Security and Privacy Aspects of Wireless Computer Networks

by

Günther Lackner

A PhD Thesis
Presented to the Faculty of Computer Science in Fulfillment of the
Requirements for the PhD Degree

Assessors

Prof. Dr. Vincent Rijmen (TU Graz, Austria and KU Leuven, Belgium)
Prof. Dr. Colin Boyd (QUT Brisbane, Australia)

June 2011

Institute for Applied Information Processing and Communications (IAIK)
Faculty of Computer Science
Graz University of Technology, Austria

Abstract

Wireless networks are one of the most influential technological accomplishments of the last thirty years. Without their help, the perceived gain in mobility our society experienced lately would have been significantly decelerated. As wireless networks in form of mobile telephony and wireless computing have pervaded our daily lives, security and privacy became very important topics. The field of IT security in general, and network security in special, draws a lot of attention from the research community and by the help of cryptography effective mechanisms were developed to form powerful building blocks for securing various different kinds of communication systems.

Part I of this thesis is dedicated to *Security in Wireless Networks* and provides an overview on these building blocks and describes and analyses their application in popular wireless network implementations, namely, Bluetooth, WiFi and WiMAX. Even as most of the used security building blocks are based on evaluated and accomplished concepts, the analysis of these standards reveals how the early designs were highly insecure and vulnerable to security related attacks. We briefly describe the evolution of these standards regarding security and evaluate the state-of-the-art.

Part II introduces the vast and blurry area of *Privacy in Wireless Networks.* In order to facilitate better understanding, we divide this topic into three subclasses, namely, *Message Related Privacy*, *Identity Related Privacy* and *Location Related Privacy.* These three concepts are presented in detail, including the state-of-the-art and our own contributions, which mainly concern identity related and location related privacy. We disclose that concerning privacy, wireless networks are still very immature and exhibit numerous vulnerabilities. In order to contribute to the enhancement of these issues, we developed attacks on the privacy protection of established standards as well as improvements for privacy preserving mechanisms.

Part III presents a collection of our contributions in the area of security and privacy in wireless networks in order to illustrate the complexity and versatility of this field. All underlying papers of these chapters have been peer-reviewed and published in the proceedings of international IT security conferences or research journals.

The main target of this dissertation is to provide a comprehensive insight on the areas of *Security and Privacy in Wireless Networks.* It includes comparative surveys and analyses of the state-of-the-art as well as original work.

Abstrakt in Deutsch

Kabellose Netzwerke gehören zweifellos zu den prägendsten technischen Errungenschaften der letzten dreißig Jahre. Ohne ihr Zutun wäre dieser hohe Grad an Mobilität unserer Gesellschaft nicht möglich. Da kabellose Netzwerke wie etwa Mobiltelefonie, Bluetooth oder WiFi in alle Bereiche unseres täglichen Lebens vorgedrungen sind, wurden Themen wie Sicherheit und Privatsphäre in kabellosen Netzwerken sehr wichtig. IT Sicherheit im Allgemeinen und Netzwerksicherheit im Speziellen haben in den letzten Jahren sehr viel Aufmerksamkeit in Forschung und Entwicklung genossen und mit der Hilfe der Kryptographie wurden effektive Konzepte zur Absicherung von mobilen Kommunikationssystemen geschaffen.

Teil I dieser Doktorarbeit ist dem Thema “Sicherheit in kabellosen Netzwerken” gewidmet und präsentiert einen Überblick über verwendete kryptographischen Konzepte und analysiert deren Implementierung als Teil von bekannten Standards wie Bluetooth, WiFi und WiMAX. Obwohl die meisten dieser Implementierungen auf bewährten Grundlagen beruhen zeigt unsere Analyse, dass vor allem ältere Designs viele Sicherheitslücken aufweisen. Weiters beschreiben wir die Evolution dieser Standards betreffend ihre Sicherheit und evaluieren den aktuellen Stand der Technik.

Teil II behandelt das umfangreiche und nur wage definierte Thema “Privatsphäre in kabellosen Netzwerken”. Um die Beschreibung zu erleichtern teilen wir das Thema in folgende Untergruppen: Nachrichten-, Identitäts- und Ortsbezogene Privatsphäre. Diese drei Konzepte werde im Detail besprochen und unsere eigenen Beiträge zu diesen Bereichen werden im Detail präsentiert. Wir kommen zum Schluss, dass kabellose Netzwerke im Bezug auf Privatsphäre noch in ihren Kinderschuhen stecken und zahlreiche Verwundbarkeit aufweisen. Um zu dieser Problematik etwas beizutragen haben wir sowohl Attacken auf, als auch Verbessrungen für bestehende Mechanismen entwickelt, die eine Verbesserung der Privatsphäre bewirken sollen.

Teil III präsentiert eine Reihe unserer Arbeiten aus dem erweiterten Bereich von Sicherheit und Privatsphäre in kabellosen Netzwerken um die Komplexität und Vielseitigkeit dieses Forschungsbereiches zu unterstreichen. Alle Kapitel beruhen auf Publikationen bei International anerkannten Konferenzen oder Forschungsjournalen.

Das Ziel dieser Doktorarbeit ist es, einen umfassenden Überblick und Detailwissen aus dem Bereich “Sicherheit und Privatsphäre in kabellosen Netzwerken” zu vermitteln. Sie enthält detaillierte Studien und Analysen der aktuellen Situation als auch neue Beiträge und bisher unveröffentlichte Forschungsergebnisse.

Acknowledgments

This thesis is dedicated to Emma!

First of all, I want to thank Silke, my bride-to-be, for her support and comprehension without which this undertaking would have been doomed and without hope.

Second I want to thank Vincent Rijmen for accepting me as a PhD student and his valuable guidance that has always been available on my way.

Further on, special thanks go to my colleagues Peter Teufl, Stefan Kraxberger, Roman Weinberger, Daniel Slamanig, Martin Eian, and others for their inestimable comments, suggestions, and discussions. They have more than once been an inspiration for my research.

I am also much obliged to Colin Boyd and the Information Security Institute at Queensland University of Technology in Brisbane, for providing me the opportunity to pursue my work for six months in a quiet and inspiring environment.

Finally, my thankfulness goes to Alois Nöbauer and Hansjörg Cohnen, who so generously allowed me to conduct my academic work during our collaboration for two years and granted me sabbatical leave to finish my thesis.

Günther Lackner
Brisbane, March 2011

Table of Contents

List of Tables

List of Figures

1

Introduction and Organization

1.1 Introduction

1.1.1 Security and Privacy at a Glance

> *Computer Security is vital. We are all Security Consumers.* [160]
>
> - Bruce Schneier

The concept of security is very complex and as old as society itself. It pervades most areas of our daily lives and the lack of it seriously affects our wellbeing and disrupts our development. Examples for corrupted security could range from global wars, influencing whole civilizations, down to being mobbed at the workplace or school, affecting just one or several individuals. As important parts of our lives, especially in western civilizations, revolve around consuming, we transformed the concept of security into a product we are able to obtain. In some cases a basic level of security is provided by an authority, like the government and also enforced by related agencies, such as the police force and justice. In other cases, we are able to decide the level of security for ourselves by regarding it as a consumer good. For example, when we buy a car, food or choose the airline for our next holiday trip, we think about our security. We prefer car companies, airlines and supermarkets with a good recommendation and neighborhoods with a low crime rate. In other words - *we are Security Consumers.*

As already mentioned, a basic level of security is dictated by government laws and regulations and guarded by the military, law enforcement agencies and justice. In the past, the desire for complete security was often strong enough to form societies with omnipotent law enforcement. In an ideal world, governments and their agencies would be inherently trustworthy, acting without exception in good faith with impeccable integrity [21]. Personal privacy could easily be sacrificed for the higher cause of absolute security and people should be *obedient good citizens* without any criminal intentions. But as the world is not ideal, all of these *social experiments* failed, and turned into totalitarian regimes - the rest of it is history.

> *Civilization is the process toward a society of privacy. The savage's whole existence is public, ruled by the laws of the tribe. Civilization is the process of setting man free from men.* [21]
>
> - Ayn Rand, The Fountainhead

> *Privacy is the right of individuals to control the collection and use of information about themselves.* [21]
>
> - Michael Caloyannides

Privacy is usually lesser valued by humans than security and we tend to have a slightly ambivalent look on it. A meaningful example is given by Michael Caloyannides in [21]:

> *Many of us cannot bear the thought that our neighbors espouse different self-evident truths and customs than we do, and we often proclaim a peculiar right to know what they are up to, while also firmly arguing our right to privacy from them.*

Advances in information and communication technologies in the last decade, have facilitated the collection, storage and retrieval of personal information. This collection is usually driven either by political or commercial motivation. Governments tend to monitor and control citizens under the guise of protecting us from each other while the commercial sector cannot pass up the lucrative opportunities of direct marketing.

This excessiveness in data collection and storage can sometimes become a boomerang as in the case of the publishing of secret and top secret military and diplomatic documents of the USA and other nations in 2010 by an Internet company called Wikileaks [8].

Recent developments are proving, that modern information technology is facilitating the steady disappearance of individual privacy. This development is sometimes accelerated by events like terrorist attacks, allowing governments to impose laws, shifting the balance of privacy rights in its direction, e.g. the US Patriot Act [1] after the incidents known as 9/11 which seriously restricted the privacy of U.S. citizens as well as travelers if a criminal or terroristic act is suspected.

As the further discussion of political aspects of security and privacy would go far beyond the scope of this theses (and of our expertise) we will now focus on security and privacy in information technology.

1.1.2 Security and Privacy in Information Technology

Computer networks have been around for several decades and evolved from interconnecting only a few computers in a single room, into integrating hundreds of millions of devices worldwide forming an entity we all know as the Internet. While in the past, computer networks usually consisted of classic hardware devices like terminals, servers and network accessories like storage systems or printers, networks nowadays cover a multitude of device classes, starting from the most powerful super computers going all the way via personal computers, laptops, household appliances to simple sensor nodes.

With the demand for mobility, wireless communication means have been developed, allowing the appearance of ultra-portable computing devices such as notebooks, tablet computers and smart phones. Thanks to advanced miniaturization, wireless devices can be even smaller than a coin, allowing to connect virtually everything to a network following the imagination of Kristofer Pister. He introduced the concept of *smart dust* [187] in 2000, which is basically a hypothetical system of tiny wireless devices coping with a variety of tasks. Although it might take another ten years to come close to his vision of interconnecting *everything*, recent developments brought small, powerful mobile devices with outstanding connectivity to the mass market and therefore into our very homes and workplaces.

Due to the convenience, people tend to switch from hardwired to wireless communication systems. The decrease of wired telephones might be a very characteristic example, as is the increase of laptop computers, who overtook desktop systems in global sales figures[1].

Following industry's euphoria to this switch of paradigms, which promised high profits, engineers started to develop protocols, standards, and devices, to satisfy the market's demands. Unfortunately, many times security was handled as an afterthought, hastily and sloppy integrated into systems which have not been designed for it. The results were technologies, that allowed attackers to circumvent security measures rather easily, gaining access to sensible data transmitted by credulous users.

[1] Gartner Group http://www.gartner.com/

In contemplation of these unsatisfying circumstances, this thesis focuses on security and privacy aspects of wireless computer networks, hoping to contribute to enhancing security. As security engineers, it has to be our goal to design systems with appropriate security. Further, we need to educate people how to get involved in the security around them, and now to maximize the amount of it they get for what the pay. But most of all, we need to educate common users about possible insecurities, preserving them from harm by technologies developed by us engineers.

The concepts of security and privacy are not easy to distinguish as they on one hand overlap in certain areas, and on the other are not always combinable. Security always involves trade-offs. It costs money, convenience, functionality and sometimes, freedoms like liberty or privacy [160].

In a general definition, IT security, with a focus on network security, involves the following concepts [191]:

1. **Confidentiality**
   An attacker should not be able to read the messages transmitted between two parties. Confidentiality in Information Technology is mainly based on cryptographic mechanisms like stream or block ciphers.

2. **Integrity**
   The receiver of a message wants to be sure that this message has not been modified during the transmission. Integrity is provided by cryptographic mechanisms like hash functions.

3. **Authentication**
   The receiver of a message wants to be assured of the identity of the sender. Authentication includes entity and data-origin authentication. It is usually achieved by authentication protocols.

4. **Access Control**
   The ability to limit and control access to devices and applications via communication links.

5. **Accountability and Non-Repudiation**
   In many cases, like electronic commerce applications, it is necessary that an entity like a customer is not able to deny the authorization of a purchase she intentionally conducted. Non-repudiation is achieved by cryptographic mechanisms like digital signatures.

6. **Availability**
   For most IT services it is important to provide high levels of availability. Adversaries often want to disturb such services by launching *denial of service* (DOS) attacks.

It is known that in different applications, the importance of the above security requirements may differ in degree but not in kind [191].

Privacy in relation to information technology is generally classified as follows [191]:

1. **Anonymity**
   The identity of a conversation's participants should be hidden from adversaries unless intentionally disclosed.

2. **Deniability**
   The ability to deny having performed a certain action like authoring and sending a message. It is closely intertwined with anonymity and non-linkability.

3. **Non-linkability**
   Different sessions by the same user should not be linkable.

4. **Context privacy**
   An adversary should not be able to learn context information like location of a user unless intentionally disclosed.

5. **Confidentiality and Integrity**
   These properties do substantial overlap with the IT security concepts of confidentiality and integrity and are usually provided by them.

It should be noted that adversaries can be classified as *outsiders* (e.g.: eavesdroppers, other legitimate users), *insiders* (e.g.: service providers) or even the other communicating party. These different groups of adversaries can have different levels of knowledge and capabilities.

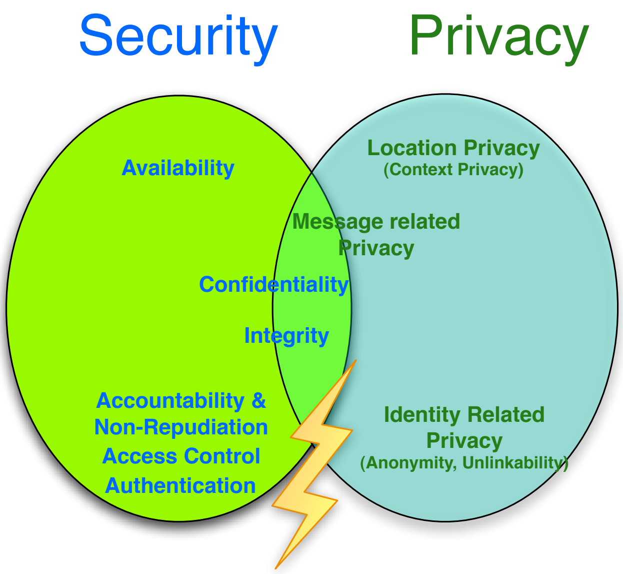


Figure 1.1: Overlapping of Security and Privacy in Information Technology

In this thesis, all privacy concerning properties are classified as follows:

1. **Message Related Privacy**
   The privacy of the content of a message.
2. **Identity Related Privacy**
   The privacy of the identity of communicating parties.
3. **Location Related Privacy**
   The privacy of the location of communicating parties.

Figure 1.1 illustrates that Confidentiality and Message Related Privacy are the same concept and fully overlap, while Accountability and Identity Related Privacy cannot be combined in the same system if perfect privacy is demanded.

As the next section will describe in detail, this thesis is organized in three parts. Part I provides basic knowledge of cryptography and a analysis of security in popular wireless network standards. Part II focuses on privacy in wireless networks and finally Part III presents a hand picked selection of relevant topics of security and privacy in wireless networks.

1.2 Organization

This thesis is organized in three main parts:

1. **Part I - Security in Wireless Networks**
2. **Part II - Privacy in Wireless Networks**
3. **Part III - Selected Chapters**

1.2.1 Part I

Part I starts with an introduction of basic knowledge on cryptography, necessary to be able to follow the discussion on security aspects of wireless network standards later in this part. Assisting as characteristic examples for the wide variety of wireless communication standards, the remainder of Part I describes and analyses security aspects of IEEE 802.11 WiFi, IEEE 802.15.1 Bluetooth and IEEE 802.16 WiMAX, belonging to the families of Wireless Local Area (WLAN), Wireless Personal Area (WPAN) and Wireless Wide Area (WWAN) Networks, respectively. This analysis has been published as [102].

1.2.2 Part II

Part II provides a discussion on three different forms of privacy:

1. Message Related Privacy, also known as Confidentiality or Secrecy;
2. Identity Related Privacy; and
3. Location Related Privacy.

Chapter 7 provides a brief introduction to the just named classifications of privacy. While Message Related Privacy strongly overlaps with the security descriptions in Part I as it is mainly a cryptographic problem, this part is held rather short.

Identity Related Privacy is about being able to determine the identity of entities taking part in a communication, regardless if it is a device or a real person.

Location Related Privacy concerns the information regarding the physical location of a device or a user. This chapter illustrates threats to location privacy and discusses some of their consequences.

After the discussion of the just named privacy classifications, Chapter 8 introduces two machine learning techniques which play an important part in the later chapters of this thesis.

Chapter 9 provides a detailed survey of techniques to attack the *Identity Related Privacy* of communicating parties in wireless networks and contains our contribution to this field called *Fingerprinting on Layer 2* (see Section 9.1.3).

Chapter 10 presents a detailed survey of mechanisms dedicated to preserve or improve the *Location Related Privacy* of participants in wireless networks. It includes our own contribution to this area called *Location Privacy Enhancement for WLANs based on Virtual Network Interfaces* (see Section 10.2).

Part II is based on our publications [100,101,105,107–110,112,170].

1.2.3 Part III

Part III presents selected chapters of security and privacy in wireless networks. It is mainly based on our publications in this area, trying to provide the reader a broader understanding on the variety of related problems.

Part III is based on our publications [100–102,104,106,111,136,174]

1.3 Contributions and Publications

This section briefly lists our publications regarding the various fields of security and privacy in wireless networks which are part of this thesis in order to clarify our scientific contributions. It also tries to illustrate the chronographic order of the publications and how the addressed research areas evolved.

1.3.1 State-of-the-Art Security in Wireless Networks (Chapters 4, 5, 6)

1. [102] G. Lackner, *''A Comparison of Security in Wireless Network Standards with a Focus on Bluetooth, WiFi and WiMAX''*, Accepted but not yet published at the International Journal of Network Security, 2011.

This journal paper presents a survey and in-depth analysis of state-of-the-art security mechanisms of the three dominating technologies (i.e. WiFi, Bluetooth and WiMAX) in the area of wireless computer networks and forms the base for the Chapters 4, 5 and 6 of this thesis.

1.3.2 Attacking identity Related Privacy (Chapter 9)

1. [107] G. Lackner, M. Lamberger, P. Teufl, and U. Payer, *''WiFi Chipset Fingerprinting''*, in Proceedings of DACH Security 2006, P. Horster, ed., Munich: 2006, pp. 41-53.

   This conference paper presents the first steps of our research in the area of wireless device fingerprinting and has the base for my master thesis and for all of our following research that is presented in Section 9.1 of this thesis.

2. [100] G. Lackner, *''IEEE 802.11 Layer 2 Address-Spoofing Protection''*, Master Thesis, 2008.

   In my masters thesis of 2008 I deepened the focus on how to counteract mac-address spoofing-attacks by the use of wireless device fingerprinting.

3. [108] G. Lackner, U. Payer, and P. Teufl, *''Combating Wireless LAN MAC-Layer Address Spoofing with Fingerprinting Methods''*, International journal of network security, Vol. 9, 2009, pp. 164-172.

   This journal paper is based on the results of the work in my masters thesis and presented the most complete survey on the topic at that time, including our own contribution presented in [107] and refined in [100].

4. [112] G. Lackner, P. Teufl, and R. Weinberger, *''User Tracking based on Behavioral Fingerprints''*, In Proceedings of the Ninth International Conference on Cryptology And Network Security, CANS 2010, Kuala Lumpur: 2010.

   Based on our intensive research in the area of device fingerprinting we started to investigate the possibilities of fingerprinting the behavior of users. This work has been published in the just named conference paper and forms the base for Section 9.2 of this thesis.

5. [110] G. Lackner and P. Teufl, *''IEEE 802.11 Chipset Fingerprinting by the Measurement of Timing Characteristics''*, In Proceedings of the Australasian Information Security Conference 2011, AISC11, Perth:2011.

This conference paper presents our most recent and advanced results in the area of device chipset fingerprinting based on the *Acknowledge Delay* first mentioned in [107].

1.3.3 Preserving Location Privacy in Wireless Networks (Chapter 10)

1. [109] G. Lackner and P. Teufl, *''Location Privacy in Kabellosen Netzwerken''*, In Proceedings of DACH Security 2010, P. Horster, ed., Vienna: 2010.

   This conference paper forms the base for our research in the area of location privacy and defines various challenges in this area which we addressed in follow up publications.

2. [98] S. Kraxberger, G. Lackner, and U. Payer, *''WLAN Location Determination without Active Client Collaboration''*, IWCMC'10: Proceedings of the 2010 International Conference on Wireless Communications and Mobile Computing, ACM, eds., ACM, Caen: 2010, pp. 1188-1192.

   This conference paper presents our contribution in the area of device location determination without any active collaboration of the client. This paper included a novel approach and it's evaluation.

3. [103] G. Lackner, *''On the Security of Location Determination and Verification Methods for Wireless Networks''*, Accepted but not yet published at the International Conference on Security and Cryptography SECRYPT 2011, Sevilla: 2011.

   Based on our research for [98], this paper presents a in-depth survey of state-of-the-art wireless location determination mechanisms and analyses their capabilities regarding their application in security relevant systems.

4. [105] G. Lackner and S. Kraxberger, *''Location Privacy Enhancement for WLANs based on Virtual Network Interfaces''*, Submitted to the Privacy Enhancing Technologies Symposium PETS2011, 2011.

   This paper presents a novel concept to enhance the location privacy in wireless networks by using virtual network interfaces.

1.3.4 Security and Privacy in Wireless Networks (Chapter 12)

1. [106] G. Lackner, S. Kraxberger, P. Teufl, and M. Eian, *''Location Aware Access Regulation for Wireless Computer Networks - A Comparative Survey''*, UNDER REVIEW! International Journal of Information Security,

2011.

This journal article presents a detailed survey on wireless location determination and verification mechanisms and analyzes their capabilities regarding their application in security relevant systems. It further on contains our own contributions presented in [103] and [98] including minor adaptations. This paper has been initiated by G. Lackner who has also written most of the text. The co-authors mainly contributed in the discussion and proof reading of the text.

2. [111] G. Lackner, P. Teufl, and R. Weinberger, *''Unterschätzes Risiko durch ultramobile Geräte''*, 7. Information Security Konferenz, Ö.C. Gesellschaft, ed., Österreichische Computer Gesellschaft, Linz:2009, pp. 43-59.

   This conference paper forms the base for Chapter 13 and has inspired all of the following papers in this chapter.

3. [174] P. Teufl, S. Kraxberger, C. Orthacker, G. Lackner, A. Marsalek, J. Leibetseder, and O. Prevenhueber, *''Android Market Analysis with Activation Patterns''*, Accepted but not yet published at MobiSEC 2011, 2011.

4. [136] C. Orthacker, P. Teufl, S. Kraxberger, A. Marsalek, J. Leibetseder, and O. Prevenhueber, *''Android Security Permissions - Can we trust them ?'*, Accepted but not yet published at MobiSEC 2011, 2011.

   These two papers have been authored by various authors and represent our combined research effort in the area of Android Market Security and Android Rights Management Security. It is very hard to distinguish the single author's contributions. Nevertheless do these papers introduce very relevant topics and are therefor part of this text.

Part I

Security in Wireless Networks

2

Introduction to Security in Wireless Networks

The number of wireless networks deployed increases every day. Due to the low cost and convenience of deploying wireless networks, they replace hardwired networks in many fields of application.

The shift from hardwired to wireless networks invalidates many established security concepts. Hardwired networks are usually integrated within structural measures, and can be protected by building security or perimeter protection. With a state-of-the-art intrusion prevention system (IPS) to protect the connection to the Internet, hardwired networks can thus be considered closed and secure, as illustrated in Figure 12.1.

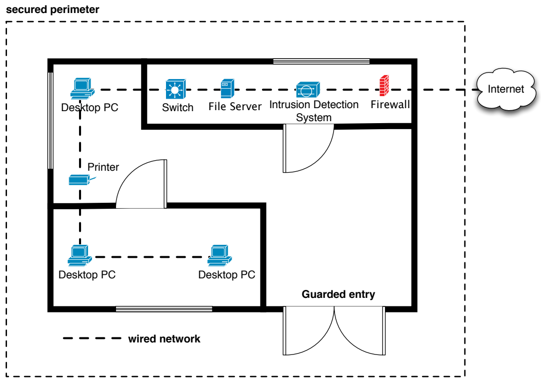


Figure 2.1: Wired-only Environment with Perimeter Protection

The nature of radio propagation makes it possible to attack wireless networks from outside the established perimeter protection. Figure 12.2 illustrates how wireless network coverage could extend to a public domain outside of a controlled building (protected area).

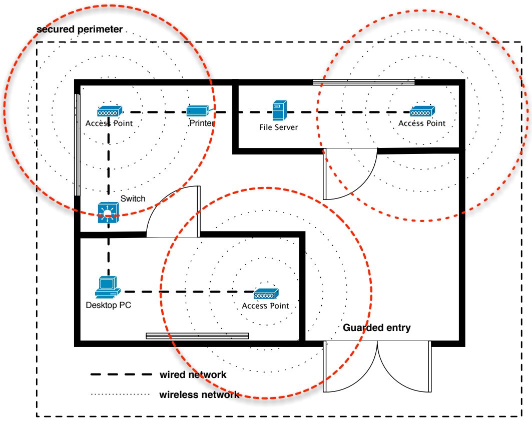


Figure 2.2: Environment with Wireless Components

As building security and perimeter protection are not sufficient to avoid attacks against the wireless network, the general approach is to secure these infrastructures by cryptographic measures. Almost all state-of-the-art wireless computer network technologies provide strong cryptographic mechanisms to provide confidentiality and integrity.

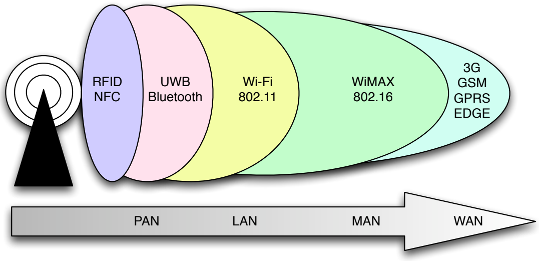


Figure 2.3: Hierarchy of popular Wireless Network Standards

Part I of this thesis is organized in the following chapters.

Chapter 3 provides the reader with basic knowledge about cryptographic methods relevant for wireless network security. It explains the principles of symmetric and asymmetric crypto schemes and presents the most important members of each. It further on discusses the idea of hash functions and their relevant fields of application. The cryptography chapter finishes with a brief introduction to authentication and presents the basic building blocks of most modern authentication protocols.

Chapters 4, 5 and 6 present popular IEEE network standards for personal area (WPAN), local area (WLAN) and wide area wireless networks (WWAN). It analyses IEEE 802.15.1 (Bluetooth), IEEE 802.11 (WiFi) and IEEE 802.16 (WiMAX) according to IT security principles like confidentiality and integrity and evaluates their over-all security levels.

- **IEEE 802.15.1 - Bluetooth**
  Concluding it has to be said, that the deployment of Bluetooth poses a serious security risk especially for enterprise settings. Even though BT can be regarded secure if all devices are configured properly, the probability of the occurrence of vulnerabilities is too high to allow its implementation in security-critical systems, apart from the fact, that it is almost impossible to properly configure security features in BT by non-professionals.
- **IEEE 802.11 - WiFi**
  Regarding WiFi it has to be said that early versions of IEEE 802.11 standards were highly insecure due to the fact that the initial design incorporated no security mechanisms at all. Later amendments were error prone and could easily be circumvented by attackers. Only with the introduction of IEEE 802.11i (see Section 4.2) in 2004, adequate security levels can be guaranteed. But unfortunately, the configuration of available implementations is often complex and not usable by standard consumers.
- **IEEE 802.16 - WiMAX**
  Even though that security was integrated in the original design of WiMAX, several serious vulnerabilities were discovered shortly after the release of the first version. These flaws have been corrected in successive standards. The actual version, IEEE 802.16e-2005 is still a young standard and currently a lot of security related research is conducted around it. As history has shown with related wireless networks, this research will uncover further vulnerabilities and design flaws.

In conclusion of the security evaluation of the example standards it has to be said, that many of them were designed with security as an afterthought. Subsequent introduction of security features often failed because of the original design and several amendments and modifications have been necessary to provide satisfactory levels of security.

From the point of view of a non IT professional technologies are inscrutable and available systems are very difficult to configure in order to provide adequate security levels. Best practice examples are often complicated and restrict the overall practicability, leaving them unattractive for standard consumers.

Before concentrating on the wireless implementations, the following chapter provides a brief introduction to cryptography and a short description of the mechanisms implemented network standards discussed later in this part of my work.

3 Cryptography in Wireless Networks

Cryptography accompanied human societies since the great empires of the ancient world. An example of early cryptographic methods was created and used by Julius Caesar in the ancient Rome to prevent enemies from learning sensitive military information by intercepting messages destined for Rome's Legions [183].

Nowadays, due to the computing power of modern hardware, more sophisticated cryptographic methods are needed to provide confidentiality for sensitive information. Protecting electronic systems is crucial to our modern way of living. Since information technology is omnipresent in modern society, exploiting security flaws has become a very profitable business for cyber-criminals.

Before describing modern cryptography implementations a closer description of the standard communication scenario with relevance for wireless computer networks is necessary. Figure 3.1 shows a scheme that is generally valid for secure data transmission via an insecure channel. Alice wants to send confidential information to Bob, and Eve is a potential eavesdropper who might mean ill.

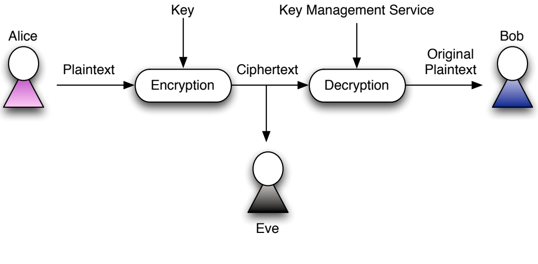


Figure 3.1: Confidential Data Channel

Let us assume Alice and Bob have already agreed on a cryptographic method to use. Alice now encrypts the plaintext message using her key and sends the ciphertext message to Bob who decrypts the ciphertext message using his own key. Depending on the used scheme, Bob's and Alice's keys might be identical or not. In *symmetric cryptography* all keys are known to Alice and Bob while in *asymmetric cryptography* Bob possesses a private key that is needed for decryption while Alice only knows the public key that can only be used for encryption [183].

As an attacker, Eve may have one of the following goals [183]:

1. Read the message.
2. Find the key and thus read all messages encrypted with that key.
3. Corrupt Alice's message into another message in such a way that Bob might think that Alice has sent the altered message.
4. Masquerade as Alice, and thus communicate with Bob so that Bob believes he is communicating with Alice.

Depending on how much information Eve possesses she might be able to perform the following four attacks on the used crypto system [183]:

1. Ciphertext only
   Eve only possesses a copy of the transmitted, encrypted message.
2. Known plaintext
   Eve got a copy of an encrypted message as well as the corresponding plaintext.
3. Chosen plaintext
   Eve is able to use the encryption system by herself but she is not able to retrieve the secret key. She can simply choose any plaintext message and try to use the resulting ciphertext to deduce the key.
4. Chosen ciphertext
   Eve is able to use the decryption system by herself but she is not able to retrieve the secret key. She can simply choose any ciphertext message and try to use the resulting plaintext to deduce the key.

Modern cryptography systems should be able endure all of these attacks. They should also follow one of the most important assumptions in cryptography, the so called *Kerckhoff's principle.*

> In assessing the security of a crypto system, one should always assume the enemy knows the method being used. The security of a system should therefore be based on the key and not on the obscurity of the algorithm [183].

Consequently one always has to assume that Eve has complete knowledge about the used cryptographic mechanisms and algorithms.

Modern cryptography can be divided in two main types, namely *symmetric* and *asymmetric* schemes. The major difference between them is the requirement for a shared secret key between the encryptor and the decryptor in symmetric cryptography. They also differ in various properties and therefore render different advantages and disadvantages [117].

The following section provides a brief description of symmetric and asymmetric crypto schemes and their relevance for mechanisms deployed in wireless LAN security.

3.1 Symmetric Cryptography

In symmetric-key cryptography schemes both, Alice and Bob know all encryption and decryption keys. In many cases the two keys are the same.

All early crypto systems were based on the symmetric scheme. The most important members nowadays are the *Data Encryption Standard* (DES) [129] and its successor the *Advanced Encryption Standard* (AES) [35].

3.1.1 Advantages of Symmetric-Key Cryptography

- Performance
  Popular asymmetric methods are several orders of magnitude slower than those of high-performance symmetric schemes.

- Key size
  Symmetric schemes may obtain the same security level as asymmetric schemes by using much shorter keys.

- Practicability
  Symmetric ciphers may be used in many other cryptographic mechanisms like *pseudo random number generators*, *hash-functions*, *message authentication codes* or *digital signature schemes.*

The most significant advantages of symmetric-key cryptography are the high efficiency and the relatively small key lengths [117].

3.1.2 Stream Cipher

A Keystream Generator provides a random series of bits that depend on some secret key. A stream cipher uses this random series to XOR it with some plaintext. Figure 3.2 illustrates this principle by the example of the RC4 algorithm.

Figure 3.2 illustrates the basic idea. Each single bit of the keystream is XOR'ed (exclusive OR) with one bit of the Plaintext and the result is the encrypted plaintext also known as the ciphertext. The decryption process works

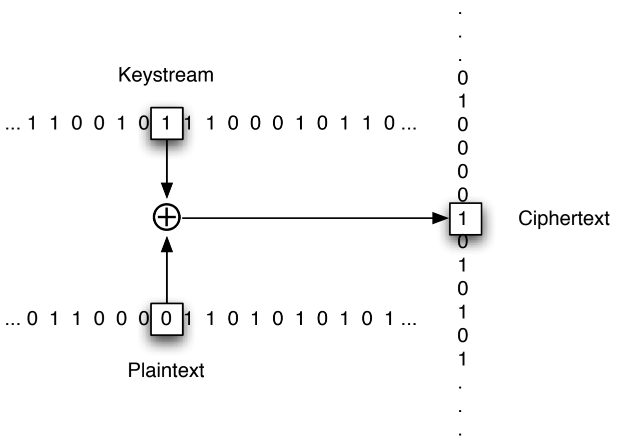


Figure 3.2: Stream Cipher

exactly the other way round. The Ciphertext is XOR'ed with the keystream and the result is the original plaintext message.

The vital part of a stream cipher is the Keystream Generator. In general, these schemes suffer from the fact, that the same secret key will cause the same keystream if used more than once which can reveal relations between two encrypted messages. It is necessary to change the secret key as often as possible. There exist stream cipher implementations that allow to change the IV or this task is can be handled by a *Key Scheduling Mechanism* (KSM) [171].

3.1.2.1 Linear Feedback Shift Register (LFSR)

As the first of two examples for stream ciphers we want to present Linear Feedback Shift Register (LFSR).

LFSRs can be used to create a binary data stream. A Linear Feedback Shift Register consists of small memory cells forming a circuit whereas its own output is transformed by some feedback function and serves as the input for the next computational round. In each circle, a predefined set the bits of the memory cells (s_{j-1} to s_{j-n}) are XORed according to the scheme in Figure 3.3. One also defines a set of bits (c_1 to c_n) which are set to one if the corresponding cell is affected. The initial state of the register has to be set by some initialization vector. All cells set on zero will result in an all zero output sequence [171]. Figure 3.3 provides a scheme of a possible LFSR.

As LFSRs are periodical and linear, they cannot be used as a stream cipher by their own as they are insecure against known plaintext attacks. LFSRs can be combined using a non-linear combining function providing a high degree of cryptographic security. They can be used as building blocks in very fast stream ciphers, especially if implemented in hardware.

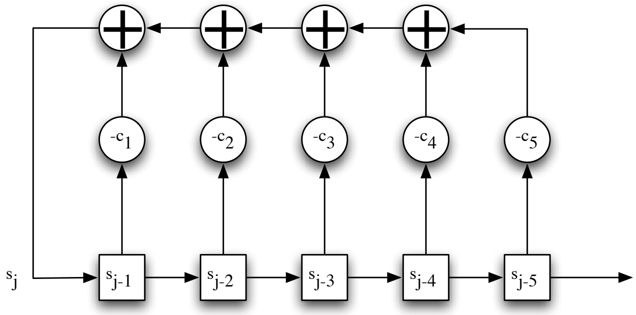


Figure 3.3: Linear Feedback Shift Register (LFSR) Stream Cipher

3.1.2.2 RC4

RC4 was released in 1987 and soon became the standard cipher for many popular application fields. It was implemented for securing Internet traffic using the *secure sockets layer* (SSL) protocol [52]. Its integration in prominent software products like Microsoft Windows or Lotus Notes supported its global spreading [117]. It consists of two parts [117]:

1. Key-scheduling algorithm (KSA)
   It turns a random key into an initial permutation.

2. Pseudo random number generator (PRNG)
   It uses the permutation of the KSA to generate a pseudo-random output sequence.

Due to design flaws in the KSA, RC4 can not be seen as strong cryptography any more. Unfortunately RC4 has been implemented in the IEEE 802.11 standard as part of the privacy service *wired equivalent privacy* (WEP) [75].

3.1.3 Block Cipher

In contrast to stream ciphers, block ciphers encrypt a plaintext on a block to block basis and not bit after bit. The size of these blocks can vary and depends on the used algorithm. Many modern block ciphers are based on the Feistel scheme. (see Figure 3.4) [171]. One exception is the Advanced Encryption Standard (AES) which will be presented later in this chapter.
The following steps are performed:

1. Equally sized blocks of the plaintext and divides each of these blocks in a left and a right part (L_0, R_0).

2. R_0 is processed by some function F, that applies the Round Key K_i in some way. The Round Key is in some way derived from the secret key.

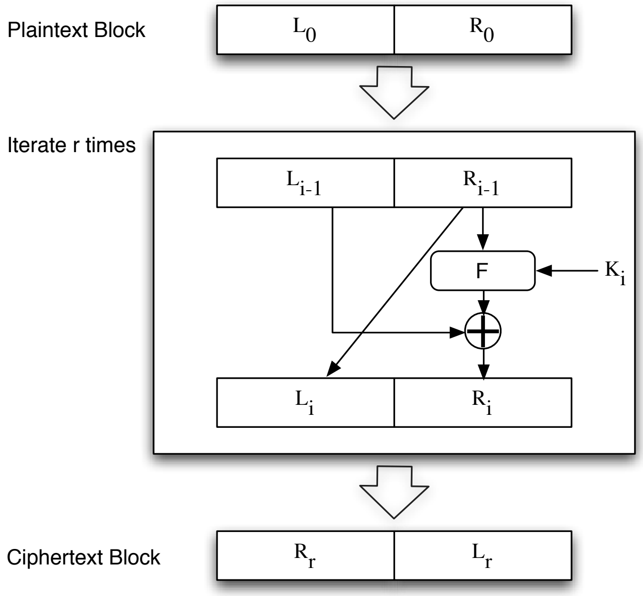


Figure 3.4: Feistel Block Cipher

3. The result of F is XOR'ed with L_0 and becomes R_1.
4. R_0 becomes L_1
5. Steps 1 to 4 are applied to the new block consisting of L_1 and R_1. This process is repeated several times and is called *a Round*.

The security of a Feistel cipher is based on:

- the generation method of the round keys
- how many iterations / Rounds are taken
- how the function F is defined

Block ciphers can be operated in various modes of operation, providing different properties regarding performance and security. Besides others, these four main *Modes of Operation* have been standardized internationally, namely:

- ECB-Mode (Electronic Code Book Mode)
- CBC-Mode (Cipher Block Chaining Mode)
- OFB-Mode (Output Feedback Mode)
- CFB-Mode (Cipher Feedback Mode)

The following sections shortly introduce three examples for block ciphers.

3.1.3.1 Data Encryption Standard (DES)

DES [129] is a variant of the basic Feistl scheme described in Section 3.1.3. It has been developed by a team in IBM and is based on a cipher named Lucifer and contributions of the National Security Agency (NSA) of the United States Government. The Data Encryption Standard has been approved by several international standards organizations as the ANSI or ISO, making DES the first publicly available cryptographic algorithm with *official status* [171].

DES uses a 56-bit keys and 16 Rounds. Due to the relatively short key lengths, DES can not be regarded state-of-the-art and secure anymore.

3.1.3.2 Advanced Encryption Standard (AES) - Rijndael

The algorithm behind the *Advanced Encryption Standard* (AES) is called Rijndael and was designed to replace the outdated *Data Encryption Standard* (DES). In 1997, the *National Institute of Standards and Technology* (NIST) launched a call for the replacement of DES. On October 2nd 2000 Rijndael was officially announced as AES. It has been chosen from 15 candidates.

Rijndael is a block cipher and was designed by Vincent Rijmen and Joan Daemen at the *Katholieke Universiteit Leuven*[1] and has become the widest spread symmetric-key cryptography algorithm [35].

Rijndael is characterized by its simplicity and outstanding performance. No attack faster than a brute-force approach has been discovered yet[2]. Brute-force simply means performing an exhaustive search over the whole keyspace. With a key-length of 256 bit the number of all possible keys is unimaginably large[3].

Rijndael is based on blocks of 128 bits and supports key lengths up to 256 bit. Its encryption process consists of four steps named:

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

For more details about the Rijndael algorithm the reader is referred to [35]. AES is implemented in several wireless network standards as IEEE 802.11i [78] or IEEE 802.16e-2005 [81].

3.1.3.3 SAFER+ (AES Candidate)

SAFER+, another block cipher, is based on the existing SAFER family of ciphers. It succeeds SAFER K-64, SAFER K-128, SAFER SK-64, SAFER SK-128 and SAFER SK-40. The algorithm was not selected for the final round in the

[1] http://www.kuleuven.be/

[2] Attacks exist only for round-reduced variants.

[3] $2^{256} = 1.15792089E10^{77}$

NIST call for the Advanced Encryption Standard. It is mentioned here due to its implementation in the IEEE 802.15.1 (Bluetooth) standard [34]. After its release, several vulnerabilities have been discovered rendering it insecure [72].

3.2 Asymmetric Cryptography

In the 1970s, the introduction of *public key algorithms* revolutionized cryptography. Suppose a scenario where Alice and Bob want to communicate securely, but they are hundreds of kilometers apart and have not agreed about a secret key yet? It seems impossible to do this without getting in contact first in order to agree on a key. They obviously can not send the key via an insecure channel so they need to meet personally or use a trusted courier to carry the key between them. The amazing fact is that asymmetric schemes provide solutions to this problem.

A key-pair consisting of a public encryption key and a private decryption key is created in such a way that it is mathematically infeasible to find the the decryption key only by knowing the encryption key. The most popular implementation is RSA[4] that is based on the difficulty of factoring large integers.

3.2.1 Advantages of Asymmetric-Key Cryptography

- Key-agreement
  For a two-party communication only one secret key is needed. The key never has to be possessed by any other party than the one who has created the key. The key-agreement can be carried out by using insecure channels.

- Number of keys
  In multiparty crypto systems using asymmetric schemes only one key-pair per client is needed while symmetric schemes would need a quadratic number of keys (one key per pair).

- Key-scheduling
  Asymmetric keys are usually secure for a very long time while symmetric keys in common practice are changed frequently. Sometimes they are changed for every session.

The most significant advantage of asymmetric-key cryptography is the possibility of key-distribution over insecure channels [117].

3.2.2 The RSA Algorithm

The RSA algorithm was the world's first (published[5]) public key encryption algorithm. It is named after its inventors Ron **R**ivest, Adi **S**hamir and Leonard

[4] Named after the inventors Ron Rivest, Adi Shamir and Leonard Adelman

[5] Clifford Cocks, a British mathematician working for the UK intelligence agency GCHQ, described an equivalent system in an internal document in 1973, but given the relatively expensive computers needed to implement it at the time, it was mostly considered a curiosity

Adelman. It was published in 1978 and has stood the test of time remarkably well [171].

Following the principal of asymmetric-key cryptography, RSA provides a key pair consisting of a private and a public key. These keys are mathematically intertwined in a way, that the private key cannot, or only with very high effort, be derived from the public key.

If RSA is implemented according to some design rules, it is based on the mathematical difficulty of finding prime factors of large integers. Solving the prime factorization of large numbers is computationally very costly and time consuming. But according to Moore's law [128], hardware performance grows exponentially, and therefore drops the time needed to get the factors of large integers. RSA circumvents this fact by using even larger keys up to 4096 bit[6] at the time this chapter was written [171].

RSA related operations are time consuming and computationally costly, rising with larger key lengths. New, more performant technologies as *Elliptic Curve Cryptography* (ECC) have been developed and start to replace RSA in many applications. Nevertheless, RSA is still the most popular algorithm in public key cryptography and will be around for many more years. It is also deployed in wireless network standards as the remainder of this chapter will show.

3.2.3 Elliptic Curve Cryptography (ECC)

ECC is a modern cryptographic method based on the difficulty of a discrete logarithm problem. It presents a very performant and secure alternative to RSA. A detailed and understandable description of ECC can be obtained from [171].

3.2.4 Public Key Infrastructures (PKI)

As the topic of public key infrastructures is a very dynamic and versatile one, this section can only provide an rough overview. For more detailed information on PKI and digital certificates we refer to [2].

Public Key Infrastructures are very popular and widely spread. They provide means to reliably identify communication parties and secure their communications.

Definition 3.1. "A Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates." [182]

and, as far as is publicly known, was never deployed. His discovery, however, was not revealed until 1998 due to its top-secret classification, and Rivest, Shamir, and Adleman devised RSA independently of Cocks' work. (Taken from http://en.wikipedia.org/wiki/RSA)

[6] A number with 1234 digits.

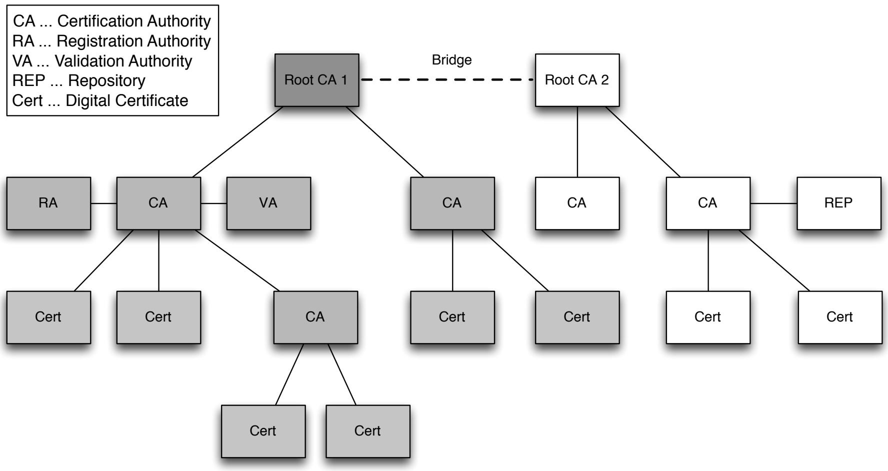

CA ... Certification Authority
RA ... Registration Authority
VA ... Validation Authority
REP ... Repository
Cert ... Digital Certificate
Root CA 1
Bridge
Root CA 2
RA
CA
VA
CA
CA
CA
REP
Cert
Cert
CA
Cert
Cert
Cert
Cert
Cert
Cert

Figure 3.5: Components of a Public Key Infrastructure

3.2.4.1 Digital Certificates

Digital certificates allow to to encrypt and digitally sign messages. They allow to prove the ownership, and other features, of a public key in a public cryptography scheme. The owner of a certificate and the belonging public key can be a person, organization or a piece of hardware. Such a certificate is usually issued by a trusted third party called *Certification Authority* (CA). The issuing CA provides a directory to allow anyone to check the validity of a questioned certificate [39].

Each certificate is *digitally signed* by the issuing Certification Authority which itself holds a digital certificate signed by a superior CA (see Figure 3.5). Certificates can be validated by checking the directory of the signing CA or an dedicated *Validation Authority* (VA).

Several standards defining digital certificates exist. The most important is X.509 defined by the International Telecommunications union (ITU-T) [86]. X.509 is implemented in most current operating systems independent from the hardware platform, ranging from personal computers, mobile phones to small electronic devices as wireless network transmitters.

3.2.4.2 Components of a PKI

Figure 3.5 presents an example structure of a PKI. Public Key Infrastructures may consist of one ore more Certification Authorities (CA) which have to be structured in a hierarchic matter an may be joined at some level. CAs are responsible for issuing certificates and may also provide services to enroll new certificates and for the validation or revocation of former issued certificates.

PKIs can furthermore contain *Registration Authorities* (RA) which may assist the CAs in providing the infrastructure for entity registration as well as

Validation Authorities (VA) to provide certificate validation capabilities in order to disburden the CA. Additional components can be *Repositories* to provide certificate directories and *Revocation Lists* to publish information about issued certificates and their validity status.

3.3 Hash Functions (MDC)

Hash functions $h()$ constitute a very important component of many cryptographic algorithms. They take a message of arbitrary length and return as output a *message digest code* (MDC) of fixed length (see Figure 3.6). A hash function needs to satisfy certain properties [183]:

1. The message digest code (MDC) $h(m)$ can be calculated very quickly.
2. h must be a *one-way* function.
   Given a y it must be computationally infeasible to find an m' with $h(m') = y$.
3. It must be computationally infeasible to find messages m_1 and m_2 with $h(m_1) = h(m_2)$. The function is then called *strongly collision-resistant.*

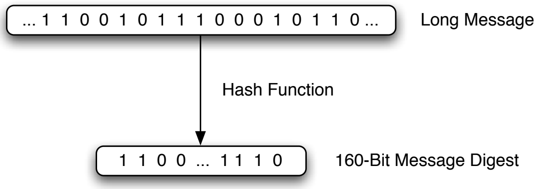


Figure 3.6: A Hash Function computing a 160-Bit Message Digest of a Long Message

Since the number of possible messages is theoretically infinitely larger than the number of possible message digests, there should also exist an infinite number of examples where $h(m_1) = h(m_2)$. Requirement number three only states that the task of finding such a fitting pair must be hard enough to be computationally infeasible [122].

The most notable hash functions are the *Secure Hash Algorithm* family (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512), the *Message Digest* family (MD[7], MD2, MD4, MD5) and the RIPEMD-160 message digest algorithm [183].

In network communication, hash functions often serve to provide *message integrity.*

Definition 3.2. Message integrity is the property whereby data has not been altered in an unauthorized manner since the time it was created, transmitted, or stored by an authorized source [122].

[7] MD has never been published

The following example should illustrate a typical scenario:

Example 3.1. Before sending a message, Alice computes a MDC of the message and delivers the message together with the MDC. After receiving this packet, Bob himself computes the MDC of the message using the same hash function as Alice. Now he just needs to compare the received MDC with the one he has computed. If they are identical, the chance that the message has been altered during transmission is extremely small. (see Figure 3.7)

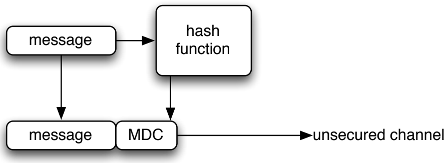


Figure 3.7: Message Integrity provided by a MDC only

Additional to message integrity, many applications demand the authentication of the origin of the message. This feature can not be provided by simple hash functions.

3.4 Message Authentication Codes (MAC)

3.4.1 Introduction to MACs

As mentioned in the last section, for assuring secure communication, providing message integrity is often not sufficient because it can not guarantee the authenticity of the data source.

Definition 3.3. Message origin authentication is a type of authentication whereby a party is corroborated as the (original) source of a specified data created at some (typically unspecified) time in the past.
By definition, message origin authentication includes data integrity [122].

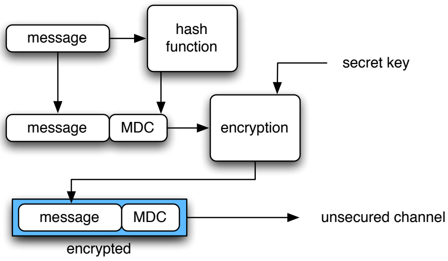


Figure 3.8: Message Origin Authentication provided by a MDC and encipherment

If we again take a look at the example 3.1 in the last section, in which Bob verified the integrity of Alice's message, Bob has no means to also verify that the message has really been sent by Alice. One possibility would be the encryption of the message together with the MDC (see Figure 3.8). In this case Bob decrypts the message and follows exactly the scheme illustrated in the example in the last section.

Sometimes it is not possible or desirable to encrypt the message. In this case, the usage of a *message authentication code* (MAC) would be the most practical way. It uses a secret key for creating the MAC of the message. Following the scheme of symmetric-key cryptography, the receiving party knows the secret key as well, and is now capable of verifying the MAC value and furthermore the message origin authenticity (see Figure 3.9) [122].

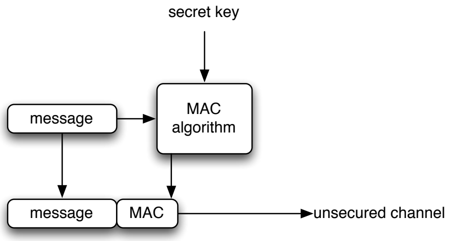


Figure 3.9: Message Origin Authentication provided by a MAC only

MAC algorithms may be constructed from different cryptographic primitives. If they are derived from hash functions they are called *HMAC*. MACs based on block cipher algorithms like AES or DES are called *OMAC, CBC-MAC* or *PMAC*. Although *dedicated MACs* have been designed, the most popular and securest MACs are CBC-MACs.

3.4.2 Michael Message Integrity Code (MIC)

In 2004 the IEEE ratified the draft of the IEEE 802.11i standard. It is an amendment to 802.11 and should replace WEP in the long run. Besides a complete new design (*Counter-Mode-CBC-MAC Protocol*, CCMP), MIC also provides a compatibility mode for legacy hardware (*Temporal Key Integrity Protocol*, TKIP). TKIP implements a keyed hash-function called *Michael* that is meant to provide message integrity (see Definition 3.2) [73].

Michael is a *message integrity code* and was designed by Niels Ferguson in 2002 [48]. It is a keyed hash-function that takes a message of arbitrary length and a 64-bit Michael key. The key is converted into two 32-bit words and the output message is partitioned in blocks of 32-bit length and padded that the message length is a multiple of four.

Like any keyed hash-function Michael should fulfill the basic requirements (see Section 3.3). Even the author of Michael knew about a flaw right from the release. Its is even mentioned in [48, page 6]:

> A known-plaintext attack will reveal the key stream for that IV, and if the second packet encrypted with the same IV is shorter than the first one, the MIC value is revealed, which can then be used to derive the authentication key.

Avishai Wool was able to create a simple function that is capable of inverting Michael, and he proposed a related-message attack [193]. In [74] Huang et al. proved that Michael is also not *collision-resistant*. In fact it is not very hard to find a collision and furthermore launch a packet-forgery attack.

Although these attacks are not practical yet, they reveal weaknesses in Michael that render it as not secure on the long run. Fortunately, TKIP was designed to be a short-term fix for WPA and it will be replaced by the 802.11i CMCP in near future.

3.5 Digital Signatures

Digital signatures can generally be compared to handwritten signatures but the number of their uses goes beyond them [171]. For example, digital signatures can be used to:

- allow users to authenticate themselves to a system;
- allow users to authenticate data; and
- sign documents.

Practically, every thing that has a binary representation can be signed, but unlike handwritten signatures, each digital signature is different and can therefor not be copied. Basically, digital signature mechanisms bind a Private Key to the content of a digital message. The consequences are that the message cannot be altered undetected after the signature has been created. Digital signatures can provide the following properties:

- **Authentication**
  Digital signatures can be used to authenticate the source of a message. A property vital in critical transactions like financial transactions.
- **Integrity**
  If a message is digitally signed, any change in the message after creating the signature can be detected as it will invalidate the signature itself.
- **Non-repudiation of origins**
  An entity that has signed some information cannot at a later time deny having signed it.

Digital signature schemes can be based on different Public Key crypto mechanisms such as RSA or ECC (see Sections 3.2.2 and 3.2.3). The most popular signature algorithm is called *Digital Signature Algorithm* (DSA) [133] which is standardized and finds it application in many real-world applications.

3.6 Authentication

Authentication is the process of establishing or confirming some entity as *authentic*. Entities in authentication processes may be humans, hardware, software, locations, timestamps, single messages and many others.

Sometimes, authorization is mistaken for authentication. Authorization is the process of determining if some entity is allowed to perform certain operations or access specific resources. Authentication, therefore, must precede authorization, and is the foundation to establish any level of trust between communicating parties. The following sections are mainly based on the book *Modern Cryptography - Theory & Practice* by Wenbo Mao [118].

In information technology, authentication is usually done by verifying if a claim made about a subject is true. This process may be done one-way or mutual and can be based on different categories of claims:

1. **To know something**
   e.g. any shared secret like a password

2. **To have something**
   e.g. any security token such as a smart card

3. **To be something**
   e.g. any personal and unique (or very rare) characteristic like a fingerprint

4. **To be somewhere**
   e.g. to be at a certain location

Based on these different kinds of claims, various authentication protocols have been developed. The following sections briefly describe basic authentication schemes and well established implementations, useful to better understand the later chapters of this thesis.

Generally, the notion of authentication in IT can be broken down to three sub-notions:

1. Data-origin authentication;

2. Entity authentication; and

3. Authenticated key establishment;

Data-origin authentication allows a receiver to verify whether a message is from a purported source. Entity authentication is a communication process by which an entity establishes a lively correspondence with a second entity whose claimed identity should meet what is sought by the first. By authenticated key establishment, one means the process of bootstrapping higher or application level secure communications. It is usually an extension of entity authentication and serves to exchange cryptographic keys used in the latter communication protocols.

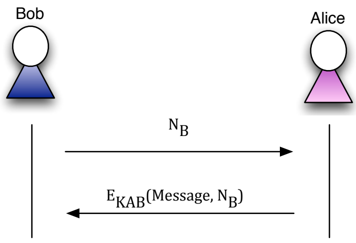


Figure 3.10: Basic Challenge-Response Mechanism based on a random Nonce and a shared Key

3.6.1 Basic Authentication Schemes

All the well established authentication protocols discussed later in this thesis are based on the following basic principles.

3.6.1.1 Challenge-Response Mechanisms

Let Bob be the entity which wants to verify the lively correspondence of Alice via the freshness of his own input. The usual form of this input could be a random number called a nonce, which generated by Bob is passed to Alice, also called *the challenge*. The answer of Alice is called *the response*. This response can be decrypted by a shared Key and has to include the challenge sent by Bob. Bob can easily decrypt the response and verify the correctness of the nonce, verifying the freshness of Alice's message because she should not be able to predict the random nonce used by Bob beforehand (see Figure 3.10).

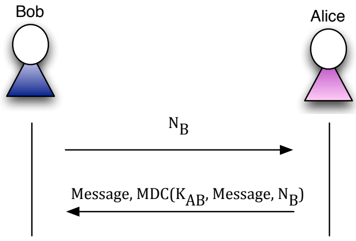


Figure 3.11: Challenge-Response Mechanism providing Data Integrity

As this protocol is very basic, it lacks serious mechanisms needed in a real-world scenario, e.g. a data integrity service in order to detect if somebody has modified the massage during its transmission. This service is usually provided by the use of a manipulation detection code MDC, also known as message digest code. A MDC is usually computed using a hash function (see Section 3.3).

Figure 3.11 illustrates such a data-integrity providing message-authentication-protocol. After Bob has send his random nonce to Alice, she computes a MDC using the shared Key K_{AB}, the message, and the nonce N_B and sends the plain text message plus the MDC to Bob. Bob now computes the MDC of the message

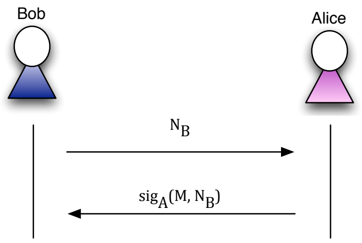


Figure 3.12: Challenge-Response Mechanism based on Digital Signatures

on his own as he also knows the shared Key and the nonce. If the MDCs are identical Bob can assume that he received the message correctly.

In scenarios where a-priori key exchange is not possible or inconvenient, challenge-response authentication can also be accomplished by using a public Key based signature (see Section 3.2.4), see Figure 3.12.

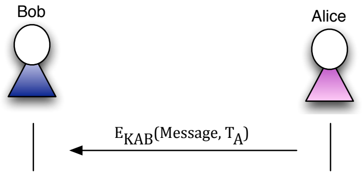


Figure 3.13: Basic Timestamp Mechanism

3.6.1.2 Timestamp Mechanisms

In a timestamp mechanism, Alice cryptographically integrates the current time into her message composition. This allows Bob to determine the exact moment of message creation. Analogous to the challenge-response protocols, three basic timestamp mechanisms can be defined. Figure 3.13 illustrates the basic protocol. Figure 3.14 show the extended version also providing message integrity.

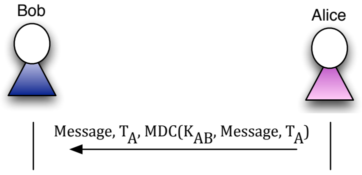


Figure 3.14: Timestamp Mechanism providing Data Integrity

Obviously, a timestamp mechanism can also be obtained by applying asymmetric cryptography techniques, see Figure 3.15.

All of the challenge-response and timestamp mechanisms described are standardized basic constructions for building authentication protocols. As the reader

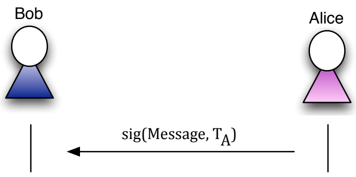


Figure 3.15: Timestamp Mechanism based on Digital Signatures

may have recognized, all of these protocols are only one-way and do not provide mutual authentication.

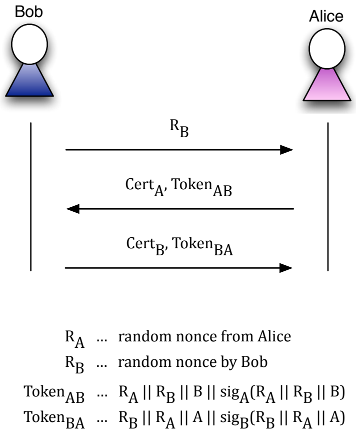

Bob
Alice
R_B
Cert_A, Token_AB
Cert_B, Token_BA
R_A ... random nonce from Alice
R_B ... random nonce by Bob
Token_AB ... R_A || R_B || B || sig_A(R_A || R_B || B)
Token_BA ... R_B || R_A || A || sig_B(R_B || R_A || A)

Figure 3.16: ISO Public Key Three-Pass Mutual Authentication Protocol

3.6.2 Mutual Authentication

The basic authentication mechanisms discussed in the last section only provide unilateral authentication, which means that only one of the two protocol participants is authenticated. One might consider that mutual authentication is simply twice unilateral authentication. This is not generally true.

A good example is the first version of the ISO Public Key Three-Pass Mutual Authentication Protocol which is based on public Key certificates for Bob and Alice, see Figure 3.16.

As the Token$_{BA}$ was originally a syntactic and symmetric mirror image from Token$_{AB}$, providing a context-sensitive link between the two tokens, a very effective attack discovered by Wiener is possible [118]. This is a so called man-in-the-middle attack where Malice is placed between the communicating parties, intercepting all messages and is in control of the channel.

Alice thinks that it is Bob who initiated the run and accepts Bob's identity; but Bob did not initiate the run, and is still awaiting for terminating a run

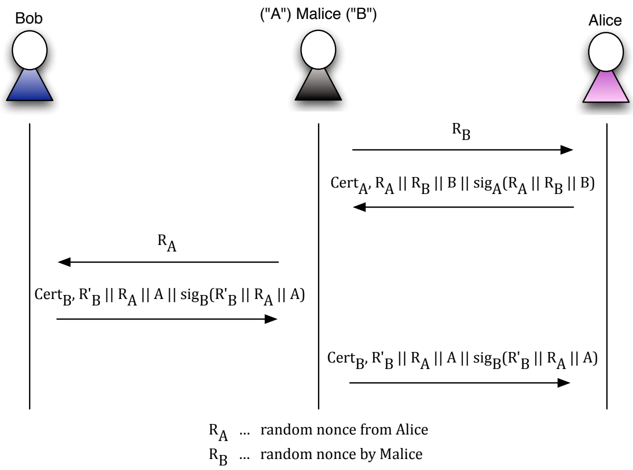

Bob
("A") Malice ("B")
Alice
R_B
$Cert_A$, R_A || R_B || B || $sig_A(R_A$ || R_B || B)
R_A
$Cert_B$, R'_B || R_A || A || $sig_B(R'_B$ || R_A || A)
$Cert_B$, R'_B || R_A || A || $sig_B(R'_B$ || R_A || A)
R_A ... random nonce from Alice
R_B ... random nonce by Malice

Figure 3.17: Wiener Attack on the ISO Public Key Three-Pass Mutual Authentication Protocol

started by Malice ("A").

This design flaw has been corrected in the current version of the protocol and Alice is explicitly instructed to maintain the state regarding Bob's nonce R_B until the current run terminates.

Many other protocols have been invented over the decades any many of them have been proven faulty. Some of them were revised and are still in use while some others were just too erroneous to be fixed and needed to be replaced.

3.6.3 Password Based Authentication

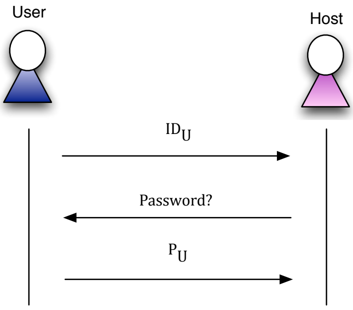


Figure 3.18: Naive Password Authentication Scheme

As many user-to-host authentication scenarios are based on passwords memorable by human brains, we will shortly discuss it here. A password can be seen as a rather small-size symmetric key which is essentially long-term valid. A user who wishes to use a service of a host must first be initialized and get a password issued. The host needs to keep track of all the issued passwords and store them linked to the belonging user's identity (ID_u, P_u). A naive protocol for password authentication is illustrated in Figure 3.18.

This protocol does not achieve any sense of entity authentication, not even unilateral authentication from U to H, because no part of the protocol includes a freshness identifier. Such a scheme offers several points for attacks such as the plaintext transmission of the password as well as the plaintext storage of the ID_u, P_u pairs on the host. A well established authentication protocol called *Challenge Handshake Authentication Protocol* (CHAP) [164] is illustrated by Figure 3.19

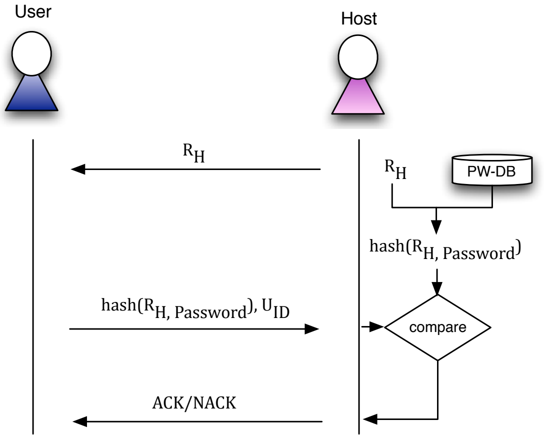


Figure 3.19: Challenge Handshake Authentication Protocol (CHAP)

Authentication is achieved in three phases and the password is never sent in plaintext but in a hash value based on the nonce R_H and the plaintext password. This process assures the liveliness of the correspondence preventing replay attacks. The security of CHAP relies mostly on the randomization of the nonce and the strength of the one-way hash function. Additionally, CHAP allows reauthentication in regular intervals.

For the sake of completeness, Password-Authenticated Key Agreement (PAKE) shall be mentioned here but not explained in detail. The interested reader shall be refered to [12].

3.6.4 Anonymous Authentication

Anonymity and Authentication seem to be two contradictory concepts [191]. To achieve authentication, a user needs to reveal her identity. This action obviously undermines anonymity. An attacker can use the user's ID to establish linkability and traceability in order to launch various attacks. A lot of research has been conducted on anonymous user authentication that is robust against privacy attacks while maintaining access security.

> *The basic idea for anonymous authentication is that through some cryptographic means, a legitimate users legitimacy of using the service can be verified, while at the same time the particular identity of the user is somehow concealed.* [191]

The following techniques have been developed in order provide anonymous authentication. They differ in the underlying cryptographic mechanism as well as in their properties concerning the trade-off between security and privacy.

While *Blind Signatures* and *Ring Signatures* provide almost perfect privacy, they suffer from degraded security protection as they do not provide source accountability resulting in the fact that anonymity is irrevocable. As a consequence, bad user behavior and insider attacks cannot be traced [191]. *Group Signatures* on the other hand allow revocation of anonymity in the case of a dispute by a designated group controller and perform therefor a trade-off between security and privacy.

3.6.4.1 Blind Signatures

Blind Signatures (BS) have been presented by Chaum in 1982 [28]. They are based on digital signatures (see Section 3.5), whereas the content of a message is disguised from the signer. BS can be based on various signature schemes like RSA or DSA based signatures. Creating a blind signature consists of the following steps [191]:

1. Alice blinds her message by using a *Blinding Function* f, typically combined with a random blinding factor.
2. Bob receives the message and signs it with a standard signing algorithm (SA) and sends the signature back to Alice.
3. Alice unblinds the signed message using a *Unblinding Function* g.

The algorithm is designed such that:

$$g(SA(f(m))) = SA(m) \tag{3.1}$$

This scheme can be used for anonymous access authentication. A legitimate user obtains a set of blind signatures from a service provider and unblinds them and can use them as authentication tokens. Blind signatures are hiding the user's true identity and can also provide non-linkability, preventing the signer from linking blinded messages to the unblinded version it may be called upon to verify. Blind signature schemes are of great use in systems where sender privacy is very important, such as electronic voting.

3.6.4.2 Ring Signatures

Ring Signatures have been presented by Rivest et al. in 2001 [154]. The signature scheme consists of the following steps [191]:

1. A set of possible signers associated with the Public Key of a standard signature scheme needs to available.
2. Bob, who wants to sign a message, declares an arbitrary set of possible signers out of the predefined set of possible signers including himself. He computes the signature of his message using only his Private Key and the Public Keys of the chosen signers.
3. Alice can verify the signature as valid from one of the declared signers, without revealing which signer actually produces the signature.

Ring signature schemes allow a legitimate user to hide her true identity in a crowd of other, selected, legitimate users. Unlinkability of multiple signatures by the same user is also provided.

3.6.4.3 Group Signatures

Group Signatures have been presented by Chaum and van Heyst in 1991 [29]. The key functionality, similar to Ring Signatures is, that one member can sign a message on behalf of the group without revealing her distinct identity. A verifier can only proof that a message was signed by a member of the group. In contrast to Ring Signatures, it is possible for a declared group controller to open the group signature and reveal the origin of the message. The group controller can for instance be incorporated by the service provider (SP) or a trusted-third-party (TTP).

The property of non-repudiation introduced by the capability of anonymity revocation allows to counteract bad user behavior and insider attacks, but poses a risk to the users privacy, especially if the SP or TTP is not highly trustworthy.

4
Security in IEEE 802.11 (WiFi)

4.1 Wired Equivalent Privacy Algorithm (WEP)

Right from the release of the first IEEE wireless LAN standard 802.11, a security mechanism called *wired equivalent privacy* was integrated. The primary goal of this mechanism was to protect the confidentiality of user data from eavesdropping. This should be gained by enforcing three properties [16]:

- Confidentiality
  Prevent casual eavesdropping by a non-authorized clients.
- Access control
  Only authorized clients should be allowed to join the network.
- Data integrity
  It should be recognized if data was altered during the transmission (see Definition 3.2)

All these properties are gained by using a secret key. The security of the WEP protocol only relies on the difficulty of discovering the secret key. If this difficulty only relies on the length of the key, and the only possibility of getting the key is an exhaustive search, the protocol is cryptographically secure.

WEP was initially designed for 40-bit keys with a resulting keyspace of $2^{40} = 1.099E9$. Using modern hardware it is no infeasible problem to discover the key with a brute-force approach in a reasonable time. As a consequence, the key length has been raised to 128-bit and an overall keyspace of $2^{128} = 3.402E38$. This extension renders an exhaustive key-search attack impossible, even with the most powerful hardware available [16].

Nevertheless, WEP owns some very critical design flaws that leave the standard practically futile. Although some feeble attempts to improve WEP were made like [67], the main vulnerabilities remained unchanged.

4.1.1 WEP Encryption / Decryption Process

Before taking a closer look at the encryption / decryption process, some terms need to be declared:

- Pseudo random-number generator (PRNG)
  Cryptography always needs some kind of random number source. In WEP, this task is done by the RC4 stream cipher (see Section 3.1.2.2). Seeded by some initialization value it creates a stream of *pseudo random-numbers*. But like all stream ciphers it will create the same keystream again if given the same seed.

- The initialization vector (IV)
  The IV is used to provide some diversion to the RC4 PRNG. It is 24-bits long and concatenated to the 40-bit secret key. In order to keep the PRNG from producing the same numbers for every packet, this IV needs to be changed as often as possible. There exist only $2^{24} = 16.777E3$ different IVs.

- The integrity check value (ICV)
  In order to provide data integrity, WEP uses the CRC32 algorithm. Before a packet gets encrypted, a *cyclic redundancy check value* with 32-bit length is computed and concatenated to the message. CRC32 is a linear function and does not provide any cryptographic security.

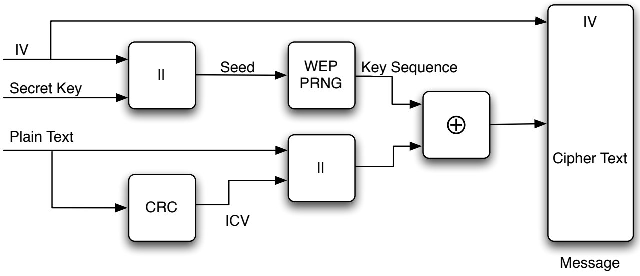


Figure 4.1: WEP Encryption Block Diagram

Figure 4.1 illustrates the message encryption process in WEP. The WEP-PRNG gets seeded by the secret key and some IVs and as the result it provides the so called *key sequence*. This key sequence is XORed with a concatenation of the plain text data and its CRC32 (ICV) value. Finally, the encrypted message is concatenated with the plaintext IV and transmitted [75].

The receiving client only needs to reverse the process to retrieve the plaintext massage, compute a CRC32 value of its on (ICV') and verify the integrity of the message by comparing the ICV and ICV'. The process is illustrated in Figure 4.2.

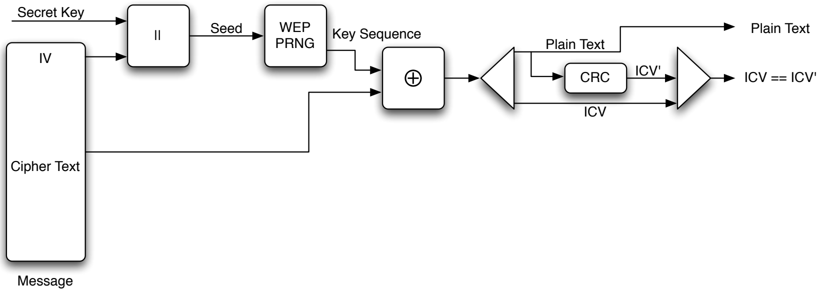


Figure 4.2: WEP Decryption Block Diagram

4.1.2 WEP Security Analysis

Several different attacks have been published during the last years. Most of them are based on the insecurity of the used RC4 stream-cipher. Although, RC4 was believed to be secure when it was integrated to WEP, it turned out to have some design flaws. While first attacks needed a high amount of collected data, more recent approaches like the attack of Andreas Klein [92] only need a relatively small number of transmitted packets. Klein's approach targeted flaws of the RC4 cipher. Erik Tews et al. [179] designed a process using Klein's approach and massive packet injection to generate enough traffic for breaking 128-bit WEP[1] in less than 60 seconds. Furthermore they do not need powerful special-purpose hardware, any contemporary personal-computer suffices. But not only RC4 may be exploited to break WEP. Also the very small number of IVs and their plaintext transmission offer a weak point. Another major vulnerability arises from the usage of the linear integrity check function CRC32. A detailed analysis of the components used in WEP is described in [16].

As a short conclusion it can be stated that WEP is highly insecure and should not be used if any other mechanism is available.

4.2 IEEE 802.11i (WPA, WPA2)

Since the publication of the WEP vulnerabilities and the upcoming of very effective attack implementations, the IEEE has begun the work on a replacement standard. On June 24th 2004, IEEE 802.11i was ratified in order to provide

[1] Due to the 24-bit plaintext IV concatenated to the key, the effective key-length is only 104-bit.

enhanced security for wireless networks. A formal verification of this standard may be found in [168]. The standard specifies two classes of security algorithms:

- Robust Security Network Association (RSNA)
- Pre Robust Security Network Association (Pre-RSNA)

Pre-RSNA consists of WEP and 802.11 entity authentication while RSNA implements two new data confidentiality protocols known as *Counter-Mode-CBC-MAC Protocol* (CCMP) and *Temporal Key Integrity Protocol* (TKIP) and the RSNA establishment procedure that includes the use of the IEEE 802.1X (see Appendix B) authentication and key management protocol [78].

TKIP is meant to bring more security to legacy hardware by using available RC4 implementations, while CCMP demands AES (see Section 3.1.3.2) compatible hardware. The WiFi-Alliance[2] certified TKIP compatible hardware under the name *Wi-Fi Protected Access* (WPA).

4.2.1 Wi-Fi Protected Access (WPA)

WPA may be seen as a short-time fix to secure legacy hardware based WLANs. TKIP is based on RC4 and includes the keyed hash-function Michael (see Section 3.4.2). TKIP can be described as a *"wrap"* around the existing WEP encryption / decryption to shield its worst vulnerabilities. Due to the inherited insecurities and flaws, it does not provide sufficient security in the long-term [78]. Figure 4.3 illustrates the TKIP encryption process while Table 4.1 explains the used notations.

Symbol	**Description**
TA	Transmitter address
TTAK	TKIP mixed transmitter address and key
TK	Temporal key
TSC	Sequence Number
IV	Initialization vector
DA	Destination address
SA	Source address
MSDU	MAC service data unit
MPDU	MAC protocol data unit

Table 4.1: TKIP Notations

The block *WEP encryption* corresponds with the WEP data encryption scheme presented in Figure 4.1. The TKIP extensions gain the security improvements only by modifying the input for the WEP encryption process. The

[2] Nonprofit international association certifying interoperability of wireless local area network products based on IEEE 802.11 specification. `http://www.wi-fi.org/`

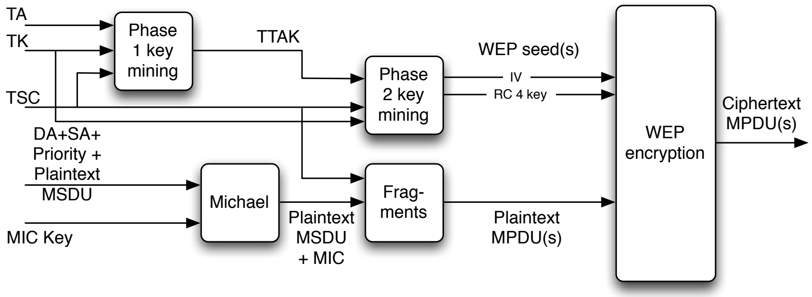


Figure 4.3: TKIP Encryption Block Diagram

most important change to classic WEP is that a new temporal key for each packet is used. This key is created by mixing together a base key, the MAC address of the transmitting station and a 48-bit serial number. The base key is newly created any time a station associates with the network and the mixing operation can be done with little computing power but provides a significant rise in cryptographic security. By adding the serial number into the key, it is assured that it will be different for each packet. The 48-bit space for the serial number prevents WEP-collision attacks and replay attacks as well. Together with IEEE 802.1X (see Appendix B), the secret keys are securely distributed between the participating STAs.

The second major vulnerability in WEP was the use of the linear CRC32 integrity check function. By implementing the Michael keyed hash-function, this problem was diminished but not solved as Michael also possesses some design flaws (see Section 3.4.2).

Figure 4.4 shows the TKIP decryption process that can be seen as a *"wrap"* around the WEP decryption scheme. It works exactly the other way round as the TKIP encryption process.

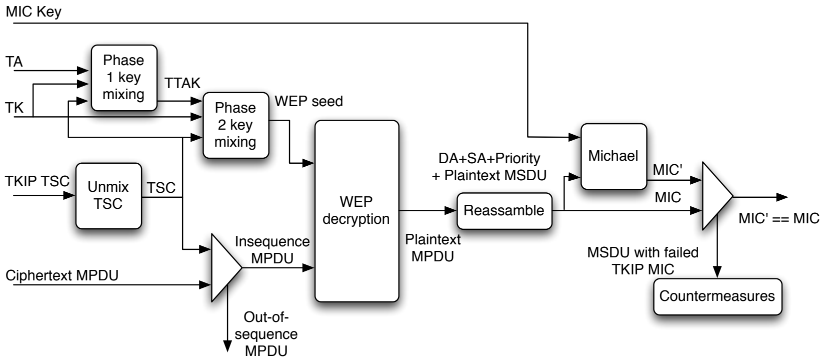


Figure 4.4: TKIP Encryption Block Diagram

4.2.2 TKIP Security Analysis

Due to the inherited WEP vulnerabilities and the fact that some parts of TKIP (like Michael) possesses known security relevant flaws, WPA cannot be assumed to be secure in the long run. However, it has always been a short-time fix for WEP and it does its job pretty well. But as mentioned before, it is just a fix and not a perfect solution. So, the use of WPA2 is preferred.

4.2.3 Wi-Fi Protected Access 2 (WPA2)

The Wi-Fi Alliance certified systems in compliance to IEEE 802.11i's *Robust Security Network Association* (RSNA) algorithm *Counter-Mode-CBC-MAC* (CCMP) under the name *Wi-Fi Protected Access 2* (WPA2). WPA2 may be seen as the first wireless network protocol that provides real cryptographic security. The only shortcoming is the need of new hardware because the WEP standard cipher RC4 has been replaced by the Advanced Encryption Standard (AES) (see Section 3.1.3.2) [78].

The use of AES brings some very significant advances. With one single 128-bit AES key one is able to encrypt all packets, eliminating the key management problems of WEP and TKIP. CCMP also provides an AES based *Message Integrity Code* (MIC) over the frame body and nearly the complete MAC header. Message confidentiality and integrity are both gained by the use of the same 128-bit AES key. Like in TKIP, CCMP also implements a 48-bit serial number (PN) to prevent replay attacks and PN collisions. Figure 4.5 illustrates the CCMP encryption process while Table 4.2 explains the used notations.

Symbol	**Description**
PN	Packet number
A2	MPDU address 2
AAD	Additional authentication data
TK	Temporal key
KeyId	Key identifier
MPDU	MAC protocol data unit

Table 4.2: CCMP Notations

The following steps explain the CCMP encryption of the payload of a plaintext MPDU and the encapsulation of the ciphertext in a MAC frame:

1. In order to obtain a new PN for each MPDU respectively for the temporal key creation, it is incremented after each packet.

2. The additional authentication data (AAD) is created from the MAC header and provided to the CCM encryption module.

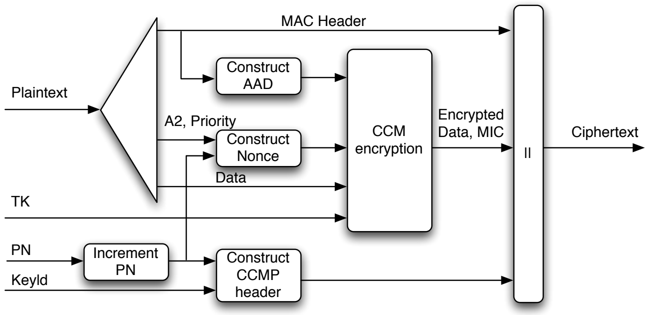


Figure 4.5: CCMP Encryption Block Diagram

3. The CCM Nonce is formed of the incremented PN, the A2 and the Priority field.
4. The key identifier (keyId) and the PN are placed in the CCMP header.
5. The TK, AAD, Nonce and MPDU data is taken by the CCM encryption to form the ciphertext and MIC. This step is also known as *CCM originator processing.*
6. The final step is to combine the results of the former steps to form the packet including the MPDU header, the CCMP header, the encrypted data and the MIC.

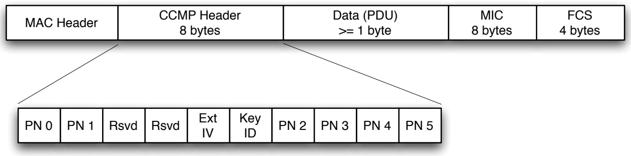


Figure 4.6: WPA2 Packet Format

Figure 4.6 shows the format of the WPA2 packet after CCMP encryption. The CCMP decryption process shown in Figure 4.7 works exactly the other way round as the encryption process.

Without the knowledge of the key, an adversary is not able to break data confidentiality or integrity. Even with a *known-plaintext-attack* (see Section 3), it is not possible to obtain any information about the key [68].

However, like any relevant cryptographic mechanism, CCMP relies on the privacy of the key. It is well known that *pre-shared key schemes* are very vulnerable. Therefore, IEEE 802.11i defines the RSNA establishment procedure to ensure strong mutual authentication by using the 802.1X protocol (see Appendix B). This mechanism is not only restricted to CCMP but may also be integrated in TKIP.

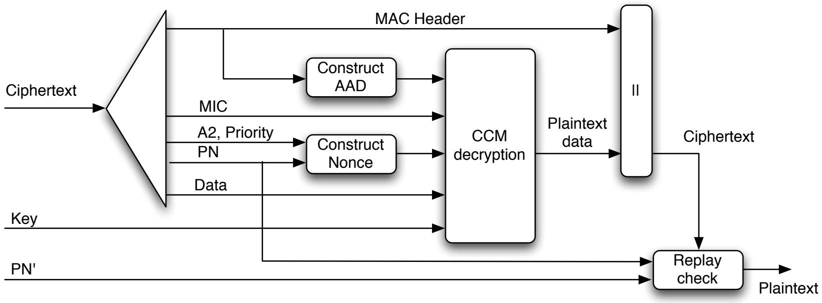


Figure 4.7: CCMP Encryption Block Diagram

4.2.4 CCMP Security Analysis

The usage of the AES introduced high levels of cryptographic security to wireless networks. Without the knowledge of the key, an adversary is not able to break CCMP data confidentiality or data integrity. Supported by the (proper) use of IEEE 802.1X (see Appendix B) the temporal keys may be exchanged securely between the communicating stations and it is not possible for an attacker to obtain a key. CCMP in combination with IEEE 802.1X is the best available security solution for wireless networks. The fact that CCMP does not protect MAC control- and management-frames leaves some inherited WEP vulnerabilities unaddressed.

4.3 Conclusion

Chapter 4 has shown that early versions of IEEE 802.11 standards were highly insecure due to the fact that the initial design bore no security mechanisms at all. Later amendments were error prone and could easily be circumvented by attackers. Only with the introduction of IEEE 802.11i (see Section 4.2) in 2004 adequate security levels could be guaranteed. But, unfortunately, the configuration of available implementations is often complex and not usable by standard consumers.

5 Security in IEEE 802.15.1 (Bluetooth)

Bluetooth is an open standard for short-range radio frequency communication. It has been designed to easily establish wireless personal area networks (WPAN), often referred to as ad-hoc or peer-to-peer networks. Initially integrated into personal computers and mobile phones, Bluetooth can nowadays be found in a wide variety of devices such as headphones, portable music-players or even in cars [159].

There have been several versions of Bluetooth, with the most recent released definition being Bluetooth 4.0. The released versions differ greatly in bandwidth and the provided security. Since most of the available devices are still implemented according to Bluetooth 2.1 and earlier, this chapter will focus on their analysis [159].

Like WiFi, Bluetooth (BT) operates in the unlicensed 2.4 GHz ISM frequency band. Therefore it is primarily vulnerable to all physical layer Denial of Service (DoS) attacks like channel jamming. As BT implements channel-hopping at a very high rate, changing frequencies about 3200 times per second, it shows some resistance against these DoS attacks [80]. The BT standard specifies the following three security services [80]:

- **Authentication:** This service authenticates the communicating devices. User authentication is not natively provided by Bluetooth.

- **Confidentiality:** Ensuring that not only authorized devices can access transmitted data, and therefore prevents all kinds of eavesdropping.

- **Authorization:** As bluetooth allows the control connected resources (printers, headphones, etc.), this service assures a device's authorization before allowing it to do so.

Other security services such as *non-repudiation* are not provided by BT [159].

5.1 Bluetooth Security Modes

Cumulatively, the BT versions up to 2.1 define four modes of security. Each of these version support some of these modes but none of them supports all four.

5.1.1 Security Mode 1

This mode is insecure. Authentication and encryption are bypassed leaving this mode without any security measures at all. Mode 1 is only supported in BT 2.0 + EDR (Enhanced Data Rates) and earlier versions [159].

5.1.2 Security Mode 2 (service-level enforced)

Mode 2 is designed as a *service-level enforced security-mode*. It is possible to grant access to some services without providing access to others. It introduces the *notion of authorization*, the process of deciding if a specific device is allowed to have access to a specific service. A centralized security manager (as defined in the BT architecture) controls access to specific services and devices. The security measures take place after the physical link has been established. Security mode 2 is supported by all Bluetooth devices [159].

5.1.3 Security Mode 3 (link-level enforced)

This mode mandates authentication and encryption for all connections to and from the device. All security measures take place before the physical link is fully established. Security mode 3 is only supported in Bluetooth 2.0 + EDR and earlier devices [159].

5.1.4 Security Mode 4 (service-level enforced)

Similar to security mode 2, this mode is enforced on the service level, after the physical link has been established.The pairing mechanism uses Elliptic Curve Diffie Hellman (ECDH) techniques. Services supported by mode 4 must be classified as one of the following:

- Authenticated Link Key required
- Unauthenticated Link Key required
- No security required.

Security mode 4 is mandatory for communication between devices in compliance to Bluetooth 2.1 + EDR or newer versions [159].

5.2 Bluetooth Key Management

The various defined Bluetooth security mechanisms require several different keys. Depending on the used security mode, some of them are used to establish the connection and derive a Link Key between two devices. This Link Key can be semi-permanent or temporary. A semi-permanent key might be stored in the nonvolatile memory of a device and therefore used for multiple sessions, while the lifetime of a temporary key is limited to the current session [80].

- $\mathbf{K}_{AB}$ - Combination Key
  The Combination Key is derived from information in both connecting devices A and B. It therefore depends on two devices. K_{AB} is derived for each new combination of two devices.

- $\mathbf{K}_{A}$ - Unit Key
  Contrary to K_{AB}, K_A is only derived from the information of a single device. It is generated at the installation of the device and usually very rarely changed.

- $\mathbf{K}_{master}$ - Master Key
  In a point-to-multipoint (Broadcast or Multicast) scenario, a common encryption key (K_{master}) may be used to replace the current Link Keys.

- $\mathbf{K}_{init}$ - Initialization Key
  The Initialization Key should be used to as the Link Key during the initialization process, when no combination or unit keys have been exchanged yet. It protects the transfer of initial parameters. In security modes 2 and 3, this key is derived from the triple of a random number, a PIN code and the device's hardware address.

- $\mathbf{K}_{link}$ - Link Key
  The Link Key is usually a 128-bit random number which is shared between two or more parties as the basis for all cryptographic transactions. It is used in the authentication routine and to derive the Encryption Key K_c.

- $\mathbf{K}_{c}$ - Encryption Key
  The Encryption Key is used for encrypting all transmissions during a session. It is usually derived from the Link Key K_{link}.

5.2.1 Link Key generation in Security Mode 2 and 3

As the Link Key must be distributed among the communicating devices in order to allow the authentication procedure, it has to be created during the initialization phase. This procedure is also called pairing and consist of the following five steps:

1. Generation of an Initialization Key

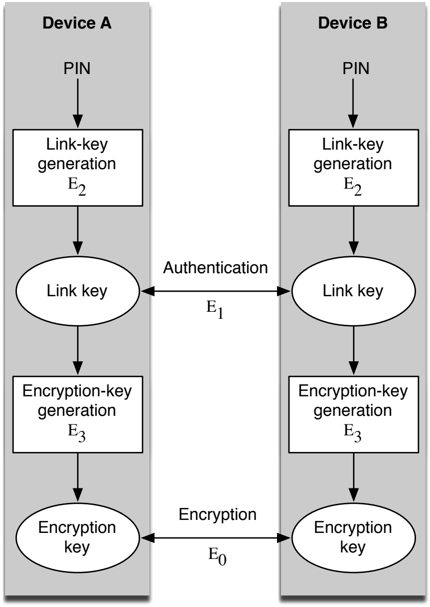

Device A
Device B
PIN
PIN
Link-key generation E_2
Link-key generation E_2
Link key
Authentication E_1
Link key
Encryption-key generation E_3
Encryption-key generation E_3
Encryption key
Encryption E_0
Encryption key

Figure 5.1: Overview of the Bluetooth Key Generation Routines for Security Modes 2 and 3 (simplified) [91]

2. Generation of a Link Key
3. Link Key exchange
4. Authentication
5. Generation of encryption keys (optional)

Bluetooth standards define a number of generic cryptographic building blocks called E_0, E_1, E_2 and E_3 [80].

- E_0 - a stream cipher function
- E_1 - the authentication function
- E_2 - the Link Key generation function
- E_3 - the Encryption Key generation function

These building blocks are mainly based on the block cipher SAFER+ (see Section 3.1.3.3) and Linear Feedback Shift Registers (LFSR) (see Section 3.1.2.1). Figure 5.1 provides an overview of the Bluetooth key generation process and the cryptographic building blocks used for security modes 2 and 3.

5.2.2 Secure Simple Pairing (SSP) in Security Mode 4

SSP was introduced in Bluetooth 2.1 + EDR for the use with security mode 4. It simplifies the pairing process by providing four flexible association models [159]:

- **Numeric Comparison**
  During pairing the user is shown a six digit number on both devices allowing her to enter a "yes" or "no" response if the numbers do match.
- **Passkey Entry**
  One of the devices shows a six digit number which the user has to enter on the second device in order to allow pairing.
- **Just works**
  Is designed for the use of devices without displays or an input possibility. Keys are exchanged in plaintext leaving a vulnerability for man-in-the-middle attacks.
- **Out of Band (OOB)**
  OOB is an extension that allows devices with additional wireless techniques like near field communication (NFC), to use them for device discovery and cryptographic value exchange. Devices can therefore be paired by simply "tapping" one device against the other.

Figure 5.2 provides an overview of the Bluetooth Secure Simple Pairing process for security mode 4.

5.3 Authentication in Bluetooth

Authentication in Bluetooth is based on a challenge-response scheme as shown in Figure 5.3. The authentication procedure takes the following steps [159]:

1. The verifier transmits a 128-bit random challenge (AU_RAND) to the claimant.
2. The claimant applies the E_1 authentication function using his unique 48-bit Bluetooth device address (BD_ADDR_A), the Link Key and AU_RAND as inputs. The verifier performs the same procedure. The 32 most significant bits of the E_1 output (SRES) are used for the authentication output while the remaining 96 bits (Authenticated Ciphering Offset - ACO) will be used later to create the Bluetooth encryption key.
3. The claimant returns the SRES to the verifier.
4. The verifier compares the received SRES with its own outcome of the E_1 algorithm.
5. If the two SRES values are equal, the authentication process is successful in one direction. To achieve mutual authentication, this process needs to be repeated with switched roles.

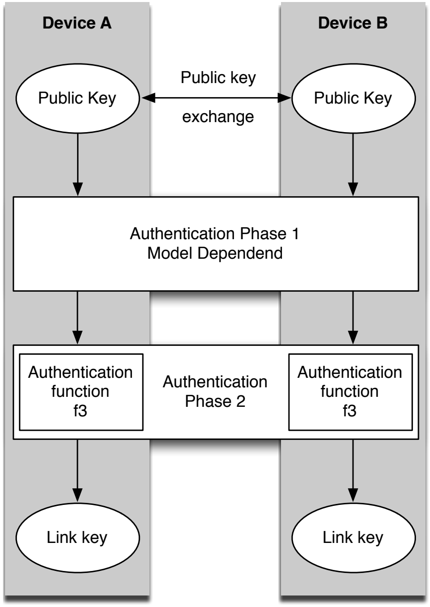


Figure 5.2: Overview of the Bluetooth Secure Simple Pairing Routines for Security Mode 4

5.4 Bluetooth Encryption Concept

As already mentioned, encryption is not mandatory for all bluetooth connections and devices. Bluetooth defines three encryption modes [159]:

1. **Encryption Mode 1**
   No encryption is performed at all.
2. **Encryption Mode 2**
   Broadcast traffic is not encrypted. Only individually traffic is encrypted using keys based on individual link keys.
3. **Encryption Mode 3**
   All traffic is encrypted using an encryption key based on the master Link Key.

Figure 5.4 illustrates the Bluetooth encryption procedure as implemented in BT versions 2.0 + EDR and earlier. Newer versions differ in the key derivation (see Section 5.2).

The key stream K_{cipher} is generated by the stream cipher function E_0, which is based on the block cipher SAFER+ (see Section 3.1.3.3). This key stream is XOR'ed with the data and transmitted to the receiver. According to the symmetric cryptography paradigm (see Section 3.1), decryption is achieved by applying the same cipher key as used for encryption.

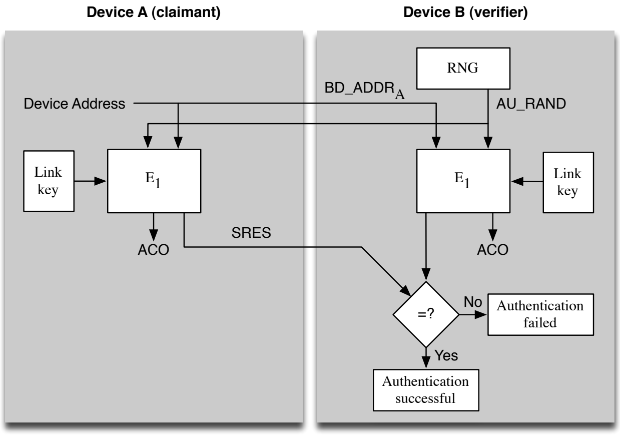


Figure 5.3: Bluetooth Authentication [159]

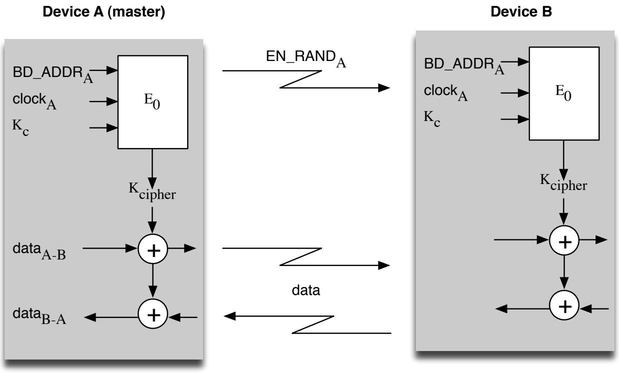


Figure 5.4: Functional Description of the Bluetooth Encryption Procedure [80]

5.5 Bluetooth Trust and Service Levels

Additionally to the four security modes, Bluetooth allows two *trust levels* and three *service security levels.* Trust levels are *trusted* and *untrusted.* Trusted devices have full access to all services provided by the connected devices while untrusted devices only receive restricted access [159]. Service Security Levels allow to configure and alter the requirements for authorization, authentication and encryption independently. Bluetooth Service Security Levels [159]:

- **Service Level 1**
  Authorization and authentication are required. Trusted devices are allowed to automatically connect to all services. Untrusted devices need manual authorization for all services.

- **Service Level 2**
  This level requires authentication only. Access to services is granted only after the authentication procedure.

- **Service Level 3**
  Access is granted automatically and to all devices with no authentication required.

Trust and service levels allow the definition of policies to set trust relationships and may also be used to initiate user-based authentication. Bluetooth core protocols usually only provide device authentication.

5.6 Analysis of Security Measures in Bluetooth

Security matters differ very strongly between the single versions of Bluetooth. Bluetooth security always depends on the weakest BT device in the communication chain. As legacy-standard devices are still widespread this section will take their vulnerabilities in account as well as of state-of-the-art implementations. Later on, this section lists and shortly describes common Bluetooth related attacks.

5.6.1 Bluetooth Version related Vulnerabilities

5.6.1.1 Versions before Bluetooth 1.2

Unit Key and Link Key vulnerability
The Unit Key is reusable and becomes public after once used. This could be circumvented by using temporary broadcast keys, derived from the Unit Key which is kept secret. The same problem occurs if a corrupt or malicious device that has communicated with either device of a new communication pair, wants to eavesdrop on this communication. The Link Key stays the same for the same device. Various kinds of replay attacks are possible.

5.6.1.2 Versions before Bluetooth 2.1

This section presents vulnerabilities in Bluetooth standards prior to version 2.1 + EDR. As newer versions, namely 3.0 and 4.0, are still in the process of being standardized, no vulnerabilities have been published yet.

- **Short PIN codes are allowed**
  Short PIN codes can easily be guessed and all derived Link end Encryption keys compromised.

- **No PIN management**
  It is hardly possible to use adequate PINs in an enterprise setting as no PIN management capabilities are defined.

- **Keystream reoccurrence**
  The keystream (as created in Figure 5.4) repeats after 23.3 hours due to a clock overrun allowing various cryptographic attacks on the ciphertext.

5.6.1.3 Regarding All Versions

- **No User Authentication**
  By default, no user authentication is defined by BT standards. Application-level security and authentication needs to be added.

- **E_0 Stream Cipher Function is weak (SAFER+)**
  The used stream cipher function SAFER+ has been subject to vulnerabilities and needs to be replaced by a more robust solution to prevent cryptographic attacks. (see Section 3.1.3.3)

- **One Way Device Authentication**
  One-way challenge-response authentication can easily be exploited by man-in-the-middle (MITM) attacks. Mutual authentication should be enforced.

- **No End-to-End Encryption**
  No end-to-end encryption is provided in multi-hop scenarios. Transmissions are only encrypted between two nodes. Higher level solutions need to be deployed.

- **Limited Security Services**
  Services such as nonrepudiation are not defined by BT standards. They can only be implemented in an overlay fashion.

5.6.2 Bluetooth related Attacks

BT attacks are best classified using the following definitions [42]:

- **Surveillance**
  Collecting information about a BT device like the provided services, device address, location and so on. No direct adverse effects to the target caused. Location tracking of users is a great potential threat.

- **Range extension**
  The range of BT devices is limited by their device class between 1 and 100 meters. Extending the transmission range of BT devices is in general against authority regulations. Attackers can use strong directional antennas to conduct BT all kinds related attacks from a great distance, even up to some kilometers.

- **Obfuscation**
  Attackers can forge their Bluetooth identities by spoofing the 48-bit device address, the device name and the device class. This can be used to obfuscate attacks.

- **Fuzzer**
  Bluetooth stack implementations are sometimes not very robust against nonstandard inputs. An attacker can create malformed data packets causing buffer-overflows or system failures at the target devices.

- **Sniffing**
  Attackers can capture all BT traffic due to its open space propagation nature in order to launch offline cryptographic attacks to recover the plaintext.

- **Denial of Service (DoS)**
  DoS attacks can target the media (e.g. channel jamming) or the devices (e.g. the energy consumption in mobile devices).

- **Malware**
  Malware is a form of malicious software that carries out various attacks such as data mining or password theft on the targeted devices. This malware can be self-replicating in the form of worms.

- **Unauthorized direct data access (UDDA)**
  UDDA attacks can gather all kinds of private data, and further on use all resources of the attacked device. They can e.g. place phone calls or send text messages if the attacked device provides these services.

- **Man in the middle (MITM)**
  An attacker could place himself between two communicating devices, relaying all their communication to each other. If the attacker is placed between a computer and a printer it can obtain all traffic sent to the printer. This attack mainly concerns the *Just Works* authentication method.

5.7 Conclusion

The deployment of Bluetooth poses a serious security risk especially for enterprise settings. Even though BT can be regarded secure if all devices are configured properly, the probability of the occurrence of vulnerabilities is too high to allow its implementation in security-critical systems.

There exist some guidelines for securing Bluetooth such as [159] or [42]. Further information of the security of Bluetooth can be obtained from the following references: [141, 158, 162, 165].

6 Security in IEEE 802.16 (WiMAX)

Whereas WiFi and Bluetooth have been around for many years now, WiMAX is a young and emerging standard. For a better understanding of its principles, the following section will provide a short introduction.

6.1 WiMAX at a Glance

WiMAX stands for *worldwide interoperability for microwave access* and is a certification mark for the IEEE 802.16 standard family. It was designed for point-to-multipoint broadband wireless access. Its original main purpose was not to connect end-users with an access-point, but to interconnect access-points with each other. It could be seen as a kind of wireless backbone network and states an alternative to cable and DSL to provide broadband access to groups of end-users [126]. In recent years, as a response to customer and industry needs, WiMAX was extended to support connections between mobile end-nodes and base-stations.

WiMAX devices are usually organized in a mesh network (see Figure 6.1). A mesh network consist of two different kinds of nodes, which perform the necessary routing tasks: *mesh routers* and *mesh users*.

The fact that mesh users and mesh routers are able to perform the same operations and therefore may switch roles, renders mesh networks very powerful and flexible. Mesh networks are usually not limited to IEEE 802.16. They are designed to integrate other standards as IEEE 802.11 or IEEE 802.15.1 and form so called *metropolitan* and *enterprise networks*. The most significant benefits of mesh networks are:

- **Scalability**

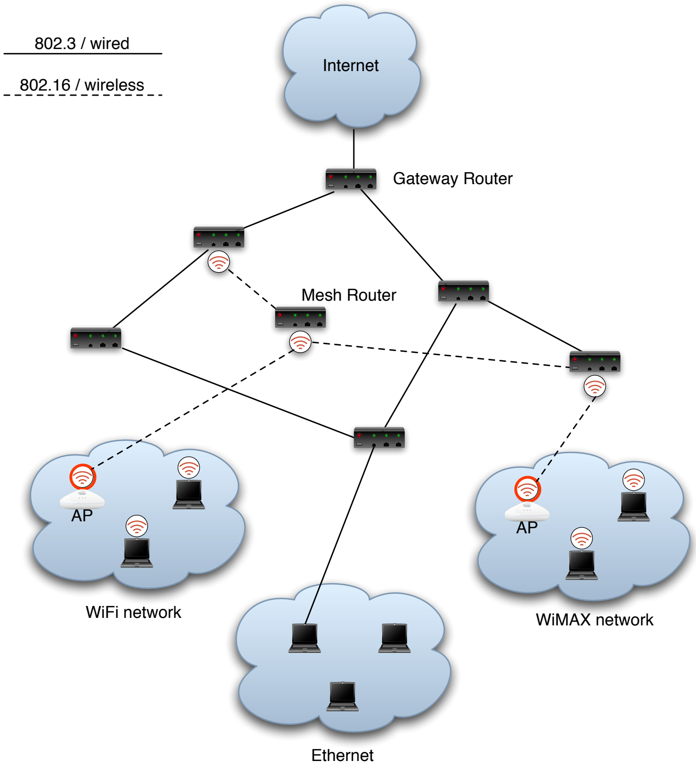


Figure 6.1: Possible WiMAX Network Setup

The whole infrastructure is designed to be scalable as the need for resources might increase over time.

- **Ad hoc networking support**
  Devices are able to join and leave the network all the time. Routing can be self organizing.
- **Mobility support of end nodes**
  End node roaming is supported.
- **Connectivity to wired infrastructure**
  Heterogeneous networks may be interconnected by mesh routers.

The IEEE 802.16 standard uses the frequency range from 10 GHz up to 66GHz which states another significant difference to WiFi, which is using the 2.4 GHz band. WiMAX is able to cover up to 50 km of connectivity services between nodes without a direct line of sight, although the practically used distance is about 5 to 10 km. The data rate provided is up to 70 Mb/s which is enough to serve about 60 T-1-type links simultaneously [126].

Probably the most significant differences between WiMAX and WiFi standards may be found at the MAC layer. WiMAX offers a remarkable improvement as it defines a MAC layer that supports multiple physical-layer specifications. This renders WiMAX as a great framework for wireless broadband communications.

The MAC layer is a so called *scheduling* MAC layer where devices need to compete for the initial entrance to the network. Once joined the network, the base station dedicates a time slot to the device which can be variable but must not be used by any other user. This method offers better bandwidth efficiency and allows the base station to offer QoS by balancing the assignments of connected devices [126]. Some of the IEEE 802.16 MAC layer properties to support mesh networking are:

- It is designed to support multi-hop communication
- It is designed for multipoint-to-multipoint communication
- Self-organizing features are provided

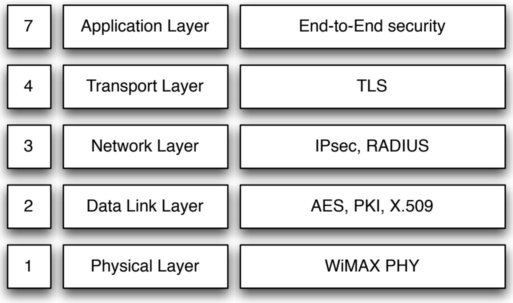
WiMAX was initially released as IEEE 802.16-2001 in April 2002 [76]. After some amendments, IEEE 802.16-2004, also known as IEEE 802.16d [77], was released and fixed many errors and initial security vulnerabilities. In 2005, IEEE 802.16e-2005 [81] was released, enabling mobility support in WiMAX networks and fixing further security issues. IEEE 802.16j [82] is the latest major release in this standard family. It mainly extends mobile support and does mot introduce new security functionality.

6.2 Overview of IEEE 802.16 Security

Lessons learned from weaknesses in WiFi security have been incorporated in WiMAX right from the beginning of its design. WiMAX provides right out-of-the-box the following security services [5]:

- Privacy - Protect from eavesdropping
- Data integrity - Protect data from being tampered in transit
- Authentication - At the user and the device level
- Authorization - At the service level

As Figure 6.2 illustrates, IEEE 802.16 allows the incorporation of security functions at various network layers [5]. Right from the beginning of the WiMAX design process, a special layer, as part of the MAC layer has been introduced. The so called *security sublayer* should provide all necessary security functionality, securing all communication on the higher layers. (see Figure 6.3). As this chapter is about security in wireless networks, it will focus on security measures which are part of the IEEE 802.16 security sublayer.



7	Application Layer	End-to-End security
4	Transport Layer	TLS
3	Network Layer	IPsec, RADIUS
2	Data Link Layer	AES, PKI, X.509
1	Physical Layer	WiMAX PHY

Figure 6.2: WiMAX supported Security Functions at various Network Layers

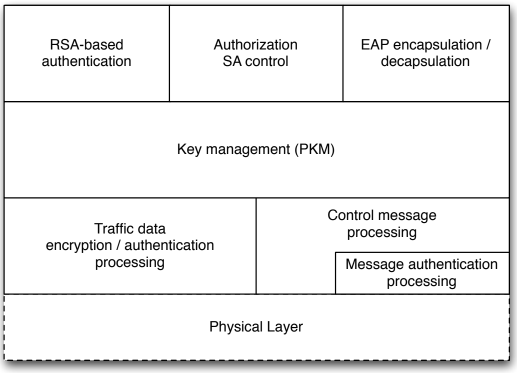

RSA-based authentication
Authorization SA control
EAP encapsulation / decapsulation
Key management (PKM)
Traffic data encryption / authentication processing
Control message processing
Message authentication processing
Physical Layer

Figure 6.3: WiMAX Security Sublayer

6.2.1 Authentication and Authorization in WiMAX

Authentication and Authorization in WiMAX is completely implemented at the security sublayer. It is achieved using the so called *public key interchange protocol* that ensures authentication and establishment of the cryptographic keys. As mentioned in Section 3.2.4, a key pair, consisting of a private and a public key is needed for each party in the public key interchange scheme.

Key exchange and key management in general had several vulnerabilities in the original IEEE 802.16 standard. As IEEE 802.16e-2005 corrected most of these problems, this section will focus on this state-of-the-art standard.

IEEE 802.16e-2005 defines two *Privacy Key Management* (PKM) protocols, PKMv1 and an enhanced version PKMv2. They basically allow three types of authentication (see Figure 6.3):

- RSA based authentication - based on X.509 certificates (see Section 3.2.4) and RSA encryption
- Extensible Authentication Protocol (EAP)

- RSA based authentication followed by EAP authentication

All security information between communicating parties are part of so called *Security Associations* (SA). SAs are a set of parameters used for authentication, authorization and encryption. The shared information depends on the chosen cryptographic suite and usually includes the encryption keys and *initialization vectors* (IV) needed for the encryption process. Three different types of SAs are defined by IEEE 802.16e-2005 [81]:

- **Primary SA**
  Each subscriber station (SS) establishes a primary SA during its initialization process.
- **Static SA**
  They are provisioned within each base station (BS).
- **Dynamic SA**
  They are established and eliminated, on the fly, in response to the initiation and termination of the specific service flows.

Each SS establishes an exclusive Primary SA with its BS and dynamic SAs for each new service flow. The lifetime of SAs is limited by the standard. Each new SA has to be newly authorized before its establishment.

Key Name	**Description**	**Derived from**
AK Authorization Key	Shared private key (between SS and BS)	not clearly defined by the standard
KEK Key Encryption Key	Key used for encrypting TEKs in the key exchange	derived from the AK
TEK Traffic Encryption Key	Used for encrypting all end to end traffic	derived from the AK
PK Public Key	public key of the BS and the SS respectively	stored in the X.509 certificate of the BS and SS respectively

Table 6.1: Overview of Cryptographic Keys used in WiMAX (excerpt)

The PKM establishes a shared key called *Authorization Key* (AK) between the subscriber (SS) and the base station (BS). After this shared AK is established between the parties, a *Key Encryption Key* (KEK) is derived from it. This KEK is then used to encrypt subsequent PKM exchanges of *Traffic Encryption Keys* (TEK). All payload encryption is based on TEKs. Table 6.1 provides an excerpt of the cryptographic keys used in WiMAX.

Figure 6.4 illustrates the authentication and authorization protocol as originally integrated in IEEE 802.16-2001. The SS uses the first message to push its manufacturer X.509 certificate (see Section 3.2.4.2) to the BS allowing it to

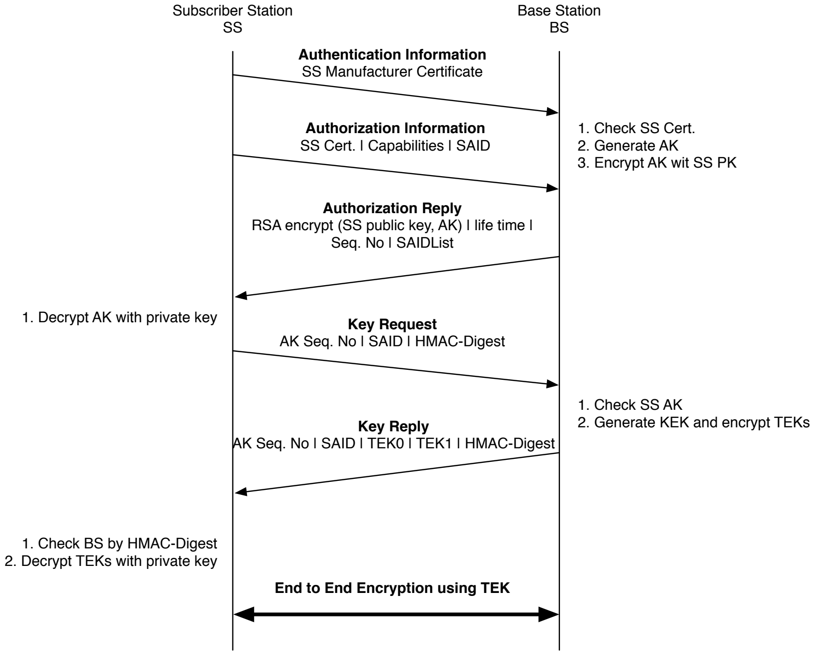

Subscriber Station
SS
Base Station
BS
Authentication Information
SS Manufacturer Certificate
Authorization Information
SS Cert. I Capabilities I SAID
1. Check SS Cert.
2. Generate AK
3. Encrypt AK wit SS PK
Authorization Reply
RSA encrypt (SS public key, AK) I life time I
Seq. No I SAIDList
1. Decrypt AK with private key
Key Request
AK Seq. No I SAID I HMAC-Digest
1. Check SS AK
2. Generate KEK and encrypt TEKs
Key Reply
AK Seq. No I SAID I TEK0 I TEK1 I HMAC-Digest
1. Check BS by HMAC-Digest
2. Decrypt TEKs with private key
End to End Encryption using TEK

Figure 6.4: WiMAX Privacy Key Management Protocol (PKM) v1

validate its identity via a Certification Authority (CA). The second message is send right after the first and includes the SS's X.509 certificate its security capabilities and the ID of the Primary Security Association (SAID). By using the SS certificates public key (PK), the BS is able to construct the Authorization Reply including the Authorization Key (AK). The following messages are to establish the keys needed for encryption [5]. PKMv1 lacks mutual authentication as only the SS provides a certificate. Problems arising due to this fact are discussed in the security analysis of WiMAX later in this chapter.

IEEE 802.16e-2005 introduced an improved version of the Privacy Key Management Protocol called PKMv2, targeted to provide mutual authentication based on X.509 certificates and to correct the vulnerabilities of PKMv1. As illustrated in Figure 6.5, the *Authorization Reply* is extended by the BS's certificate an digital signature and random seeds from the SS and BS respectively. These additional parameters aim to harden the protocol against replay and man-in-the-middle attacks [88] .

PKMv2 also allows the usage of *Cipher based Message Authentication Codes* (CMAC) instead of *Hashed Message Authentication Codes* (HMAC) [115]. (see Section 3.3)

Additionally to RSA based authentication, WiMAX allows the use of the *Extensible Authentication Protocol* (EAP). The EAP method can use a particular kind of credential, such as an X.509 certificate in the case of EAP-TLS

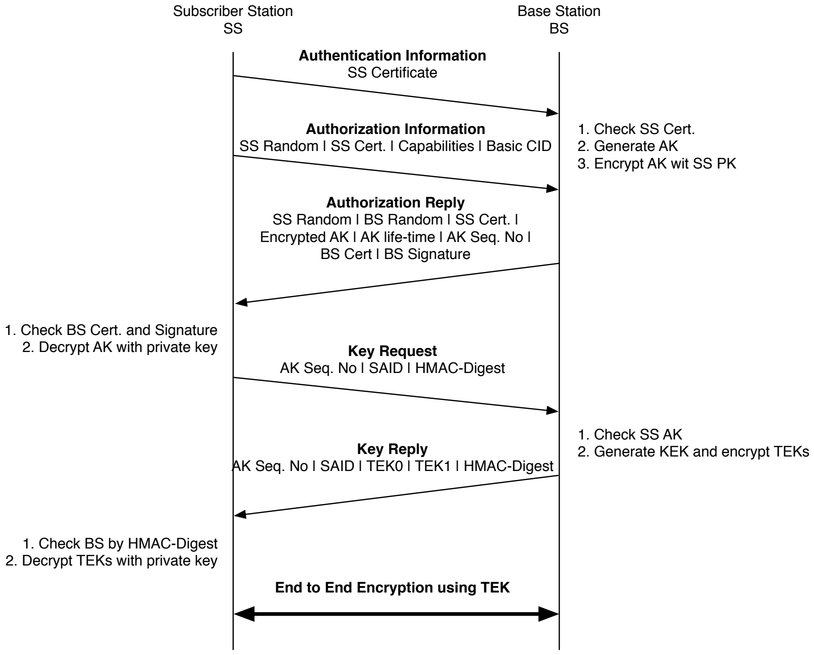

Subscriber Station
SS
Base Station
BS
Authentication Information
SS Certificate
Authorization Information
SS Random I SS Cert. I Capabilities I Basic CID
1. Check SS Cert.
2. Generate AK
3. Encrypt AK wit SS PK
Authorization Reply
SS Random I BS Random I SS Cert. I
Encrypted AK I AK life-time I AK Seq. No I
BS Cert I BS Signature
1. Check BS Cert. and Signature
2. Decrypt AK with private key
Key Request
AK Seq. No I SAID I HMAC-Digest
1. Check SS AK
2. Generate KEK and encrypt TEKs
Key Reply
AK Seq. No I SAID I TEK0 I TEK1 I HMAC-Digest
1. Check BS by HMAC-Digest
2. Decrypt TEKs with private key
End to End Encryption using TEK

Figure 6.5: WiMAX Privacy Key Management Protocol (PKM) v2 [3]

or a *Subscriber Identity Module* (SIM card) in the case of EAP-SIM [81]. The definition of the EAP protocol is outside of the WiMAX standard and can be obtained from RFC 4017 [172].

6.2.2 Encryption in WiMAX

The initial standard defined encryption based on the *Data Encryption Standard* (DES) with a default key length of 56 bit. Figure 6.6 illustrates the encryption process of IEEE 802.16-2001.

DES is operated in *Cipher Block Chaining* (CBC) mode using the TEK as encryption key, an *initialization vector* derived from the SA's IV and the value of a field in the PHY header. Both of these last named values are predictable.

IEEE 802.16e-2005 introduced the usage of the *Advanced Encryption Standard* (AES) (see Section 3.1.3.2) in *Counter mode with CBC-Message Authentication Code* (CCM) mode for authentication and AES in *Counter* mode (CTR) for encryption purposes (see Figure 6.7). AES-CCM and AES-CTR are faster in their operation than 3DES and the security increase is significant.

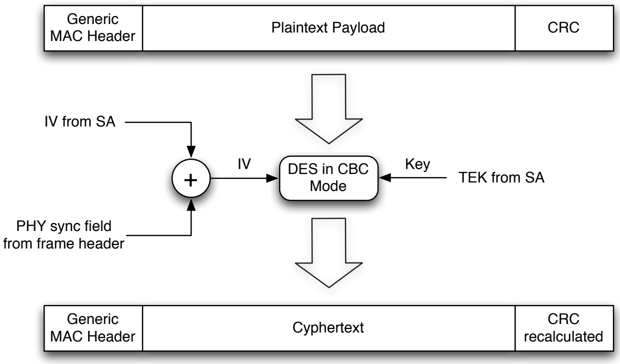


Figure 6.6: IEEE 802.16-2001 Encryption Process [88]

6.3 Analysis of IEEE 802.16 Security

WiMAX was originally developed to address the *last mile* problem. The IEEE 802.16 Working Group tried to avoid design mistakes like done by defining WiFi standards by incorporating a pre-existing standard, *Data Over Cable Service Interface Specifications* (DOCSIS). DOCSIS was designed to solve the last mile problem for wired connections. This fact allows the assumption, that it might not work in wireless networks without problems. The result was, that IEEE 802.16-2001 failed to properly protect the wireless links [88]. The major security flaws of the initial standard are the following [88]:

- Only data transport is encrypted, leaving management frames vulnerable for attacks.
- The focus on the encryption of the packet payload left the authorization protocol neglected and thus vulnerable.
- The standard allowed one-way authentication leaving many loop-holes for replay attacks.
- Several security related parts of the standard such as key generation, lacked explicit definitions and could therefore be implemented imperfect by hardware vendors.
- Triple DES was used in CBC mode. While DES itself is not unbreakable anymore, very short keys as used in IEEE 802.16-2001 are a serious vulnerability. Further on, the encryption process (see Figure 6.6) exhibits a severe error by using predictable initialization vectors (IV). CBC mode would require a random IV to secure the scheme [156].
- Vulnerabilities introduced by the weak encryption scheme and lacking mutual authentication allow several attacks on the privacy and integrity

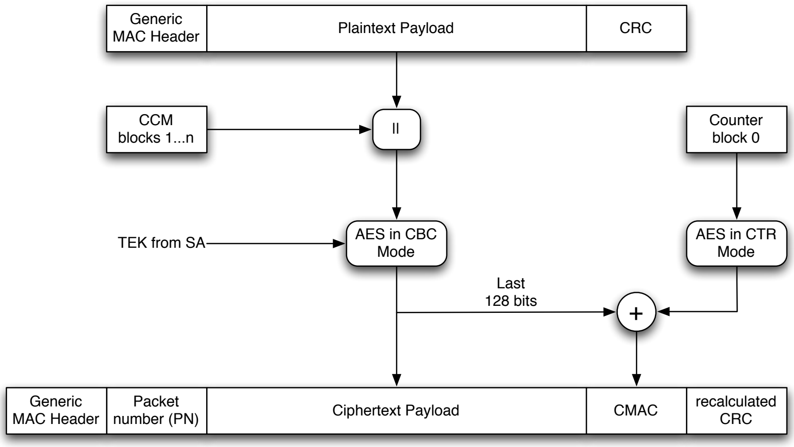


Figure 6.7: IEEE 802.16e-2005 Encryption Process based on AES [115]

of the communications. It furthermore leaves the topology of the network exposed to mesh-network attacks. The interested reader is referred to [11, 88, 116, 195, 198].

IEEE 802.16e-2005 corrects these errors described above by incorporating the following mechanisms:

- Encryption of management frames.
- Improving the authentication protocol by introducing PKMv2.
- Implementing mutual, PKI based authentication.
- Rendering definitions on key generation more precise.
- Replacing DES-CBC with AES-CBC.
- Introducing AES-CCM for message authentication.

As mentioned before, IEEE 802.16e-2005 is still a young standard and currently a lot of security related research is conducted around it. As history has shown with related wireless networks, this research will uncover further vulnerabilities and design flaws.

6.4 Conclusion

Even though that security was integrated in the original design of WiMAX, several serious vulnerabilities were discovered shortly after the release of the first version. These flaws have been corrected in successive standards. The actual version, IEEE 802.16e-2005 is still a young standard and currently a lot of security related research is conducted around it. As history has shown with related wireless networks, this research will uncover further vulnerabilities and design flaws.

Part II

Privacy in Wireless Networks

7 Introduction to Privacy in Wireless Networks

As Part I has illustrated, message related privacy is a well addressed topic in wireless networks regardless which (state-of-the-art) standard they are based on. It has also shown, that most of the presented wireless technologies possessed severe security related vulnerabilities in their original design.

Wireless computer networks are one of the most influential inventions of the last thirty years, and are pervading our everyday life. We are to hold that they deserve security researcher's attention in equal measure if not more as any other field in IT communications.

Multiple evolutionary iterations were necessary to secure them against even the most trivial threats. This points out the fact, that the designers did not focus on security aspects during the design and specification phases. It seems to be the same issue with privacy matters.

Since within the IT security community the main focus regarding wireless computer networks was set on securing communication channels from eavesdropping, privacy aspects have been neglected. Thanks to an upcoming discussion about privacy in general, research also moves over to pay more attention to this topic. We differentiate three forms of privacy in the context of wireless computer networks:

- **The privacy of the communication content - Message Related Privacy (Confidentiality)**

- **The privacy of the identity of the communication participants -**

Identity Related Privacy

- **The privacy of the location of the communication participants - Location Related Privacy**

Our main contributions concerning privacy in wireless networks are found in the chapters about *Attacking Identity Related Privacy* and *Mechanisms to Preserve Location Privacy in Wireless Networks*. But before we focus on these topics we will provide an introduction to the problem of privacy in wireless networks and an excursus to machine learning techniques which play a fundamental part in our contributions presented later in this part of the thesis.

Many privacy concerning attacks are based on exploiting the identity of a device or a user. In Chapter 9 we illustrate naive approaches as well as sophisticated device and user fingerprinting methods. We present our own contributions for device fingerprinting based on interface timing characteristics and for user fingerprinting, which is based on user behavior.

After the discussion of identity related privacy we turn to location related privacy and present a large scale example scenario to outline possible consequences or attacks on the location privacy. Many attacks on the latter are based on exploiting the device's or user's identity in the first place and establish linkability between the timely and spatial occurrences of it. As a consequence, all current location privacy enhancing mechanisms are based on disposable pseudonyms, regardless if WPAN or WLAN. Chapter 10 presents a detailed analysis of state-of-the-art mechanisms for Bluetooth and WiFi. It also includes our own approach to enhance location privacy in WiFi based on multiple virtual wireless network interfaces.

After presenting our analyses and contributions regarding *Privacy in Wireless Networks*, we have to conclude that there still seems to be a long way to go until we will reach a satisfying state of things. While the community's focus has been on security, privacy related research needs to intensified in all related areas. Fortunately does the upcoming discussion about privacy in general boost privacy in information technology and allows an optimistic glance into the future.

7.1 Message Related Privacy

Message related privacy overlaps strongly with the security related concept of confidentiality. As Chapters 3 to 6 have illustrated, confidentiality in state-of-the-art wireless networks is achieved by deploying cryptographic techniques such as symmetric or asymmetric encryption. The goal in general is to prevent potential attackers from obtaining the plaintext content of a conversation. Wireless network confidentiality mechanisms usually follow one of two different schemes:

- Shared Key Encryption; or
- Private Key Encryption.

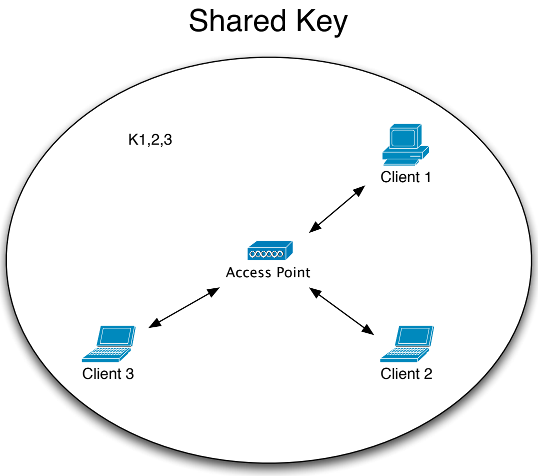


Figure 7.1: Every Message is encrypted with the Shared Key (K1,2,3)

7.1.1 Shared Key Schemes

Shared Key encryption schemes are characterized by the fact that only one cryptographic key (for symmetric cryptography) or key pair (for asymmetric encryption) is used in the domain. Private Key encryption schemes are characterized by the fact that the access point shares a unique key or key pair with each client in the domain.

If the encryption mechanisms used in a wireless setup is secure, we can assume perfect confidentiality between the communicating parties. As many security schemes in wireless networks are based on Shared Keys (see Figure 7.1), everybody who owns the Shared Key and is within transmission range is able to decrypt the conversation, as all messages are encrypted by the same cryptographic primitives.

Most system administrators and security personnel would not see a problem in such a configuration, as the network is secured against outside attacks. But the stakeholders of the communication might see the privacy of their conversation compromised as it is generally readable by other people knowing the Shared Key.

7.1.2 Private Key Schemes

Figure 7.2 illustrates the concept of using individual keys for encrypting the communication between the access point and each client. In this case the privacy of the conversation is protected against being overheard by other participants of the network.

Individual key systems are more secure and privacy preserving but the management overhead is considerably larger as in Shared Key based solutions, especially if the number of clients increases.

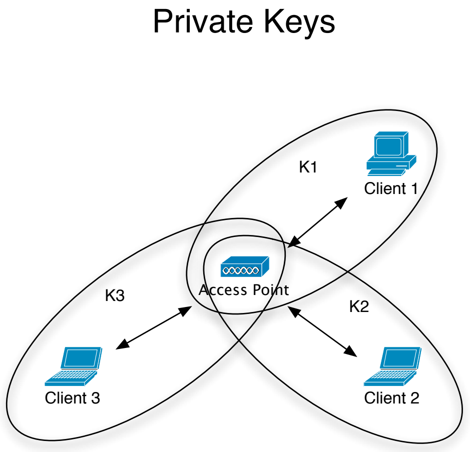


Figure 7.2: Communication between the Access Point and each Client is encrypted by an individual Key (K1, K2, K3)

Many real-world implementations of wireless networks user multiple layers of confidentiality providing mechanisms. Usually hop-to-hop encryption is implemented on lower layers and additional end-to-end encryption is available on higher layers (see Figure 7.3). Such configurations protect the wireless channel from outside attacks as well as the individual conversations by the use of individual keys.

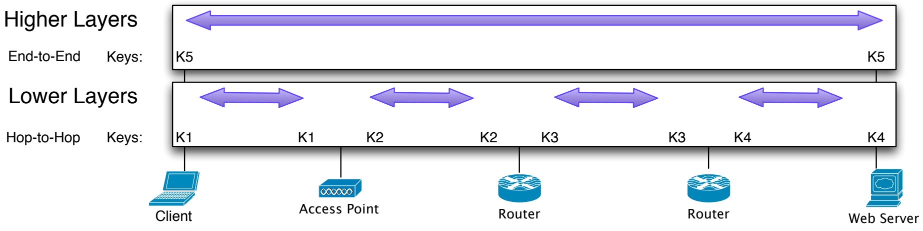


Figure 7.3: End-to-End and Hop-to-Hop Encryption Schemes

7.1.3 Conclusion

Message related privacy in wireless networks depends on two factors:

- The security of the implemented encryption scheme; and
- The choice of the key management scheme.

While for most private and corporate network security responsibles the cryptographic security against outsider attacks is the more important point, operators of public available networks should also strongly consider the internal privacy and choose a privacy preserving key management scheme.

7.2 Identity Related Privacy

Many security and privacy related attacks on wireless networks start by identifying the connected parties. Protecting the *Identity Related Privacy* of communicating parties provides a major improvement to overall security in wireless network and must therefore be regarded as very important.

Simplified, all messages transmitted in a wireless network consist of a packet header and a packet payload. While the payload is usually protected by message confidentiality mechanism such as presented in the former section, the packet privacy of headers is often neglected. As packet headers usually contain information necessary for network management such as routing, they reveal privacy concerning data like the ID of the sender and the receiver of this packet. Further on, they contain information like sequence numbers and time stamps and provide the base for several attacks [100].

Generally all participants of wireless networks are subject to ID privacy threats. We define *devices* and *users* as our main stake holders and therefore analyze their particular situation. These stakeholders can be identified using different techniques even if privacy preserving mechanisms are in place as we will describe in detail in Chapter 9.

After a survey on naive identification techniques and sophisticated fingerprinting approaches, we present our own contribution in this area called *Fingerprinting on Layer 2* (see Section 9.1.3).

7.3 Location Related Privacy

Location privacy means to be able to decide if other people are allowed to know about someone's current physical location. A lot of sensitive information may be obtained from knowing a persons location and her movements. It may reveal daily routines, social contacts, shopping preferences and many other private characteristics.

Existing wireless communication protocols are very vulnerable to attacks on location privacy as they are all based on fixed identifiers like hardware addresses. If such an address has been linked to a person using this device, locating and tracking this person is an easy task. A lot of research in the area of privacy preserving protocols has been done in the recent years and is still ongoing. But as chapter 9 illustrates, by deploying sophisticated mechanisms, it is technically feasible to reliably identify devices, and consequently circumventing current privacy preserving methods on the lower layers [109]. Section 7.2 also shows, that it is even not mandatory to be physically near to the target (e.g. in the same wireless networking cell), as some identification and fingerprinting techniques rely on traffic analysis in the backbone of the connected wireless networks.

It can be argued that the capability of tracking devices does not consequently mean that a user can be tracked because the device may be used by multiple individuals. But as we have proved in [112], it is possible to identify the particular individual using a device at a certain moment by creating behavioral fingerprints (see Section 9.2), this discussion can be dismissed.

7.4 Outlook and Conclusion

As message related privacy overlaps greatly with the security concept of confidentiality, it is only briefly discussed in the Introduction. The focus was laid on Group Key and Shared Key schemes and their privacy related properties as well as on end-to-end and per-hop encryption fundamentals.

Our contributions regarding privacy in wireless networks are presented in the chapters *Attacking Identity Related Privacy* (see Chapter 9) and *Preserving Location Privacy in Wireless Networks* (see Chapter 10).

Many attacks are based on exploiting the identity of a device or a user. In Chapter 9 we illustrate naive approaches as well as sophisticated device and user fingerprinting methods. We present our own contributions for device fingerprinting based on interface timing characteristics and for user fingerprinting, which is based on user behavior.

After the discussion of identity related privacy we turn to location related privacy and present a large scale example scenario to outline possible consequences or attacks on the location privacy. Many attacks on the latter are based on exploiting the device's or user's identity in the first place and establish linkability between the timely and spatial occurrences of it. As a consequence, all current location privacy enhancing mechanisms are based on disposable pseudonyms, regardless if WPAN or WLAN. Chapter 10 presents a detailed analysis of state-of-the-art mechanisms for Bluetooth and WiFi. It also includes our own approach to enhance location privacy in WiFi based on multiple virtual wireless network interfaces.

Due to the fact that sophisticated device and user fingerprinting techniques are quiet powerful and very hard to counteract. Even performant pseudonym approaches theoretically lack the robustness regarding some fingerprinting mechanisms.

After presenting our analyses and contributions regarding *Privacy in Wireless Networks*, we have to conclude that there still seems to be a long way to go until we will reach a satisfying state of things. While the community's focus has been on security, privacy related research needs to be intensified in all related areas. Fortunately, the upcoming discussion about privacy in general boosts privacy concerns in information technology and allows an optimistic glance into the future.

8 Excursus - Machine Learning Techniques

8.1 Introduction to Machine Learning and Privacy

Using the Internet leaves tracks that might lead to the unwanted disclosure of information resulting in the violation of user privacy – ranging from user tracking over collecting web-usage data to modeling user behavior and putting it into relation to other users or user groups. When looking at the available information that can be logged (e.g. due to using a Wi-Fi access point) there are numerous features that might disclose information about the user's privacy. Although, there are obvious ones (e.g. the MAC address of a network interface card), there are other features or their combination that cannot be identified by simple methods[1]. However, in order to protect user privacy, we must understand how privacy information is disclosed. Only with this understanding it is feasible to deploy effective countermeasures. Here, machine learning can play an important role. Due to the fuzzy nature of machine learning algorithms, patterns within the data can be identified and analyzed without the need for a human understanding of the raw data.

This chapter provides detailed information about two machine learning techniques called *Self Organizing Maps* and *Activation Patterns*, which play an important part in our contributions presented later in this part of the thesis.

[1] E.g. simple key word matching signatures, or searching for unique IDs within the data (e.g. MAC address) etc.

8.2 Self Organizing Maps

Self Organizing Maps (SOM) belong to the broader category of neural networks [94]. They are mainly used for unsupervised learning and the visualization of high-dimensional data. In our approach in Section 9.1.3, we employ SOMs for a supervised classifier to classify different WLAN chipsets. Although, other supervised algorithms like neural networks or support vector machines might be better suited for such classification tasks, we still focus on the SOM due to one main reason: The visual representation of the data in a 2D map allows us to quickly gain insight on the analyzed data (an example is given in Figure 8.2), which proves very useful for prototype development.

8.2.1 The SOM Tree Algorithm

By labeling the SOM units during the training process according to the class labels of the data they represent, the SOM can also be employed for supervised learning. However, due to the unsupervised nature of the SOM, the class information is not taken into account during the training process. Therefore, the accuracy of the trained model might be inadequate for the separation of data belonging to different classes. This data is mapped by the same units and leads to classification errors that decrease the accuracy of the SOM. In order to cope with this issue, our classifier utilizes multiple SOMs arranged in a tree.

Whenever the model of a trained SOM is not precise enough to separate data of different classes accurately, we extract this data, train a new SOM on this data and link the units of the old SOM covering this data to the new SOM. Therefore, we do not need to deal with SOM model complexity manually. If the model of a trained SOM is not accurate enough, the algorithm simply trains a new SOM that is only trained on the data which requires more complex modeling (indicated by a higher misclassification rate). The multiple SOMs are trained and arranged in a tree according to this algorithm:

1. Train a SOM on the input data

2. Label the units according to the classes they represent

3. Calculate misclassification rates for all classes

4. Extract the data of classes that cannot be separated with an error rate lower than a given threshold

5. Mark the units that cover the extracted data to indicate that the actual classification will be made in the next SOM.

6. Go to step one and train a SOM for new extracted data. Repeat these steps until the error conditions are met or only two classes remain in one SOM.

A simple example with five classes is shown in Figure 8.1. The first SOM is trained on the complete data set and the misclassification rates are determined. The example shows that classes A/B/C and D/E cannot be separated accurately by the first SOM. Therefore, two data sets for A/B/C and D/E are extracted. For both data sets, new SOMs are trained and the units corresponding to these classes in the first SOM are linked to the newly trained SOMs. In case of A/B/C, the second SOM is able to separate the class C from A/B but the misclassifications rates for A/B are still too high. Therefore, another SOM is trained that increases the classification performance. The picture indicates that the SOM for A/B still has some misclassification errors, which cannot be removed without losing generalization (and thereby overfitting the data). The trained SOM hierarchy of SOM Tree is used for the classification of unknown data in this way:

- Present the data to the root SOM of the tree and determine the best matching unit (BMU)
- If the unit is linked to another SOM further down in the hierarchy, load this SOM and go to the previous step. If the unit is not linked to another SOM, return the class label of the unit.

This procedure is indicated in the example by the two classification paths for data vectors from class B end E. Besides our classification mechanism presented later in this thesis, we already successfully applied to other classification problems, especially for network traffic classification [142]. For SOM training in our prototype implementations, the SOM toolbox [186] which is available for Matlab® [120] was used for.

8.2.2 SOM Training and Classification

Before we describe the training/classification algorithm, we need to explain several terms:

- **Best Matching Unit (BMU)**: For any given feature vector[2] at least one unit on the map can be found which has the smallest distance to the given feature vector[3].
- **Hit based classification**: The existing SOM algorithm was extended, so that it can be used for supervised learning. During the training phase the BMUs of each feature vector are assigned to the class, the feature vector belongs to. During classification, this class information is used to determine the class of unknown feature vectors. Units which are hit by more than one class are either ignored or the class which has the majority

[2] A feature vector consists of n entries, where n is the number of features used for the classification task.

[3] There can only be more than one BMU if the feature vector has the same distance to several map units.

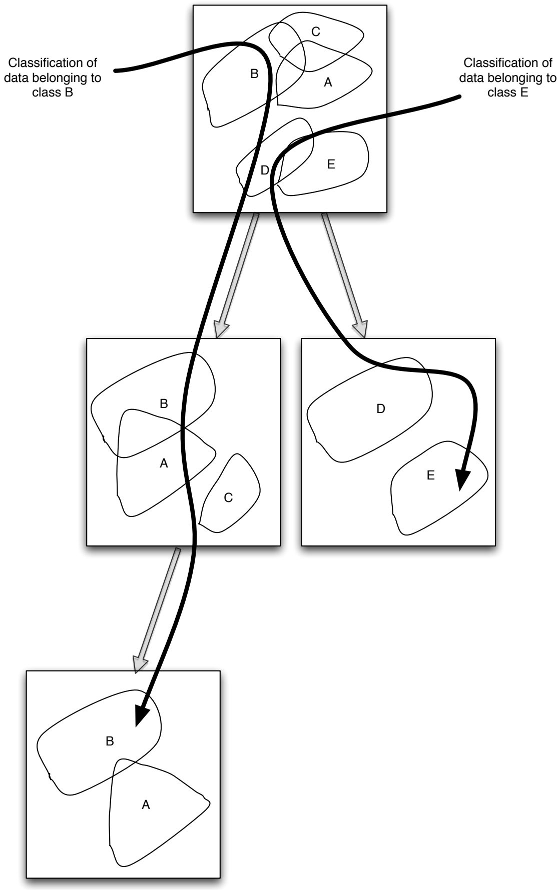

Classification of data belonging to class B
Classification of data belonging to class E
C
B
A
D
E
B
A
C
D
E
B
A

Figure 8.1: SOM Tree example

of hits[4] is taken into account. Figure 8.2 shows a SOM trained with data from three different WLAN chipsets. The colored units represent those three different chipsets. The size of the colored dots indicate the number of hits during the training process. Units which have more than one color are hit by different classes during the training process. This hit information is used for the classification of unknown feature vectors.

- **SOM Tree**: During previous work we realized, that a single SOM is not accurate enough for most of the classification tasks. Thus, the algorithm adds and trains new SOMs dynamically, whenever the classification error rate gets too high.

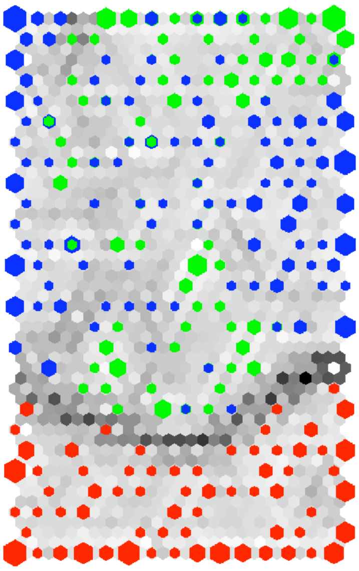


Figure 8.2: SOM trained with data from three WLAN chipsets

This is the pseudo code of the training algorithm:

- **Step 1**: Train a SOM with all training data.
- **Step 2**: Label each unit according to the labels provided by the training data. Classify the training data by using hit based classification (described above). The classification result is compared to the labels of the training data and the classification errors for each class are determined.
- **Step 3**: Partitions (one partition has one or more classes) are created according to a given error rate. The error rate is increased until at least two partitions can be found.

[4] The number of hits for each class is normed with the number of total examples in one class.

- **Step 4**: If each partition has a single class, the algorithm stops. Otherwise it is called recursively for each partition and continues with **step 1**.

This is the pseudo code of the classification algorithm:

- **Step 1**: Take the topmost SOM of the tree.
- **Step 2**: Determine the BMU of the SOM for the given feature vector.
- **Step 3**: If the BMU belongs to a partition which needs to be classified with another SOM use this SOM and go to **step 2**. If no further SOM can be found return the classification result for the feature vector.

8.3 Activation Patterns

Activation Patterns are generated by utilizing three different techniques from the areas of machine learning and artificial intelligence. These building blocks include unsupervised learning algorithms, associative networks and SA algorithms. For the analysis and discretization of single features and feature groups we require unsupervised learning algorithms based on prototypes. Examples for such algorithms are Neural Gas (NG) [119] and its successors Growing Neural Gas, Robust Neural Gas and Robust Growing Neural Gas (RGNG) [147]).

The following definitions are used through this Section:

- *distance-based features:* These are features that are represented by continuous values for which it makes sense to define a distance measure (e.g temperature values, connection duration, etc.)
- *nominal features:* These are features that are represented by values that cannot be brought into relation via a distance measure (e.g. protocol identifiers such as UDP, ICMP, TCP or email addresses).
- *Activation Pattern:* This is as an n-dimensional vector that represents the activation values of the n nodes of an associative network.

We use this technique to classify the behavior of wireless network users in Section 9.2. Associative networks [148] are directed or undirected graphs that store information in the network nodes and use edges (links) to present the relation between these nodes. Typically, these links are weighted according to a weighting scheme. Spreading activation (SA) algorithms [33] can be used to extract information from associative networks. Associative networks and SA algorithms play an important role within Information Retrieval (IR) systems such as [14, 47, 96, 97, 184]. By applying SA algorithms we are able to extract *Activation Patterns* from trained associative networks. These *Activation Patterns* can then be analyzed by arbitrary unsupervised learning algorithms such as Self Organizing Maps (SOM) [94], Hierarchical Agglomerative Clustering (HAC), Expectation Maximization (EM), k-means, etc.

Unsupervised learning algorithms rely on some kind of distance measure to find clusters of similar data within a given dataset. It is easy to define such distance measures for datasets based on continuous features. However, as soon as categorical features need to be analyzed, these distance measures might not make sense. Typically, it is not possible to define a meaningful distance for the values of such features. Therefore, several unsupervised algorithms for the analysis of categorical data have been developed. Some examples are COOLCAT [10], or Kernel K-Means [45]. Typically such techniques analyze the co-occurences of attributes and use this information for unsupervised clustering. Couto [45] introduces the Kernel K-Means algorithm for categorical data and gives a good overview of other unsupervised algorithms for categorical data. If continuous features need to be analyzed with such algorithms, the values need to discretized first. Such discretization methods range from very simple methods that put the categorical data into n bins to more complex methods based on entropy or fuzzy techniques.

The transformation of raw feature vectors into *Activation Patterns* involves the mapping of categorial and continuous features into an associative network. Similar to the other methods the continuous features need to be discretized. Although there are several discretization methods available, we have decided to employ an NG based algorithm for the discretization. This comes with certain advantages that will be explained later.
As mentioned above, applying SA to the associative network generates the *Activation Patterns*. These patterns can then be analyzed by standard unsupervised learning algorithms with conventional distance measures. In addition to unsupervised analysis, the information stored in the associative network can be used directly to gain information about the relations between features. Furthermore, we are able to execute search queries that retrieve similar *Activation Patterns*. These additional benefits are not given by the other algorithms.

8.3.1 From Feature Vectors to Activation Patterns

The process of generating and analyzing *Activation Patterns* is separated into five processing layers. The general idea is to extract the co-occurence information of different features (L1, L2), to store this information in an associative network (L3) and to generate *Activation Patterns* by applying SA strategies (L4). Various analysis techniques can then be applied to the generated patterns (L5).

8.3.1.1 L1 - Feature extraction

As mentioned before, features of any data set can be separated into the categories *distance-based features* and *nominal features*. These two types of features are handled differently by subsequent processing steps and need to be identified correctly at L1. For *distance-based features* groups that represent features with similar meanings and value ranges can be created. This grouping is not a requirement for further analysis, but reduces the computational complexity.

8.3.1.2 L2 - Node generation

This process layer creates the nodes of the associative network that will be generated in the next layer. The process of mapping feature values to nodes depends on the type of the particular feature. For *nominal features* the possible values are directly mapped to separate nodes. For *distance-based features* we need to apply some kind of discretization operation to map values onto nodes. Although there is a wide range of discretization algorithms available, we have chosen the RGNG algorithm. It is applied to the continuous values and the trained prototypes are used as nodes for the associative network.

Basically any prototype based unsupervised learning algorithm could be used for the discretization process. RGNG was selected, since it includes several advanced method and employs the Minimum Description Length (MDL) [153] to automatically determine the model complexity. Since the performance of RGNG and similar algorithms has been evaluated by applying them to a wide range of datasets, we can assume that these algorithms will produce good results for the low dimensional data represented by single features or selected feature groups. Although the computational complexity of RGNG is high, the benefits justify its application and improve the employed analysis techniques. In other more specific scenarios, the RGNG algorithm can be replaced with a simple adequate discretization method.

The node generation process of L2 can be summarized as follows : Values of *nominal-features* are directly mapped to unique nodes within the associative network. For each of the feature groups or single features (defined in L1) of the category *distance-based features*, an RGNG-map is trained and prototypes are incorporated as new nodes into the associative network.

8.3.1.3 L3 - Network generation

In this layer, links are created between the nodes according to the relations between the nodes:

1. The features are analyzed according to the two categories determined in L1. *Nominal features* are directly mapped to nodes according to the mapping from the previous step. For *distance-based features* (single values or groups) the prototype of the corresponding RGNG-map with the smallest distance to the data vector, is located. This prototype is called the Best Matching Unit (BMU). Its corresponding node in the network is found according to the mapping generated in L2.

2. All these nodes are now linked within the associative network. Newly created links between two nodes are initialized with weight 1. The weight of existing links is increased by 1. This linking represents the co-occurence of different values of distinct features. The link weight represents the strength of this relation.

The weight of the links within the network represents the number of times two nodes co-occur. In order to apply the SA-algorithm in L4, we need to normalize

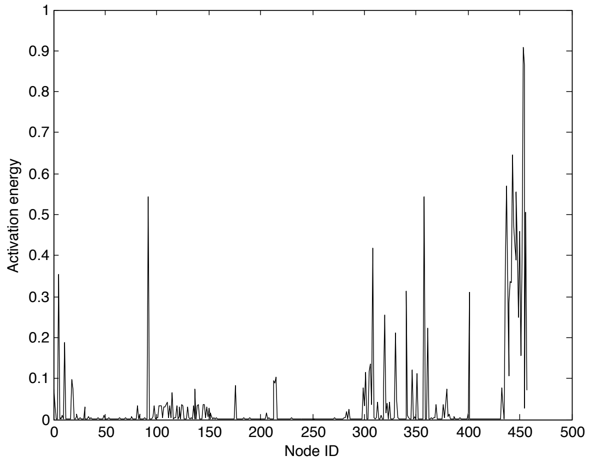


Figure 8.3: Activation Patterns Example 1

the link weights within the associative network, so that the maximum weight is equal to 1. We can apply different strategies here that normalize the links locally or globally.

8.3.1.4 L4 - Activation Pattern Generation

The links of the associative network created in L3 represent the relations between features and values of the features are represented as the nodes of the associative network. The information about relations can be extracted by applying the SA-algorithm to the network. For each data vector, the nodes in the network that represent the values stored in the data vector, are determined. By activating these nodes for a given data vector, the activation can be spread over the network according to the links and their associated weights for a predefined number of iterations. After this spreading process, we can determine the activation value for each of the nodes in the network and present this information in a vector - the *Activation Pattern*. The areas of the associative network that are activated and the strength of the activation gives information about which feature values occurred and how they co-occur. Examples for different patterns are shown in Figures[5] 8.3 and 8.4. By applying distance measures, such as the cosine similarity, the patterns can easily be compared.

8.3.1.5 L5 - Analysis

- *Unsupervised clustering and semantic search*: Due to the transformation of the raw data into *Activation Patterns* we can apply standard distance-

[5] The x-axis represents the nodes within the network, the y-axis represents the activation energy of these nodes after applying the spreading activation process to the activated nodes.

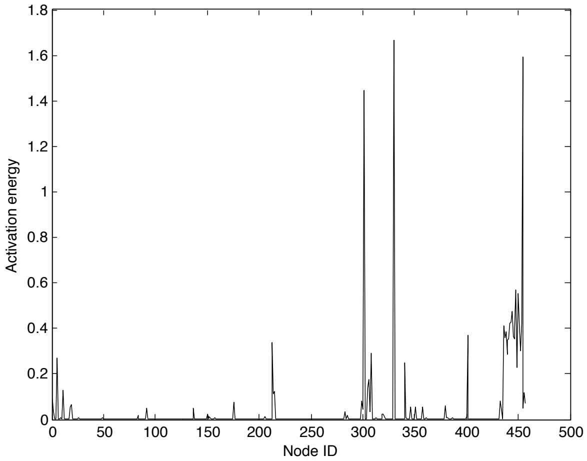


Figure 8.4: Activation Patterns Example 2

based unsupervised clustering algorithms while keeping the information about semantic relations within the data. This allows us to find clusters of similar behavior patterns and to deduce common features within a cluster. By varying the model complexity, we are able to build a hierarchy from a very coarse grained categorization down to a very detailed representation of the analyzed data. The distance between the *Activation Patterns* can be used to implement semantic search algorithms that retrieve similar behavior patterns. These search queries can also be used to specify certain feature values and find closely related patterns (e.g. given a user: which other users use similar recipients within their emails).

- *Feature relevance*: The relations within the semantic network are created according to the co-occurrence of feature values within the analyzed data set. The strength of these relations are represented by the associated weights within the network. Given a feature value that is represented by a node and the number of emerging/incoming links and their weights, we are able to deduce the importance of the information carried by the node. Nodes that are connected to a large number of other nodes typically do not add information for subsequent analysis processes. This is highlighted by a simple example: Assuming a data set that describes features of various vehicles (bikes, cars, trains), a node that describes that the vehicle has wheels does not carry any information at all. The reason is that all the mentioned vehicles have wheels and thus the node is connected to all other possible feature values. In the analysis section we will show some examples of such values in the case of the analyzed email data. Nodes that do not carry information can be penalized by introducing so called fanout factors that attenuate the spread activation.

- *Feature relations*: The semantic network describes arbitrary relations between feature values. By activating one or more nodes (corresponding to feature values) within the semantic network, and spreading their activation via the links to the neighbors, we are able to extract details about the relations between various feature values (e.g. between a given time and the typical users that write emails at this time).

8.3.2 A simple example - Artificial Clusters

This section presents a simple example based on four distinct clusters that does not include *nominal-features* (see Figure 8.5). Although this example data could easily be analyzed with standard unsupervised analysis methods, we use it to show the basic properties of the *Activation Patterns* and the influence of single parameters. For visualization of the clusters, Self Organizing Maps (SOMs) [94] are used.

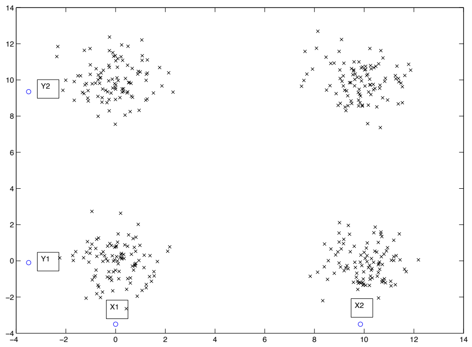


Figure 8.5: Artificial dataset consisting of four distinct clusters. There are two RGNG prototypes for each dimension (X_1 and X_2 for the first feature and Y_1 and Y_2 for the second feature).

- **L1:** The data-set consists of 2D data-vectors representing two different features. In this case there are only *distance-based features* meaning that the values of these features can be related with a distance measure.

- **L2:** In this layer we create the nodes for the associative network. Since both features belong to the category of *distance-based features*, we cannot map their values directly to nodes within the associative network. Instead, we apply the RGNG algorithm to the values of the first feature (represented by the x-axis) and to the values of the second feature (represented by the y-axis). The RGNG employs the MDL in order to control model complexity and finds one prototype per cluster.

- **L3:** We now create the links of the associative network and determine their strength by analyzing the co-occurence information of the two features within the training data.

- **L4:** We now make use of the SA algorithm to generate *Activation Patterns* for each data vector within the training set.

- **L5:** The generated *Activation Patterns* can now be analyzed with unsupervised learning techniques. In order to provide a meaningful visualization, we train SOMs for both *Activation Pattern* sets.

Coping with different ranges of features: Due to the transformation of the raw data into the *Activation Patterns*, we get information about the co-occurence of different features. The values of the features are represented by different nodes in the associative network. The information about the relations between them is stored in the links between the nodes and their strength. Therefore, the framework does not require any kind of normalization operation applied to the raw data. In order to show that we use the same data-set as in Figure 8.5, we multiply the value of the second feature with a constant (1000). Again, we train SOMs on the unmodified and modified raw data-set. For the unmodified data-set, the four different clusters are very easy to recognize within the trained SOM (see Figure 8.6). However, for the modified data-set the trained SOM only shows two distinct clusters (see Figure 8.7). This behavior is due to the fact that the much larger values of the second feature have more influence on the Euclidean distance and hide the relative small distances between the values of the first feature. In contrast, by utilizing the *Activation Patterns*, the SOM is always able to find four clusters regardless of the range differences between the features (see Figure 8.8).

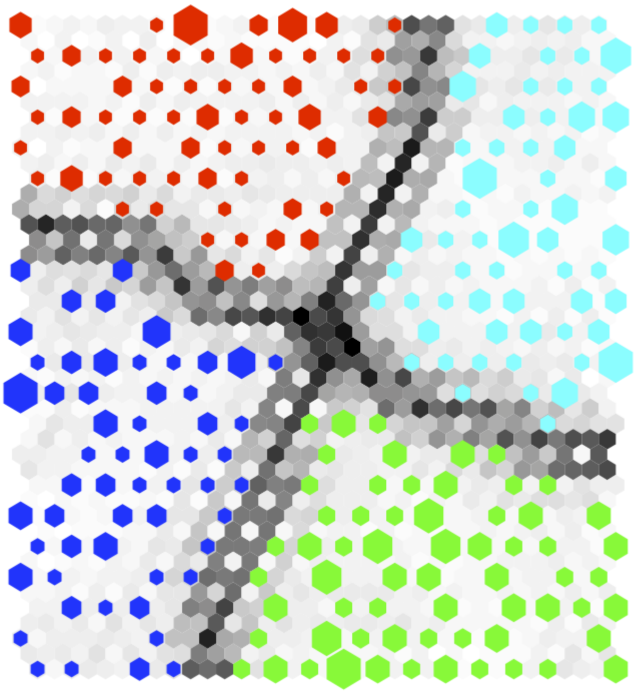


Figure 8.6: SOM trained on 2D raw-data

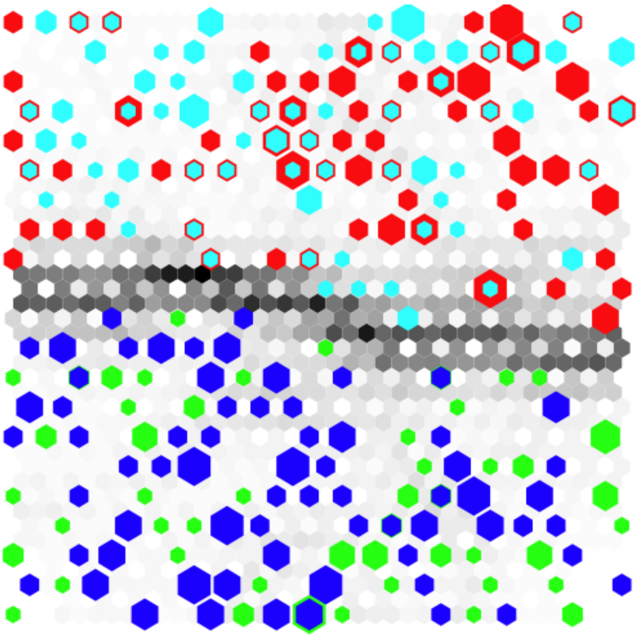


Figure 8.7: SOM trained on 4D *Activation Patterns*

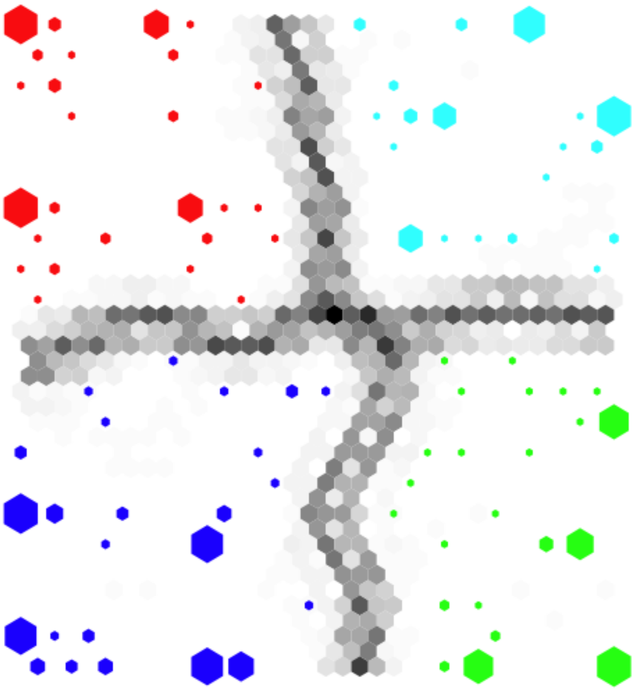


Figure 8.8: Activation Patterns Clustering

9

Attacking Identity related Privacy

As mentioned before, many attacks on wireless networks start with attacking the identity related privacy of the connected parties. In order to identify a device or a user, some externally observable characteristics need to be defined. A set of these features, preferably unique per entity is called a fingerprint. The following two sections present means of creating fingerprints of devices and users respectively.

9.1 Device Identification

As many communication protocols are based on hardware addresses, they provide a solid base for device identification. Standards such as Bluetooth and WiFi rely on globally unique device identifiers and assume hardware MAC addresses as satisfactory to this requirement. Randomly changing device addresses seem therefore to be a trivial solution to provide protection against unwanted device identification.

Hardware address and device identity forgery can be the basis for several other attacks, not directly related to privacy. A straightforward approach for device identification is to utilize the device addresses such as the MAC (Media Access Control) address (layer 2) or the IP address (layer 3). This can easily be achieved by analyzing relevant ARP (Address Resolution Protocol) traffic [144]. Unfortunately, this approach has the major drawback such as most devices allow to modify their assigned MAC address with easy to use, free software tools.

Additionally to device addresses, there exists a variety of externally observable features to create characteristic fingerprints of wireless hardware. The following section presents an overview of state-of-the-art approaches and a short evaluation of their practicability and performance.

9.1.1 Device Fingerprinting on the Physical Layer

Fingerprinting of physical layer characteristics could be a very effective way to counteract MAC address spoofing. It is the reflection of defects or the unique design of the hardware on the transmitted waveforms. Three main classifications of these methods are identified by Zeng et al. in [197]:

1. **Radiometric Fingerprinting**
   The unique characteristics of a hardware transceiver cannot easily be replicated or copied from one device to another. These characteristics can either be of based on signal transients or signal modulation which are different from device to device due to deviations in the manufacturing process. Device mobility or strong interferences may seriously downgrade the performance of this classification technique. Further on, software defined radio or high-end arbitrary waveform generators may be used to fool the system. The latter require a powerful and well equipped attacker.

2. **Clock Skew Fingerprinting**
   The clocks in state-of-the-art wireless devices are built with inexpensive crystal oscillators that are affected by a number of environmental factors and aging effects, leading to the fact, that no two clocks run the same. Therefor all clocks have some skew compared to a reference clock. As this skew is unique for different devices it can serve as a fingerprinting feature. A drawback is that attackers could alter the time values reported by their device and overwrite the transmitted timestamps.

3. **Physically Unclonable Functions**
   PUFs are a new approach to generate signatures. They are based on the complex physical characteristics of the integrated circuits (ICs) in wireless devices. To carry out PUFs, it needs a very sophisticated attacker to mimic these characteristics. The need for specially manufactured ICs for the wireless network devices would be to expensive for a broad deployment.

Concluding it can be said that PHY layer based methods would be an effective additional layer for device fingerprinting. The major drawback is, that for most implementation, special purpose hardware is needed. As these approaches belong to the disciplines of electronic engineering and signal processing, they will not be described in detail in this thesis. But for the sake of completeness, a Radiometric Fingerprinting approach based on turn on signal transients published by Hall et al. [64] will be presented later in this chapter (see Section 9.1.1.2).

9.1.1.1 Remote Physical Device Fingerprinting

One of the most significant papers in the field of device fingerprinting has been published by Tadayoshi Kohno and his team at UC San Diego [93]. Kohno developed a method to identify remote devices by exploiting small, microscopic deviations in the hardware: clock skews. By analyzing the deviation of TCP or ICMP timestamps over a certain period of time it is feasible to distinct different hardware clocks and thus different devices. Unfortunately, this approach is not applicable in an encrypted wireless environment as it needs plaintext TCP or ICMP payloads for analysis. Nevertheless, Kohno's method is a powerful fingerprinting approach for most wired networks and it does not rely on physical proximity to the target device as most others do.

9.1.1.2 Radio Frequency Fingerprinting in Wireless Networks

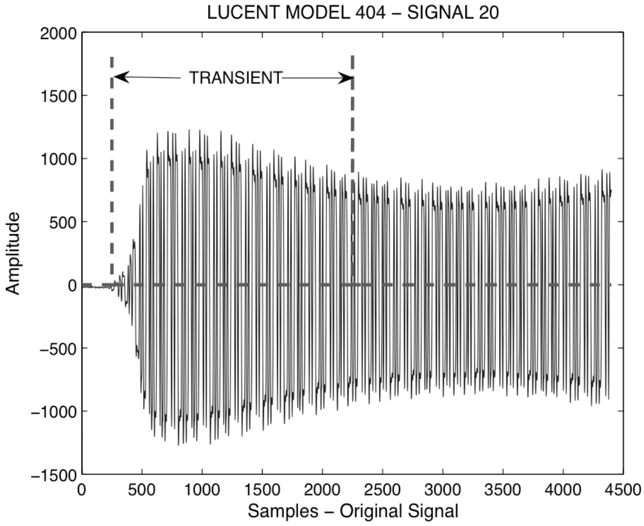


Figure 9.1: Signal from a 802.11b Transceiver [64]

This section describes a WLAN fingerprinting method presented by Hall [64] et al. As opposed to the two other methods described later in this work, we did not implement the method due to the lack of hardware which is needed for the approach. Thus, we only give a short overview and refer the reader to [63] and [64] for further details.

9.1.1.2.1 Background The presented fingerprinting technique is based on the signal characteristics of turn-on transients of wireless transceivers. These transients are specific to each different transceiver and thus are perfectly suited as data source for fingerprint generation. Figure 9.1 shows an example for the turn-on transient of an Orinoco chipset. In preceding work [63] the authors describe significant features that are extracted from the turn-on transient and are used for fingerprint creation. Transient capturing and analysis requires a special infrastructure for signal capturing, which is depicted in Figure 9.2.

The method extracts basic signal components - the DWT (Discrete Wavelet

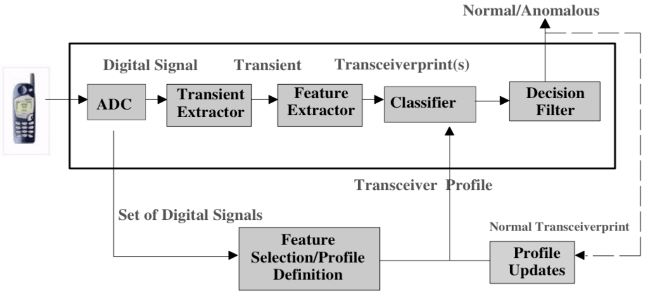


Figure 9.2: System overview [64]

Transformation) coefficients, signal phase and signal amplitude - and generates features used for the classification process (see Figure 9.3) The extraction and the computation of the features and their further analysis is done with Matlab™on a standard laptop. The fingerprint for each device is represented by these features. Fingerprint classification is based on a statistical classifier.

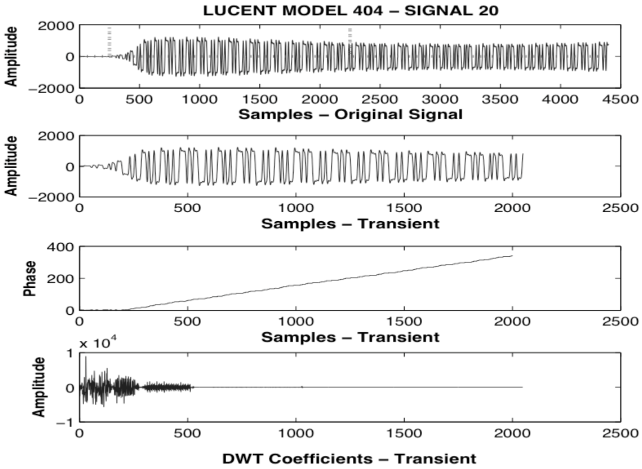


Figure 9.3: Signal components [64]

9.1.1.2.2 Discussion Hall et al. evaluated the performance of the fingerprinting method with 30 transceivers. For each transceiver 120 signals were captured and used for the performance evaluation. The results indicate that the method is capable of achieving a very low false positive rate (0% during the evaluation) and a high detection accuracy (95% during the evaluation). However, the biggest disadvantage of this method is the hardware which is needed for signal capturing. This drawback also limits the usability in intrusion detection

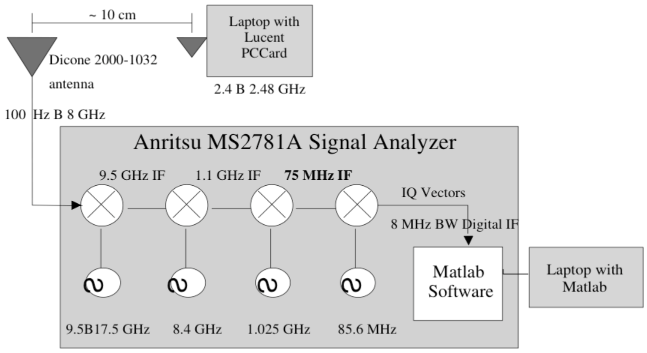


Figure 9.4: Evaluation setup [64]

systems.

9.1.2 Passive Data Link Layer Fingerprinting

This section describes the wireless NIC fingerprinting approach developed by Jason Franklin and his team, published in 2006 [51]. Franklin identified an imprecision in the IEEE 802.11 Media Access Control specification that was interpreted differently by wireless NIC firmware developers. The following section will explain this flaw and its use for fingerprinting in more detail.

9.1.2.1 Background

Typically, an activated wireless NIC instantly starts to look around for available wireless networks. This search is performed by broadcasting *probe-request frames*. The IEEE 802.11 standard describes this so-called *active scan function* as follows.

> For each channel, the client broadcasts a probe request and starts a timer. If the timer reaches *MinChannelTime* and the channel is idle, the client scans the next channel. Otherwise, the client waits until the timer reaches *MaxChannelTime*, processes the received probe response frames and then scans the next channel [51].

Due to this loose definition many drivers with slightly different probing techniques have been implemented. Jason Franklin and his team found out that these varieties are externally observable characteristics that allow the creation of fingerprints.

Figure 9.5 and Figure 9.6 visualize the time difference between arriving probe frames as transmitted by two different wireless drivers. One can observe unique cyclic patterns with different time deltas between the probe requests for each wireless NIC. Small variations in these patterns which complicate the creation of good fingerprints are due to two main reasons, packet loss caused by signal interference and the fact that wireless drivers by default constantly circle through all eleven channels in the 2.4 GHz ISM band in search of other access points. The first source of information loss can easily be avoided by using higher gain antennas while the second can be compensated by using statistical methods.

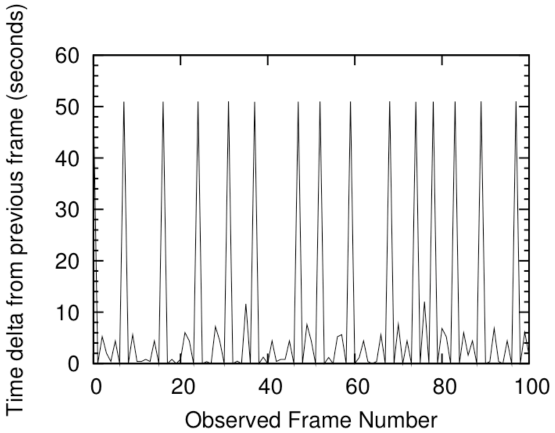


Figure 9.5: D-Link driver for the D-Link DWL-G520 (802.11b/g PCI wireless NIC [51]

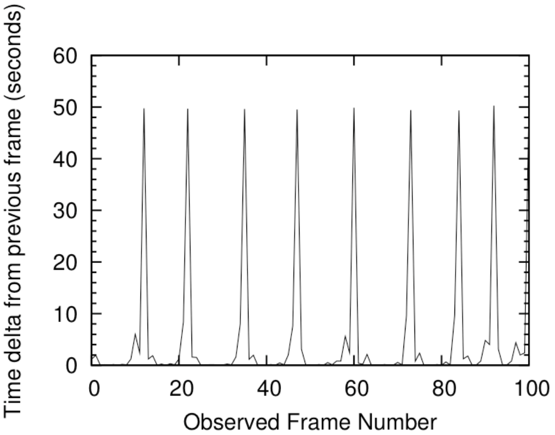


Figure 9.6: Cisco driver for the Aironet AIR-CB21AG-A-K9 (802.11 a/b/g) PCI wireless NIC [51]

In order to create a fingerprint the presented method needs to *capture the trace* of a wireless NIC. This is done by capturing a series of probe request frames of a specific NIC with a WLAN sniffer. For characterizing the time deltas between the probe requests a binning approach has been chosen. Binning works by translating an interval of continuous data points into discrete bins. A bin is an internal value used in place of the true value of an attribute. The distributions of the observed deltas in these bins of equal size allow the creation of a stable signature [51].

By analyzing this collected data, Franklin et al. identified two attributes from the probing rate that are essential for fingerprinting the NIC respectively its driver. The first attribute is the bin frequency of delta arrival time values between probe request frames that characterizes the size of each bin. The second attribute was the average for each bin, of all actual (non-rounded) delta arrival time values of the probe request frames placed in that bin. This characterizes the actual mean of each bin. The next step was to create a signature for each driver. The authors decided to use a Bayesian model because it is simple and well tested [51].

Franklin et al. were now able to create signatures of 17 different NIC drivers which they named *master signatures.* Unknown signatures can now be compared to the master signatures in order to determine the closest matching NIC driver. This is done by calculating the closest distance between the captured signature and a master signature [51].

Let p_n be the percentage of probe request frames in the n-th bin of the signature T and let m_n be the mean of all probe request frames in the n-th bin. Let S be the set of all master signatures and let s be a single signature in this set. Let v_n be the percentage of probe request frames in the n-th bin of s and let w_n be the mean of all probe request frames in the n-th bin of s. The following equation was used to calculate the distance between the observed signature T and all known master signatures, assigning to C the distance value of the closest signature in S to T [51].

$$C = min(\forall s \in S \sum_{0}^{n}(|p_n - v_n| + v_n|m_n - w_n|)) \tag{9.1}$$

9.1.2.2 Proof-of-Concept

To prove their concept empirically, Franklin et al. have chosen three different evaluation setups. The first two (named Test Set 1 and 2 in Table 9.1) were used to create the master signatures and evaluate them. These tests have been performed in a laboratory environment. No background traffic or other WLAN activity interfered with the measurement. The authors say that Test Set three (cf. Figure 9.7) could be seen as a *real world scenario.*

Testset	**Successful**	**Total**	**Accuracy [%]**
1	55	57	96
2	48	57	84
3	**44**	**57**	**77**

Table 9.1: Accuracy of fingerprinting technique by scenario [51]

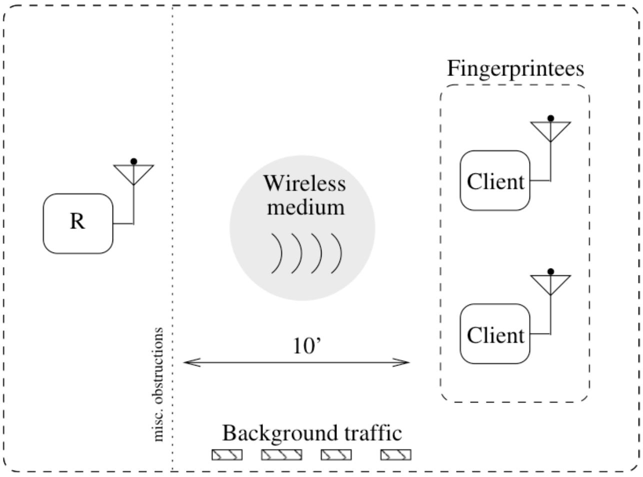


Figure 9.7: Evaluation setup [51]

These results may be obtained after 30 minutes of trace capturing per NIC. The authors also say that after one minute of scanning the accuracy of their technique is at least 60 % in all test cases [51].

9.1.2.3 Discussion

Franklin et al.'s approach uses an uncertainty in the IEEE 802.11 specification. It is able to classify different firmware versions instead of the underlying hardware. For creating a meaningful fingerprint a large number of probe-requests need to be captured. Typically, a NIC - willing to join a network - usually just needs a hand-full of these requests. Consequently, it could take a rather long time to obtain a suitable amount of data. Another significant draw-back is that fingerprinting may easily be avoided by using passive-scanning or altering the device firmware. Some improvements to this approach have been developed by Desmond Loh et al. [38]. The investigated the influence of the host machine and the operating system on the fingerprinting approach based on beacon frames. They were able to improve the classification results.

9.1.3 Fingerprinting on Layer 2

Our contribution to the area of device fingerprinting is based on observing their timing behavior.

The approach examines the timing behavior of IEEE 802.11 devices generating so called *acknowledge packets (ACK)*. Due to the fact that IEEE 802.11 standards follow the principle of *half-duplex* communication, a *collision avoidance technique* is generally needed to be deployed. If a participant A (client) has sent a data frame to participant B (access point), A is not able to observe if its message was transmitted correctly or collided with a data frame sent from another participant at the same time. IEEE 802.11 standards are based on the so called *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) mechanism [75]. In this work we just shortly describe the IEEE 802.11 media access system. For further details consult the specifications document [75].

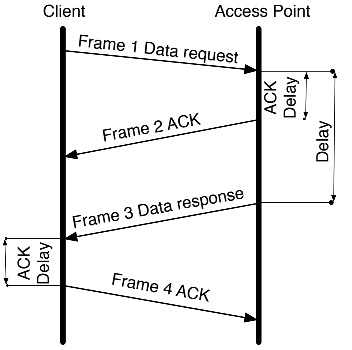


Figure 9.8: ACK delay

To inform *A* that its data frame was transmitted correctly, *B* generates and transmits an ACK packet after having correctly evaluated the CRC checksum of *A's* data frame. If the CRC check fails, no ACK will be sent and *A* retransmits the data frame after a certain time [75].

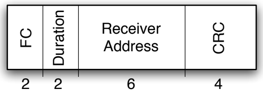


Figure 9.9: ACK packet

The computing and evaluation of the CRC checksum plus the generation of the ACK packet takes a certain amount of time. This amount depends on the hardware implementation of the CRC algorithm, the firmware and some other components of the used wireless network device. We call this delay *The Acknowledge Delay.* If one regards the distribution of a certain number of ACK delay values the outcome represents a significant property of the used wireless device. This outcome is called *Significant Histogram.*

Based on these Significant Histograms it is possible to distinct between different IEEE 802.11 device chipsets. The classification results in lab environment were very promising.

9.1.3.1 Classification

This section describes how we applied the concept of self organizing maps (SOMs) as presented in Section 8.2, to classify different wireless network chipsets.

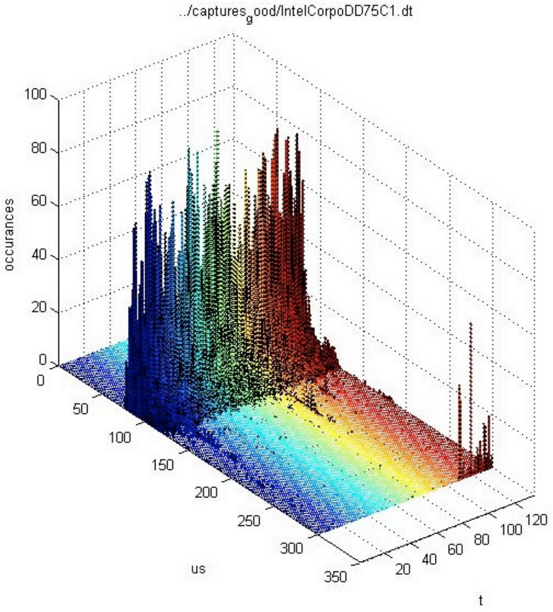


Figure 9.10: Significant Histograms of 400 ACK delays each, over several time periods t

9.1.3.1.1 Features An initial evaluation of WLAN traffic showed us, that the ACK delays of different packets vary from WLAN chipset to chipset and therefore could be used to identify such chipsets. By analyzing the spectrum of the ACK delays of the same chipset we can derive a histogram that represents the number of packets over the various observed delay times. In addition we capture the packet size in order to find out whether the ACK delay also depends on the packet size. The packets of a session – the time frame, where packets of a given chipset are captured – are arranged in the histograms in the following way:

1. Collect the ACK delays for each session of traffic generated by different WLAN chipsets.
2. For each 50 packets, create a 3D histogram which stores the frequency of the packets with a specific ACK delay and packet size. Each histogram is converted into a feature vector used for SOM training and classification.
3. Train a SOM tree with the histograms of the different WLAN chipsets.
4. The trained SOM tree is used for the classification of new traffic.

The length of the feature vectors depend on the number of analyzed ACK delay values (indicated as n) and packet size values (indicated as m). By storing the number of packets for given delay values and packet sizes we gain a 3D histogram that can be converted into a feature vector with $f = n \times m$ entries. In order to keep the feature vectors at a feasible length, we need to map delay and packet size ranges into single values. E.g. by considering ACK delay values from 1 ms to 300 ms ($n = 300$) and packet size values from 1 byte to 1600 bytes ($m = 1600$) we would get feature vectors with $f = 300 \times 1600 = 480000$ entries, which is not feasible. However, this resolution is not needed and on the contrary would decrease the accuracy of the classifier. Therefore, we reduce the number of features by mapping several ACK delay values and packet size values into bins representing value ranges. E.g. if we use a bin size of 10 ms for the ACK delay (then $n = 30$) and a bin size of 40 bytes for the packet size (then $m = 40$) the feature vector length is reduced to $f = 30 \times 40 = 1200$.

In Figures 9.11 and 9.12 two histograms based on delay information only (packet size is ignored) are shown. We observe that there is a significant difference between the analyzed chipsets. The role of the packet size combined with the delay values is visualized in Figures 9.13 and 9.14. Here we observer that the ACK delay values also depend on the packet size – at least for certain chipsets. In Figure 9.13 the captured data of the Agere chipset clearly shows that there is such a dependence. In contrast this dependence cannot be observed when analyzing the Edimax chipset (Figure 9.14).

By integrating both features into the classification process we are able to increase the accuracy compared to classifiers based on the delay information only. For further details and evaluation results we refer to the results section.

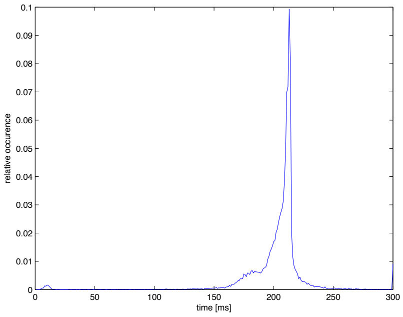


Figure 9.11: Delay histogram without packet size for an Orinoco chipset

9.1.3.2 WiFinger

As proof of concept of our approach, a small linux command line utility called WiFinger was developed and implemented in C/C++. The implementation was kept small in anticipation of possible use on handheld PCs as passive scanning devices. Additionally to our code just a small number of libraries was used (*libpcap*[1], *libSom* [142] and *ncurses*).

[1] http://www.tcpdump.org/

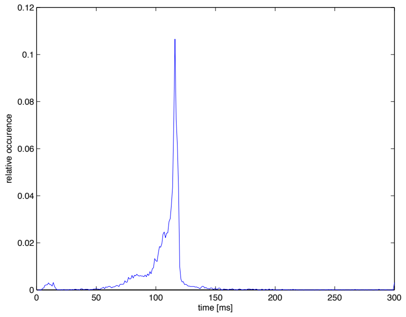


Figure 9.12: Delay histogram without packet size for a Broadcom chipset

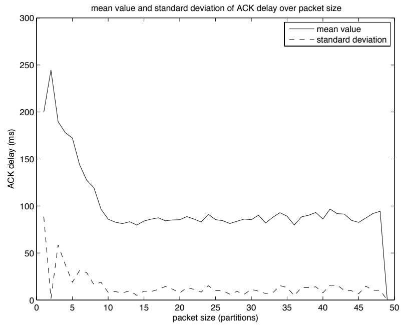


Figure 9.13: Agere (Chipset 2): Dependency between packet size and ACK delay values

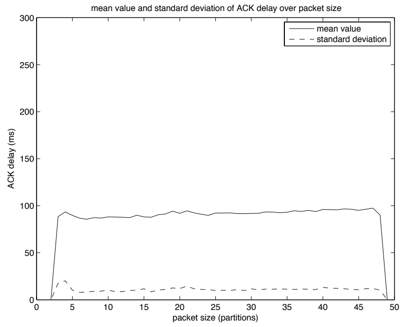


Figure 9.14: EdimaxTech (Chipset 6): Here, we cannot observer a dependency between the packet size and the ACK delay values

Two possible modes are provided by the application. Frames may either be captured live from the wireless network or be loaded from a previous capture-session file in *libpcap* format. Only the first 24 bytes of each IEEE 802.11 frame e.g. only the frame header bytes are examined. The payload itself plays no role in the classification task but the overall packet size does because of its influence to the CRC processing time. The architecture allows the collection and analysis of data on distributed devices.

9.1.3.2.1 Data Processing Figure 9.15 shows how data is processed by WiFinger. After *capturing* and *feature measurement* the measured delay and associated host information is handled in two steps. First, the collected data is saved unfiltered onto the hard-disk and converted in a format that could be imported by Matlab®. Second, the data is filtered and added to the *Significant Histogram*. Hosts are distinguished by their MAC address. The accuracy of the *Significant Histogram* increases with time. Per default, classification is run every 1000 measurements.

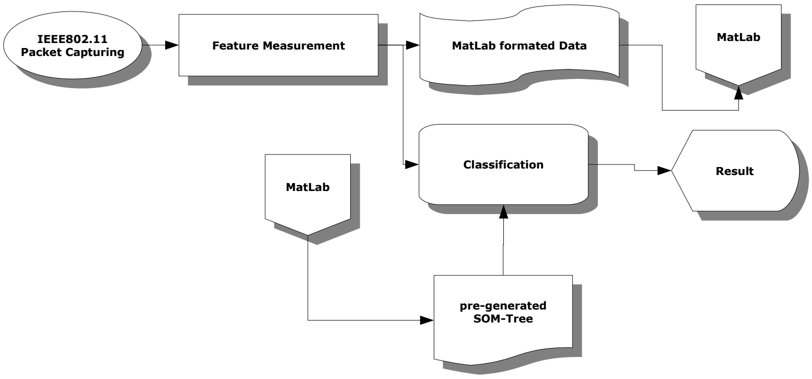


Figure 9.15: WiFinger Dataflow

9.1.3.2.2 Feature Measurement Figure 9.16 illustrates the process of feature measurement. Depending on the type of the captured frame, one of three states is entered: IDLE, CAPTURED DATA FRAME and CAPTURED ACK OF DATA. Recognized types are data frames and frames containing an ACK. During *contention free periods*, DATA frames can contain *contention free acknowledgments*, hence referred to as CF-ACK. During these periods DATA frames with embedded CF-ACKs may appear in direct succession of each other. Thus as a special case, the states LAST FRAME WAS DATA and DATA/ACK PAIR CAPTURED can be entered during the same pass. MAC addresses are read from the frame header's *address 1 field* which always contains the wireless destination station and the *address 2 field* which always contains the sending

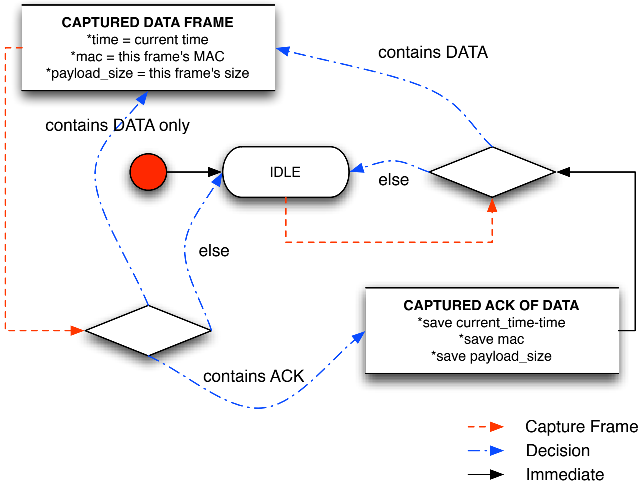


Figure 9.16: WiFinger Feature Measurement FSM

wireless station [75]. Both fields are 6 bytes long and start at byte 4 and 10 respectively.

The destination station's MAC address, payload size and the time of reception of the latest data frame are saved in temporary variables. If the frame acknowledges the immediately previously sent data packet, the *acknowledge delay* is measured as time between the reception of the last data frame and the reception of the acknowledging frame (see Figure 9.8). Note that interval is longer than the *Shortest InterFrame Space*, since the delay of receiving the data frame is added.

Measurements on broadcast addresses are discarded. Since an ACK frame carries only a destination address (see figure 9.9), it is possible that on-air data is missed and a later ACK frame is mistaken for an expected acknowledgment. To minimize such mistakes, the sending stations address of the previous data frame is checked against the destination address of the following ACK frame. This only works outside of *contention free periods* since the destination address of frames containing CF-ACK does not need to match the address of the previously transmitting station. The following frame types are relevant during *contention free periods* [75]:

- CF-End + CF-ACK
- Data + CF-ACK
- Data + CF-ACK + CF-Poll
- NoData + CF-ACK
- NoData + CF-ACK + CF-Poll

9.1.3.2.3 Usage of Matlab®for SOM training Figure 9.15 depicts the interaction between WiFinger and Matlab®. For each host WiFinger writes a simple tab and newline delimited text file. Measurements exported for use in MatLab®are not preprocessed by WiFinger. During the experimentation process this provided more flexibility in finding the best parameters for classification. Like in [142] the used scripts produce a SOM-tree which is then automatically loaded by WiFinger.

9.1.3.2.4 Usage of libSom for SOM classification *libSom* [142] provides loading of SOM-trees, datatypes for SOMs and vectors as well as classification functionality. Classification works as described above in section 9.1.3.1. For each host a SOM-Vector is used to save a histogram of ACK delays. The file *somconfig* generated by the scripts was extended to save the parameters used in the training of the SOM-tree. These are:

- minimum
- maximum
- number of subdivisions

for each of the two features, acknowledge delay and data frame size.

Captured values above the given limits (500ms) are discarded. If the number of values between minimum and maximum differ from the number of subdivision, values are scaled to fit the chosen resolution. After a variable number of measured ACK delays, a copy of the SOM-Vector is normalized and the resulting *Significant Histogram* is classified. Optionally, a number of measurements can be defined after which the histogram is reset.

9.1.3.3 Results

For performance evaluation, the packet size and ACK delay time was taken from real traffic data with WiFinger. To facilitate this, an Aironet 350 access point was set up with Internet access. Traffic was measured using a 802.11b wireless NIC with Orinoco chipset, capable of capturing all layer 2 data in *rfmon* mode. In turn, six different wireless NICs with known chipsets were used to generate the traffic. The access point shows up in the classification results as chipset 3.

60% of this data was used for the SOM training process. The remaining test data was used to create simulated sessions with 500 respectively 1000 packets. This session based classification should give a hint on how much data from a chipset is needed to get an accurate classification result. Table 9.2 gives details on the evaluated chipsets and the training respectively test sets.

For performance evaluation several parameters were evaluated:

- **Number of used features**: By using the packet size information in addition to the ACK delay time information, the feature space can easily

Chipset	**training/test data** in packets	**test data** sessions 500/1000 pkts
1: Atheros	35105/23404	47/24
2: Agere	53006/35337	71/36
3: Aironet	292858/195239	391/196
6: RaLink	19265/12843	26/13
7: BCM4306	149864/99910	200/100
8: Intel2100	41390/27594	56/28
9: PRISM	16076/10714	22/11

Table 9.2: training/test data sets

exceed reasonable boundaries. Thus, the ACK delay and packet size information needs to be grouped. This is done by specifying the number of delay partitions and the number of size partitions. A short example explains this grouping: If the packet size from 1 byte to 1600 bytes and the delay time from 70 ms to 500 ms are taken into consideration, the size of the feature space would be $1600 \times 430 = 688000$. By using 40 partitions for the delay time and 40 partitions for the packet size, this size can be lowered to $40 \times 40 = 1600$ features. The grouping is done by mapping 1 byte to 40 bytes to the first feature, 41 bytes to 80 bytes to the second feature and so on. The same procedure is applied to the ACK delay information.

- **Time/packet size range**: These parameters set the range of ACK time delay and packet size which is used for feature generation. For the evaluation of WiFinger we used a range of 70 ms to 500 ms for the delay information and 1 byte to 1600 bytes for the size information.
- **Histogram size:** This parameter is used to set the number of packets which are used to create a histogram. For our tests we evaluated a setting of 50 packets per histogram.
- **Session size:** This parameter is used to create sessions from the test data sets. The evaluation of different sessions sizes gives information about how many packets need to be analyzed before an accurate classification can be made.
- **Training factor:** This factor is used to separate the whole data set into training and test data. We used a setting of 0.6 for all tests.

The classification results were obtained in this way:

- Data was collected with WiFinger for seven different chipsets.
- The parameters described above were tuned to evaluate the impact of delay and packet size features.

- The overall classification results were obtained by getting the number of correctly classified time/size histograms for the whole test dataset.
- The dataset was split into sessions with 500 and 1000 packets. This should give an indication on how many packets are needed in order to get an accurate classification result.
- The combination of delay and packet size features which gave the best overall results was evaluated with sessions 500 and 1000 packets.

The first row of Table 9.3 shows the results when using delay information only. Tests from row 2 - 5 evaluate the performance of different feature sets and indicate that adding packet size information can significantly increase the classification performance. In case of 40 time slots and 40 packet size slots 80% of all histograms were classified correctly, which is an increase in classification accuracy of 64% over the first version, which only classifies 51,4% of all histograms correctly. These parameters result in a feature vector with 1600 entries, which is quite large. However, row 2 shows, that the classification performance only drops slightly when using just 10 features for packet size. The feature vector size is also reduced to 25% (400 instead of 1600), which is even lower than the feature vector size used in row 1, where only delay information is taken into consideration.

feature range	time slots	size slots	vector size	results
70-500 ms (time only)	430	1	430	51,4%
70-500 ms, 1-1600 bytes	**40**	**10**	**400**	**74,4%**
70-500 ms, 1-1600 bytes	10	40	400	69,8%
70-500 ms, 1-1600 bytes	20	20	400	68,7%
70-500 ms, 1-1600 bytes	**40**	**40**	**1600**	**80,0%**

Table 9.3: Comparing the impact of different features

It is necessary to be careful with the number of features used for the packet size. As real data is used for SOM training, it is not guaranteed that this data has an equal distribution of packet sizes over all chipsets. Thus, by using a too fine resolution for the packet size (meaning a large feature space), the algorithm learns to classify the chipsets according to the packet size.

The parameters which gave the best classification accuracy (row 5) were used to rerun the experiment with 500/1000 packets per session. These sessions were created by using data from the test sets. The results of this evaluation can be seen in Table 9.4 and 9.5. The following conclusions can be drawn from the results:

- There is only a very small performance increase if 1000 packets instead of 500 are used per session. The session size needed for an accurate classification result depends on the type of analyzed data. Generally, increasing the session size increases classification accuracy as noise is reduced.

- Most of the classification errors are made, when chipsets are classified as chipset 7 (PRISM 3). It seems that this chipset is quite similar to the other ones tested. Furthermore, the training set for chipset 7 was rather small. As the classification is based on the hits on the SOM, noise plays a larger role when smaller training sets are used.

	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**89,4**	0	2,1	8,5	0,0	0,0	0,0
2	9,9	**16,9**	0,0	1,4	5,6	28,2	38,0
3	0,0	0,0	**93.6**	0,0	6,4	0,0	0,0
4	0,0	0,0	0,0	**34,6**	0,0	0,0	65,4
5	0,0	0,0	0,0	0,0	**100,0**	0,0	0,0
6	0,0	3,6	0,0	0,0	0,0	**82,1**	14,3
7	0,0	0,0	0,0	0,0	0,0	0	**100,0**

Table 9.4: Confusion matrix for 500 packet sessions. E.g. 89,4 % of chipset 1 (in row 1) sessions are classified correctly as chipset 1, 0 % as chipset 2, 2.1 % as chipset 3, etc.

	1	**2**	**3**	**4**	**5**	**6**	**7**
1	**95,8**	0	4,1	0,0	0,0	0,0	0,0
2	11,1	**13,9**	0,0	0,0	5,6	22,2	47,2
3	0,0	0,0	**93.4**	0,0	6,6	0,0	0,0
4	0,0	0,0	0,0	**30,8**	0,0	0,0	69,2
5	0,0	0,0	0,0	0,0	**100,0**	0,0	0,0
6	0,0	0,0	0,0	0,0	0,0	**85,7**	14,3
7	0,0	0,0	0,0	0,0	0,0	0	**100,0**

Table 9.5: Confusion matrix for 1000 packet sessions. E.g. 95.8 % of chipset 1 (in row 1) sessions are classified correctly as chipset 1, 0 % as chipset 2, 4.1 % as chipset 3, etc.

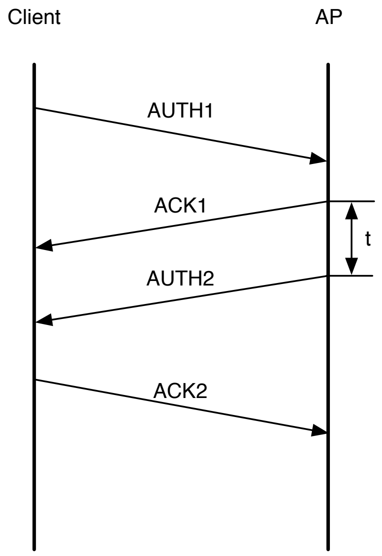


Figure 9.17: Radio Frame Exchange During the Authentication Procedure

9.1.3.4 Discussion

The proposed method uses the delay time between a data frame and the belonging ACK to identify chipsets. For an accurate classification result, 500 to 1000 values are needed. This values can be obtained by passive monitoring or by actively sending packets to the chipset that needs to be identified. In contrast to the method of Franklin et al. the amount of data needed for the accurate representation of the chipset fingerprint can be obtained quite easily, due to the fact that each packet needs to be acknowledged with an ACK packet.

The proposed fingerprinting method cannot differentiate between WLAN NICs containing the same chipset since in this case the extracted ACK delay information is the same. This represents a limitation to the broad use of this technique.

9.1.4 Active Fingerprinting by Timing Analysis

Bartolomiej Sieka's work on device fingerprinting [163] is probably the one closest related to our presented in Section 9.1.3 approach. It uses the time t that elapses between the first acknowledgement $ACK1$ is sent and the moment the authentication response $AUTH2$ is sent (see Figure 9.17). For classification purpose, support vector machines are used. The drawback of this approach is its limitation to the authentication phase for measurements. As this phase only occurs during the initialization of the connection, Sieka actively needs to provoke the repetition of it by sending specifically crafted 802.11 frames. This could be detected by an intrusion detection system or the device to fingerprint, allowing it to counteract. Our approach is immune against such countermeasures as it is absolutely passive. Sieka's classification results are comparable to our results.

9.2 User Identification

The area of user identification is still a rather uncharted one. This section shortly describes techniques related to our own contribution in this field.

9.2.1 Related Work

Eagle and Pentland [43] of MITs Media Lab developed a vector based scheme called *Eigenbehavior* that allows to quantify the behavior of a user in order to predict her next actions. This should introduce more interactivity in browsing webpages and allow networks and services to prepare contents in advance. As far as we know this approach has yet not been used for identifying users in order to attack their privacy.

Liu and Peng [114] from the University of North Carolina were addressing the problem of mutual trust in pervasive computer systems. They used unique identifiers as hardware addresses to identify devices. Further on they track their behavior by analyzing network event logs to find participants with hostile behavior and consequently level down trust to this nodes to prevent them from harming the rest of the network. Although the fingerprinting method is not very sophisticated, the approach of creating user profiles in order to assess trustworthiness is related to our ideas.

Pang et al. [139] followed a similar approach as we did as they tried to find characteristics in user behavior by analyzing their network traffic. The main difference is that most of their features are based on IEEE 802.11 MAC Layer properties while we focus on information extracted from higher layers. While Pang et al. need to analyze traffic captured in the wireless cell the target is actually stationed, we are able to collect all traffic at a central place in the backbone of the underlaying network.

9.2.2 Behavioral Fingerprints and Knowledge Mining

Based on the discussion in Section 9.1, we conclude that device fingerprints are rather unreliable or too costly to be applied in real world scenarios. Therefore, this approach concentrates on another approach – behavioral fingerprints. Taking such a fingerprint of any user means to find characteristic features that describe her behavior and thereby allows the identification and tracking of the user. Such features can be derived from a wide variety of available data, ranging from lower layer network packets to high level application related traffic. In addition the extracted features can be subject to further sophisticated analysis processes that extract information about users or user groups. Such analysis methods are also employed in other research areas that are focused on knowledge mining. In previous and ongoing work in the areas of e-participation [178], event correlation [177], malware analysis [175] and semantic RDF analysis [176], we presented the concept of *Activation Patterns* (see Section 8.3 for a detailed description) that allows us to use a single model as a basis for a wide range of analysis methods. The basic idea behind this concept is to transform raw data into a new representation that models the relation between the analyzed features. Although, there are machine learning methods that could be used for particular analysis procedures, none of these methods has the flexibility of *Activation Patterns* and their wide range of applications.

9.2.3 Email Analysis

In this approach we utilize the *Activation Pattern* (see Section 8.3) concept for the analysis of email data. In addition to generating behavioral fingerprints for user identification and tracking the technique enables us to extract further valuable information about the underlying dataset. The decision to utilize emails – to be precise the headers of emails – for this first evaluation is based on the following reasons:

- We need to get a better understanding of the capabilities of the employed *Activation Patterns* before we can apply them to other data. In case of emails the extracted features are easy to understand for humans and therefore the results gained by employing the *Activation Patterns* concept can easily be verified.
- The lessons learned by the application of *Activation Patterns* will be of benefit for future work that will concentrate on a wide range of features extracted from different abstraction layers.
- For emails, one could use the from address as unique ID for tracking, however this address can easily be forged. Therefore, we do not take it into consideration for the fingerprinting process.
- Although, VPNs, HTTPS and TLS POP/IMAP/SMTP connections are an effective countermeasure against extracting the analyzed features, there is still a large number of unencrypted POP3, IMAP and especially SMTP traffic that is vulnerable to this kind of analysis.
- In case of unencrypted connections, SMIME and other email encryption techniques are not an effective countermeasure, since the proposed method relies completely on information extracted from the email headers which are always transmitted in plaintext.

The transformation of raw emails into *Activation Patterns* is based on five process layers depicted in Figure 9.18. After extracting and preprocessing the email features we apply the four layers L1-L4 to the raw feature data in order to determine the *Activation Patterns.* The techniques within these layers are based on various concepts related to machine learning and artificial intelligence: semantic networks for modeling relations within data [47, 148, 184], spreading activation algorithms (SA) [33] for extracting knowledge from semantic networks,

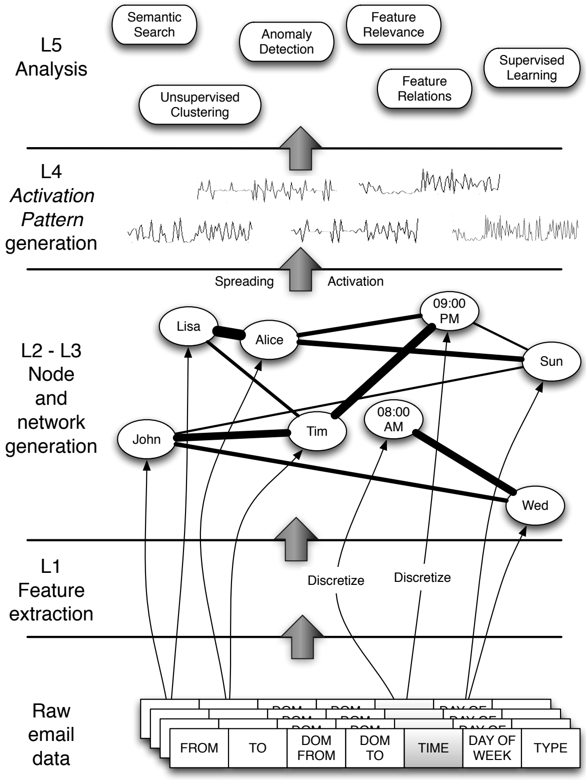


Figure 9.18: Transformation of raw data into *Activation Patterns*

and supervised/unsupervised learning algorithms to analyze data extracted from the semantic network [119, 147]. For a detailed description of the *Activation Pattern* technique, we refer to Section 8.3.

9.2.4 Fingerprinting and Further Analysis

In this section we apply various analysis methods to the transformed email *Activation Patterns*. For the analysis we have extracted 8 features (see Table 9.6 for a complete list) for each of the 1708 emails belonging to 13 users. For *Feature relations* and *Semantic Search* the UF (user from) feature was included in the analyzed patterns. For *Feature Relevance*, *Unsupervised Clustering* and *Supervised Learning* we excluded the UF feature in order to find out to what extend UF could be identified by the other features.

Feature	**Type**	**Abbr.**	**Description**
User From	Nominal	UF	Sender address without domain name
Domain From	Nominal	DF	Domain name of the sender address
User To	Nominal	UT	Receiver address
Domain To	Nominal	DT	Domain name of the receiver address
Time of Day	Distance Based	TD	Timestamp of the message without the date
Day of Week	Nominal	DW	Date of the message time stamp
Content Type	Nominal	CT	Content-type field of the email header

Table 9.6: Features

User	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
FN	0.0	0.31	0.23	0.16	0.12	0.40	0.07	0.23	0.01	0.14
FP	0.07	0.31	0.18	0.1	0.18	0.47	0.16	0.04	0.12	0.13
TP	**1.0**	**0.69**	**0.77**	**0.84**	**0.88**	**0.60**	**0.93**	**0.77**	**0.99**	**0.86**

Table 9.7: Performance per user (excerpt): FN - false negatives, FP - false positives, TP - true positives. Since these are the mean values for the 10 iterations, the sum over the columns is not 1

9.2.4.1 Supervised learning for the creation of behavioral fingerprints

The *Activation Patterns* of emails that belong to a given user represent a fingerprint that can be used to identify and thereby track users. Further knowledge, such as the feature relevance, extracted by the previous methods can be integrated into the *Activation Patterns* in order to create more robust fingerprints.

For the creation of fingerprints, we apply a neural network[2] to the *Activation Patterns* of 13 users. The training data is generated by randomly taking 50% of the emails of each user. This results in 857 emails in the training data set and 851 in the test data set. The network is then trained and evaluated on these two data sets. In order to increase robustness, the whole process is repeated 10 times. Table 9.7 shows the results for each user. The mean value of correctly classified emails is calculated for the 10 iterations and yields 88.36%. We note, that user sessions, which typically contain more than one email, are not taken into account. By taking this information into account, the classification accuracy will be increased further.

[2] The Matlab™Neural Network toolbox is utilized. Except for the validation check parameter, the standard parameters are used: Training: Scaled Conjugate Gradient, Performance: Mean Squared Error, Validation Checks: 20 instead of 6, Max Epochs: 1000, 20 hidden units, the training data is separated into 60% training data, 20% validation data and 20% independent test data.

Relation 1	DT:wizzards.com
UF	gandalf (0.5), merlin (0.2), saruman (0.2)
DF	wizzards.com (1.0), dragons.com (0.1), hobbits.com (0.1)
DT	dwarfs.com (0.1), elfs.com (0.0), orks.com (0.0)
TD	08:46 (0.4), 16:08 (0.3), 13:12 (0.3)
DW	Mon (0.3), Wed (0.3), Tue (0.3)
CT	5 (1.0), 1 (0.4), 3 (0.3)
Relation 2	TD:60 (in minutes, meaning 01:00 a.m.)
UF	aragorn (0.2), ermurazor (0.1), fellowship (0.1)
DF	giants.com(0.3), nazgul.com (0.3), elfs.com (0.1)
UT	ermurazor (0.3), denetor (0.3), aragorn(0.3)
DT	nazgul.com (0.9), giants.com (0.6), elfs.com (0.2)
DW	Thu (0.4), Fri (0.4), Wed (0.3)
CT	5 (1.0), 1 (0.4), 3 (0.3)
Relation 3	TD:600 (in minutes, meaning 10:00 a.m.)
UF	tower (0.3), gandalf (0.2), gimli (0.2)
DF	wizzards.com (0.4), horadrim.com (0.2), dwarfs.com (0.2)
UT	tower (0.7), gandalf (0.3), mithrandir(0.1)
DT	dwarfs.com (0.6), wizzards.com (0.5), nazgul.com (0.4)
DW	Mon (0.6), Tue (0.6), Wed (0.5)
CT	1 (1.0), 5 (0.8), 3 (0.5)

Table 9.8: Relation between feature values

9.2.4.2 Feature relations

The links within the semantic network represent the relations between the different feature values. By activating one or more nodes within the network and applying spreading activation the related nodes receive activation energy. The strength of the received activation energy depends on the strength of the relations. In Table 9.8 we show several examples for analyzing the relations between different features. In all examples three nodes with the strongest activation values are extracted for each feature.

Relation 1: DT: The node representing the domain *wizzards.com* is activated and the links to other nodes are analyzed, the results reveal other users that are closely related to the given domain: *gandalf*, *merlin* and *saruman*.

Relation 2: TD: In this case the node representing the time 01:00 a.m. was selected. The results allow us to find users and domains that are typical for emails written at that time.

Relation 3: TD: In this example the time 10:00 a.m. was selected. When compared to the results of Relation 2, we can see that other users and domains are involved in the morning than during the night. This difference indicates that the time of day feature adds valuable information for discriminating users.

9.2.4.3 Semantic search

Due to the relations stored in the semantic network, we are able to apply semantic search queries to the data in the following way: We activate one or more nodes within the network, spread their activation to neighboring nodes and extract the generated *Activation Pattern* from the network. For this extracted *Activation Pattern* the distance to the other patterns can be calculated. As distance measure, the cosine similarity is used:

Query 1: UT: We activate the node for the user *lembas*, which only occurs once in the whole data set, spread the activation and compare the resulting *Activation Pattern* to the others. Obviously, the best matching result contains the user *lembas*. Since the *Activation Patterns* maintain the information about semantic relations, we are also able to retrieve other search results, that do not contain the given user, but are related to this user due to other features.

Query 2: DT: In this example we activate the node for the domain *wizzards.com* and search for emails that are semantically related. As the results show, we are also able to retrieve emails that contain other domains, but are still related to *wizzards.com* due to the involved users.

Query 3: DT and UT: In this case we activate the same nodes as in Relation 4 (UT: *gandalf* and DT: *wizzards.com*). Obviously, the first results contain the feature values specified in the search query. However, as we can see in results 648 and 650, we are also able to retrieve mails without those feature values, but that are still semantically related due to other features. For 648 this is the user *saruman* and for 650 it is the user *merlin*. We have already found out in Relation 3, that those users are strongly related to UT: *gandalf* and DT: *wizzards.com*.

9.2.4.4 Feature Relevance

By analyzing the number and the strength of the links emanating from a given unit, we are able to filter out features that do not carry information. In this case the different content-types, certain days and domains are identified by the analysis (see Table 9.10). This is not surprising, since these features and values are shared by a large percentage of users and thus do not carry important information.

9.2.4.5 Unsupervised clustering

Unsupervised clustering groups similar instances into clusters. These clusters enable the user to gain a quick overview of the whole data set. For this evaluation we apply the Neural Gas based RGNG algorithm [147] to the *Activation Patterns*. However, it would also be possible to apply any other clustering algorithm to the patterns. In Table 9.11 we give two examples for the 22 clusters that were found by the RGNG algorithm. For each feature we extract the three *most activate feature values*. Cluster 1 covers the 98 emails of a single user, whereas Cluster 2 covers 4 users that have similar communication partners.

Query 1	**Query for UT *lembas***				
Result	**UF**	**DF**	**UT**	**DT**	**TD**
1	gandalf	hobbits.com	lembas	hobbits.com	11:10 a.m.
2	gandalf	hobbits.com	frodo	hobbits.com	01:12 p.m.
9	frodo	hobbits.com	gandalf	hobbits.com	11:10 a.m.
50	frodo	hobbits.com	mithrandir	trolls.com	11:10 a.m.
1496	sauron	eagles.com	boromir	elfs.com	01:12 p.m.

Query 2	**Query for DT *wizzards.com***				
Result	**UF**	**DF**	**UT**	**DT**	**TD**
1	gandalf	wizzards.com	merlin	wizzards.com	08:30 a.m.
35	saruman	wizzards.com	gandalf	wizzards.com	08:30 a.m.
210	gandalf	wizzards.com	tower	dwarfs.com	01:12 p.m.
321	saruman	wizzards.com	ankantoiel	gmail.com	06:42 p.m.
996	gollum	ents.com	tower	dwarfs.com	06:42 p.m.

Query 3	**Query for DT *wizzards.com* and UT *gandalf***				
Result	**UF**	**DF**	**UT**	**DT**	**TD**
1	saruman	wizzards.com	gandalf	wizzards.com	08:30 a.m.
648	saruman	wizzards.com	durin	ringwraiths.com	01:12 p.m.
650	merlin	wizzards.com	faramir	urukhais.com	01:12 p.m.

Table 9.9: Semantic search queries (excerpt)

CT	5 (unknown)
CT	1 (text-plain)
DT	nazgul.com
CT	3 (multipart-alternative)
DW	Tue
DW	Mon
DT	giants.com
DW	Fri

Table 9.10: Examples for feature values with low relevance

9.3 Conclusion

This chapter shows that identifying devices and users by creating and classifying fingerprints is technically feasible. It presents various techniques for device fingerprinting on different layers and analyses their performance and applicability. The focus of device identification is pointed on MAC layer fingerprinting techniques as those promise the best results and real world applicabilities.

The chapter further on presents our approach on creating fingerprints of users

Cluster 1	98 emails and 1 user
DF	nazgul.com (0.8), wizzards.com (0.1), giants.com (0.1)
UT	ermurazor (0.8), denetor (0.2), tower (0.2)
DT	nazgul.com (1.0), wizzards.com (0.2), dwarfs.com (0.2)
TD	03:33 p.m. (0.3), 10:57 a.m. (0.3), 01:03 p.m. (0.2)
DW	Wed (0.3), Thu (0.3), Tue (0.3)
CT	5 (0.9), 1 (0.4), 2 (0.2)
UF/emails	ermurazor/98

Cluster 2	101 emails and 4 users
DF	wizzards.com (0.9), dragons.com (0.1), hobbits.com (0.1)
UT	merlin (0.7), tower (0.2), gandalf (0.2)
DT	wizzards.com (1.0), dwarfs.com (0.2), nazgul.com (0.1)
TD	03:33 p.m. (0.4), 08:52 a.m. (0.3), 01:03 p.m.(0.3)
DW	Tue (0.5), Wed (0.3), Mon (0.3)
CT	6 (0.8), 5 (0.8), 1 (0.3)
UF/emails	gandalf/86, saruman/8, stormcraw/5, merlin/2

Table 9.11: Examples for clusters, the activation strength is normed and denoted within parentheses, 1.0 represents the strongest activation

based on their behavior. It discusses the results of our prototype based on email analyses and possible future improvements.

Concluding it has to be said that fingerprinting techniques can be used by attackers and security professionals as well. In the same way as they are a threat to privacy they can act as an additional layer in intrusion detection systems.

While this additional security my be desirable in high security environments, the possible loss of privacy is unacceptable for publicly accessible networks. It is therefore necessary to develop countermeasures against fingerprinting in order to be able to preserve privacy when required.

10 Preserving Location Privacy in Wireless Networks

Although, the problem of attacking the *Location Related Privacy* has been briefly presented in the introduction of the part, implications to location privacy by tracking users and devices may best be explained by an example scenario:

With the growing requirement of pervasive connectivity, new business models have emerged for network service providers. Cellular network based services are usually too expensive for exhaustive broadband usage. Therefore, wireless computer networks such as IEEE 802.11, or better known as *wireless fidelity* (WiFi) offer higher bandwidth at a lower cost, especially if a high-speed wired backbone network is available. This is mostly the case in metropolitan areas.

More and more Internet service providers (ISP) are boarding this train and install large-scale wireless networks in metropolitan areas. But also cellular network operators are using their backbone infrastructure to install WiFi access points in crowded areas such as city centers, shopping malls or airports. This trend is also promoted by the availability of low-cost mobile hardware like smart phones and netbooks, with excellent abilities to connect to WiFi networks.

As many large metropolitan areas provide area-wide wireless network access, it is massively used by a growing crowd of people. However, a lot of these people are not aware of the security and privacy risks posed by wireless communications. Even if a user uses state-of-the-art encryption for her communication, a lot of personal information can be obtained from analyzing her traffic.

This example scenario illustrates privacy violations due to the threat of tracking the location of a user in a large-scale wireless computer network. Apart from determining the location of a user, the tracking process enables the collection of user-data at different times of days and locations. This data can then be used for further sophisticated analysis processes that disclose information about users or user groups.

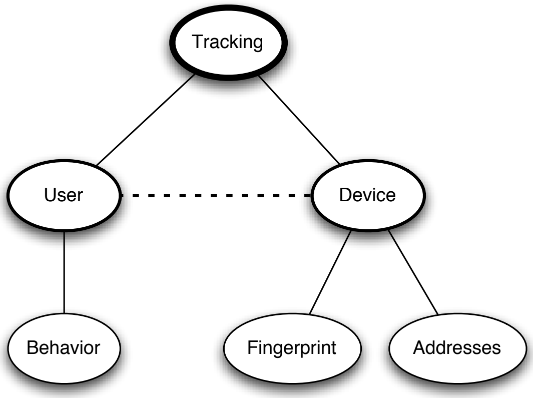


Figure 10.1: Overview of the Term Tracking [112]

Figure 10.1 tries to classify the term tracking as it can be regarded in connection with wireless networks. The dotted line illustrates the loose connection between the identification of a device and its user.

The main problem of tracking in a massive multi user setting is to identify the traffic of a particular user out of a myriad of network packets. It may be possible to identify a user by the device she is using to connect to the network. In the best case, a device can be identified by its hardware address as briefly discussed in Section 9.1. If we cannot bind a user to a single device or a device is forging its hardware address, more sophisticated methods as described in Section 9.2 need to be applied.

Figure 10.2 contains a map of public WiFi hotspots in New York City. It should only serve as an illustration of how a public metropolitan area wireless network could be spread out over an urban area. If an attacker would be able the determine the connection of a wireless device with any of these access points the movements of this device could be tracked in the whole city. As will be further discussed in Section 13, the usage of ultra mobile devices as PDAs or smart phones provides seamless, non-stop connectivity, seriously favoring the threat of being tracking. It can easily be imagined which consequences can arise if e.g.:

- a terrorist can track his potential targets;
- a criminal can determine the position of law enforcement units;
- a company can track the whereabouts of their employees; or
- a jealous person could monitor his or her partner.

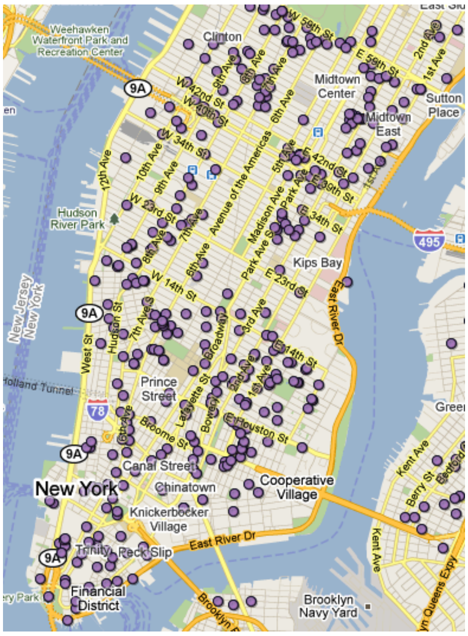


Figure 10.2: WiFi Hotspots in Manhattan [56]

These are just some trivial examples illustrating the danger posed by compromised location privacy. Dave Singelee states in [165], that the goal of location privacy enhancing/preserving techniques has to be, to establish *untraceability* and *unlinkability* [... to the device ...] at the protocol level. For the following discussion, we use the following extended definitions by [165]:

> **Untraceability**: It should be computationally hard for an attacker, who observes the exchanged messages, to detect which specific device/user is participating in the communication.
>
> **Unlinkability**: It should be computationally hard for an attacker to link messages to one sender and/or receiver, even without knowing the exact identity of this device/user.

10.1 Preserving Location Privacy in Wireless Networks

We agree with the assumption of Dave Singelee [165] that untraceability and unlinkability at lower layers improve location privacy significantly. This accounts for mesh as well as for infrastructure based wireless networks regardless their transmission range.

A naive approach to achieve these properties would be to frequently and randomly change the identification of the communication devices. As presented in

Part I, major network management mechanisms like *association*, *authentication*, *routing*, *roaming* and *device pairing* rely on singular device identifiers like MAC addresses in the case of WiFi or BD_ADDR for Bluetooth. Changing device IDs without considering these functionalities could seriously degrade network performance and user acceptance. Nevertheless, changing device IDs seems to be the best option. The following sections present current approaches to achieve or improve location privacy in WPANs and WLANs.

10.1.1 Privacy Preserving Techniques for Wireless Personal Area Networks

This section provides a short overview on privacy preserving protocols by Gerhrmann and Nyberg [54], Wong and Stajano [192], and Singelee and Preneel [167].

10.1.1.1 Anonymity Mode

Gerhrmann and Nyberg were the first to propose changing device IDs to improve location privacy in Bluetooth [54]. Their approach called *Anonymity Mode* is based on the usage of *pseudonyms*. They replace long-lived hardware identifiers (BD_ADDR) by short-lived, randomly chosen addresses (BD_ADDR_ACTIVE). Update intervals are following fixed time intervals and at device power-up. To avoid the necessity of re-pairing Bluetooth devices after each random address change, they assume that the master device has knowledge of the slave device's BD_ADDR and that after the connection is established on the baseband, special messages are exchanged to agree on temporary pseudonyms. This first approach to the location privacy problematic in Bluetooth was rather theoretical but inspired significant improvements, presented in the following sections.

10.1.1.2 Protected Stateful Pseudonyms

The approach by Wong and Stajano [192] is based on Gerhrmann's and Nyberg's *Anonymity Mode*, where they uncovered three privacy weaknesses and proposed an enhanced solution called *Protected Stateful Pseudonyms*.

Their formal goal was to develop a privacy framework that provides sender and destination anonymity in a peer-to-peer ad-hoc wireless environment. Pseudonyms are used, and unlinkability between these pseudonyms should be provided.

While Anonymity Mode is stateless, this approach requires the clients to keep a database of tuples each containing the temporary pseudonyms of the communication parties and a shared link key. The first pairing is done according to Anonymity Mode, using random pseudonyms. Figure 10.3 illustrates the protocol run for repeated pairing.

The protocol uses three ID packets called ID1, ID2 and ID3. The past pseudonyms of Alice and Bob are I_A and I_B. H is a hash function and R_{1-3} are random nonces. $K_{A,B}$ is a shared Link Key formed by Alice and Bob in a former connection.

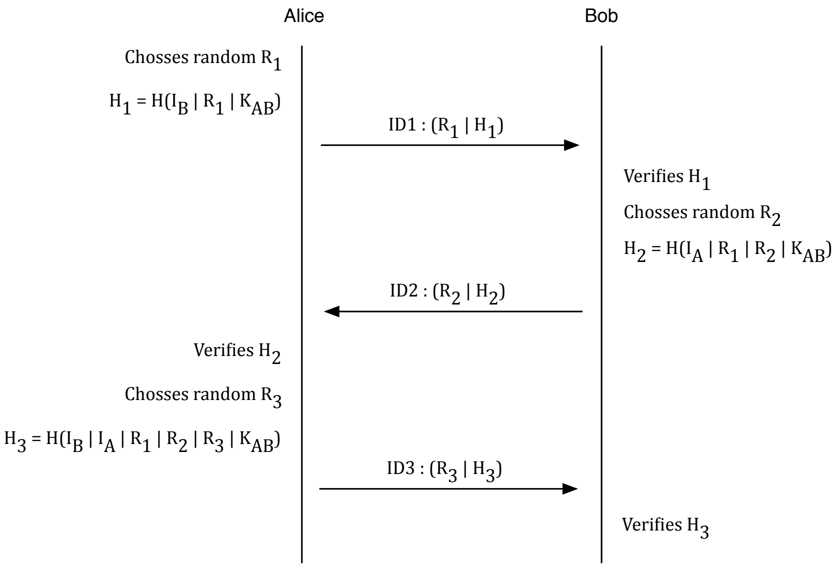


Figure 10.3: Protected Stateful Pseudonyms Protocol by Wong and Stajano [192]

If Alice wants to page Bob she starts by selecting a random nonce R_1, computes the hash H_1 and sends the packet ID1. The hash value in ID1 hides Bob's past pseudonym. Bob can now verify the hash using the list of the paired pseudonyms stored in his database. When he finds a match he choses nonce R_2, computes H_2 and responds with packet ID2. As the next step, Alice verifies H_2 using her database and R_2. She further on computes H_3 and sends ID3 to Bob who ends the protocol run by verifying H_3.

Following the principles of mutual authentication this protocol protects against a variety of attacks (see Section 3.6). Several triggers for a pseudonym change are defined by the authors:

- a manual request of the operator;
- a random time interval; or
- when the device discovers a certain number of other devices in transmission range to *hide in the crowd*.

10.1.1.3 Enabling Location Privacy

Dave Singelee investigated the topic of location privacy in wireless personal area networks very thoroughly [165]. He approached the problem by separating the possible communication scenarios and proposed an optimized solution for each of them:

10.1.1.3.1 Mobile devices share a symmetric key This scenario assumes that the communicating devices have some how established a shared key. Every device has in its memory a list of those keys, shared with other devices and a identifier R which is based on an initialization vector IV and computed as follows:

$$R = PRF_k(IV) \tag{10.1}$$

PRF is a pseudo-random function and k is the Shared Key. Singelee assumes that all devices in the WPAN use the same IV. The identifiers R are stored in the memory. When Alice wants to send a message to Bob for the first time she takes the corresponding R in the header of the message. Bob will recognize the identifier and knows that the message is for him. Now both devices can update their identifier R as follows:

$$R_{new} = PRF_k(R_{old}) \tag{10.2}$$

In order to avoid tracking, R should be updates for each new message. An eavesdropper can just intercept random identifiers and is not able to link them. This approach allows devices to calculate identifiers offline and in advance and provides the highest degree of location privacy.

10.1.1.3.2 Address known by other mobile device This scenario assumes that Alice knows the identifier of Bob but they do not share a key. One has to remark that identity based cryptography will not be considered here do to the fact that it is computationally expensive and consumes too much energy.

As in the first scenario, Alice includes Bob's identifier R_B into the initial message, letting Bob know that she intends a communication with him. In order to avoid tracking, this identifier as well as the used nonce has to be different in every message. R_B is computed as follows:

$$R_B = H(addrB, nonce) \tag{10.3}$$

Whereas H is a one-way hash function (see Section 3.3) and $addrB$ is the pre-known identifier of B. As the sent messages also include the sender address R_A, it also has to be different in each message and initially set to a random number by Alice. The reply of Bob will include an identifier R_{reply} which is computed as follows:

$$R_{reply} = H(R_A, addrB, nonce) \tag{10.4}$$

If Alice now wants to reply again, she also needs to compute a new identifier:

$$R_{reply} = H(addrB, R_A, nonce) \tag{10.5}$$

This is exactly done as in equation 10.4. Bob has stored the random identifier R_A and can hence detect that the reply comes from Alice.

Contrary to scenario one, this solution allows an attacker to track a specific device if she knows the hardware address. This attack cannot be avoided as the only difference between an attacker and Alice is the knowledge of the destination (Bob).

10.1.1.3.3 Secure out-of-band channel available This scenario assumes that a secure out-of-band (OOB) channel between the communicating devices exists. Depending on the latency of this channel it can be used in several ways, from being used to provide a secure way for pairing the devices up to being used to update the device identifiers in regular intervals.

The most comfortable way is to use the OOB channel to establish a secret shared key between Alice and Bob and use the solution from scenario one to provide location privacy. In [167], Singelee also analyses various pairing protocols.

10.1.1.3.4 No shared data available This scenario is the most general as it does not make any assumptions at all. No shared keys, no known hardware addresses or secure out-of-band channels are available to Alice and Bob. But it is still possible to improve location privacy in this situation.

A trivial approach would be that all messages are broadcasted with the disadvantage that every device has to check the content of each message and decide if it was intended for it. The high energy consumption of this technique renders it non practicable.

A more sophisticated approach presented by Singelee is to use random identifiers for Alice and Bob for each new communication session. Mutual updating of the IDs by the communication parties during the session is insecure as the attacker can perform the same steps and has the same knowledge as Alice and Bob. It is therefore possible for the attacker to track the devices during one session but she has no possibility to link the identities across different sessions.

10.1.2 Privacy Preserving Techniques for Wireless Local Area Networks

The idea of providing untraceability and unlinkability of mobile nodes to improve location privacy also counts for wireless local area networks. The approach of using pseudonyms seems to be nearby. Unfortunately, the domain of wireless local area networks is significantly more complex than of personal area networks.

While in WPANs like Bluetooth, the main concern is to provide (location) privacy against attackers intercepting wireless transmissions, WLAN scenarios might also require to provide (location) privacy against the infrastructure provider. This fact extends the necessity to use changing identifiers to higher layers. WLAN routing takes place at the IP layer and therefore IP addresses need to be taken in to account.

The improvement of location privacy, especially for WLANs, based on disposable identifiers does not only rely on the ID changing mechanisms but also on the following factors:

- **Mobility**
  If a network and the connected clients are very static (regarding spacial mobility), signal characteristics such as the signal strength can easily be used to detect the location of a node. Changing identifiers can be correlated

with the approximate position of the client and link the pseudonyms to the physical device. If clients move and change their positions and pseudonyms frequently, its hard for an attacker to link them to a physical devices as long as their is a crowd to hide.

- **Size of the Crowd**
  If there is only one client (or a small number) associated with the network, it is pointless to use pseudonyms as an attacker has no problems linking those identifiers to the same device.

The following paragraphs present the evolution of mechanisms and their issues and performance.

10.1.2.1 Short-lived Disposable MAC Addresses

In 2003, Grutser and Grunwald [60] proposed the usage of short-lived, disposable MAC addresses to reduce the opportunities of location tracking. They identified the goals of their approach as the following:

- **Unlinkable Identifiers**
  An attacker should not be able to link different pseudonyms to real devices.

- **Minimal Network Disruption**
  The change of pseudonyms should cause no or minimal disruption to the network. These disruptions include degrading the throughput or the latency and the user experience.

- **Applicability**
  The solution has to be readily applicable to state-of-the-art IEEE 802.11 implementations. No special purpose hardware should be necessary.

They further on identified the following key challenges:

- **Identifier Selection**
  MAC address pseudonyms must be valid under the IEEE 802.11 standard in order to not interfere with various intrusion detection mechanisms and to make it harder for adversaries to detect pseudonyms. A MAC address consists of 48-bit whereas the bits 45 to 24 represent the so called *Organizationally Unique Identifier* (OUI). This identifier must be requested by a device manufacturer and licensed by the IEEE [85]. The remaining bits of the address usually serve as a counter for the manufacturer and is incremented with each produced device. As only a small number of valid OUIs exist compared to the 22 bits address space available, random generation would almost certainly result in an invalid MAC address. If the just named limitations are taking into account, valid identifiers can be generated in advance on the device. The authors proposed a MD5 forward hash chain with an unpredictable random seed.

- **Identifier Uniqueness**
  As all major network management mechanism are based on MAC addresses, it is vital that they are unique in the network. Duplicate address detection mechanisms mitigate the problems caused by address collisions. Randomly chosen MAC addresses may not cause collisions if a small number of devices with a slower pseudonyms refresh rate are connected, but in busy environments this could be an issues and needs to be covered.

- **Integration with Port Authentication**
  In some cases it is necessary to employ port authentication mechanisms like IEEE 802.1X (see Appendix B for more details) for authentication and accounting. Instead of relying on MAC addresses during the authentication protocol, the authors propose an additional token that comprises a service provider identifier and a temporary client identifier. This way, the subscribers location privacy is protected against the exposure of statical identifiers by the port authentication protocol.

- **Implementation Issues**
  The authors proposed to change the identifier not only during fixed time intervals but also at power up of the device and at significant changes in the signal strength to make it harder for attackers to correlate new and old identifiers.

  Another issue which comes to mind is the fact that it is necessary to reset the link-layer sequence numbers. Otherwise an attacker could easily link pseudonyms based on the surveillance of such counter values.

Discussion The usage of client MAC address pseudonyms with frequent changes causes serious network disruptions. When changing the identifier during association, all connections like TCP streams will abort because the client also needs to obtain a new IP address. This process could take several seconds to complete.

A possible improvement suggested by the authors would be the use of multiple network cards simultaneously. One card could keep the connection till all active sessions are finished and the other card switches the identifier and is used for the next sessions. This way the client would be connected at all times not suffering from the disturbances.

This first approach with disposable identifiers seem to be an appropriate way if the attacker is not physically present in the transmission range, which means he has no way to obtain signaling properties, but is only able to analyze the traffic inside of the connected network, after the access point. But in contemplation to the possibility of device and user fingerprinting, this mechanism can be circumvented and rendered useless.

10.1.2.2 Silent Periods

Huang et al. analyzed the short-lived, disposable MAC addresses presented in the former section and proposed an extension in [121]. They introduce *silent* *periods* as a transition period between the use of new and old pseudonym in order to the temporal and spacial relations between them. Beresford and Stajano claimed in [13] that it is possible to find strong correlation between a trails left by an old and a new address, if a device can be accurately positioned within a wireless cell. If an attacker is able to determine the position of a device accurately she might be able to correlate two pseudonyms that are used separately by the same device moving trough space. Temporal correlation can be used because the period during ID changes may be short. Spatial correlation can be used if one assumes that mobile stations continue in the same direction with the same speed after a pseudonym change.

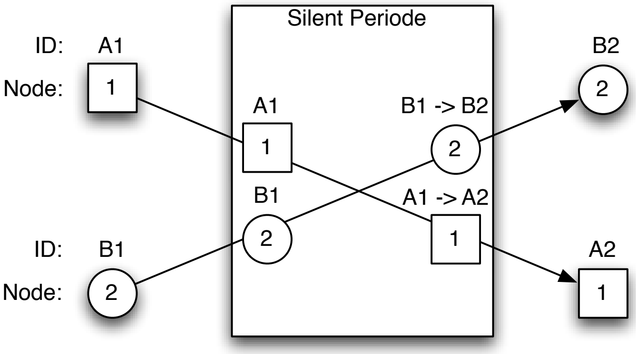


Figure 10.4: Movement of two Clients which change their IDs during a Silent Period [121]

In order to combat such correlation attacks, the authors propose the use of silent periods in order to create a kind of *virtual mix zone* as described in [13]. A silent period in which a client is not allowed to transmit any packets but change his pseudonym while continuing his spatial path, introduces ambiguity into the determinations of the time and/or place at which a change of pseudonyms occurred (see Figure 10.4). This makes it more difficult to link pseudonyms to a physical device as the spatial and temporal correlation is disrupted.

In order to enhance the degree of correlation disruption the authors propose to use silent periods with variable length additionally to periods with fixed length. The effect of the constant length period is to mix the spatial relation while the effect of the variable period is to mix the temporal relation between disappearing and reappearing of the nodes.

Discussion The authors argue that the problems of valid MAC addresses, duplicate identifiers and verifiable addresses need to addressed and solved before such a pseudonym mechanism may be used in real-world applications.

10.1.2.3 Privacy Enhancement by User Cooperation

In [36], Defrawy and Soriente propose an extended approach based on disposable identifiers. They assume the cooperation of the clients and a so called *Privacy Enhancer* (PE). Their system is called PEUC-WiN and the basic architecture is illustrated in Figure 10.5. They consider a scenario with a group of K users,

$$G_j = \{u_1, ..., u_k\} \tag{10.6}$$

registered with the same access point and a thrusted third party called privacy enhancer (PE) which is assisting the clients to enhance location privacy. The PE can also be seen as a proxy as all communication to and from the clients is forwarded by it.

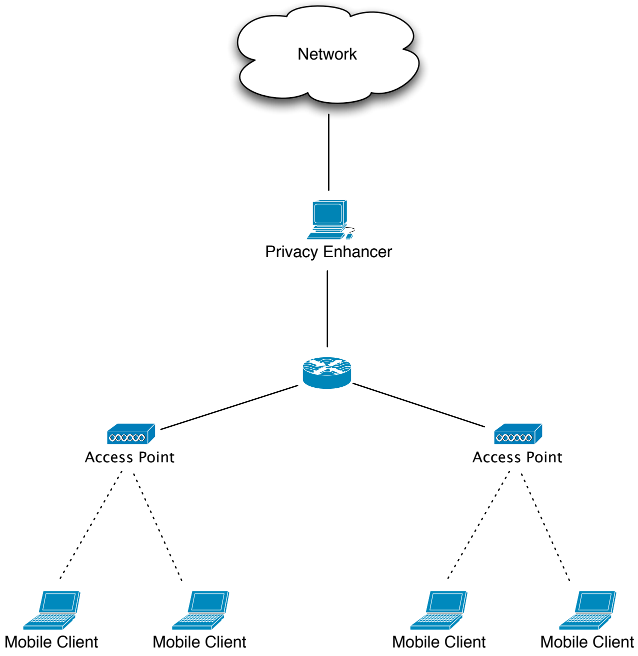


Figure 10.5: PEUC-WiN basic Elements

They further on assume that each user u_i has a private/public key pair, a Group Key that is the same for all users and a permanent IP address. The PE also has its own key pair and knows the Group Keys of all groups it manages. Time is separated in slots and each client uses an [IP, MAC] addresses pair during such a slot. All of these pairs are stored in a table T_j which is known to all members of a group and the PE. If a new user joins the group, a new address pair is created by the PE and broadcasted to all members.

All clients and the PE share the same hash function $H_j()$ which is used to create a hash chain for the future values of the address pair. Each client calculates these values and changes its identifiers when the PE broadcasts an update message. In order to stay synchronized, all clients have to reset their clocks at the reception of the update message.

The PE stores the current address pair as well as the history of the updates for each client. After an identifier update of the group, the PE, which acts as a proxy, modifies the address headers of the incoming and outgoing packets accordingly to the newly obtained addresses of the clients. This mechanism allows the clients to keep all TCP connections alive and they do not need to re-associate with the wireless network which avoids severe disruptions.

All communication between the client and the PE is encrypted, as well as the group broadcasts done by the PE.

Discussion The approach by Defrawy and Soriente promises minimal disruptions for the clients as no re-association with the wireless network is needed. An address collision detection mechanism needs to be implemented as hash chain values for the identifiers might cause collisions. Given a crowd, the mechanism seems to provide a high degree on location privacy against outside attackers, other clients and even malicious access points. But as a trusted third party is needed that acts as a central point of management and also keeps all relevant information, an attacker or malicious provider being able to control the PE could exploit all location privacy relevant information.

10.1.2.4 Dynamic MAC Address Exchanging

In 2007, Lei et al. presented their approach called DMAS (Dynamical Mac Assignment with Shuffle) in [113]. Similar the approach of Defrawy and Soriente described in the last section (see Section 10.1.2.3), the key idea is that clients obtain MAC addresses dynamically similar to how IP addresses are distributed in the *Dynamic Host Configuration Protocol* (DHCP) [40]. Further on, the assigned MAC addresses will be periodically shuffled between the connected clients. This achieves the same results as an individual nodes periodically updating its MAC address like in the approaches described in the sections above, but without the problems of MAC address duplication, AP access control conflict.

The system architecture of Lei et al. requires a *Dynamic MAC Assignment Server* (DMAS) and the cooperation of the access points (see Figure 10.6). DMASs can be logical entities implemented in one central physical location where security is guaranteed.

The IEEE 802.11 standard demands of a client to inform the AP of its intention to connect, and therefore reveal its MAC address, the authors introduced a public *Special MAC Address* (SMA) which serves for this purpose and may be used by each new client to establish the initial connection. The AP relays the association request to the authentication server (AS) who sends a MAC request to a DMAS. The DMAS is responsible for assigning a free MAC address out of its address pool to the client.

In order to improve location privacy, a dynamic MAC address exchange between the connected clients is performed. All clients C_i wishing to update their

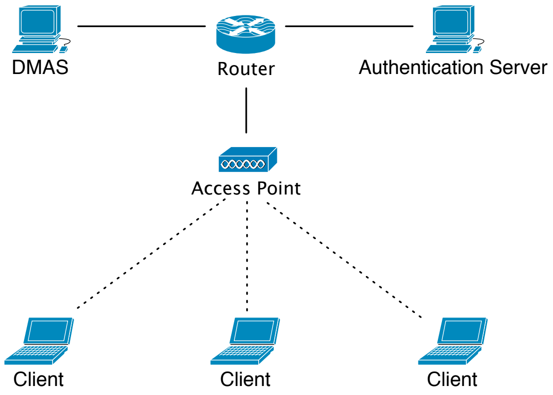


Figure 10.6: DMAS System Architecture [113]

identifier randomly form a directed circle:

$$C_1 \to C_2 \to C_3 \to ... \to C_n \to C_1 \tag{10.7}$$

The client on the left changes his MAC address to the one of the client on the right of it. Via this uni-directed exchanging circuit, every participant will know to which address it changes but is unable to determine who is going to use his former identifier. The authors also suggest that each client who updated its MAC address should also perform a silent period as presented in Section 10.1.2.2.

Clients are allowed to decide if they want to participate in a shuffle or not. This allows for continuing active connections and sessions and avoids unwanted disruptions.

Discussion The approach of Lei et al. has the advantage of preventing MAC address duplication issues but depends on the cooperation of the infrastructure and therefore of the provider. Location privacy is only provided against and outside attacker and not against the infrastructure provider. The main advantages of this approach versus the classic disposable pseudonyms method and the approach by Defrawy and Soriente (see Section 10.1.2.3), are that not all nodes need to change their addresses at the same time and no address collision scheme is required. But a significant draw back is the fact, that it requires identity checking and authentication by the infrastructure before exchanging addresses and that due to IP address requesting after each MAC address change, the network performance is degraded. A proxy similar to PEUC-WiN could solve this problem.

10.1.2.5 Enhancing WLAN location Privacy using Mobile Behavior

In 2003, Beresford and Stajano [13] proposed the use of a trusted middleware in order to provide location privacy for wireless clients using location based services. This might sound confusing as location based services depend on the location information received from the client who wants to use them. The authors present the concept of so called *Mix Zones*. Based on the cooperation of the infrastructure, which provides mix nodes with routing capabilities according to the concept of Chaum's mix networks [27]. All communication with service providers is done by a anonymizing proxy between the user and the location based applications.

2010, Horng et al. [71] proposed a further improvement for disposable identifier mechanisms concerning the triggering of pseudonym updates and by incorporating the idea of the just presented Mix Zones. They call their approach *Enhancing WLAN location Privacy using Mobile Behavior*. As Section 10.1.2.2 about silent periods illustrates, correlation attacks are a powerful threat to pseudonym mechanisms and sophisticated update schemes are necessary to counter act. An update scheme is proposed, based on the behavior of neighboring nodes and their relative position. The authors define the *relative position* as the position of the connected access point plus an uncertainty of the maximum transmission range.

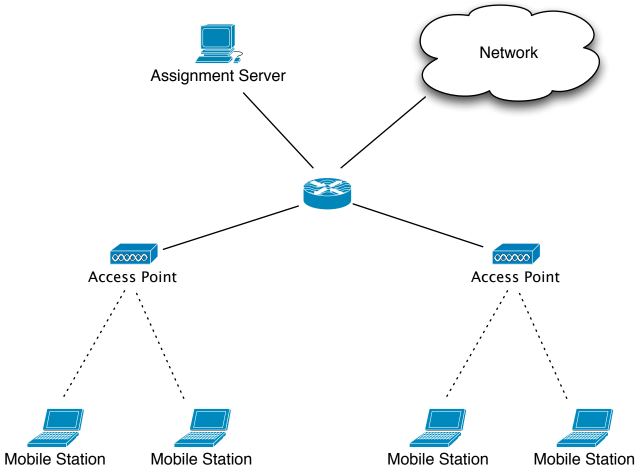


Figure 10.7: System Architecture [71]

Clients are organized in groups (one per access point (AP)) and they posses individual and Group Keys similar to the approach by Defrawy and Soriente (see Section 10.1.2.3). Figure 10.7 illustrates the basic elements of the design which seems to be quit similar to the PEUC-WiN setup (see Figure 10.5). The main difference is, that only the client's MAC addresses are used as disposable identifiers while the IP addresses are not updated. The authentication server keeps a connection table to record each connection from the AP to the clients. The approach relies on the cooperation of the APs which need to inform the AS of each new connecting client, and further on keeps a table of the MAC addresses of the client connected to it.

As the update scheme is based on the relative location of the clients, a signal strength approach was chosen to determine the necessary location information. Each client collects signal strength information about all other clients and the AP in his group and transmits it to the AS. The AS is now able to calculate the relative position of all clients in a group by collecting the signal strength values in a so called distance matrix (DM).

$$DM = \begin{bmatrix} d_{11} & d_{12} & ... & d_{1j} & ... & d_{1n} & d_{1AP} \\ d_{21} & d_{22} & ... & d_{2j} & ... & d_{2n} & d_{2AP} \\ ... & ... & ... & ... & ... & ... & ... \\ d_{n1} & d_{n2} & ... & d_{nj} & ... & d_{nn} & d_{nAP} \end{bmatrix}$$

By transforming this distance matrix into an angle matrix (AM),

$$AM = \begin{bmatrix} \Theta_{11} & \Theta_{12} & ... & \Theta_{1j} & ... & \Theta_{1n} \\ \Theta_{21} & \Theta_{22} & ... & \Theta_{2j} & ... & \Theta_{2n} \\ ... & ... & ... & ... & ... & ... \\ \Theta_{n1} & \Theta_{n2} & ... & \Theta_{nj} & ... & \Theta_{nn} \end{bmatrix}$$

and by observing the changes over time, the AS is also able to determine the direction of the mobile clients as illustrated in Figure 10.8.

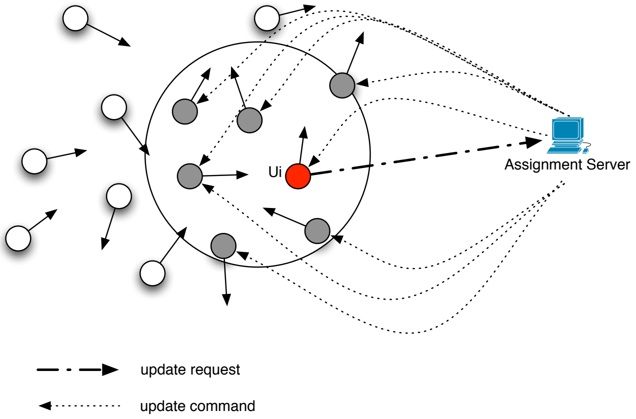


Figure 10.8: Client Movements [71]

Instead of using fixed time intervals for the updates, clients are able to request them. If a client U_i (red) requests such an update, the AS analyses the area around the client and sends update commands to his neighbors (grey). By doing this, the AS creates a mix zone around U_i. All effected clients need to keep a silent period depending on their speed and direction in order to improve the result.

Discussion The proposed scheme introduces a five percent overhead in network traffic due to the collection of the signal strength values. Similar to the

DMAS approach presented in Section 10.1.2.4, the cooperation of the infrastructure is necessary and offers a loop hole to malicious infrastructure providers. The proposal also needs the access points to perform special tasks which can not be handled by current state-of-the-art hardware. The authors assume that IP addresses are in the encrypted payload of the wireless MAC layer packets and therefore pose no privacy leak to outside attackers.

10.2 Location Privacy Enhancement for WLANs based on Virtual Network Interfaces

This section presents our own contribution to the field of location privacy preserving mechanisms in wireless local area networks. We propose a new technique, based on the usage of multiple (virtual) network interfaces, with the capability of using multiple disposable identifiers at the same time, with the same access point.

10.2.1 Motivation

IEEE 802.11 based wireless networks rely on the usage of hardware identifiers (MAC address) for management and routing purpose. These identifiers are defined in the production process of the network interface and not meant to be changed during the device's live-cycle. As these addresses are broadcasted in plaintext and therefore are observable by potential attackers, they constitute a serious privacy breach. By tracking these identifiers one can exploit the location privacy of the device and hence of its user.

As mentioned in the Section 10.1.2, all current approaches are based more or less on disposable identifiers also known as pseudonyms. The initial idea of using pseudonyms to enhance privacy in information technology comes from Chaum [27]. The key to provide privacy is to establish *untraceability* between different transmissions and *unlinkability* to communicating parties.

The direct approach would be, to frequently and randomly change the MAC address of a device. Singelee [165] has shown that this approach works perfect for one-to-one communication scenarios in wireless personal area networks, but our analysis of the current techniques in WLANs revealed serious issues regarding performance and usability.

In order to understand why Singelee's idea is not directly applicable to IEEE 802.11 networks, one needs to take a look at the connection process. Figure 10.9 illustrates (simplified) the steps which are necessary to establish a connection between a client and an access point (AP) and to enable IP based communication.

The connection process starts with *Association* and *Authentication* on the MAC layer. These steps consist each of several messages and take in the magnitude of milliseconds to complete. As MAC layer authentication does not provide any sophisticated security features, another layer of authentication is necessary [79]. In current WLAN implementations, this task is handled by IEEE

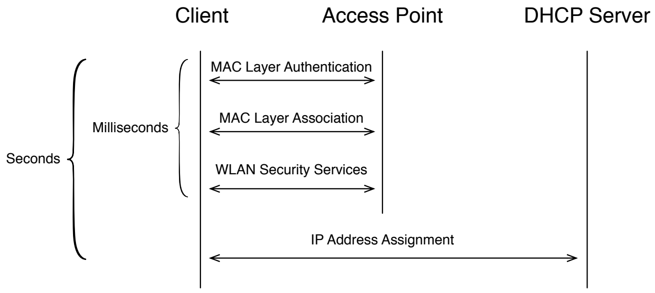


Figure 10.9: Necessary steps to connect to an IEEE 802.11 network (simplified)

802.11i, also known WPA/WPA2 and also takes in the magnitude of milliseconds to be completed [78].

After establishing the connection with the AP, the client needs to obtain an IP address in order to enable IP based communication with the network. In common real-world implementations based on the *dynamic host configuration protocol* (DHCP) [40] this IP address assignment takes at least several seconds.

Association and authentication at the MAC layer are based on the MAC address of the device which is stored in tables on the access point(s) and router(s). Network communication on higher layers, like TCP or RTP sessions are based on the IP address which is interconnected and relies on the MAC address for routing in at least the last network segment [143].

If a device changes its MAC address it needs to reconnect with the AP, and repeat all steps illustrated in Figure 10.9. Additionally, all disrupted sessions need to be reestablished, causing additional latency. There is no need to tell, that such a step seriously downgrades the perceived network performance and user acceptance.

Nevertheless, this direct approach can theoretically provide high levels of location privacy against attackers who are able to intercept the communication with the wireless infrastructure, as well as against the infrastructure provider.

Current approaches address this reconnection problem by relying on a trusted-third-party (TPP) which acts as a proxy, modifying and relaying network traffic in order to circumvent session breakdowns. By introducing MAC address exchange between the connected nodes [113], it is also possible to avoid reassociation and re-authentication with the APs. But all of these approaches have in common that they rely on some form of trusted infrastructure, exploiting privacy against the infrastructure provider.

The intention of our approach is to minimize or avoid these disruption without depending on any kind of infrastructure in order to provide location privacy against all outsiders including the infrastructure.

10.2.2 Enhancing Location Privacy using multiple virtual Network Interfaces

In order to understand the idea behind our approach we should take a quick look at the location privacy preserving solution for one-to-one communication in WPANs, developed by Singelee and Preneel [167] and the concept of virtualized wireless network interfaces developed for MultiNet by Chandra et al. [26] as they inspired us to develop this solution.

10.2.2.1 Location Privacy for Bluetooth based on Disposable Pseudonyms

This scenario assumes that the communicating Bluetooth devices have some how established a shared key. Every device has in its memory a list of those keys, shared with other devices and an identifier R (short-lived pseudonym) which is based on an initialization vector IV and computed as follows:

$$R = PRF_k(IV) \tag{10.8}$$

RPF is a pseudo-random function and k is the Shared Key. Singelee and Preneel assume that all devices in the WPAN use the same IV. The identifiers R are stored in the memory. When Alice wants to send a message to Bob for the first time she puts the corresponding R in the header of the message. Bob will recognize the identifier and knows that the message is for him. Now both devices can update their identifier R as follows:

$$R_{new} = PRF_k(R_{old}) \tag{10.9}$$

In order to provide a very high degree of location privacy, R should be updated for each new message. An eavesdropper just intercepts random identifiers and is not able to link them. This approach allows devices to calculate disposable identifiers offline and in advance.

If one is assured of the trustworthiness of the infrastructure we assume that this approach can be implemented for WLANs as well but needs major firmware modifications for access point and client network interfaces.

10.2.2.2 The Concept of virtualized Wireless Network Interfaces

Chandra et al. developed a system called *MultiNet* that allows to connect a single physical wireless network interface to multiple wireless networks simultaneously. MultiNet is implemented as an additional layer in the network stack (see Figure 10.10) just below the IP layer and pretends the availability of multiple network interfaces.

In fact, only one physical interface is switching between the networks following some algorithm, creating the illusion of serving multiple connections simultaneously. The motivating scenarios for developing MultiNet were:

- **Concurrent connectivity** to multiple physical networks;

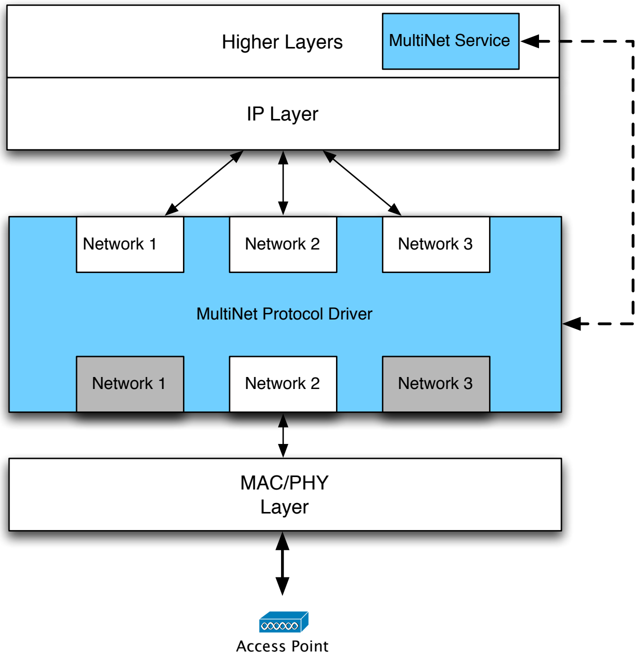


Figure 10.10: MultiNet modified Network Stack [26]

- **Network Elasticity** - The range of infrastructure networks can be extended by creating relay nodes;
- **Gateway Nodes** - Connecting ad-hoc networks to access points via gateway nodes;
- **Increased Capacity** - Network capacity can be increased when nodes can communicate on orthogonal channels ; and
- **Virtual Machines** - Virtual machines can be connected to different physical networks.

In MultiNet, the virtualized interface connects to this multiple networks using the same hardware identifier.

10.2.2.3 The Idea behind our Approach

As mentioned before, our approach has been inspired by the just presented concepts of disposable identifiers, changed as often as possible, and the concept of virtual network interfaces.

The remarkable part of Singelee's and Preneel's approach is, that a new random identifier is used for each new message providing the highest degree of unlinkability and untraceability. It seems to be very desirable to convert this feature to WLAN scenarios. In Section 10.2.1, we identified two issues with disposable identifiers in WLANs that need to be considered:

1. The necessity of reconnecting to the network on the MAC layer, and obtaining a new IP address; and

2. The disruption of communication streams like TCP or RTP sessions.

These problems seem to be inevitable if one does not want to rely on a trusted-third-party or the infrastructure. Our approach therefore focuses on the sustainment of active sessions and is based on multiple network interfaces similar to MultiNet. In contemplation of the current IEEE 802.11 standards, it seems not to be possible to provide unlinkability on a per-message basis. Our aim is therefore to achieve it on a per-session basis. Let's assume that a device D has n network interfaces,

$$IF_1, IF_2, ..., IF_n \tag{10.10}$$

regardless if physical or virtual, each identified by a random disposable pseudonym (MAC addresses) and administered by some sort of middleware or intermediate layer (see Figure 10.11), similar to the MultiNet approach. Each of these interfaces connects to the available wireless network and obtains an IP address. When some application issues a *communication request*, it is reached down the network stack to the middleware, which assigns one of the interfaces IF_i to carry-out the communication process. The next request for another communication will be assigned to another interface and so forth. After a connection or session is terminated, the middleware instructs the assigned interface to dispose its identifier and reconnect to the network. After having successfully done so, the interface is ready to be assigned the next communication session.

This approach provides unlinkability between communication sessions without causing perceptible disruptions for the client as there is always at least one interface connected and ready for duty.

10.2.3 Real-world Approach and Middleware Architecture

Applying our theoretical approach to real-world scenarios affords some considerations. Regardless if we use physical or virtual network interfaces, their number will be restricted depending on the resources of the client device. While the number of physical interfaces is likely to be restricted to be smaller than a handful due to limited space and power in mobile devices, it seems nearby that the number of virtual interfaces can be significantly higher.

As briefly described in Section 10.2.2.2, Chandra et al. [26] proved the possibility of virtual network interfaces to connect quasi simultaneously to multiple networks at orthogonal channels. We modify their approach and take it one step further by extending it to the MAC layer. Our proposed system is able to directly interact with the MAC layer implementation of the physical wireless interface in order to apply the disposable pseudonyms concept. MultiNet does not mind to manipulate the MAC address of the physical device and is therefore

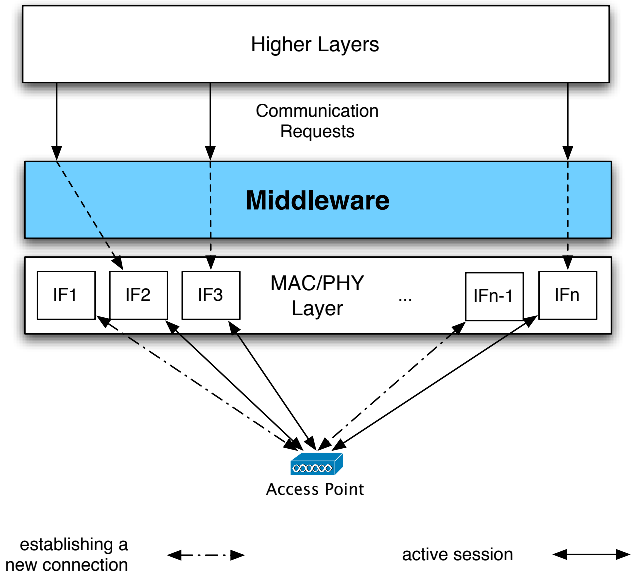


Figure 10.11: Idea of multiple network interfaces

using the same identifier for all communications, exploiting the location privacy of the device on all used communication channels. The core of our approach is a middleware that manages the virtualization of the network interfaces as well as the communication with the higher layers. The following list presents the requirements of the middleware design, and describes the key properties:

10.2.3.0.1 Media Access Control Due to characteristic properties of open air propagation, like the *hidden* or *exposed node problem* [108], IEEE 802.11 networks need to implement a collision avoidance mechanism [79].

The consequence arising of this MAC implementation is, that the middleware needs to sense the media at all times in order to enable standard conform media access for all virtual interfaces nevertheless they are active or not at a certain moment. As a detailed discussion of this topic would go beyond the scope of this work, the interested reader may be referred to [79].

10.2.3.0.2 Network Management The access point (AP) is responsible for managing the network consisting of the connected clients in its transmission range. Various functionalities like *authentication*, *association* or *power management* are administered by management messages broadcasted by the AP. The middleware needs to receive these management messages regardless which virtual interface is active at the moment and apply them to the relevant virtual interfaces.

10.2.3.0.3 Session Tracking As we aim to provide unlinkability on a per session basis, the middleware needs to track all forms of communication that rely on sessions such as TCP or RTP. Further on, it needs to wait until such a session has been closed by the client before it is allowed to change the identifier of the assigned virtual interface. Otherwise it would disrupt the session and cause undesired performance aggravation.

10.2.3.0.4 Logical Switching and Scheduling As the middleware only virtualizes one physical network interface, it has to apply some sort of switching algorithm to assign to a virtual interface the physical resource. This switching mechanism needs to be tunable in order to enable priority scheduling for communication sessions.

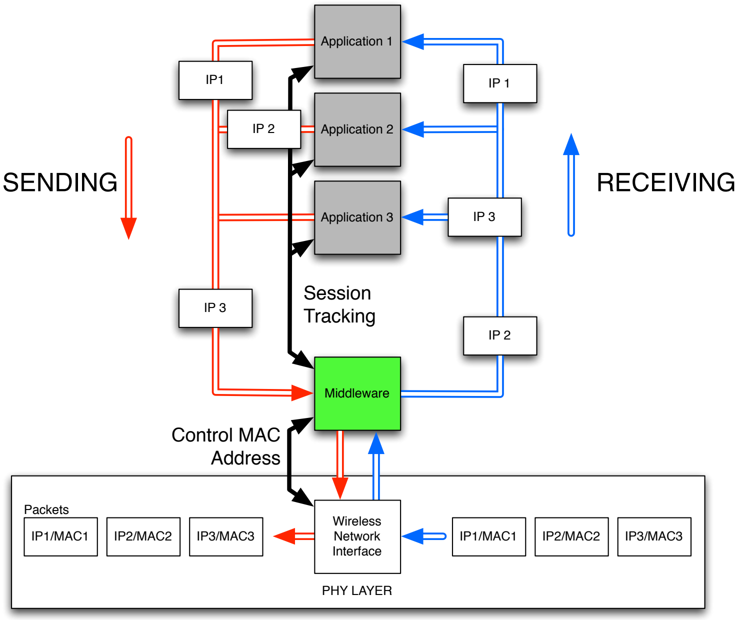


Figure 10.12: Middleware Workflow (simplified)

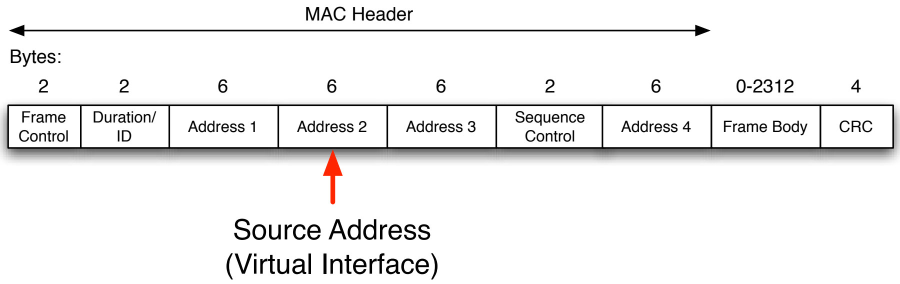
Figure 10.12 illustrates the simplified flow of data packets in the network stack. Applications usually only know the IP address of their communication pear and don't care about the MAC address of the network interfaces. A packet meant to be transmitted is handed down the stack to the middleware which assigns the appropriate virtual network interface by considering whether it belongs to an active session or not.

The knowledge about active sessions is gathered by a *Session Tracking* mechanism and the virtual interface assignment is accomplished with a *Control MAC Address* mechanism.

The reception of a packet is simple as the middleware just checks if the MAC address in the header of the received packet matches one of the identifiers of the virtual interfaces. If yes, it just hands it to the higher layers and further to the appropriate application.

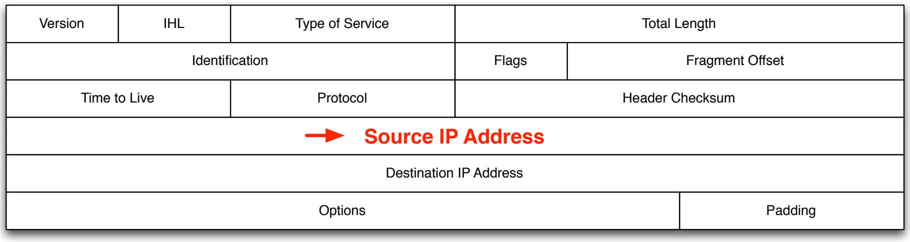
10.2.3.0.5 Virtual Network Interface The term *Virtual Network Interface* has already been mentioned some times in the last sections. In fact it is just a convenient abstraction. It merely exists as an entry in a table administered by the middleware. Every time a transmission request for a certain IP addresses is handed down to the middleware, it looks up the belonging MAC address and modifies the MAC source address field in the packet header, before the physical interface submits it to the media. As most current interfaces allow this kind of manipulation it is straight forward to implement it [25]. Figure 10.13 illustrates the structure of an IEEE 802.11 MAC header, highlighting the source address field while Figure 10.14 does the same with the IP header which is part of the payload of the MAC data frame.

As just mentioned, a virtualized interface only constitutes a change of the source address fields in the MAC and IP headers and a table entry of the address pair stored in the middleware.



Frame Control	Duration/ ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	Frame Body	CRC
2	2	6	6	6	2	6	0-2312	4

Figure 10.13: IEEE 802.11 MAC Header



Version	IHL	Type of Service		Total Length			
Identification				Flags	Fragment Offset		
Time to Live		Protocol		Header Checksum			
Source IP Address							
Destination IP Address							
Options						Padding	

Figure 10.14: IP Packet Header

10.2.4 Evaluation

Now that we have presented the theoretical background of our solution and outlined the considerations for a real-world approach we want to evaluate the implications for real-world applications. We have to distinguish between two different application scenarios or more precisely two different views on the problem of location privacy.

The common question asked about this issue is that what would happen if we use the same disposable identifiers, in our case MAC addresses, in every network we connect with?

First, communication infrastructure providers such as mobile network companies (T-Mobile, Vodafone, Sprint, Verizon,...) or even public free Wifi providers (Starbucks, McDonalds, ...) would be able to acquire location and other privacy sensitive data for one particular user on a global scale. Thus, the first thing we have to verify and guarantee is that with our approach users are able to obliterate their tracks on a global scale but are still able to use all the desired services. Such an behavior can be achieved if we use truly random identifiers.

Second, even if users are able to provide different identities for various scenarios or in different networks, profiling techniques would enable the previously mentioned service and infrastructure providers to create profiles for specific behaviors and assign them virtual identities, which then can be used to again track the users. Thus, it is not only necessary to switch, iterate and randomize identifiers on a global basis but also on a local or more precisely on session basis. In our understanding session can mean a lot of different things and we will go more into the specifics about what implications the different session contexts mean for the location privacy issue subsequently. For now just think about a common request-response protocol versus a typical TCP session versus a HTTP request which involves actions such as resolving the domain name, obtaining data from different services and asynchronous java script. All of these processes can be viewed as sessions in some sense, it only depends on the sophistication of the profiling technique and the complexity of the privacy middleware.

We have outlined the two different scopes which always have to be considered together if a particular location privacy solution must be evaluated according its efficiency and effectiveness. We now go on to evaluate the session concept and its implications for our solution as it is understood at the different layers of the internet protocol suite. We also verify if the mechanisms required for obfuscating tracks from the session concept do not interfere with the usability of services working on a global scale (e.g. IPSec, VPN, NAT, ...).

10.2.4.1 Session Management

Whatever kind of session concept is used there are several different possibilities how to manage these sessions inside our middleware. We have currently identified two major session management mechanisms which are as follows:

- **One session per virtual network interface** - For each new session a new virtual network interface is initialized. The slot stays active until the session has ended. The virtual network interface is thereafter torn down and its resources are freed. This may be the most simple approach but since the amount of sessions can be tremendous it may result in too much overhead for managing the virtual network interfaces as well as problems with available physical network resources such as MAC or IP addresses in a particular subnet.

- **Time division of sessions into virtual network interfaces** - In this case each virtual network interface slot is assigned a specific time interval at which it is active or more precisely takes new session. Every new session which is initiated is handled by one specific virtual network interface. Before the activity period of this slot is over a new slot is initialized. After the activity period is over all new sessions are then managed by the new slot. The old slot stays active until all the sessions of that slot have ended. Thereafter, the slot is torn down and the virtual network interface is released.

10.2.4.2 Internet Layer Sessions

The most notable issues related to sessions on the internet layer are concerned with IPSec and NAT. Although NAT is not only restricted to the internet layer and also doesn't really fit into the *session context* we also have to address possible problems with our approach if NAT is used for IPs.

The main issue is that the communication with services in the internet is rewritten by the NAT server and it thus holds translation tables for outgoing and incoming connections. Thus, it is necessary to provide the same network interface on the host as long as it is used/stored for any incoming data on the NAT server. NAT servers use specific timeouts to remove bindings between internal and external addresses. This behavior must be mirrored by our middleware.

Another issue, which is the same for NAT itself, is with protocols such as FTP or SIP which use out-of-bound messages to negotiate addresses and ports for the reverse data connection. More sophisticated clients also allow to schedule data transfer which would require the middleware to hold the network interface available until that point. For instance, the connection is opened over one connection but data is transferred on a different connection which is usually opened by the remote service at some later time.

It may also be crucial to know what kind of NAT is used in order to not exhaust the possible translated outgoing connections. For instance, if to many internal addresses (iAddr:iPort) are mapped to external addresses (eAddr:ePort) in a specific amount of time, this may lead to an *unintended internal* denial of service attack. Thus, it would not be possible to open anymore connections to remote services until a timeout at the NAT server has freed resources again.

Another, more general problem arises in the context of IPSec and all other services which require long-lived connections or sessions. Our middleware has to identify these long-lived connections and should be able to tear down them if the user would like to be aggressive about privacy. Thus, it would for instance tear down the IPSec connection after some specific amount of time, which would trigger a new IPSec connection which could then be initiated over a new virtual interface. But since it may still be possible for an attacker to identify such an behavior, because the same remote IPSec provider is used, the attacker could again link several virtual interfaces together and identify a particular user.

10.2.4.3 Transport Layer Sessions

On the transport layer the most important type of session is the TCP session. Also the both upcoming protocols DCCP and SCTP have the notion of sessions which can be treated similar to TCP but since they are currently not used very often we consider only TCP sessions.

TCP sessions can very easily be identified since they are an integral part of almost any networking application or tool for networking forensics. For instance Wireshark[1], the widely used network analyzer, has a built in feature to follow TCP streams and sessions. Every TCP session is started using a three-way handshake and torn down using a specific sequence of messages. This allows us to associate a session with a particular network interface and IP address combination. Also TCP sessions have a timeout mechanisms which triggers a session tear-down if the sessions stays idle for a specific amount of time. Thus, it seems that using TCP sessions as basis for the session concept for our middleware would be the most simple and straightforward. The only drawback could be that since application protocols very often initiate and use a lot of TCP sessions available physical network resources could get exhausted. For instance every HTTP request triggers often several TCP sessions we might run out of available free IP addresses in the WLAN network and thus we would need to reuse already acquired IP addresses which may lead again to traceability.

10.2.4.4 Application Layer Sessions

Sessions in the context of the application layer are much more difficult to identify since a lot of different application protocols exist and there is no common notion of how these sessions are constituted. For instance, a simple client FTP session opens a connection to the FTP server. Inside of this session at least one extra data connection is negotiated. This data connection can be established immediately or can also be scheduled to become operational at a specific time. For HTTP sessions it is even more complicated since it usually consists of several distinct processes such as host name resolution, several TCP streams, delayed and asynchronous data transfer and also response messages which are initiated by the client on user request. Thus, identifying these sessions is hard and associating them with a specific virtual network requires the middleware to have exact knowledge of all the application protocols which are used on top of it.

[1] Wireshark Network Protocol Analyzer http://www.wireshark.org/

10.2.4.5 Resource Allocation Issues

The first limitation is the amount of wireless connections a particular access point is able to handle. State-of-the art wireless routers are able to handle literal connections anywhere between 50 and 253 depending on the manufacturer. The question of how many usable *simultaneous connections* is a completely different issue because that number is drastically smaller. With *usable simultaneous connections* we are referring to how many connections the wireless router can handle before the connectivity speed gets too slow to be usable. Although each node may have only one physical wireless network interface we are not sure at this point if access points treat virtual network interfaces as distinct connections.

Then there is the issue of available MAC address and address space exhaustion. Although, the amount of theoretically available MAC addresses is big enough, not all of them can be used since a big chunk of space is not assigned to any particular or existing manufacturer, also some of them are also not producing wireless network interfaces. Thus, if only one node uses our privacy protection middleware, we can only choose randomly amongst a limited set since. Because otherwise it would be possible for a tracker to identify the user based on these invalid MAC addresses. On the other hand, if all of the nodes in a particular WLAN use our technique we may experience collisions of MAC addresses.

One of the more critical issues is the availability of IP addresses. Since we relay on the automatic provision of IP addresses from the access point we need to take into account that the available address space will be restricted. Usually wireless routers provide the connected nodes with IP addresses from the ranges which are reserved for private networks. Most common is the use of 192.168.0.0. In that private network 65,534 IP addresses are available for the connecting nodes. Depending on the implementation of the wireless router the IP addresses are assigned to a network interface for a specific amount of time. If either too many other nodes are trying to connect or if the middleware is requesting too many IP addresses we may exhaust the address space to fast and perform unwittingly a denial of service (DoS) attack. Thus, we also must take this issue into account in the design of our middleware.

10.2.5 Conclusion and Outlook

In this chapter we have presented a location privacy enhancing mechanism based on disposable identifiers and virtual network interfaces. Our approach promises significant improvements for location privacy of wireless clients without relying on the cooperation of a trusted-third-party or the network infrastructure. Our solution is based on using an arbitrary amount of virtual network interfaces to which we assign random MAC addresses and obtain different IP addresses. Each of these virtual network interfaces is then used for communication for a particular amount of sessions depending on the environmental constraints as explained in Section 10.2.4.

We have outlined the architecture of a middleware, capable of realizing our idea. Currently, we are in the process of defining the implementation details which should lead to a proof-of-concept prototype. With the help of this prototype we plan to obtain detailed data to evaluate the performance and privacy enhancement of our approach in real world environments.

Future steps will include large scale application and evaluation. We plan to incorporate other current approaches presented in the Related Work section like *silent periods* proposed by Huang et al. [121] and the *behavior of neighboring nodes* presented by Horng et al. [71].

We anticipate two major challenges, the session tracking and the access point's capabilities. Especially application layer session management affords sophisticated mechanisms in order to provide accurate functionality and a untarnished user experience.

10.3 Conclusion

This chapter has illustrated some problems arising if location privacy in wireless networks is compromised. It has presented state-of-the art mechanisms for wireless personal and wireless local area networks. All current approaches are based on the concept of disposable pseudonym which seems to be suitable to improve location privacy by providing unlinkability and untraceability on lower layers. But in contemplation to sophisticated device and user fingerprinting mechanisms (see Chapter 9), all of these mechanisms suffer from the possibility to be circumvented. It is necessary to combine identity privacy improving mechanisms and location privacy approaches in order to circumvent attacks by fingerprinting techniques.

Further on, Section 10.2 presented our own contribution to enhance location privacy in WLANs. It is also based on disposable identifiers but incorporates ideas from WPANs and the concept of virtual network interfaces. It provides high levels of location privacy without suffering frequent communication disruptions and the need of a trusted-third-party (TTP).

Part III

Security and Privacy in Wireless Networks

11 Introduction

This part of the thesis illustrates the complexity of *Security and Privacy in Wireless Networks*. We present a number of selected articles out of different related fields such as mobile security, and location based access control.

All chapters are based on peer-reviewed journal or conference submissions and have been authored or co-authored by the author of this thesis during his PhD studies. The following sections present the abstracts of the original versions:

Location Aware Access Regulation for Wireless Networks - A Comparative Survey [106]

Due to the low cost and convenience of deploying wireless networks, they have replaced wired networks in many fields of application. This shift from wired to wireless networks invalidates some established security concepts that rely on the physical inaccessibility of wired connections, as the nature of radio propagation makes it possible to attack wireless networks from outside the established perimeter protection. As recent years have been proving, wireless network security should not rely solely on cryptographic measures, and therefore, additional mechanisms are needed.

The introduction of *location awareness* into wireless intrusion prevention systems could bring existing building access restrictions into play, with the consequence, that potential attackers would need to intrude the perimeter or spoof their location to gain full access to the infrastructure. In this article we present a survey of state-of-the-art security mechanisms for wireless networks based on location awareness. We provide an up-to-date overview on location determination based on wireless computer networks, location verification methods and on how these can be incorporated in security mechanisms and policies. This article provides a sound basis for all researchers planning to advance the field of location aware access control as well as related areas.

Threats Posed by Ultra-mobile Devices [111]

As information technology has gone increasingly mobile during the last years, ultra-mobile devices such as smart phones, PDAs and tablet computers are widely deployed in private and corporate settings. With the introduction of such devices into corporate infrastructures, new threats are arising which have to be considered by infrastructure administrators and IT security personnel. This chapter provides a detailed overview on attacks and threats and is aimed on administrators and security responsibles to help them estimate the risks imposed by these ultra-mobile devices.

Android Market Analysis with Activation Patterns [174]

The increasing market share of the Android platform is partly caused by a growing number of applications (apps) available on the Android market: by now (January 2011) roughly 200.000. This popularity in combination with the lax market approval process attracts the injection of malicious apps into the market. Android features a fine-grained permission system allowing the user to review the permissions an app requests and grant or deny access to resources prior to installation. In this paper, we extract these security permissions along other metadata of 130.211 apps and apply a new analysis method called Activation Patterns. Thereby, we are able to gain a new understanding of the apps through extracting knowledge about security permissions, their relations and possible anomalies, executing semantic search queries, finding relations between the description and the employed security permissions, or identifying clusters of similar apps. The paper describes the employed method and highlights its benefits in several analysis examples – e.g. screening the market for possible malicious apps that should be further investigated.

Android Security Permissions – Can we trust them? [136]

The popularity of the Android System in combination with the lax market approval process may attract the injection of malicious applications (apps) into the market. Android features a permission system allowing a user to review the permissions an app requests and grant or deny access to resources prior to installation. This system conveys a level of trust due to the fact that an app only has access to resources granted by the stated permissions. Thereby, not only the meaning of single permissions, but especially their combination plays an important role for understanding the possible implications. In this paper we present a method that circumvents the permission system by spreading permissions over two or more apps that communicate with each other via arbitrary communication channels. We discuss relevant details of the Android system, describe the permission spreading process, possible implications and countermeasures. Furthermore, we present three apps that demonstrate the problem and a possible detection method.

12
Location Based Access Regulation

12.1 Introduction

The number of deployed wireless networks increases every day. Due to the low cost and convenience of deploying wireless networks, they have replaced wired networks in many fields of application. The shift from wired to wireless networks invalidates some established security concepts. Hardwired networks are usually integrated structurally, and can be protected by building security or perimeter protection. With a state-of-the art intrusion prevention system (IPS) to protect the connection to the Internet, wired networks can thus be considered closed and therefore, more secure, as illustrated in Figure 12.1.

The nature of radio propagation makes it possible to attack wireless networks from outside the established perimeter protection. Figure 12.2 illustrates how wireless network coverage could extend to a public domain outside of a controlled building (protected area).

As building security and perimeter protection are not sufficient to avoid attacks against the wireless network, the general approach is to secure these infrastructures by cryptographic measures. Almost all state-of-the-art wireless computer network technologies provide strong cryptographic mechanisms. Following is a list of wireless network technologies and references to their cryptographic mechanisms:

- IEEE 802.11 (WLAN) [78,155,168];
- IEEE 802.15 (Bluetooth, ZigBee) [80,158,165];
- IEEE 802.16 (WiMAX) [83,88,116,195,198]; and

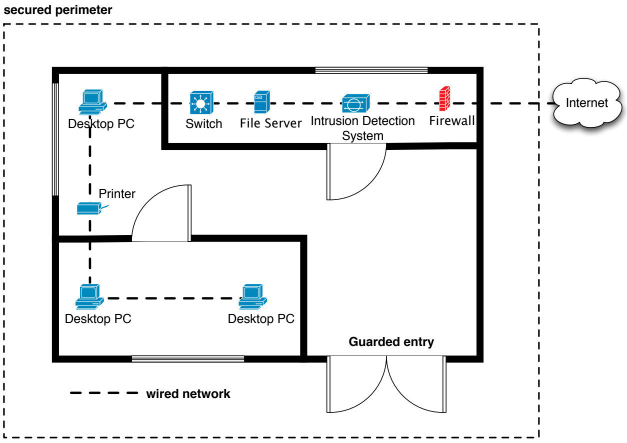


Figure 12.1: Wired-only Environment with Perimeter Protection

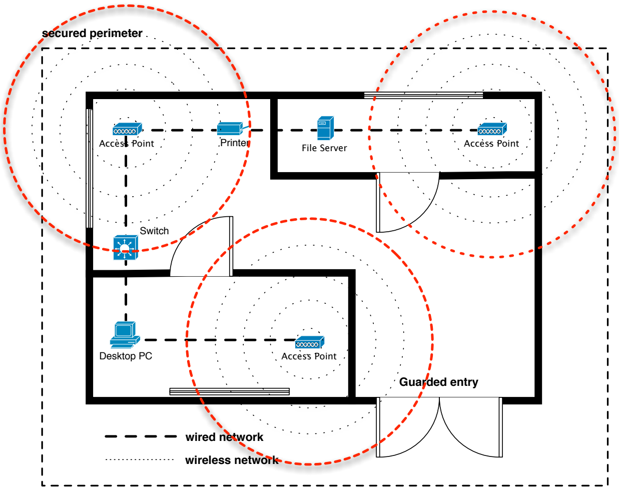


Figure 12.2: Environment with Wireless Components

- Cellular Based Networks [15, 17].

As history shows, attackers can fool many of these concepts by simply bypassing them. Passwords and digital certificates could be stolen or lost and legitimate users may be tricked by social engineering techniques into revealing their authentication credentials. Wireless network security should not rely solely on cryptographic measures.

The introduction of *location awareness* into wireless intrusion prevention systems could bring existing building access restrictions into play (see Figure 12.3). Location awareness means that the network knows to some degree about the geographical position of all connected clients. The network may then deny or limit connections to clients which are not in legitimate locations. Potential attackers would need to intrude the perimeter or spoof their location to gain full access to the infrastructure. This could be detected and prevented more easily.

In this article we present a survey of state-of-the-art security mechanisms for wireless networks based on location awareness. It provides an up-to-date overview on location determination based on wireless computer networks, location verification methods and on how these can be incorporated in security mechanisms and policies. Although various wireless networks are discussed, the focus lies on the IEEE 802.11 standards family.

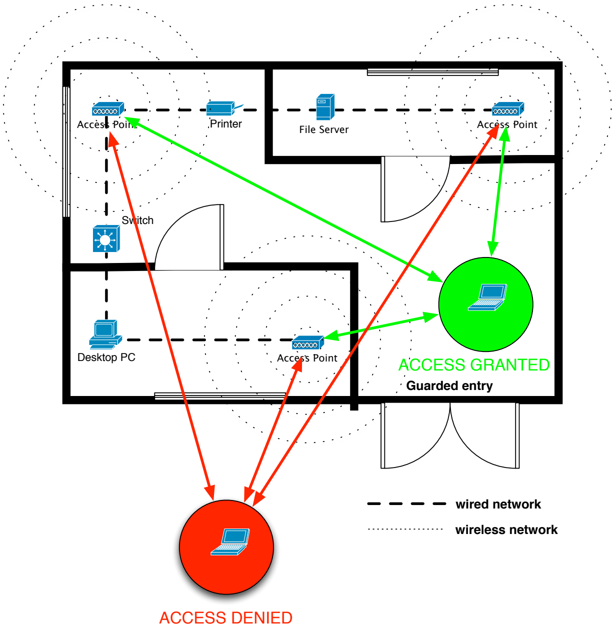


Figure 12.3: Location Aware Access Control System

This work is not the first survey that includes location determination methods in wireless networks, but in contrast to [69, 114, 138], it sets a focus on security and incorporates location determination and verification with location based security policies and regulations (see Figure 12.4).

12.2 Organization and General Assumptions

This section describes the organization of the article and defines necessary assumptions needed for the evaluation of the location determination and verification methods described later in this article.

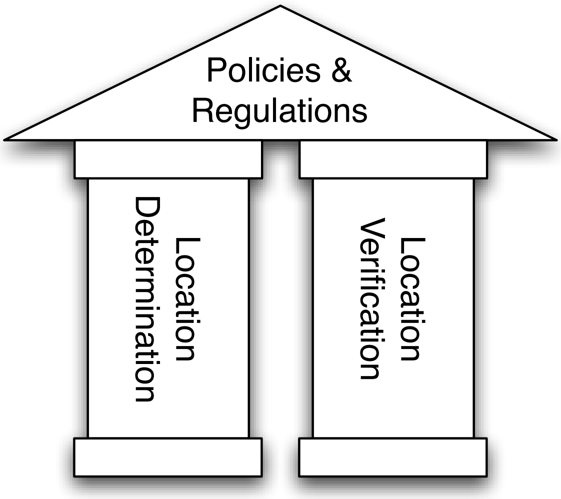


Figure 12.4: Building Blocks of a Location Aware Access Regulation System

12.2.1 Organization

The remainder of this article is structured as follows. Section 12.3 briefly describes possible location determination methods and evaluates them regarding security and practicability in security related systems. Section 12.4 presents location verification methods and popular implementations. It also evaluates their security related properties. Section 12.5 discusses well-known approaches using different access regulation philosophies associated with various location determination methods. Finally, Section 12.6 concludes the paper and presents future trends in this area.

12.2.2 General Assumptions

As this article evaluates location determination and verification methods regarding their applicability and practicability in security related systems, we need to define an *Adversary Model.*

Most of the following definitions are adopted from [23] as it represents a very general model used in many scenarios.

An attacker is called *external* if she cannot authenticate to the network. We call an attacker *internal* if a legal participant of the network has been compromised or is malicious.

Further on, we define two corresponding kinds of attacks to location systems: *internal attacks*, where an internal attacker convinces the location system that she is in a false position; and *external attacks*, where an external attacker convinces a legal participant and the location system, that this participant is at a different position from its true position.

Depending on the location system, an attacker can completely spoof its location (*location-spoofing attack*), enlarge its distance (*distance-enlargement attack*) or reduce its distance (*distance-reduction attack*) to a location determination sensor.

We assume that attackers are able to use directed antennas and amplifiers in order to alter their RF signal features (like the signal strength). They might further access and alter the firmware and drivers of their wireless hardware. This allows the alteration of all transmitted data packets on every layer in the network stack. Examples are MAC or IP address spoofing, packet-header flag tampering or checksum modifications.

12.3 Location Determination Methods

This section provides an overview and classification of location determination principles and algorithms. Figure 12.5 presents a breakdown of the approaches presented in this article.

Location determination in wireless networks has found a number of fields of application in recent years. Besides security related topics, context-aware computing in general is highly dependent upon location information. Especially in indoor environments or areas with a high density of elevated buildings classic positioning systems like GPS are not reliable. Alternative methods need to take their place.

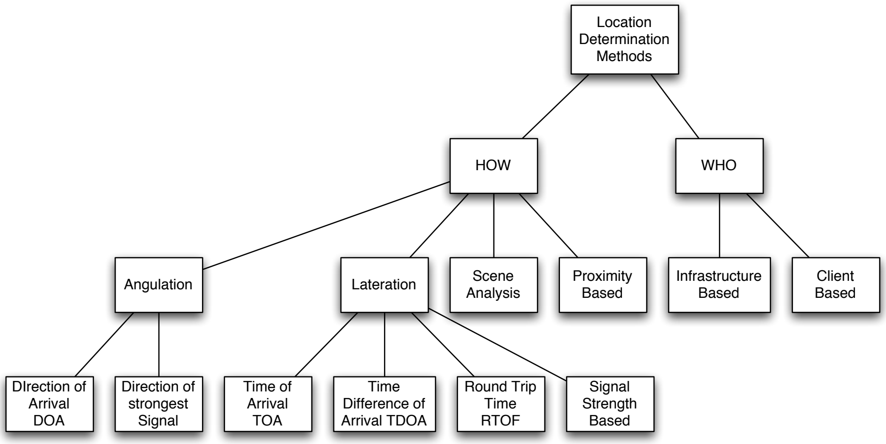


Figure 12.5: Classification of Location Determination Methods

As the pervasiveness of wireless network infrastructure dramatically increases, precisely in the just named areas, it is convenient to use the existing infrastructure for location determination. This article focuses on the use of wireless computer networks of the IEEE 802.11 family. A number of other technologies also offer the possibility of location determination. The interested reader may refer to the following literature to learn more about location determination on the listed technologies:

- GPS and DGPS [127];
- Cellular-Based [20, 137];
- Bluetooth [65, 95];
- RFID based localization [70, 132]; and
- Near Field Communication/UWB [50].

The article also includes an evaluation of above methods regarding security-related aspects.

12.3.1 Client or Infrastructure Based Methods

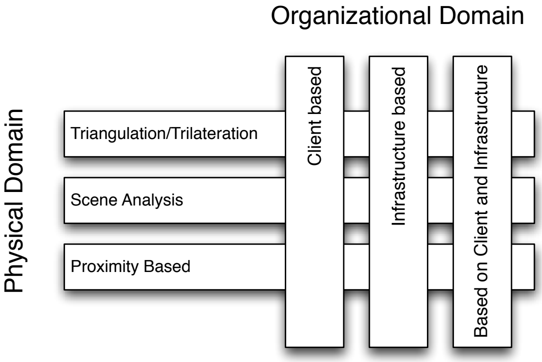


Figure 12.6: Relation of Classification Methods

As Figure 12.6 shows, this particular distinction of the *"WHO is carrying out the location determination process"*, we call it the *Organizational Domain*, can be seen as orthogonal to the classifications described later in this paper.

Generally, all approaches described in this section can be carried out either by the client or by the infrastructure. Some need the collaboration of the other party and some can be used without the client or the infrastructure noticing it. Deciding if a client- or an infrastructure based method is used in an implementation strongly depends on the use case. Both approaches offer advantages and disadvantages. Hybrid methods, combining the collaboration of the client and the infrastructure complete the discussion about the organizational domain.

12.3.1.1 Client Based Methods

This class of methods is characterized by the fact, that the location determination process is carried out fully by the device being located. The work load of data collection and position computation is handled by the device alone and presents no additional burden for the network infrastructure. This property promises good scalability and a decent base for many kinds of location based services. However, mobile devices with limited power and computational resources could be disadvantaged by this architecture. [30]

Many client-based positioning solutions are based on an agent-server architecture [9]. An autonomous agent is installed on all participating clients. Its purpose is to collect necessary data as signal strength values or time measurements [61] and compute the current location of the client by using this information.

Advantages: The computational load can be spread over the clients and does not burden the infrastructure. This scenario finds application in ad-hoc networks like wireless sensor networks which are often based on trust level models [49].

Disadvantages: As the client is responsible for the location determination it is easy to spoof the result. As this describes client only methods, the infrastructure is not allowed to participate in the process and provide for example location verification.

Conclusion: From a security point of view, these methods do have a significant weakness as they rely on the collaboration of the client who could be malicious. As attackers are usually not cooperative, these methods are not suitable to be part of a robust security solution [98].

12.3.1.2 Infrastructure Based Methods

In contrast to *client based methods*, *infrastructure-based approaches* work without any collaboration and generally even without any notice by the connected clients. Besides the favorable conditions for mobile devices, which are greatly unburdened regarding the usage of their resources, these methods do have a good suitability to be part of a network security architecture [32, 98].

Advantages: The network infrastructure alone is responsible for determining the clients location. No collaboration and computational power by the client is needed. This is an important advantage if low power mobile devices are included in the scenario. Depending on the physical location determination method, infrastructure based systems are harder to deceive.

Disadvantages: By using infrastructure based methods, the system cannot use additional information provided by the clients, such as GPS signals or scene related data such as the proximity to signal beacons. This could result in a decrease of performance and flexibility.

Conclusion: Infrastructure-based location-determination methods are generally more suited to be implemented in security related architectures than client based ones. They are considered robust against attacks by malicious clients as the overall security mainly relies on the used physical location-determination method.

12.3.1.3 Hybrid Methods

Hybrid methods generally rely on collaboration between the client and the infrastructure. The location determination process may be carried out by the client and the infrastructure is able to verify the client's position claim by some means. This process is called location verification and will be discussed in Section 12.4.

Advantages: The burden of collecting data and computing the location can be shared between the client and the infrastructure. Hybrid methods generally provide more flexibility than client or infrastructure based approaches.

Disadvantages: Due to the required information provided by the client, hybrid methods may be less robust against attacks than infrastructure based methods.

Conclusion: Hybrid methods can combine the advantages of infrastructure and client based approaches. The slight rise in complexity provides a gain in flexibility by preserving a high degree of security.

12.3.2 Triangulation or Trilateration

This section briefly describes the principles behind the mathematical concepts of *trilateration* and *triangulation*. It provides enough detail to obtain a general overview. For further insights on these topics refer to [114].

Lateration and angulation are geometrical techniques based on measured signal and communication properties. The accuracy of this data is vital to carry out reliable location determination. In indoor environments this accuracy is not only influenced by the precision of the used hardware, but also by radio propagation properties such as *multipath propagation* and a low probability for availability of *line-of-sight*. Until today, no reliable model for indoor RF multipath communication exists [138].

This fact is still the major limiting factor for all indoor location determination methods based on wireless computer networks as it greatly downgrades their performance.

Both, trilateration and triangulation based methods can be implemented as client only, infrastructure only or hybrid systems.

12.3.2.1 Triangulation

As the name *triangulation* hints, this concept is based on the geometric properties of the triangle. By determining the *Direction of Arrival* of a signal, from at least two different points of view (A and B, three points in three dimensional space), which are not located on a straight line to the target, one can calculate an intersection an thus locate the source of this signal (see Figure 12.7).

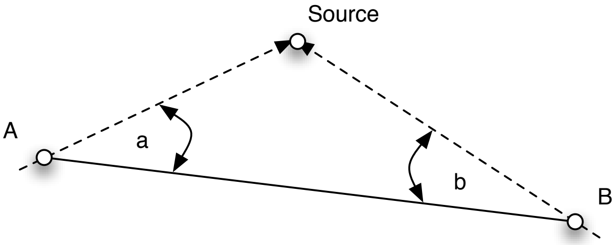


Figure 12.7: Principle of Triangulation

Besides the already mentioned issues in RF based indoor-localization, an additional major weakness of triangulation is its requirement for directional antennas.

Advantages: Triangulation based methods are very robust against all kinds of location attacks. The location determination can easily be carried out in real time and without any a priori measurements or computations.

Disadvantages: The main disadvantage is the need of directional antennas to determine the direction of the client's signal. In non-line-of-sight scenarios, RF signal propagation features as multipath propagation, can seriously downgrade the performance.

Conclusion: If the determination process is carried out by the infrastructure, triangulation methods prove very robust against all kinds of location attacks [23]. Due to their need for directional antennas, they turn out to be cost intensive.

12.3.2.2 Trilateration

Similar to triangulation, the concept of *trilateration* is also based on the geometric properties of the triangle. Instead of determining the angles between the source and the observation points, one measures the distance between them. This approach requires at least three points for measurement in two-dimensional space. (see Figure 12.8)

The process of distance measurement could be based on the following wireless communication properties:

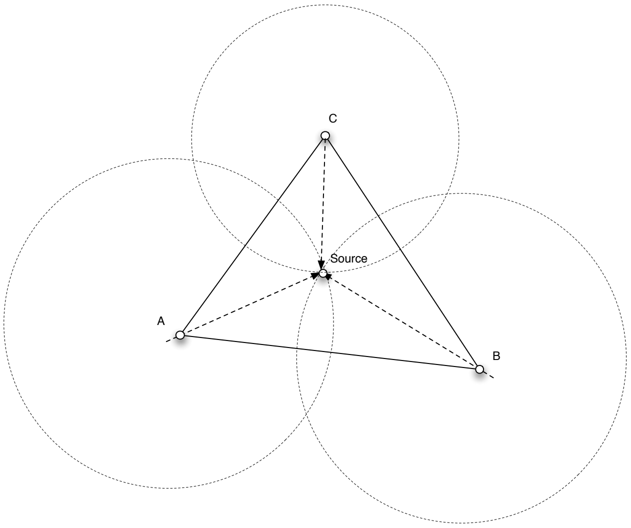


Figure 12.8: Principle of Trilateration

1. Time of Arrival - TOA
2. Time Difference of Arrival - TDOA
3. Return-Time-Of-Flight - RTOF (Round-Trip Time)
4. Received Signal Strength - RSS

Methods 1 to 3 generally require precise timing. The favorable method, from a cost oriented point of view is one based on measuring the received signal strength as most of off-the-shelf WLAN hardware is able to accomplish it with an adequate accuracy. State-of-the-art IEEE 802.11 hardware only provides a resolution of 1 microsecond in the time-domain. Due to the fact that the RF signal is traveling with light-speed, this resolution cannot be used to carry out distance measurements in the standard range of WiFi networks which is generally below 100 meters. Upcoming technologies as IEEE 802.11n could provide timing resolutions up to 1 nano second and therefore allow precise location determination based on signal traveling times.

Advantages: Trilateration can be based on various communication properties, leaving it a very flexible method. If RSS values are used, generally no special purpose hardware is needed as most wireless devices register the signal strength of incoming transmissions.

Disadvantages: If attackers use amplifiers or directional antennas they can easily spoof their location if no countermeasures, such as anomaly detection, are applied. Since multiple sensors at different locations are needed for RSS trilateration, it is not suitable for client based implementations. Similar to triangulation, signal distortions like multipath propagation can downgrade the performance of this method, as they disrupt the proportionality of signal strength and distance.

Conclusion: Trilateration techniques are very flexible as they can be based on various signaling and communication features. As generally no special purpose hardware is needed, they can be implemented at a low cost. If lateration methods are implemented without any additional security improvements, they are easy to deceive and not appropriate for security related applications.

12.3.3 Scene Analysis

RF-based *Scene Analysis* can be seen as a kind of *Location Fingerprinting* where various communication and signal features of certain locations or areas are collected and combined. These combined datasets need to be collected and stored in a system *a priori*. To determine one's location, it is necessary to collect these same features, combine them in the exact same way and try to match or approximate the outcome with the formerly created datasets [114].

A common example of these RF features is the usage of received signal strength (RSS) values. Figure 12.9 shows the RSS value map of one wireless access point in a certain area. The measured RSS values provide information about the possible location of a device. If two or more of these maps, belonging to different signal sources, are superposed, multidimensional vectors of RSS values can be assigned to each single location.

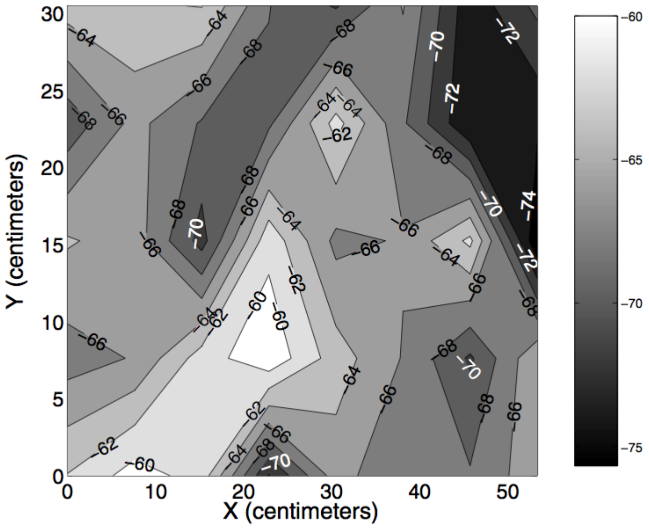


Figure 12.9: Visualization of RSS values in a 2D Area [196]

The major drawback of implementations based on location fingerprinting is, that these fingerprint collections (maps) have to be created in advance.

For further details and implementation examples please consult the following references: [89, 140, 194]

Advantages: Very robust against attacks with directional antennas and signal amplifiers. Generally no special hardware is needed. No complex computations are needed for the location determination process.

Disadvantages: RSS maps (see Figure 12.9) are needed for each sensor device. They have to be created a priori and require frequent re-calibrations.

Conclusion: RSS based scene analysis can be seen as an advancement of simple trilateration as it proves more resistant against signal distortions if more RSS maps are superposed. Anomaly detection mechanisms can easily be deployed, checking the plausibility of RSS values. This is a powerful tool against directed antennas and signal amplifiers. Nevertheless, this method is not sufficiently robust against location attacks, and therefore not appropriate for being part of a security relevant system.

12.3.4 Proximity Based

Proximity based techniques are the most basic approaches in location determination. They provide relative location information in a symbolic manner. *Landmarks* with well known coordinates, like base stations of the wireless network, represent points in a virtual grid over the environment. The distance between these *landmarks* represent the resolution of the location determination process based on this network. The outcome could be the ID of the transmitter, which detected the proximity of the mobile device [114]. Figure 12.10 presents an example demonstrating how the location of the client (Laptop PC) can be determined with Cell-ID 1.

Proximity based techniques are simple to implement and can be integrated with different types of physical media such as infrared radiation (IR), bluetooth, ultrasound and radio frequency identification (RFID).

An example is the deployment of IR beacons in every room of a building. These beacons could transmit their ID which is only receivable within their deployment area.

Advantages: This approach is generally very easy to implement if existing network infrastructure is used for cell identification. IR and ultrasound are usually delimited by room boundaries allowing these approaches to reach precision on the room level.

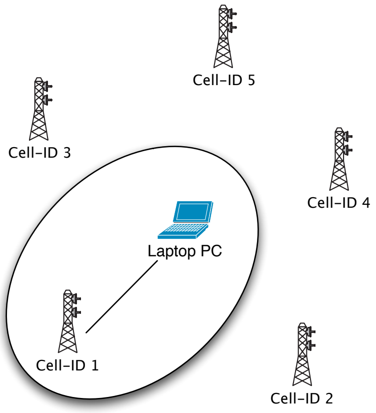


Figure 12.10: Mobile device connected to an RF Cell with a unique ID

Disadvantages: Infrastructure has to be deployed finely grained for higher precision. The distribution of the base stations define the resolution of the location determination.

Conclusion: Proximity based methods can be based on low cost hardware such as IR, ultrasound or Bluetooth beacons, allowing their cheap implementation. In order to be part of a security relevant system, they need to be combined with a location verification approach.

An alternative application for proximity based systems are *public WLAN-based positioning systems* (WPS). As a prominent member we note the Skyhook localization system [169] which is widely deployed on mobile platforms. WPS uses existing wireless access-points and a database holding their geodetic positions. A client reports all currently received SSIDs to the system which uses a *multiple nearest-neighbor* approach to interpolate the possible position of the client. In areas with a high density of registered access-points the localization error could be as low as some meters [180].

12.4 Location Verification

Section 12.3 provided a general overview on how the approximate location (plus or minus some error distance) of a certain device, in a wireless network can be determined. Based on this location, various security mechanisms can be implemented.

A major concept in this field is the *verification of location claims*. It is described best by an example (see Figure 12.11): A certain *device R* claims to be located in a certain *area A*. This area can be a single room or even a building. Every device located in this very area should be granted access to a specific resource whereas devices that are out of its boundaries must not get access. The network infrastructure provides a *verifier v*. An entity that is able to validate the claim of R according to the *in-region verification problem* [157].

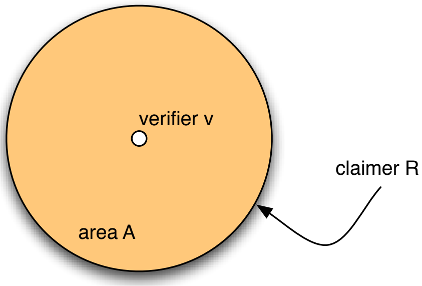


Figure 12.11: Location Verification

Location verification can be used to extend location determination methods and improve the security and reliability. In some cases it may work without requiring a dedicated location determination method and be the base for a security related system.

The next sections present popular research and implementations in the field of location verification.

12.4.1 Distance-Bounding Protocols

Stefan Brands and David Chaum proposed the first solution to the problem of *verifying the distance of a prover to a verifier* [18] in 1994 by presenting the *distance bounding protocol*. It is based on the timing delay between sending out a challenge and receiving back the corresponding response[1]. In the following, Srdjan Čapkun et al. [22] extended the protocol to SECTOR, a mutual authentication protocol using distance bounding. As vulnerabilities to this protocol have been discovered, Dave Singelee and Bart Preneel of the K.U. Leuven presented modifications to render it secure against the so called *terrorist fraud attacks* [166]. Another solutions, similar to the approach of Singelee and Preneel has been published by Laurent Bussard [19]. In 2006, it was again Srdjan Čapkun, this time with Jean-Pierre Hubaux [23], who advanced this distance-bounding location verification by pairing it with multilateration. They assume that an increasing number of verifiers also increases the trustworthiness of a location claim (see Figure 12.12) as an attacker needs to trick all verifiers at the same time and with coherent spoofs.

In 2010, Rasmussen and Čapkun demonstrated a practical implementation of a distance bounding protocol [149]. The implementation used custom hardware

[1] In practice, a series of these rapid exchanges is used.

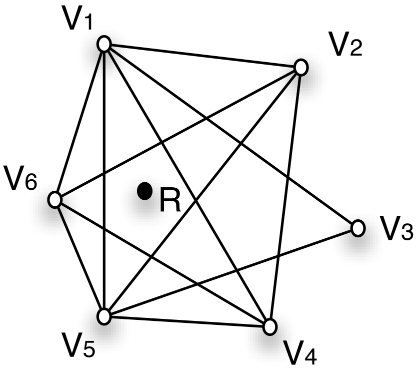


Figure 12.12: Multilateral Verification by [23] with 6 verifiers

with sub-nanosecond processing delay, and provided a precision of approximately $15cm$.

Advantages: The distance bounding protocols measure the propagation delay of radio waves. Since these waves travel at the speed of light, an attacker is not able to mount a distance reduction attack. Furthermore, some of the proposed protocols cryptographically bind the distance bound to the prover , so that even man-in-the-middle attacks are unfeasible.

Disadvantages: Distance bounding protocols are extremely sensitive to processing delays. A processing delay of $1ns$ adds approximately $30cm$ to the distance bound. A practical implementation of such a protocol thus requires extremely fast hardware.

Conclusion: Distance bounding protocols are robust against distance reduction attacks. However, such protocols require fast hardware, so they may not be suitable for implementation in current network deployments. As an example, the 802.11 standard with the a/b/g amendments has a time resolution of $1\mu s$, which corresponds to a distance error of approximately $300m$.

12.4.2 Geodetic Location

Dorothy Denning and Peter MacDoran proposed that the *geodetic location* (to be somewhere) could extend the established triple in authentication:

- to know something - e.g. passwords
- to have something - e.g. security tokens
- to be someone - biometric property

They developed a system called *CyberLocator* for achieving location-based authentication. The location determination process is based on the *global positioning system* (GPS) and the location verification relies on the unpredictability of GPS data fluctuations due to subtle satellite orbit perturbations, which are unknowable in real-time, and intentional signal instabilities (dithering) imposed by the U.S. Department of Defense selective availability (SA) security policy [37].

Advantages: The *CyberLocator* system could potentially verify the location of the client with a high degree of precision. According to the claims of the authors, it prevents distance reduction, enlargement and location spoofing attacks.

Disadvantages: The system is not suitable for indoor use, due to the dependence on GPS signal reception. Furthermore, ordinary GPS receivers used for navigation are not suitable for use with the *CyberLocator* system. Special purpose GPS receivers are needed in both the infrastructure and all clients.

Conclusion: The description of the *CyberLocator* system in [37] does not provide enough details to explain why the location verification is secure. If a verifier is able to verify that a location claim is correct, a prover might calculate a spoofed position based on the GPS signals received as well, as long as the prover is within 2000 to $3000km$ of the spoofed position. It is not clear whether the security of the system is due to the difficulty of performing such a calculation in real time or if there are other security mechanisms involved.

12.4.3 The Echo Protocol

Naveen Sastry et al. [157] from the University of California, Berkeley developed the *Echo Protocol* in 2003. The *Echo Protocol* is extremely lightweight, and it does not require time synchronization, cryptography or very precise clocks. It is well suited for use in small, cheap, mobile devices. The location determination process requires RF and ultrasound transceivers. The protocol is similar to the RF based distance bounding protocol, the difference is that the response from the prover to the verifier is transmitted as ultrasound rather than RF. Since ultrasound travels at a much slower speed than light, this approach allows for a higher degree of precision when the processing time makes RF based protocols unreliable. The *Echo Protocol* is vulnerable to distance reduction attacks if the attacker is able to connect to an ultrasound transceiver, e.g. a speakerphone, inside the controlled area. Using a speed of light physical medium, e.g. RF, to connect to an ultrasound transceiver inside the controlled area violates one of the fundamental assumptions of the *Echo Protocol*.

Advantages: The *Echo Protocol* requires no pre-established trust relationship between the prover and verifier. It is extremely lightweight, and thus suitable for devices that are not able to meet the stringent processing time demands of the RF based distance bounding protocol.

Disadvantages: The *Echo Protocol* requires ultrasound transceivers in both the provers and verifiers. The protocol is vulnerable to distance reduction attacks.

Conclusion: One avenue of future research could be to determine if the microphones and loudspeakers in laptops, PDAs and mobile phones could be used as (ultra)sound transceivers.

12.4.4 Proximity-Proving Protocol

Brent Waters and Edward Felten [189] developed a protocol to determine the proximity of wireless devices by measuring signal round-trip times, including party identification based on X.509 certificates and a PKI. Waters et al. presume that the *location claimer* and *location verifier* are temper-proof devices. This approach has been improved by the authors and published in [188].

Advantages: The method offers certificate based authentication which provides the whole power of PKI systems including their established capabilities and implementations. Further on, it proves resistant against severe distance-reduction attacks due the use of triangulation for location determination.

Disadvantages: The main disadvantage of this approach is the need for special, temper-proof hardware. Further on, it depends on the availability of a PKI and involves multiple parties in the verification process.

Conclusion: Waters and Felten were the first to develop a location verification system based on wireless networks that offers integrity and privacy. They incorporated authentication and identification in a round-trip-time based distance bounding approach. However, the requirement for tamper-proof devices and a PKI limits the flexibility and practicability.

12.4.5 The Secure Location Verification Proof Gathering Protocol (SLVPGP)

Michelle Graham and David Gray from Dublin City University propose a location verification approach based on distance-bounding protocols by Brand and Chaum [18]. They believe that enlisting the aid of neighboring devices provides the basis of such a method. Their solution allows a *claimant* to make a location claim, and then have this claim verified by an independent *proof provider*. They state that using this approach, a verifier can determine if a location claim is possible and by selecting suitable proof providers one can limit the size of the area in which the claimant can be located. They extend the original protocol by security measures and rely on tamper proof devices to keep the devices cryptographic keys [58, 59].

Advantages: SLVPGP does not mandatorily require any network infrastructure as other participants of the network can act as dedicated *proof-providers*. Even the required central authentication authority could be deployed on a node in the ad-hoc network. This offers a high degree on flexibility and allows the deployment in wireless sensor networks.

Disadvantages: The main disadvantage of this method is the need for temper-proof devices to hold the cryptographic keys required for authentication purposes.

Conclusion: The SLVPG protocol can be seen as a variation of Walters and Feltons Proximity-Proving Protocol. As their location proof is gathered from neighboring devices, a large-scale environment with multiple participants is required. Their approach is very suitable for ad-hoc networks as mobile sensor networks but is limited by the need of tamper-proof devices.

12.4.6 RFID based Distance Bounding Protocol

Gerhard Hancke and Markus Kuhn from Cambridge proposed an RFID based distance bounding approach in [66]. They argue that standard RFID tokens are not suitable for proximity authentication as the are vulnerable to relay attacks. Attackers can circumvent the limited range of the radio channel by using transponders that forward signals via larger distances. They introduced a distance bounding based protocol allowing to cryptographically verify round-trip time delays to assure the physical proximity of the RFID tag. Hancke and Kunh's protocol has been extended by Jason Reid et al. from the Queensland University of Technology. They call their approach *Protocol 2* [151] and it resembles the first symmetric key based distance bounding protocol resistant to so called *terrorist fraud*. Unfortunately does their protocol still miss formal verification.

As these protocols are based on close range scenarios, they cannot be transferred to wireless local area networks but have been shortly described for the sake of completeness.

12.5 Access Regulations

Many well-known methods exist which authenticate a users identity. Usually they are based on either something the user knows (e.g. password), has (e.g. smart cards) or is (e.g. fingerprint, retina profile). In the last decade, research in the area of access control and regulation has begun to integrate traditional access control mechanisms with conditions based on the physical location of users. However, location information of users should also be considered sensitive in the context of privacy protection. Thus, access control solutions should be developed which not only provide authentication based on this information but also prevent its disclosure. In this section we will discuss well-known approaches using different access regulation philosophies associated with various location determination methods.

12.5.1 Cricket Location-Support System

One of the first and most cited publications in this area is the one of Priyantha et al. [146] about the Cricket Location-Support System. This is one of the first work in the area of context-aware applications. It focuses on radio and ultrasonic signals available from off-the-shelf components rather than on technologies like Global Positioning System (GPS) or cellular infrastructure. Using radio and ultrasonic signals made it possible to deploy location-dependent applications also inside buildings and residential homes which can be seen as major step towards ubiquitous computing [190]. The Cricket Location-Support System has been designed to allow applications running on user devices and service nodes to learn their position relative to each other without relying on centralized management or control and explicit coordination between beacons. Their system is based on exploiting the difference in signal propagation times of radio-frequency and ultrasonic signals. Based on these difference each node correlates the received signals and infers the space it is currently in. Although their work focuses primarily on detecting location and reducing interference they provide some thoughts on how to use their system for gaining access to nearby services.

The first shortcoming, in terms of access regulation, of the Cricket approach is that all location claims are sent in plain and are not secured by any cryptographic means. Thus, it is not possible to verify the identity of any node. This behavior is somehow intended, since privacy was one of the authors main design goal. Also with using their system it is not possible for the system itself to determine the location of a node but rather the node is able to detect in which space it supposedly is in. Thus, it is not possible without modification to let the system decide if a node should get access or not. Another disadvantage of their approach is the requirement for an additional ultrasonic signal transceiver, which is not commonly deployed in standard computing devices such as personal computers, notebooks or mobile phones. Nevertheless, the use of such an ultrasonic signal simplifies the location determination and access regulation process.

12.5.2 Cooltown based approaches

The HP Cooltown project explored future context and location-aware applications based on a vision where people, places and things all have web-based representations [90].

In [24] they have described the fundamental requirements for a generic framework which incorporates location into its fabric and allows to create an entity, called *Place Manager*. Such a place manager can be used to develop location-based applications for a diverse set of places in a generic manner. These fundamental requirements are:

- Relationships with other entities - The web representation of a place needs to represent the people and things present within the place.

- Open-ended entity description for programmatic search - The ability to dynamically discover other resources which are needed by a specific service without a prior specification of the possible arguments.

- Security policy - A place should be able to authenticate a person/device's presence in that place. It is also advisable to be able to specify and control what kinds of relationships are allowed in a place and with which other entities.

In their work they realized that different places have different security requirements. Their solution was to process all these requirements through the place manager. Thus, their place manager must be aware in which network a specific user is and issue URL's with the appropriate addresses that can be reached by that user. Then a proxy located on the corporate firewall, would recognize the public URLs issued by the place manager and reissue the request to the corresponding private URLs in order to access place resources which are not on the public network. Also, some services must be reserved for designated people who must authenticate themselves. The need for services to be able to provide different authorization levels is not a new requirement for places.

The approach they advocate to provide location authorization for places does not require the act of proving location to be atomic with requesting access. A short-range wireless beacon emits a token that contains an encrypted timestamp along with the URL of the place manager. The token is placed in a cookie by the beacon receiver and is presented to the place manager on each access. The cookie is valid until the current time passes the time in the timestamp. When the cookie becomes invalid, the beacon receiver must get a new token and create a new cookie from it. Basically, they are concerned with location security in the sense that there are no precautions against forwarding the token over a wide-area network, which would enable others to spoof the location. However, they outline if this would be required the place manager would need to enforce that all communication must be done over a secure link. Additionally tokens must be generated individually for specific users. By encrypting the token using a secret shared between the user and the place manager, retransmitting the token will not enable outsiders to spoof their location. If it is important to guard against an inside attack, then client authentication must be required as well, and the secret key is tied to the user's identity.

Their approach outlines some general theoretical foundations of the problem itself but does not provide any particular technical solutions for the very complex problem. The theoretical foundation of their work is similar to others presented in this survey but since no concrete statements are made it is not possible to verify their approach and evaluate the security of their system.

12.5.3 PAC: Pervasive Access Control

In the work of Michalakis [124] PAC, a location access control system for pervasive computing environments, is outlined. The PAC system uses the Cricket system for location determination but provides location dependent access regulation using random short-term cryptographic keys disseminated through the beacons. In addition to the Cricket system, each beacon broadcasts a Location ID (`LID`) and a time-varying Location Code (`LIDCODE`). This information is used by the nodes to create ticket requests which are authenticated with a *HMAC* using the received `LIDCODE` as key. A central authority verifies the request and grants a ticket for a specific time, service and network address. Thus, the requesting node can access a specific service for a certain amount of time. In order to prolong the use of the service the node has to renew the ticket using the same process. With this system it is possible to restrict access to services based on the nodes location using anonymous authentication. Nodes which are not in the vicinity of the beacon can not learn the LIDCODE and are thus not able to generate a legitimate ticket request. With the other information included in the request it is also guaranteed that malicious nodes can not get access to services other then the requested ones. Also no other nodes, then the requesting one, can get access to services due to the binding of the granted ticket to the network address.

The PAC system is susceptible to man-in-the-middle attacks (MITM) since it does not require entity authentication but only authentication of the transmitted messages using the distributed LIDCODE. If a malicious node is in the proximity of a beacon and acts on its own as beacon other nodes can be fooled to use the malicious beacon as access point. The malicious node then requests services on behalf of the other nodes and works as proxy for the requested services or provides services itself. Without entity authentication it is not possible for requesting nodes to verify if the access point with which it is connected is a legitimate one. One attack which is also possible is the forwarding attack. In the forwarding attack a node is placed in the vicinity of the beacon and registers itself for all possible services the beacon provides. If the node possesses another wireless network interface it can provide these services to nodes within in the range of its wireless network. Thus, the range of the original beacon can be extended significantly without the PAC system being aware of this fact.

12.5.4 LRBAC: Location-aware Role-based Access Control System

Ray et al. [150] show in their work how the Role-Based Access Control (RBAC) model can be extended to incorporate the notion of location. They show how different components in the RBAC model are related with location and how this location information can be used to determine whether a subject has access to a given object. Their model is suitable for applications consisting of static and dynamic objects, where location of the subject and object must be considered before granting access. The constraints specified by Core RBAC are present in any RBAC application. The model requires that users (human) be assigned to roles (job function), roles be associated with permissions (approval to perform an operation on an object), and users acquire permissions by being members of roles. The model does not place any constraint on the cardinalities of the user-role assignment relation or the permission-role association. Core RBAC also includes the notion of user sessions. A user establishes a session during which he activates a subset of the roles assigned to him. Each user can activate multiple sessions; however, each session is associated with only one user. The operations that a user can perform in a session depend on the roles activated in that session and the permissions associated with those roles. The different components of RBAC are Users, Roles, Sessions, Permissions, Objects and Operations. Figure 12.13 illustrates how these components are related with Location. The multiplicity of these relationships are indicated by presence or absence of an arrowhead. The absence of an arrowhead indicates a multiplicity of ***one*** and the presence of arrowhead indicates a multiplicity of ***many***.

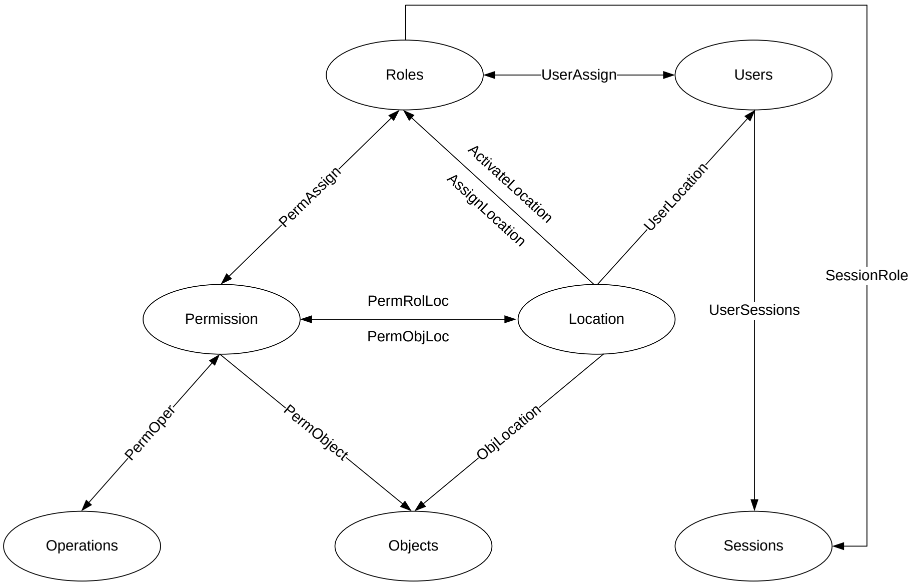


Figure 12.13: Relationship of RBAC entities with Location

In addition they have formalized their access control model based on RBAC which integrates location information using the Z specification language [145]. The specification assumes the given types USER, ROLE, SESSION, PERMISSION, LOCATION, OBJECT, OPERATION, which enumerate all possible users, roles, sessions, permissions, locations, objects and operations respectively.

They showed how in theory location could be integrated into an existing access control strategy such as RBAC. They provided textual description of how location can be integrated with the existing types of RBAC and also provided a formal model to show them. Thus, it is a good starting point for other solutions. Most of the subsequent solutions are rather practical than formal and none has explicitly made any connection to the LBRAC model although they all provide some sort of access control mechanism. Nevertheless, they all can be considered related work.

12.5.5 LAAC: Location Aware Access Control

In [31] a novel Location-Aware Access Control protocol (LAAC) based on a coarsely defined location area that is enclosed by overlapping areas of multiple access points is proposed. They derive a location key from the overlapping access points beacon information. Thus, in their model it is possible to track the location of mobile devices using this location key. LAAC does not require additional hardware such as GPS or ultrasonic devices but they assume an infrastructure WLAN system based on IEEE 802.11, and that mobile devices can communicate with each other through the access points. In addition, they require that access points can be equipped with directional antennas in order to have precise signal coverage and that signals do not bounce. Since they also assume that mobile devices have sufficient computational capabilities they designed LAAC as a client-based protocol. They also state that each mobile device may carry the public key of each access point if necessary.

Access control areas are divided into two types namely access-granted and access-denied areas. The access-granted areas are created from zones which are covered by multiple access points. Inside these zones specific geometric shapes are specified which represent the areas where access is granted. The access-denied areas are the spaces outside of the access-granted areas. Each access point broadcasts its nonce generated by a random number generator periodically. The mobile devices collect all nonces of the access points in the area, and derives its location key k_i by XOR-ing all the nonces. Then the mobile device constructs an access request (AR) using a strong collision-resistant hash function value of k_i and sends the request to the associated access point. The access points of the same AP group are aware of each other, and exchange their nonces. Thus, the associated access point can derive the same location key as the mobile device and authenticate the mobile device.

Each access point must select a nonce which is not already used by other access points in order to not allow for nonce cancelation in the location key calculation. In order to protect against replay attacks and allow for session termination if users move out of access-granted areas two timeouts are introduced. One timeout specifies the beacon broadcasting interval of nonces and the other the location key lifetime. Protection against forwarding or wormhole attacks are not included. Also Sybil attacks, although in their case it would be more appropriate to refer to an mutual authentication problem, are also not covered in their initial protocol. In general, this protocol is similar to the PAC approach with its strengths and caveats with the addition of more fine-grained access-granted ares rather than to allow access for the whole coverage area.

12.5.6 CA-RBAC: Context-aware RBAC

Kulkarni et al. [99] present a context-aware RBAC (CA-RBAC) model for pervasive computing applications. They have modified the common RBAC model in order to allow for dynamic integration of context-based services with an application. Context-sensitive information is used in several parts of the new CA-RBAC such as in role admission policies, execution permission policies, and dynamic service access policies. Due to the dynamic nature of context information they have integrated mechanisms for revocation for roles and permissions in the case of changes to the context conditions in their model. In addition, they have designed a developer framework, which is based on their model, for building context-aware applications. Their framework provides mechanisms for specifying and enforcing context-based access control requirements.

They consider a broad range of context information which may be relevant for their access control model, such as temporal constraints, role membership history and user location. They use two representative context-aware applications to demonstrate the necessity and applicability of their context-aware RBAC model. These two applications are:

- Context-aware Patient Information System
- Context-aware Music Player

In terms of context awareness, location was the most dominant information used inside their demo applications and is used in all extensions. They integrated methods for dynamically discover services and providing access to them under certain context conditions. Due to the impossibility of knowing a priori which services will be accessed they created an abstract object which is used as proxy. Also, the notion of role permissions has been updated to reflect the requirements of pervasive computing. Thus, now it is possible that a permission which is invoked by different role members is executed on different objects in compliance with their context. Previously it was rather static in the sense that a permission has been invoked always on the same object regardless which member of the role performed it. Another modification is the introduction of context information to role operations. Role operations may require that context information is checked before the operations are invoked and that sessions which are created during such operations are terminated as the context information changes. An addition called *resource access constraints* has also been proposed in other work, called *parameterized roles*, but with less attention on context information.

With the proposed theoretical framework for context-aware role based access control several attributes required to address pervasive computing are now available for specifying context-aware policies. Although, the model has addressed most of the issues raised in such dynamic environments some of their features still seem to inflexible to account for real-world scenarios. However, the proposed modifications can increase the security of RBAC systems under such circumstances.

Method	**Attack**
Client Based	Complete location spoofing
Infrastructure Based	Robust against Attacks, dependent on physical domain method
Hybrid	Inherits vulnerabilities from Client and Infrastructure based methods
Triangulation	Robust against location attacks
Trilateration	Complete location spoofing
Scene Analysis	Complete location spoofing
Proximity Based	Distance reduction and distance enlargement

Table 12.1: Location Determination Methods and possible Attacks

Method	**Attack**
Original Distance Bounding Protocol	Distance enlargement
Improved Distance Bounding Protocol	Robust against location attacks
Geodetic Location	Robust against location attacks
The Echo Protocol	Distance enlargement and Distance reduction
Proximity-Proving Protocol	Robust against location attacks
SLVPGP	Robust against location attacks

Table 12.2: Location Verification Methods and possible Attacks

12.6 Conclusion and Future Trends

In this chapter we described and evaluated the components required to build location aware access regulation systems. Section 12.3 discusses location determination methods for wireless networks with a focus on indoor environments and evaluates their applicability in security related systems. Section 12.4 provides insights on location verification techniques and also evaluates them according to their security and applicability properties. Section 12.5 completes the description of all necessary building blocks for location aware access regulation systems by discussing available location based access regulation policies and systems.

Table 12.1 provides an overview of all described location determination methods and possible attacks against them. Our conclusion on these approaches is, that most of the current solutions are not suitable for usage in security related systems. Either their lack of precision is not tolerable or they are too easy to deceive. Only triangulation based on signal-runtime measurements theoretically provides a satisfactory degree of reliability and security.

Table 12.2 provides an overview of the discussed location verification methods, and their relevant vulnerabilities. Some of these approaches are fit for application and provide a high degree of security. In particular cases, location aware access regulation systems could use one of the location verification methods described in this article, without a dedicated precise location determination method. For all other applications, hybrid approaches are needed. They could be composed of multiple location determination methods, strengthened by location verification. [41, 62, 87]

A number of security policies, guidelines and frameworks capable of addressing location awareness already exist. Unfortunately, localization tools and techniques[2] are not technically mature.

One may assume that location determination and verification will be addressed in future wireless LAN standards, and therefore the hardware limitations for location determination and verification methods will be overcome. As precise locations are the foundation of location aware access regulation systems, this fact will seriously improve their development and performance.

[2] regarding state-of-the-art wireless computer networks such as IEEE 802.11

13
Threats Posed by Ultra-mobile Devices

Information technology has gone mobile in the last years. Laptops and notebooks are replacing classic desktop computers in many places, especially in the business sector. Although, this did not happen over night, IT security had great difficulties adopting their strategies in order to get a grip on new attack vectors introduced by the rising mobility of their infrastructure.

Methods as *harddrive encryption*, *user authentication based on smart cards* or *personal firewalls* and *anti-virus software* are very well established in corporate environments. They are able to protect critical data very effectively against unauthorized access, especially in connection with corporate IT-security infrastructure like firewalls and intrusion detection systems (IDS).

With the broad deployment of so called *ultra-mobile devices* like smart phones, personal digital assistants (PDA) and tablet computers, new threats to critical data and protected infrastructures are arising. Some of these devices are computationally very powerful, to some degree even comparable to Laptops and desktop PCs. They provide large storage capabilities up to several gigabytes, allowing them to carry all kinds of data, private as well as business related. Due to missing access protection, loss or theft of such a device could result in the exposition of critical corporate data.

Another important aspect is the excellent connectivity of these devices. Many of them provide hardware interfaces to several different wireless networks simultaneously. State-of-the-art devices implement Bluetooth for personal area, WiFi for local area and cellular-network technologies like GSM or UMTS for wide area network access. RFID, infra-red (IR) and WiMAX might also be available. Localization technologies as GPS and integrated cameras are completing the portfolio of these devices.

All these properties of new generation ultra-mobile devices introduce new attack vectors which have to be addressed by IT security professionals in order to adopt security policies and guidelines.

13.1 Threats related to Smart Phones

Smart phones, as well as other ultra-mobile devices differ in at least three points from laptops (regarding IT security):

1. The lack of mature security software like storage encryption, personal firewalls and anti-virus programs.
2. Their compact size allows for more mobility which rises the risk of loss and theft.
3. Outstanding energy resources and excellent connectivity lead to non-stop connections with public networks as the Internet through mostly unsecured channels.

These properties lead to major threats regarding the *loss of critical data* and several *privacy issues*, which are discussed at this place.

13.1.1 Loss of Critical Data

The loss of critical data has lately not only been a topic for security professionals as stories of thousands of lost credit card numbers has been circulating the press. Lost compact discs in hotel rooms and laptops forgotten in yellow cabs have caused serious discussions in the privacy and security communities. Full encryption for hard drives and portable media like compact disks and USB sticks has since become widely deployed in the business sector.

Since smart phones and PDAs play an important part in the daily routines of business people, they usually hold communication related data like address books, call history and business emails including their attachments. But in contrary to laptops, storage encryption is but sparely in use. This is partly because adequate systems are not available or IT security responsibles are simply not aware of the risk they are facing.

13.1.2 Privacy Issues

This section discusses threats to the user's privacy, if an attacker is able to access or control an ultra-mobile device without the user noticing it.

The excellent technical abilities of modern mobile devices intend to provide the user various information facilitating his daily routines. They allow making phone calls, sending and receiving emails, view and edit files and documents, providing the exact location of the device for navigation means, taking pictures and videos when ever wanted. This is just to name a few.

The almost non-stop connection to the Internet exposes ultra-mobile devices to the same attack vectors as personal computers and servers, but they are usually not protected by corporate firewalls and intrusion detection systems and are therefore very vulnerable and sometimes an easy target.

Once an attacker gains control over such a device, she can access all data stored on it and furthermore utilize the device's hardware. This allows to eavesdrop on phone calls, intercept *short-message-service* (SMS) messages, track the user's movements and even to activate the device's camera and taking pictures whenever the attacker wants to.

The consequences to the user's privacy are easy to estimate.

13.2 Attack Scenarios

Threats to ultra-mobile devices can be caused by different attack scenarios. Attackers may place malicious software on the device, similar to trojan horses, or exploit vulnerabilities in communication services via remote access. Furthermore, smart phones controlled by an attacker may be used to attack secondary systems as corporate networks and bypassing security systems as firewalls.

13.2.1 Malicious Codes

To provide state-of-the-art service, mobile device operating systems became very complex and powerful. The most important implementations are closely aligned to classic computer operating systems as Apples Mac OSX, in the case of the iPhones iOS [6]. Microsoft's mobile systems [125] are related with Windows and Google's Android [57] can be seen as a Linux derivation.

The usage of these sophisticated operating systems provides a variety of new fields of application, especially for software developers. They can implement their products based on shared libraries which also facilities the transformation of desktop software to the related mobile platform. Many of these systems also provide so called *Mobile Application Stores* to distribute software fast and convenient to the mobile devices. Examples for these stores are the *Apple App Store* [7], the *Android Market* [55] or the *Blackberry App World* [152].

Derivations of desktop operating systems generally inherit many vulnerabilities from their ancestors exposing them to the same or similar attack vectors. An attacker will try to place malicious codes on the device to use it for following actions. This intrusion mainly happens based on one of the following subsystems :

1. Remote Access
2. Installed Applications

13.2.1.1 Remote Network Attack

In remote network attacks, malicious code is placed on the target device directly via an active network connection. The requirement for this attack is some application running on the mobile device, that is listening on a TCP or UDP port for incoming transmissions. If this application offers some vulnerability, an attacker could use it to inject malicious code on the device.

in the desktop PC and server sector, such attacks are very common and lead to millions of infected computers all over the world. These computers are sometimes centrally controlled, operating as so called *botnets*. For further information on botnets we refer to [46].

Comparing ultra-mobile devices to classic personal computers, important risk circumstances for mobile devices are:

- **Public IP addresses**
  A great part of the connectivity of mobile devices is based on cellular networks like UMTS or HSDPA. As these protocols rely on IP for data traffic, IP addresses have to be assigned to each connected device. As these IP addresses are generally taken from a public address pool, they are reachable from all over the world. An attacker who knows which address pools a cellular network provider uses for mobile devices is able to identify her targets in order to launch well aimed attacks.

- **Missing Firewalls**
  Mostly due to limited resources, the vast majority of ultra-mobile devices is not equipped with a firewall. This poses a serious vulnerability, especially in combination with a public IP address. While laptops or desktop computers are generally protected by corporation and personal firewalls even though they usually possess private IP addresses which are not routed in the internet, most ultra-mobile devices are left unprotected. Even if laptops are connected using cellular networks the are usually protected by personal firewalls.

- **Security Updates**
  State-of-the-art desktop and server operating systems are generally offering automatic update functionalities. They allow fast and reliable distribution of security relevant patches soon after a new vulnerability has been identified. Current ultra-mobile device operating systems (OS) are still missing this capability. Although some systems provide automatic update functionalities, they usually replace the whole OS and do not apply delta fixes. As OS updates are usually published much less frequently as new security vulnerabilities are uncovered, they leave the mobile device exposed for a longer period.

- **Anti-Virus Software**
  Comparable to the problem with firewalls, mainly limited system resources are responsible for the lack of anti-virus solutions for ultra-mobile devices.

The just described issues are mainly responsible for the distribution of botnets and need therefore be addresses by security responsibles and researchers. Remote network attacks can be carried-out fully automated and allow the infection of a large number of devices in a very short time.

13.2.1.2 Installed Applications

One of the major advantages of modern ultra-mobile devices is their ability to fast and conveniently install new software. This functionality is generally provided by some kind of online application store or market. Developers can distribute their software to numerous clients in a very easy way.

Unfortunately, this can be abused by attackers. Although, many of the application stores try to check their items for malicious code, it cannot be guaranteed that none such is being delivered to clients. This could serve as a very convenient way for attackers to spread their dangerous software to many devices.

Application store providers do have some organizational and technical measures to minimize the risk :

- **Organizational Measures:**
  - **General Guidelines**
    The store provider needs to define formal guidelines informing developers what kinds of products are not allowed because they might pose security risks (e.g. virtual machines).
  - **Predefined APIs**
    Developers must only use predefined and evaluated APIs. This allows the provider to control system calls and prevent the usage of unwanted actions.
  - **Scan for Malicious Code**
    The provider must assure that all applications are scanned for known malicious codes like viruses, worms and shellcodes.
- **Technical Measures:**
  - **Kill Switches**
    Some providers integrate a so called *kill switch* into their mobile operating systems, allowing them to deactivate a certain application on all relevant platforms and devices.
  - **Sandboxes**
    Sandboxes allow the execution of applications in an isolated environment. This measure prevents that applications access resources, like memory stacks, of the device which are not assigned to it.

13.2.2 Smart Phones as Hacking Tools

The last sections illustrated, that mobile devices which have been subject to external attacks are posing serious security risks. But these devices can also be used as tools to prepare sophisticated attacks on corporate network infrastructures. As state-of-the-art devices provide computational powers comparable to personal computers, and as their operating systems are closely related, it is possible to port existing hacking software onto these platforms.

Several very popular and powerful *security tools* like *aircrack* [4], *nmap* [134], *metasploit* [123], *nectat* [131] or *tcpdump* [173] are already available for ultra-mobile devices.

Even though some devices like Apple's iPhone ships with protection against installing software that is not approved by Apple, it is possible to circumvent this protection by a procedure called *jailbreaking.* This action provides root access to the core of the UNIX based operating system, leaving the user in complete control of the device.

These ultra-mobile hacking tools are very well suited for analyzing network infrastructures, called *footprinting.* Footprinting allows an attacker to obtain a detailed profile of security measures implemented in the target network.

Mobile probes as smart phones can be discretely placed in communication range or plugged into the network. The can collect information about connected devices, network segments, wireless access points, used communication protocols, provided services and implemented security measures.

Attackers often need this kind of information to be able to launch sophisticated attacks against the target infrastructure.

13.3 Conclusion

This chapter provided an overview on threats posed by the careless incorporation of ultra-mobile devices into corporate environments. The following two chapters will discuss a number of these risks by the example of Android's Application Market.

14
Android Market Analysis with Activation Patterns

As mobile operating systems start to spread from the classic smartphone platforms onto tablets and ultra mobile computers, they are experiencing a significant gain in market share. Consumer acceptance and therefore commercial success of a mobile operating system depends on several factors. Beside the quality and usability of the user interface, the availability of applications (apps) may be the most important feature demanded by the market. While traditional systems provided a preinstalled set of apps, modern solutions like Apple's iOS, Google's Android, RIM's Blackberry or Microsoft's Windows Phone 7 offer the possibility to access and install a wide variety of apps from different genres, ranging from games to powerful business appliances.

While e.g. Apple enforces tight policies on software distributed via their App Store for iOS, regarding security and content, Google emphasizes a more *open* philosophy, providing many liberties to Android developers, distributing their products via the Android Market.

Newly developed iOS apps are thoroughly examined by Apple engineers to keep the platform as secure as possible. This approach sometimes limits the developers access to hardware resources like GPS receivers, cell phone functionalities or integrated cameras. As recent events have shown, even this strict approach has loopholes and apps not in line with the policies can find their ways onto the customer's devices[1]. In order to confine the impact of such incidents, Apple supposedly implemented a *kill switch* to deactivate installed apps on all devices.

[1] http://news.cnet.com/8301-13579_3-10464021-37.html

Apps submitted to the Android Market are rudimentarily checked but the process is not as strict as it is for the App Store. Google seems to pursuit a *delete afterwards* strategy if apps have been found of low quality or malicious. Android implements a similar functionality to Apple's kill switch to remotely remove apps installed on customer devices[2].

Google introduced a fine grained permission system for their Android platform, allowing developers to precisely define the necessary resources and permissions for their products. The customer can decide during the installation whether she wants to grant or deny access to these requested resources such as the address book, the GPS subsystem or telephone functionalities[3].

This user centric process is sometimes challenging and inducing customers to accept whatever is requested by the app, opening potential loopholes for attackers. In order to gain a better understanding on how permissions are used throughout the Android Market, this paper presents our analysis of publicly available metadata of 130.211 apps in the Android Market. The extracted metadata is comprised of several features that are displayed when the user opens an app for installation. Among the security permissions, which where of primary interest, we have extracted the description, download count, price and the category of each app. For the analysis, we propose a sophisticated method based on *Activation Patterns* that allows us to answer a wide range of questions such as:

- *Extract all wall papers that have a non-typical combination of security permissions - meaning they are anomalies.*
- *Is the usage of security permissions different in free/payed apps?*
- *Which permissions are typical when the term "navigation" is used within the description?*
- *What are the most relevant features when analyzing popular security apps?*
- *Cluster popular apps according to their description or security permissions.*

The paper first describes the Android permission system and discusses related work, then describes the *Activation Pattern* concept and finally shows how it is applied to the metadata of 130.211 Android Market apps.

14.1 Android Permission Mechanism

Android's security architecture ensures the isolation of apps from each other as well as from the system. Communication and resource sharing are subject to well-defined access restrictions. Android apps are executed in their own Linux process with their own unique user- and group ID (UID), which allows for protection of file system resources. Access restrictions on specific resources and

[2] http://www.engadget.com/2010/06/25/google-flexes-biceps-flicks-android-remote-kill-switch-for-the/

[3] The user can accept either all permissions and install the app or reject all permission by not installing the app. Accepting or rejecting just a subset of these permissions is not possible.

functions are enforced via a fine-grained permission mechanism. Apps are allowed access to resources if they are granted the respective permissions by the user.

This isolation of apps, called sandboxing, is enforced by the kernel, not the Dalvik VM. Java as well as native apps run within a sandbox and are not allowed to access resources from other processes or execute operations that affect other apps.

Apps must declare required permissions for such resources within their app manifest file. These permissions are granted or denied by the user during the installation of the app. The user does not deny or grant permissions during the runtime of the app[4]. Permissions are enforced during the execution of the program when a resource or function is accessed, possibly producing an error if the app was not granted the respective permission. The Android system defines a set of permissions to access system resources such as for reading the GPS location, or for inter-app communication. Additionally, apps may define their own permissions that may be used by other apps.

14.2 Related Work

Toninelli et al. [181] have investigated current methods of specifying security policies for smartphones. From their assumption, that in mobile computing scenarios users will be required to manage the security on their own, they deduct that the foremost requirement must be to design a simple security model which allows mobile end users to understand their security decisions. Otherwise, this would lead users to define or accept security policies which they do not comply with or also to turn off troublesome security features. Thus, they introduced a semantic-based policy model solution as one step towards a usable security for smartphones. They assess the efficiency and practicality of their security model by applying it to typical security related use cases. They have first analyzed typical mobile use cases and derived critical requirements for the design of a usable access control model. Their proposed solution relies on the assumption that understandability of the policy model is a necessary condition for usability of the access control system.

Their approach relies on a semantic-based policy representation. Such a semantic-based approach improves the understanding of security policies, they argue, since users would be more aware of their implications. Although their work is not directly concerned with apps from the Android Market, their results can be applied to mobile computing and the smartphone community in particular. However, they also outline that the users tolerance to failure remains a crucial issue for a usable access control framework.

SMobile has done some research on the Android Market and its permission system. They have documented specific types of malicious apps and threats. In their latest paper [185] they have analyzed about 50,000 apps in the Android

[4] There is no dynamic permission granting as with the Blackberry system.

Market. They looked for apps which could be considered malicious or suspicious based on the requested permissions and some other attributes.

Their key findings are that a big number of apps, available from the market, are requesting permissions that have the potential of being misused to locate mobile devices, obtain arbitrary user-related data and putting the carrier networks or mobile device at risk. Although the Android Operating System and Android Market prompt users for permissions before the installation, users are usually not experienced in making decisions about the permissions they are allowing or more precisely what permissions an app should have. But most often users do not take the time or have not the proper knowledge of the security implications. The most important statement they make is that fundamental security concerns and increase in malicious apps can be related to poor decisions of the user. Their work was the most comprehensive security analysis of the Android Market to date. Their conclusion was that end-users need to make educated decisions regarding the apps they are installing and that third-party security technology could assist them in making better decisions. This was one of the motivations for us to make a more in-depth analysis and to provide an open-source framework for automated permission analysis.

14.3 Activation Patterns

The analysis of app permissions in the Android Market is based on the *Activation Patterns* framework that we have developed during the last three years. This framework has been applied successfully to a wide range of domains such as event correlation [177], text classification [178] or semantic web analysis [175]. The idea behind this technique is to transform a raw data vector containing arbitrary symbolic and real valued features into a pattern, which forms the basis for a wide range of subsequent analyses. This transformation process is depicted in Figure 14.1 which shows several processing layers that:

1. extract features and feature values from instances[5] and store the information as nodes in a semantic network [148],
2. represent relations between these feature values and the strength of these relations (e.g. defined by the number of co-occurences within a data-vector) as weighted links within this network,
3. apply spreading activation techniques [33] for each instance, which stimulate the network and spread the activation of selected nodes according to their links to other regions of the network,
4. and finally extract the activation values for each instance from the network and store them within a vector that we call the *Activation Pattern.*

[5] An instance is a data vector containing all feature values describing the instance. For the Android market, an app instance would be described by various features such as permissions, description terms or download count that have different feature values (e.g. different permissions).

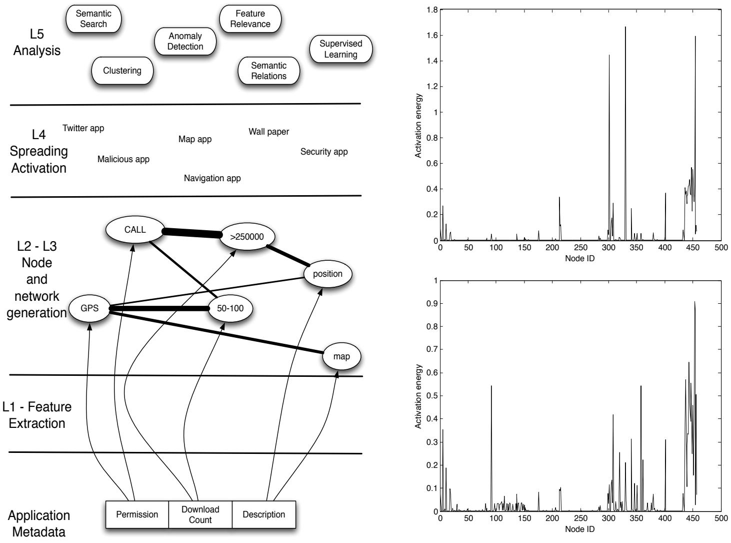


Figure 14.1: Activation Patterns and Analysis

The generated patterns represent the activation values of different regions within the network that are activated due to different input stimuli (e.g. the feature values of an app). The similarity between two patterns can be calculated by distance measures (e.g. the cosine similarity) and expressed as a simple distance value. This allows us to apply a wide range of standard machine learning algorithms and thereby cover various analyses with a single model:

Semantic *search*: The distance between the *Activation Patterns* can be used to implement semantic search algorithms that retrieve semantically related instances. These search queries can also be used to specify certain feature values and find closely related patterns. *Example: Retrieve all description terms and permissions that are semantically related to the GPS permission.*

Feature *relation*: The semantic network describes arbitrary relations between feature values. By activating one or more nodes (corresponding to feature values) within the semantic network, and spreading their activation via the links to the neighbors, we are able to extract details about the relations between various feature values and the strength of these relations. *Example: Which security permissions are strongly related to the term "GPS"?*

Feature *relevance*: The relations within the semantic network are created according to the co-occurrence of feature values within the analyzed data set. The strength of these relations are represented by the associated weights within the network. Given a feature value that is represented by a node and the number of emerging/incoming links and their weights, we are able to deduce the importance of the information carried by the node. Nodes that are connected to a large number of other nodes typically do not add information for subsequent analysis

processes. *Example: How relevant is the security permission for accessing the GPS when analyzing wall papers?*
***Anomaly* detection:** The sum of all activation values within a pattern represents the activation energy of the whole pattern, which is a measurement for the response of the network. Anomalies can be detected in two ways: First, if features are combined in a non-typical way, the activation energy is lower than for normal combinations. Second, a large number of inputs (e.g. excessive usage of permissions) causes more activations and thereby higher total activation energies. The first anomaly detection method is more suitable when the number of feature values for each instance is constant. For the market analysis we concentrate on the second method, since the number of features varies from app to app. *Example: Find all wall paper apps that use non-typical permissions.*
***Typical instances*:** An *Activation Pattern* is characterized by the activation values for the different feature values. By inspecting these values we can determine the strength of the activation and thereby the significance of the feature values. By combining several *Activation Patterns* with simple operations (e.g. mean, variance etc.) we gain knowledge about complete sets of patterns. *Example: What are the typical security permissions of free GPS apps?*
Unsupervised *clustering*: Due to the transformation into *Activation Patterns* we can directly apply unsupervised clustering algorithms without the need to apply normalization and discretization strategies to the raw feature values. For this work we apply a simple Growing Neural Gas (GNG) algorithm [53] that is extended by the Minimum Description Length (MDL) criterion as used by the Robust Growing Neural Gas (RGNG) algorithm [147]. This extension enables the algorithm to automatically detect the necessary model complexity, without the requirement to specify the number of clusters in advance[6]. *Example: Cluster all apps with a price larger that 10 Euros according to their description and employed security permissions.*

For a more detailed description of the *Activation Patterns* technique we refer to Section 8.3.

14.4 Analyzing the Android Market with *Activation Patterns*

For our analyses we have extracted the metadata of 130.211 apps in December 2010[7]. All subsequent analyses have been conducted according to the following steps:

1. Apply an arbitrary filter to the app database (e.g. apps with certain

[6] The Activation Patterns concept is not limited to the NG algorithm family – an arbitrary unsupervised algorithm can be applied to the patterns. Obviously, one could also apply supervised algorithms to the patterns for training a classifier, which is not shown in this paper.

[7] The app identifiers where collected from various web sites and the metadata itself was extracted from the Android Market via the android-market-api: http://code.google.com/p/android-market-api/.

permission, with a given download count, price or apps that are described with certain keywords).

2. Extract the features and their values from the remaining apps and apply the *Activation Pattern* transformation process.
3. Use the generated patterns for the analyses described in the previous section.

We have found several apps that have either a uncommon combination of permissions or make excessive use of permissions, however we must emphasize that having such permissions does not necessarily mean that the app abuses these permissions. Such abuses cannot be detected by the conducted analyses, but only by inspecting the code of the apps.

Due to space constraints we only highlight some prominent examples that demonstrate the capabilities of the *Activation Patterns* concept and refer the reader to our website where the complete analyses can be downloaded[8].

Q1: Retrieve apps that use the terms "hot" and "girl" in their description and find permission anomalies[9] ***(anomaly)****:* In contrast to the Apple App Store the Android Market allows apps with mature content. Similar to PC software or websites offering such content, we assume that such apps might be infected with trojans, spyware or are built deliberately in order to extract private information from the user.

By applying the ***anomaly*** analysis to the patterns generated for the filtered apps, we are able to gain the following information: A large number of these apps just come with pictures that are displayed within the app or can be set as a wall paper. Some require a connection to the internet in order to grab pictures from web sites. The normal behavior and therefore the typical activation energy within a pattern is defined by this majority of apps. Anomalies deviate from this typical energy and are highlighted by two examples:

The first one is based on a group of apps that describe themselves as apps that "change the picture whenever the user receives an SMS with certain keywords". This description would suggest that the apps are required to have some permissions related to receiving and reading an SMS only, however they make excessive use of permissions related to writing SMSs, reading the contact data, accessing the internet, determining the user's position, using Google auth and many other additional permissions.

The second anomaly refers to an app that includes "hot puzzles and videos". However the app has access to the camera, is allowed to record audio, read and write contact data and has access to the GPS.

Q2: Is there a difference in the typical permissions when comparing free and payed apps with the terms "hot" and "girl" in their description[10] ***(typical in-***

[8] http://www.carbonblade.at/wordpress/research/android-market/

[9] Hot Girls ALL Without Description.txt

[10] Hot Girls FREE Without Description.txt and Hot Girls PAYED Without Description.txt

stances)? In the second example we assume that free apps would be a better target for capturing private information, since they typically have a larger user base. After applying the filters we get the following results for the most active permissions, where the value within the parentheses represents the activation value. A higher value indicates a stronger activation within the network and therefore a higher significance of the corresponding feature value:

- **Payed (644 apps)**: *internet* (0.74), *set wallpaper* (0.53), *get tasks* (0.45), *write external storage* (0.45), *get receive completed*[11] (0.42), *access network state* (0.28), *wake lock* (0.23),

- **Free (534 apps)**: *internet* (2.95), *set wallpaper* (1.44), *read phone state* (1.19), *access network state* (1.15), *write external storage* (1.06), *access coarse location* (0.95), *access fine location* (0.64), *send sms* (0.46), *read contacts* (0.45), *wake lock* (0.42), *receive boot completed* (0.40), *receive sms* (0.38), *read sms* (0.37), *write sms* (0.37)

These results indicate a gap between free and payed apps with mature content. There might be several reasons for this difference: At first if these apps really include code that capture your private information than it would make more sense to include such code in free apps since they have a larger user base. The second explanation might be found in the light of various ad clients. Developers often deploy free apps and generate revenue by using ad clients that display advertisements within the app. These ad clients tend to accumulate personal data and often need access to several permissions (see Q9 for more details). The third explanation might be that the developers simply added too many permissions and forgot to remove them. However, this does not seem likely since it would not explain the gap between free and payed apps. To determine what these apps really do, we need to go deeper, decompile these apps and inspect the code and all the calls made to Android APIs.

Q3: Extract popular (more than 5000-10000 downloads) apps with access fine or coarse location permissions and find permissions and terms that occur in the same semantical context as the term "GPS"[12] ***(search):*** In this example we demonstrate the semantic search capability of the *Activation Patterns* concept. After applying the filter, we get 3204 apps with 5522 terms used for the description. We now execute a semantic search query for the term "GPS" yielding the following results, separated into terms and permissions:

- **Terms**: location, altitude, accuracy, coordinate, track, strength, position, sensor, range, program, technology, compass, satellite, longitude

[11] This permission does not exist in the Android system. It might be a spelling mistake and therefore useless or a self defined permission that is used by multiple apps accessing each other. 88 of the 644 payed apps employ this permission.

[12] LOCATION - GPS With Description.txt

- **Permissions**: *access fine/coarse location, internet, control location updates, access mock location, wake lock*, followed by permissions for accessing the camera, calling phones, receiving SMS etc.

The related terms are pretty obvious and show that the semantic search queries retrieve significant results. For the permissions it seems that most of the apps need to have access to the internet, prevent the phone from sleeping or dimming the screen (*wake lock*), or simulate a location update (*access mock location*). The last permission is needed especially during development in order to get a position in the emulator and probably was not removed after app deployment.

An example for retrieving semantically related instances would be a query that retrieves apps which do not contain the term "gps" but are semantically related to apps that have the term "gps" in their description (e.g. due to a description that contains "position" or "location"). Other interesting examples are the terms "wife" and "husband" which are closely related to the term "position". This semantic relation is created by apps that are used to spy on someone's wife/husband by tracking her/his location.

*Q4: How relevant are the feature values of the apps retrieved in Q3 "GPS" **(relevance)**?*

- **Most relevant feature values**: The most relevant feature values are those that occur rarely. In case of the permissions the most relevant ones are either permissions for special purposes, permissions with a wrong spelling, or self-defined permissions.

- **Least relevant feature values**: These are values that firstly occur within most of the analyzed apps and secondly that are randomly connected to other feature values. An example is the *internet* permission that is required by a large percentage of apps and is not correlated to other feature values – meaning it co-occurs randomly with those values.

Q5: How is the term "navigation" related to security permissions[13] ***(relation)**?*
In this example we extract the semantic network links to security permission nodes emanating from the node "navigation" and use their weight to determine the strength of these relations: The following permissions are sorted according to the strength of their relation with "navigation": *access coarse/fine location, internet, read contacts, read phone state, write external storage, wake lock, access network state, call phone.* The contact and phone related permissions are typical required in order to control incoming calls/messages from the navigation app. The "navigation" term is also strongly related to the category "travel", high download counts, and GPS relevant terms similar to those of Q4.

[13] LOCATION - NAVIGATION With Description.txt

Q6: Filter free apps that use the terms "wall" and "paper" in the description and have the permission set wallpaper. Find permission anomalies[14] ***(anomaly):*** The biggest anomalies are caused by apps that replace the standard launcher of Android. Due to their functionality such apps require a large number of permissions and are therefore considered as anomaly. However, they are followed by several other interesting examples: One of these apps is called *FoxSaver* and describes itself as an app that allows you to browse photos from the website *foxsaver.com* and install them as wall paper. However, the app also has the *receive/read SMS, read contacts, access fine/coarse location* and *call phone* permissions. These additional permissions do not make sense when reading the description of the app.

Q7: Filter apps that have the permissions for reading, receiving and sending SMS messages but do not contain terms related to these permissions in their description (e.g. "sms", "message" etc.)[15] ***(anomaly):*** The biggest anomalies are caused by apps related to Android development, security and backup. For most of these apps it makes sense to use the permissions, however there are other apps where the description does not match the required permissions: a large number of themes that are just described as "theme" with a certain keyword and several other apps do not state anything about messaging within their description (including games, fitness apps, travel guides etc.).

Q8: Cluster popular apps (more than 250.000 downloads) according to their description and security permissions[16] ***(clustering):*** After applying the filter, 1079 apps remain that are grouped in the following categories (clusters):

- **7 description clusters C1 (55)** ringtone apps; **C2 (64)** apps with German description[17]; **C3 (116)** social network apps; **C4 (339)** tools, widgets, games, browsers; **C5 (123)** translation, reference, books; **C6 (46)** music players, streaming; **C7 (336)** games.

- **8 permission clusters C1 (326)** games with a few permissions (*internet, access network state, read phone state)*; **C2 (31)** apps that have access to account data, contacts (e.g. social network related); **C3 (217)** mostly apps with internet access only (reference, games); **C4 (160)** also social network related, but with a bias to location related permissions; **C5 (99)** music and video apps, various permissions, but a strong activation of the *wake lock* permission, which is required to prevent the phone from sleeping; **C6 (59)** mostly ringtones apps with a strong activation of the *read phone state* permission. This permission is required by an app for recognizing

[14] Wallpaper Without Description.txt

[15] SMS PERMS Without Description.txt

[16] DOWNLOAD COUNT - Only Description.txt and
DOWNLOAD COUNT - Without Description.txt

[17] The distance between the German descriptions and the English ones is so large, that the 64 apps are only represented by one cluster. This could be changed by a more complex model.

that the phone is ringing; **C7 (126)** various apps that require a mixture of different permissions; **C8 (61)** phone and SMS related apps indicated by related permissions.

Especially, the permission cluster results are quite promising, since they allow us to gain knowledge on how permissions are typical used by various application categories and find outliers within a given category.

Q9: Identification and tracking of users: In order read out the unique ID of your smartphone, SIM ID, telephone number or cell ID, the permission *read phone state* is required[18]. In combination with the *internet* permission and possible the location related permissions[19], this enables an app to transmit information which allows user identification and tracking. 31865 of 130211 apps have these two permissions. For the 1079 apps that have more than 250.000 downloads, 362 have these two permissions. This corresponds to the results presented in an analysis of 101 popular apps, which focuses on private data sent to various companies (mostly related to advertisements)[20].

14.5 Conclusions and Outlook

In this paper we have applied the *Activation Patterns* concept to the Android Market apps. This new technique allows us to extract detailed knowledge about the apps and relations between the security permissions, description terms, download counts etc. Since, the Android Market share and therefore the number of apps are growing steadily we argue that the Android platform is an obvious target for malicious activities[21]. For this reason we deem it necessary to get a better understanding of the available apps, their employed security permissions and existing anomalies.

[18] We refer to http://developer.android.com/reference/android/telephony/TelephonyManager.html for a detailed list of extractable information.

[19] The *read phone state* permission grants access to your current cell ID, which could already be used to determine the user's position if an appropriate database containing cell tower locations is available: e.g. http://www.skyhookwireless.com/. Therefore, the location can also be determined without the *fine and coarse location* permissions.

[20] http://online.wsj.com/article/SB10001424052748704694004576020083703574602.html

[21] ...and also legal apps that identify and track you due to advertisements.

15
Android Security Permissions – Can we trust them?

The opening of mobile device platforms to third party developers was probably the most significant move of the IT industry in the last years. The availability of a multitude of applications (apps) has boosted user acceptance and usefulness of mobile devices like smartphones and tablet computers. Regardless whether for business or private use, these devices and their apps have the potential to facilitate and enrich the user's everyday life. Mobile platform vendors recognized the importance of opening their systems to third party developers in order to attract a wide range of apps. However, the large number of third party developers make it very hard to provide uniform quality standards for the repository providers.

While e.g. Apple enforces tight policies on software distributed via their AppStore for iOS, regarding security and content, Google emphasizes a more *open* philosophy, providing many liberties to Android developers, distributing their products via the Android Market. Apps submitted to the Android Market are rudimentarily checked but the process is not as strict as it is for the AppStore. Google seems to pursuit a *delete afterwards* strategy if apps have been found, which are malicious or are of low quality. Android implements a kill switch to remotely remove apps installed on customer devices[1].

Google introduced a permission system for their Android platform, allowing developers to define the necessary resources and permissions for their products. The customer can decide during the installation whether she wants to grant or deny access to these requested resources such as the address book, the GPS

[1] http://www.engadget.com/2010/06/25/google-flexes-biceps-flicks-android-remote-kill-switch-for-the/

subsystem or the phone functionalities. Although, this process is challenging to the standard user, at least an expert will have the ability to draw conclusions about the theoretical capabilities of an app based on its permissions.

Thereby, the security of the permission system is mainly based on two different aspects: the meaning of the permission itself and even more important, the meaning of combined permissions. For example, when the internet permission is combined with the read contacts permission, a possible malicious app could transfer your private contact data to the internet. This implication and the functionality is lost when both permissions are not used in the same application. Therefore, a large part of the security of the permission system and the trust in it is based on the assumption that an app only gains access to the resources that are declared via permissions.

In this work we give a detailed description of the possible communication paths between applications. We discuss the issue of what we call *spreading of permissions* which exploits interprocess communication to allow a transfer[2] of security permissions to apps which did not request them at installation time. We substantiate this threat by presenting three prototype apps that highlight the permission spreading problem and demonstrate the detection of a Service based communication path[3].

15.1 Related Work

In the article *Understanding Android Security* [44] Enck et al. took a look at the Android application framework and the associated features of the security system. One pitfall of Android, as the authors describe, is that it does not provide information flow guarantees. Also the possibilities of defining access policies in the source code introduces problems because it clouds the app security since the manifest file does not provide a comprehensive view of the application's security anymore.

SMobile has done some research on the Android Market and its permission system. They have documented specific types of malicious apps and threats. In their latest paper [185] they have analyzed about 50,000 apps in the Android Market. They looked for apps which could be considered malicious or suspicious based on the requested permissions and some other attributes.

Their key findings are that a big number of apps, available from the market, are requesting permissions that have the potential of being misused to locate mobile devices, obtain arbitrary user-related data and putting the carrier networks or mobile device at risk. Although the Android OS and Android Market prompt users for permissions before the installation, users are usually not ready to make decisions about the permissions they are granting. The most important statement they make is that fundamental security concerns as well as increase

[2] This is not an actual transfer of the permission, but has the same effect.

[3] These apps can be downloaded from:
http://www.carbonblade.at/wordpress/research/android-market/

in malicious apps can be related to poor decisions of the user. Toninelli et al. came to the same conclusion in [181].

Nauman et al. [130] investigated the Android permission system with special focus on introducing more fine-grained permission assignment mechanisms. The authors argue that the current permission system is to static since it does not take into account runtime constraints and that the accept all or none strategy is not adequate. Thus, they propose Apex, an extension to the Android policy enforcement framework which allows users to grant permissions more selectively as well as to impose constraints on the usage of resources at runtime. Their extension provides some additional security features which allows a more fine-grained security policy for Android. A similar approach is outlined in the work of Ongtang et al. [135].

Similar to [44], Shabtai et al. [161] performed a security assessment of the Android framework in the light of emerging threats to smartphones. They made a qualitative risk analysis, identified and prioritized the threats to which and Android device might be exposed. In addition, they outlined the five most important threat categories which should be countered by employing proper security solutions. They provide a listing with adequate countermeasures and existing solutions for the specific threat categories. One of the main propositions is that the permission system should be hardened to protect the platform better from misuse of granted permissions.

15.2 Interprocess Communication

Android apps may activate components of any other app if the other app allows for it. The four different types of components providing entry points for other apps are Activities[4], Services, Broadcast Receivers and Content Providers.

If a certain component is requested, the Android system checks whether the corresponding app process is running and the component is instantiated. If either of both is not available, it is created by the system. Thus, if a requesting app is allowed to access a background Service provided by another application, it can access it at any time, without the Service having to be started by the user.

15.2.1 RPC Communication

The Android system provides means for interprocess communication (IPC) via remote procedure calls (RPC). Since different processes are not allowed to access each other's address space, methods on remote objects are called on proxy objects provided by the system. These proxy objects decompose (marshal) the passed arguments and hand them over to the remote process. The method call is then executed within the remote app component and the result marshaled and returned back to the calling process. The app programmer merely defines

[4] For all subsequent Android specific terms we refer to the Android developer documentation located at http://developer.android.com/index.html

and implements the interface. The entire RPC functionality is generated by the system based on the defined interface and transparent to the application.

Interfaces for interprocess communication are defined using the Android interface definition language (AIDL). The resulting Java interface contains two inner classes, for the local and remote part, respectively.

Typically, the remote part is implemented within an app component called *Service*, allowing clients to *bind* to it in order to receive the proxy object for communication with the remote part. The Service returns the Stub class in its `onBind()` method called by the system upon a connection request from the client. The client, on the other hand provides a `ServiceConnection` callback object along with the bind request in order to receive the proxy object for interprocess communication with the Stub.

Specific messages, called *Intents*, identifying the targeted Service, represent bind requests. A client's Intent is passed to the Service's `onBind()`, so that the Service can decide whether or not to accept the connection. Upon a successful connection establishment, the system passes the proxy object corresponding to the Stub returned from `onBind()` to the client's `ServiceConnection` callback.

Services may declare required *permissions*[5] that are enforced during the binding. Applications binding to the Service have to declare the use of these permissions correspondingly.

15.2.2 Communication via Intents

Intents are logical descriptions of operations to perform and are used to activate Activities, Broadcast Receivers and Services. Since these components may be part of different applications, Intents are designed to cross process boundaries and may transmit information between applications. Note that Intents are mainly intended to identify a component, optionally adding a limited amount of additional information to more precisely specify the targeted operation.

Neither Activities, nor Broadcast Receivers or (unbound) Services provide persistent RPC connections for interprocess communication. Still, they may all be activated by Intents which will be passed to their respective activation methods `startActivity()`, `startService()`, `sendBroadcast()` and others. Via their `extras` attribute, Intents therefore provide a simple means for transmitting a limited amount of data to another process' component. Additionally, Activity components provide a way to return a result back to their caller. If launched via `startActivityForResult()` the calling process may receive a result Intent via its `onResult()` method, thus allowing for simple two-way communication.

15.2.3 Alternative Communication Paths

Content Providers are intended to share data between applications. They are uniquely identified by URIs and can be accessed via `ContentResolver` objects

[5] see Section 15.3

provided by the system. Note that Content Providers are not activated by Intents. Reading and writing to Content Providers allows for two-way interprocess communication if the participating processes have the required permissions.

Apart from the described methods, apps may also communicate by exposing data via the filesystem and setting global (world) read/write permissions on the files. Alternatively, apps may share resources if they request the same UID. In that case they are treated as being the same app with the same file system permissions. UID sharing is possible for apps signed by the same developer.

We consider these communication approaches as *side-channel* communication.

15.2.4 Information Flow Overview

Based on the IPC methods described above, there are four possible communication paths between two apps A and B (see Figure 15.1). The S block depicts any of the aforementioned interprocess interfaces that provides or receives information. The arrows indicate the information flow between an app and the remote component.

- **One-Way**: An app A can either transfer/receive information to/from an app B, by using a one-way communication method. This could also be described as pushing or pulling information to/from an application.
- **Real Two-Way**: App A exchanges information with app B, by using two-way communication. Thereby both apps can receive and transmit information from A to B and vice versa. This involves an RPC interface as provided by Services that a client can bind to.
- **Pseudo Two-Way**: In this variant, two one-way communication channels are combined to create a two-way communication channel. In this example, app A transmits information to app B via interface S2 provided by B. In addition, app B pushes information to A by calling its S1 interface. For this example two push channels were used, however an arbitrary combination of push and pull channels could be used. On an abstract information flow level, this method is equal to the real two-way communication method (regardless of the employed push/pull combination).

Especially the pseudo two-way method and one-way push method can be used to transmit information over side channels (e.g. communication by reading and writing to logging facilities). The available communication paths influence how potential malicious apps can avoid detection and where they locate the actual malicious code.

15.3 Android Permission Mechanism

Application isolation, distinct UIDs for all apps and permissions are the three building blocks of Android's security architecture. Isolation of apps from each

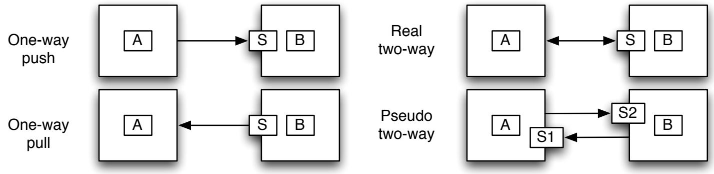


Figure 15.1: Two apps A and B can exchange information via one-way or two-way communication.

other as well as from the system is assured by executing every app within its own Linux process. Further, every app runs with a distinct user- and group ID (UID), assigned at app installation. This allows for protection of memory and file system resources. Communication and resource sharing are subject to access restrictions enforced via a fine-grained permission mechanism. Applications are allowed access to resources if they are granted the respective permissions by the user.

The isolation of applications, called *sandboxing*, is enforced by the kernel, not the Dalvik VM. Java as well as native apps run within a sandbox and are not allowed to access resources from other processes or execute operations that affect other apps.

Applications must declare required permissions for such resources within their manifest file. These permissions are granted or denied by the user during the installation of the application. The user does not deny or grant permissions during the runtime of the application[6]. Permissions are enforced during the execution of the program when a resource or function is accessed, possibly producing an error if the app was not granted the respective permission. The Android system defines a set of permissions to access system resources such as for reading the GPS location, or for inter-application/process communication. Additionally, apps may define their own permissions that may be used by other apps.

There is no central point for permission enforcement, it is scattered over many parts of the Android system. At the highest level, if permissions are declared in an application's manifest file for a component, they are enforced at access points to that component. These are calls to `startActivity()` or `bindService()` for activities or services, respectively, that would cause security exceptions to be thrown if the caller is not granted the required permission.

Permissions may control the delivery of broadcast messages by restricting who may send broadcasts to a receiver or which receivers may get the broadcast. In the first case a permission for the protected receiver is declared in the manifest file (or when registering the receiver programmatically, respectively). It gets enforced *after* a sender's call to `sendBroadcast()` and will not cause an exception to be thrown. Rather, the message will simply not be delivered to that

[6] There is no dynamic permission granting as with the Blackberry system.

receiver if the sender does not have the required permission. A sender, on the other hand, may declare a permission within the `sendBroadcast()` call, which will also be enforced without the sender noticing.

Permissions for granting read or write access to Content Providers are declared within the manifest file. Apart from that, content providers allow for a finer grained access control mechanism, via URI permissions. They control access to subsets of the content provider's data, allowing a content provider's direct clients to temporarily pass on specific data elements (identified by a URI) to other applications. A dedicated flag on an Intent[7] indicates that the recipient of the Intent is granted permission to the URI in the Intent's data field, identifying a specific resource, such as a single address book entry. The granted URI permission is finally enforced once the recipient of the Intent queries the content provider holding the URI by calling on a `ContentResolver` object[8]. Content Providers declare support for URI permissions in the manifest file. Enforcement of URI permissions results in security exceptions being thrown if the caller does not have the required permissions.

Applications may at any time query their context whether a calling PID or package (name) is granted a permission. This allows for custom-tailored permission enforcement for specific app requirements. Certain system permissions are mapped to Linux groups. On app installation, the application's UID is added to the respective group (GID). Permission enforcement involves GID checks on the underlying OS level. The permission to GID mapping is declared within the system's `platform.xml` file. Another specificity are *protected broadcasts*, which only the system may initiate.

15.4 Permission Spreading

As the user grants permissions on installation of an application, it is crucial to consider all permissions an app requests *in context*. The combination of specific permissions may indicate security flaws to the user. Within this work, we consider the user to be capable of critically analyzing an application's declared permissions and understand the implications of granting permissions.

Therefore, the permission system and the decision of the user is based on the assumption that an app can only use the functionality for which the appropriate permissions are available. We argue, that this security function can be circumvented by spreading security permissions over two or more apps that use interprocess communication. Thereby the apps are able to gain additional functionality for which they do not have the corresponding permissions.

15.4.1 Demonstration

In this section we describe two demo apps that hide the transmission of private location data to the internet by employing permission spreading via an imple-

[7] Intents are the entities used to activate app components, cf. Section 15.2.2

[8] Content Providers are not activated via Intents.

mented Service (see Figure 15.2). The app *Backdoor* requires the *access fine location* (GPS) permission, which could be justified by posing as GPS app that displays the current position. However, not detectable by the user the app also implements an Android Service that provides the GPS position to other apps. The second app – *TwitterApp* – has the *internet* permission and could pose as a simple app for accessing Twitter, which again would not raise any suspicion during its installation. However, the user is not able to see that *TwitterApp* has malicious code that determines the current GPS position by calling the Service of app *Backdoor*. This GPS position is then posted to a Twitter account[9], which requires the *internet* permission. Therefore, the app *TwitterApp* gains additional capabilities by calling a Service of another app and uses its own *internet* permission to publish this information. Although the permission system is not directly circumvented (the app is still not able to get the GPS position without the Service of the second app), there are serious implications when analyzing this method in the context with malicious applications.

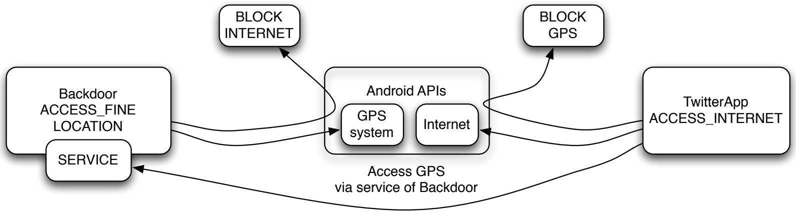

BLOCK INTERNET
BLOCK GPS
Backdoor ACCESS_FINE LOCATION
SERVICE
Android APIs
GPS system
Internet
TwitterApp ACCESS_INTERNET
Access GPS via service of Backdoor

Figure 15.2: The app *TwitterApp* uses the Service of *Backdoor* to determine the users's GPS position and submits it to Twitter via its own *internet* permission.

15.4.2 Implications for malware

When taking a closer look at these two demo apps and the permission spreading method, we come to several conclusions:

15.4.2.1 Losing the permission context in the Android Market?

The Android Market permission system is intended to support the user in her decision whether to trust an app before its installation.

Thereby, primarily the context in which multiple permissions are used and not only the permissions themselves, will alarm a user when inspecting a possible malicious application. However, exactly this context is lost in our presented attack, since permissions can be distributed over different apps and do not occur within the same context.

[9] http://twitter.com/demolocator

15.4.2.2 Trust in the permission system?

The Android permission system conveys a level of trust when an app is installed, since system functionality can only be accessed when the appropriate permissions are available. However, when employing permission spreading, this trust leads to a wrong sense of security since in this case even an app without any permissions at all can gain additional functionality by calling functions within other apps that have these permissions.

15.4.2.3 Where is the malicious code?

Assuming, we have a malicious app that transmits private information information to an attacker without using permission spreading, then, firstly this app must declare all the required permissions (e.g. read contact data and internet access) and secondly it must contain the malicious code that reads the private data and transmits it to the internet. Such an app might raise suspicions due to the employed permissions and the lack of an adequate description or app use case that would necessitate these permissions. When analyzed thoroughly, the malicious activity could be detected by decompiling the application, capturing network traffic or employing other methods for detecting malware.

However, when employing permission spreading it is not necessary that the malicious code is contained within the app that has the permissions required for the malicious activity. For instance, app *Backdoor* only contains a Service that provides the GPS position, but not the malicious code that transmits this data to an attacker. Therefore, an arbitrary malware detection/analysis method would never detect the malicious activity when inspecting *Backdoor* only. In fact, *TwitterApp* carries the malicious code and uses *Backdoors's* permissions and Services to gain the information required for the malicious activity.

Furthermore, the app *TwitterApp* could come without any permission at all and just acquire the functionality through calling Services on multiple other apps (e.g. providing contact data, GPS position, internet access).

15.4.2.4 Backdoors?

Regardless of the malicious code's location and how the various available communication paths are employed, permission spreading still requires the installation of multiple apps by the smartphone users. At a first glance this might make the likelihood of a successful attack smaller. However, permission spreading could also be viewed as a classic backdoor that could either be injected or integrated on purpose into a popular application:

- Such a backdoor Service could be injected into existing source code that is not protected adequately. Since, the malicious code is not present but only the code required for providing certain information or functionalities, it might be difficult to detect such a backdoor – especially when communication side channels are employed (e.g. by writing data to a system log).

- The backdoor could be injected on purpose into a popular app by the company developing the app itself, by a developer that is involved in the development of an popular app or by a government[10].
- The backdoors could be integrated into multiple apps created by the same developer who then convinces the user to install more than one of the apps (e.g. by promoting add-ons, splitting functionality, additional levels for a game, by using a common API for multiple apps, by providing demo/full versions of applications, etc.).

15.5 Countermeasures

During the analysis of the permission spreading problem, we have also investigated countermeasures and implemented another demonstration app that focuses on the detection of a communication path between permission spreading apps. Concerning detection, we need to deal with the following question: *How can we detect malware that employs permission spreading?* The answer strongly depends on the employed communication method. We will give a short overview about the possible detection methods in the following sections.

15.5.1 Service detection

Android Services are the simplest method for establishing a two-way communication path in Android. Services must be declared within the app manifest and get an identifier that is used when calling the Service. The detection therefore can be categorized into the detection of a service and the detection of a call to this service:

Detection of a Service

- **Android Market - by the user**: The Android market does not state which Services are provided within the application. Therefore, the user is not able to get information about the Services employed by an application.
- **Android Market - by Google**: Since the manifest of each app is readable, Google would be able to gain information about the employed Services within all market applications.
- **Android smartphone - by the user**: The user is able to get information about running Services from the Android system. However, non-running Services are not displayed by the standard apps bundled with the Android system.
- **Android smartphone - by an app (e.g. a virus scanner)**: An arbitrary app without any permissions can query the Android `PackageManager`

[10] The possible attempt to catch Facebook account data in Tunisia is a good example for such an attack: http://www.wired.com/threatlevel/2011/01/tunisia/

for installed applications. For each of these installed apps it is possible to list the declared Services. In addition the ActivityManager can be used to list the running Services.

Detection of Service calls

Services are called by using Intents as parameters for the startService() or bindService() methods. The bindService() method enables the calling app to maintain a communication channel that is used for information exchange. The direct detection of such an Intent, the activation of a Service, or an established communication channel would require direct access to the Android system, which to our current knowledge is not possible without adding appropriate functions to the Android source code. At least for the bindService() method we have discovered a simple method, that allows us to determine when a Service is called and limit the possible apps that issued this call. We have also created a simple demo app (*ServiceBindDetection*) that can be installed as background Service and notifies the user whenever a Service is called by another application:

- The Android system Service allows an app to retrieve a list of all running Services (extracted via the AcitivityManager). In addition, for each Service the number of connected clients can be extracted. A client is connected when a *ServiceConnection* is maintained between a Service and the app that calls this Service.
- The detection app runs a loop in the background that in each iteration stores the running Services and their client count. Whenever the client count changes the user is notified within the Android notification bar.
- Whenever such a change occurs the detector could also get a list of currently running tasks (also via the ActivityManager) and thereby limit the possible callers[11]. By observing different calls to the same Service over time the possible perpetrators could be narrowed down.

We emphasize that this method does not work when startService() is used, since it does not maintain a communication channel and therefore does not list the calling app as connected client. Furthermore, we might miss certain calls when the duration of a bindService() ServiceConnection is smaller than the idle time of the Service checker loop.

15.5.2 Detection of Alternative Communication Paths

As described in Section 15.2, numerous ways for communication between applications exist. Acquiring information on interprocess communication other than via binding to Services turns out to be difficult. Information about Intents being sent would be valuable to detection of permission spreading. However, there does not seem to be a user-mode facility to obtain such information. Ongoing research will focus in this area.

[11] The most probable caller is the app that is currently running, however it cannot be ruled out that another app running in the background issues the call.

Detection of communication via side channels like reading and writing to Content Providers seems to be even more difficult. Only in-depth analysis of the involved applications might yield satisfactory results.

15.6 Conclusions and Outlook

The security of the Android permission system and the trust placed into the system is based on the assumptions that an app only has access to the functionality defined by the stated permissions and that all employed permission are displayed to the user within the same context (the app to be installed). As we show, these assumptions are not valid, since permissions can be spread over multiple apps that use arbitrary communication paths to gain functionality for which they do not have the appropriate permissions.

The intention of this work is to highlight the possible dangers and the wrong sense of security when trusting the permission system. Thereby, possible countermeasure range

- from making changes to the permission system including requiring permissions when using IPC between applications, or displaying communication interfaces prior to app installation,
- over implementing automatic detection systems within the Android Market, or performing an in-depth analysis of APK files,
- to shift the detection to the Android smartphone, by detecting communication events caused by permission spreading.

Addressing the detection on smartphones, we have presented a method to detect a covert communication channel involving Services. However, further investigations are necessary, since there is a large number of possible communication channels, ranging from documented IPC to not so obvious side channels.

Part IV

Appendices

A

Overview of Available Wireless Network Standards

Name	**Type**	**Data Rate**	**Frequency**	**Description**
802.11	WLAN	2 Mbit/s	2.4 GHz	First IEEE WLAN standard release in 1997.
802.11a	WLAN	54 Mbit/s	5 GHz	Speed up to 54 Mbit/s, 12 non-overlapping channels, OFDM modulation.
802.11b	WLAN	11 Mbit/s	2.4 GHz	Speed up to 11 Mbit/s, 3 non-overlapping channels.
802.11b+	WLAN	22 Mbit/s	2.4 GHz	Speed up to 22 Mbit/s, PBCC modulation, based on TI-ACX100 Chipset.
802.11c	WLAN			Inter access-point communication.
802.11d	WLAN			Adaptation for regional regulations.
802.11e	WLAN			Enhancements for the MAC. QoS added.
802.11f	WLAN			Roaming between APs form different manufacturers.
802.11g	WLAN	54 Mbit/s	2.4 GHz	54 Mbit/s in the 2.4 GHz band. Compatible to 802.11b.
802.11h	WLAN	54 Mbit/s	5 GHz	802.11a modifications for compliance to european regulations.
802.11i	WLAN			Enhanced Security, integration of AES and 802.1X (WPA2).
802.11j	WLAN		4.9-5 GHz	Japanese version of 802.11a.
802.11k	WLAN			PHY layer enhancements, support for location-based-services.
802.11m	WLAN			Maintenance
802.11n	WLAN	- 600 Mbit/s	2.4 - 5 GHz	Next generation WLAN. Est. 2009

Table A.1: Overview of Wireless Networking Standards (excerpt) Part 1

Name	Type	Data Rate	Frequency	Description
802.15	WPAN	+ 480 Mbit/s	0.1 - 10.6 GHz	Bluetooth standards. up to 12m range, wireless USB
802.15.3	WPAN	480 Mbit/s	2.4 GHz	Bluetooth, guaranteed level of service. wireless fire-wire.

Table A.2: Overview of Wireless Networking Standards (excerpt) Part 2

Name	Type	Data Rate	Frequency	Description
802.16	WMAN	- 70 Mbit/s	10 - 60 GHz	Air interface for fixed Broadband Wireless Access Systems. WiMax (World Interoperability for Microwave Access), Range up to 70km
802.16a	WMAN	- 134 Mbit/s	2 - 11 GHz	Licensed WiMax frequencies for fixed receivers.
802.16b	WMAN	- 134 Mbit/s	5 - 6 GHz	Licensed Exempt Frequencies, WirelessHUMAN (High Speed Unlicensed MAN).
802.16c	WMAN	- 134 Mbit/s	10 - 66 GHz	Conformity standard to support 802.16 interoperability specifications .
802.16d				Extended Version of 802.16, also known as IEEE 802.16-2004.
802.16e	WMAN	- 70 Mbit/s		Mobile WirelessMAN, extensions for 802.16a MAC/PHY, mobility support up to 120 Km/h .
802.16.2	WMAN			Coexistence .
802.16.2a	WMAN		10 - 66 GHz	Recommended Practice for coexistence of Fixed Broadband Wireless Access Systems.
802.16.3	WMAN		- 11 GHz	Air Interface for Fixed Broadband Wireless Access Systems below 11 GHz.

Table A.3: Overview of Wireless Networking Standards (excerpt) Part 3

Name	**Type**	**Data Rate**	**Frequency**	**Description**
802.18				Radio Regulatory Technical Advisory Group (RRTAG).
802.19				Coexistence TAG.
802.20		- 16 Mbit/s		Mobile Broadband Wireless Access (MBWA). For trains and cars up to 250 km/h.

Table A.4: Overview of Wireless Networking Standards (excerpt) Part 4

B
Abstract of IEEE 802.1X

This appendix is a brief introduction to the IEEE 802.1X standard [84]

IEEE 802.1X is a standard for reliable authentication in IEEE 802 based computer networks, wired as well as wireless. A client is able to authenticate via an *authenticator* who is checking the validity of the client, also called the *supplicant* by the help of an authentication server (i.e using the RADIUS protocol). If the authentication was successful, the authenticator may provide various services like access to the network. This procedure allows the usage of decentralized authentication information.

The standard recommends the *extensible authentication protocol* (EAP) or the *PPP-EAP-TLS* extension for the authentication task due to the lack of own defined authentication protocols.

A IEEE 802.1X network setup consists of four functional entities:

1. **Supplicant**
   Supplicants are all devices that want to gain access to an IEEE 802.1X secured network. Usually this device is realized as a software implementation sometimes integrated into the operating system.

2. **Authenticator**
   The authenticator renders the barrier between the supplicant and the secured network. The supplicant provides authentication credentials which are usually validated by the authenticator by double-checking with an authentication server using the RADIUS protocol.

3. **Port Access Entity (PAE)**
   The PAE is mostly a logical port provided by the authenticator. Depending

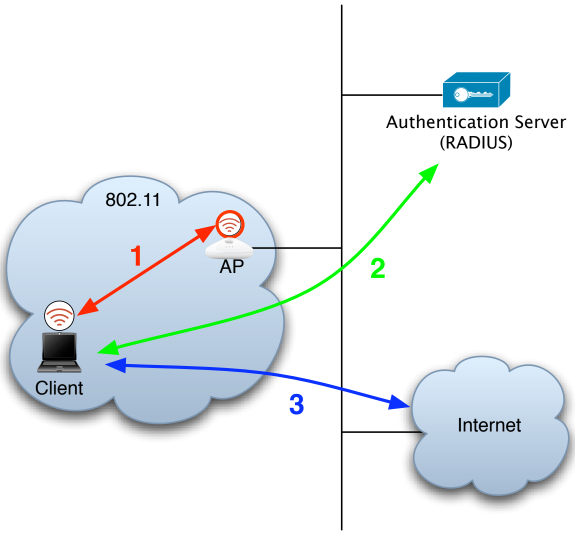


Figure B.1: IEEE 802.1X Connection Scheme

on the authorization state of the supplicant, the PAE provides access to services located in the secured network.

4. **Authentication Server (AS)**
   The AS provides the authentication service to the authenticator from inside the secured network. Most off the popular implementations follow the RADIUS protocol. Authentication Servers are provided by all major network hardware vendors and are included in several popular server operating systems. As told before, the AS is validating the supplicants authentication credentials. These credentials may either be stored at the AS it self or decentralized somewhere in the network. Most RADIUS implementations support the LDAP protocol.

Figure B.1 illustrates the authentication process defined in IEEE 802.1X. It consists of three basic steps.

1. The supplicant joins the network at the restricted port of the access-point (AP) which provides the authenticator functionality.

2. The AP provides connection to the authentication server in order to validate the authentication credentials sent by the supplicant and informs the AP weather the authentication was successful or not.

3. Depending on the success of the authentication and the access level of the supplicant, the AP now provides access to the services in the secured network e.g. a gateway to the Internet.

IEEE 802.1X is a simple but powerful security extension for business networks. It is supported by all major software and hardware vendors.

Bibliography

[1] 107th Congress of the United States of America. USA Patriot Act, 2010.

[2] C. Adams and S. Lloyd. *Understanding PKI: Concepts, Standards, and Deployment Considerations (2nd Edition).* Addison-Wesley Professional, 2010.

[3] S. Adibi, B. Lin, P.-h. Ho, G. Agnew, and S. Erfani. Authentication Authorization and Accounting (AAA) Schemes in WiMAX. In *2006 IEEE International Conference on Electro/Information Technology*, pages 210–215. IEEE, May 2006.

[4] Aircrack-ng. Aircrack-ng, 2010.

[5] Airspan Inc. Mobile WiMAX Security, 2007.

[6] Apple Inc. Apple - iPhone - iOS 4 is the worlds most advanced mobile OS, 2010.

[7] Apple Inc. Apple (Republic of Ireland) - iPhone 4 - Find hundreds of thousands of apps on the App Store, 2010.

[8] J. Assange. WikiLeaks, 2010.

[9] P. Bahl and V. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, pages 775–784. IEEE, 2000.

[10] D. Barbara, J. Couto, and Y. Li. COOLCAT: an entropy-based algorithm for categorical clustering, 2002.

[11] M. Barbeau. WiMax/802.16 threat analysis. *International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, page 8, 2005.

[12] M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attacks. In *EUROCRYPT*, pages 139–155, 2000.

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

[14] H. Berger, M. Dittenbach, and D. Merkl. An adaptive information retrieval system based on associative networks. *Conferences in Research and Practice in Information Technology Series; Vol. 59*, page 27, 2004.

[15] K. Boman, G. Horn, P. Howard, and V. Niemi. UMTS security. *Electronics & Communications Engineering Journal*, 14(5):191, 2002.

[16] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: the insecurity of 802.11. *International Conference on Mobile Computing and Networking*, page 180, 2001.

[17] P. Bouska and M. Drahanský. *Communication Security in GSM Networks*. IEEE, Dec. 2008.

[18] S. Brands and D. Chaum. Distance-Bounding Protocols (Extended Abstract). In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Technique*, pages 344–359. Lecture Notes in Computer Science 765, 1993.

[19] L. Bussard. *Trust Establishment Protocols for Communicating Devices*. Phd thesis, Eurecom-ENST, 2004.

[20] J. J. Caffery and G. L. Stüber. Overview of Radiolocation in CDMA Cellular Systems, 1998.

[21] M. Caloyannides. Privacy vs. information technology. *IEEE Security & Privacy Magazine*, 1(1):100–103, Jan. 2003.

[22] S. Capkun, L. Buttyán, and J.-P. Hubaux. SECTOR: secure tracking of node encounters in multi-hop wireless networks. In *ACM Workshop on Security of Ad Hoc and Sensor Networks (SANS)*, pages 21–32, Washington, Oct. 2003. ACM.

[23] S. Capkun and J.-P. Hubaux. Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(2):221–232, Feb. 2006.

[24] D. Caswell and P. Debaty. Creating Web Representations for Places. *Lecture Notes In Computer Science*, page 114, 2000.

[25] R. Chandra. *A virtualization architecture for wireless network cards*. Phd thesis, Cornell University, 2006.

[26] R. Chandra, P. Bahl, and P. Bahl. MultiNet: connecting to multiple IEEE 802.11 networks using a single wireless card. In *IEEE INFOCOM 2004*, pages 882–893. IEEE, 2004.

[27] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24:84–88, 1981.

[28] D. Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto*, volume 82, pages 199–203, 1982.

[29] D. Chaum and E. Van Heyst. Group Signatures. In *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, pages 257–265. Springer-Verlag, 1991.

[30] Y.-C. Chen, Y.-J. Chan, and C.-W. She. Enabling location-based services in wireless LAN hotspots. *International Journal of Network Management*, 15(3):163, 2005.

[31] Y. Cho, L. Bao, and M. T. Goodrich. LAAC: A Location-Aware Access Control Protocol. In *Annual International Conference on Mobile and Ubiquitous Systems*, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[32] M. Ciurana, F. Barceló, and S. Cugno. Indoor tracking in WLAN location with TOA measurements. *Proceedings of the international workshop on Mobility management and wireless access - MobiWac '06*, page 121, 2006.

[33] F. Crestani. Application of Spreading Activation Techniques in InformationRetrieval. *Artificial Intelligence Review*, 11(6):453, 1997.

[34] Cylink Corporation. Safer administration of insulin: summary of a safety report from the National Patient Safety Agency. In NIST, editor, *First Advanced Encryption Standard Candidate Conference*, volume 341, page c5269, Jan. 1998.

[35] J. Daemen and V. Rijmen. *The Design of Rijndael.* 2002.

[36] K. Defrawy and C. Soriente. PEUC-WiN: Privacy Enhancement by User Cooperation in Wireless Networks. In *2006 2nd IEEE Workshop on Secure Network Protocols*, pages 38–43. IEEE, Nov. 2006.

[37] D. E. Denning and P. F. MacDoran. Location-based authentication: grounding cyberspace for better security. *Internet besieged: countering cyberspace scofflaws*, page 167, 1997.

[38] L. C. C. Desmond, C. C. Yuan, T. C. Pheng, and R. S. Lee. Identifying unique devices through wireless fingerprinting. *Proceedings of the first ACM conference on Wireless network security - WiSec '08*, page 46, 2008.

[39] C. Douligeris and D. Serpanos. *Network Security : Current Status and Future Directions.* Wiley-IEEE Press, 1 edition, 2007.

[40] R. Droms. RFC 2131 - Dynamic Host Configuration Protocol (DHCP), 1997.

[41] B. Dundar and P. Varaiya. Hybrid algorithm for indoor positioning using wireless LAN. *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*, pages 4625–4629, 2004.

[42] J. Dunning. Taming the Blue Beast: A Survey of Bluetooth Based Threats. *IEEE Security & Privacy Magazine*, 8(2):20–27, Mar. 2010.

[43] N. Eagle and A. Pentland. Eigenbehaviors: Identifying Structure in Routine. In *In Proceedings of UBICOMP06*, 2006.

[44] W. Enck, M. Ongtang, and P. McDaniel. Understanding Android Security. *IEEE Security and Privacy*, 7:50–57, 2009.

[45] A. F. Famili, J. N. Kok, J. M. Peña, A. Siebes, A. Feelders, and J. Couto. *Advances in Intelligent Data Analysis VI*, volume 3646 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[46] M. Feily, A. Shahrestani, and S. Ramadass. A Survey of Botnet and Botnet Detection. In *2009 Third International Conference on Emerging Security Information, Systems and Technologies*, pages 268–273. IEEE, June 2009.

[47] C. Fellbaum. *WordNet: An Electronic Lexical Database (Language, Speech & Communication) (Language, Speech and Communication)*. MIT Press, 1998.

[48] N. Ferguson. Michael: an improved MIC for 802.11 WEP. Technical report, IEEE, 2002.

[49] M. C. Fernandez-Gago, R. Roman, and J. Lopez. *A Survey on the Applicability of Trust Management Systems forWireless Sensor Networks*. IEEE, July 2007.

[50] R. Fontana, E. Richley, and J. Barney. Commercialization of an ultra wideband precision asset location system. In *IEEE Conference on Ultra Wideband Systems and Technologies, 2003*, pages 369–373. IEEE, 2003.

[51] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *USENIX Security Symposium*, 2006.

[52] A. Freier, P. Karlton, and P. Kocher. The SSL Protocol Version 3.9, 1996.

[53] B. Fritzke. A growing neural gas learns topologies. *Advances in Neural Information Processing Systems*, pages 1211–1216, 2005.

[54] C. Gehrmann and K. Nyberg. Enhancements to Bluetooth baseband security. In *Proceedings of Nordsec*, volume 2001, pages 191–230, 2001.

[55] Google Inc. Android Market, 2010.

[56] Google Inc. Google Maps, 2010.

[57] Google Inc. What is Android? — Android Developers, 2010.

[58] M. Graham. Poster Abstract : A System for Secure Verification of Location Claims. *Mobile Computing and Communications Review*, 12(2):47–49, 2009.

[59] M. Graham and D. Gray. Protecting Privacy and Securing the Gathering of Location Proofs – The Secure Location Verification Proof Gathering Protocol. *Security and Privacy in Mobile Information and Communication Systems*, pages 160–171, 2009.

[60] M. Gruteser. Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis. In *Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots WMASH'03*, pages 46–55. ACM Press, 2003.

[61] A. Günther and C. Hoene. Measuring Round Trip Times to Determine the Distance Between WLAN Nodes. *Networking 2005*, pages 768–779, 2005.

[62] Y. Gwon and R. Jain. Error characteristics and calibration-free techniques for wireless LAN-based location estimation. In *Proceedings of the second international workshop on Mobility management & wireless access protocols - MobiWac '04*, page 2, New York, New York, USA, 2004. ACM Press.

[63] J. Hall, M. Barbeau, and E. Kranakis. Detection of Transient in Radio Frequency Fingerprinting using Signal Phase. in Proc. 3rd IASTED International Conference on Wireless and Optical Communications (WOC), 2003.

[64] J. Hall, M. Barbeau, and E. Kranakis. Radio Frequency Fingerprinting for Intrusion Detection in Wireless Networks. In *IEEE Transaction on dependable and secure Computing*, 2006.

[65] J. Hallberg, M. Nilsson, and K. Synnes. Positioning with Bluetooth. In *10th International Conference on Telecommunications, 2003. ICT 2003.*, pages 954–958. IEEE, 2003.

[66] G. Hancke and M. Kuhn. An RFID Distance Bounding Protocol. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 67–73, 2005.

[67] H. Hassan and Y. Challal. Enhanced WEP: an efficient solution to WEP threats. In *Second IFIP International Conference on Wireless and Optical Communications Networks, 2005. WOCN 2005.*, pages 594–599. IEEE, 2005.

[68] C. He and J. C. Mitchell. Security analysis and improvements for IEEE 802.11i. In *In Proceedings of the 12th Annual Network and Distributed System Security Symposium*, pages 90–110. In Proceedings of the 12th Annual Network and Distributed System Security Symposium, 2005.

[69] J. Hightower and G. Borriello. Location Systems for Ubiquitous Computing. *Computer*, 34(8):57, 2001.

[70] J. Hightower, C. Vakili, G. Borriello, and R. Want. Design and Calibration of the SpotON Ad-Hoc Location Sensing System, 2001.

[71] S.-j. Horng, C. Chen, H.-w. Ferng, T.-w. Kao, and M.-h. Li. Enhancing WLAN location privacy using mobile behavior. *Expert Systems With Applications*, 38(1):175–183, 2011.

[72] Y. Hu, Y. Zhang, and G. Xiao. Integral cryptanalysis of SAFER+. *Electronics Letters*, 35(17):1458, 1999.

[73] J. Huang, W. Susilo, and J. Seberry. Observations on the Message Integrity Code in IEEE 802.11 Wireless LANs, 2008.

[74] J. Huang, W. Susilo, J. Seberry, and M. Bunder. On the Security of the IEEE 802.11i Message Integrity Code Michael. Technical report, 2004.

[75] IEEE. *IEEE 802.11, 1999 Edition (ISO/IEC 8802-11: 1999) IEEE Standards for Information Technology Telecommunications and Information Exchange between Systems Local and Metropolitan Area Network Specific Requirements Part 11: Wireless LAN Medium Access Control (*. IEEE, 1999.

[76] IEEE. 802.16-2001 IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems, 2001.

[77] IEEE. 802.16-2004 IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems, 2004.

[78] IEEE. *IEEE 802.11i-2004 Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003). IEEE Standard for Information technology Telecommunications and information exchange between system Local and metropolitan area networks Specific requirements Part 11: Wireless LAN*. IEEE, 2004.

[79] IEEE. P802.11j/D1.6, Aug 2004 Draft Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003) Amendment 7: 4.9 GHz5 GHz Operation in Japan (as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, 802.11d-2001, 802.11g-2003, 802.11h-2003), and. 2004.

[80] IEEE. 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) specificat, 2005.

[81] IEEE. 802.16e-2005 and IEEE Std 802.16-2004/Cor1-2005 IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed, 2005.

[82] IEEE. 802.16j-2009 IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems Amendment 1: Multiple Relay Specification, 2009.

[83] IEEE. IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems Revision of IEEE Std 802.16-2004, 2009.

[84] IEEE. 802.1X-2010 IEEE Standard for Local and metropolitan area networks - Port-Based Network Access Control, 2010.

[85] IEEE. Complete List of IEEE OIUs, 2010.

[86] ITU-T. X.509: Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks, 2010.

[87] R. Jain and T. Kawahara. Robust indoor location estimation of stationary and mobile users. *Ieee Infocom 2004*, pages 1032–1043, 2004.

[88] D. Johnston and J. Walker. Overview of IEEE 802.16 security. *IEEE Security & Privacy Magazine*, 2(3):40–48, May 2004.

[89] K. Kaemarungsi. Distribution of WLAN Received Signal Strength Indication for Indoor Location Determination. *2006 1st International Symposium on Wireless Pervasive Computing*, pages 1–6, 2006.

[90] T. Kindberg, K. Zhang, and N. Shankar. Context authentication using constrained channels. In *Proceedings Fourth IEEE Workshop on Mobile Computing Systems and Applications*, pages 14–21. IEEE Comput. Soc, 2002.

[91] P. Kitsos, N. Sklavos, K. Papadomanolakis, and O. Koufopavlou. Hardware Implementation of Bluetooth Security. *IEEE Pervasive Computing*, 2(1):21, 2003.

[92] A. Klein. Attacks on the RC4 stream cipher. *Designs, Codes and Cryptography*, 48(3):269, 2008.

[93] T. Kohno, A. Broido, and K. C. Claffy. Remote Physical Device Fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2(2):93, 2005.

[94] T. Kohonen. *Self-Organizing Maps.* Springer, 2008.

[95] A. Kotanen, M. Hannikainen, H. Leppakoski, and T. Hamalainen. Experiments on local positioning with Bluetooth. In *Proceedings ITCC 2003. International Conference on Information Technology: Coding and Computing*, pages 297–303. IEEE Comput. Soc, 2003.

[96] H. Kozima and T. Furugori. Similarity between words computed by spreading activation on an English dictionary. *European Chapter Meeting of the ACL*, page 232, 1993.

[97] H. Kozima and A. Ito. Context-Sensitive Measurement of Word Distance by Adaptive Scaling of a Semantic Space. In *In Proceedings of RANLP-95*, pages 161–168. In Proceedings of RANLP-95, 1995.

[98] S. Kraxberger, G. Lackner, and U. Payer. WLAN Location Determination without Active Client Collaboration. In ACM, editor, *IWCMC '10: Proceedings of the 2010 International Conference on Wireless Communications and Mobile Computing*, pages 1188–1192. ACM, 2010.

[99] D. Kulkarni and A. Tripathi. Context-aware role-based access control in pervasive computing systems. In *SACMAT '08: Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 113–122, New York, NY, USA, 2008. ACM.

[100] G. Lackner. *IEEE 802.11 Layer 2 Address-Spoofing Protection.* Master thesis, Graz University of Technology, 2008.

[101] G. Lackner. *Wireless-Network Security: Basics Knowledge, Cryptography, MAC-Layer Security, Best Practices.* VDM Verlag Dr. Müller, 2009.

[102] G. Lackner. A Comparison of Security in Wireless Network Standards with a Focus on Bluetooth , WiFi and WiMAX. *Accapted but not yet published at the International Journal of Network Security*, 2011.

[103] G. Lackner. On the Security of Location Determination and Verification Methods for Wireless Networks. In *Accapted but not yet published at International Conference on Security and Cryptography SECRYPT 2011*, Sevilla, 2011.

[104] G. Lackner. Security Related Evaluation of Location Determination and Verification Methods for Wireless Networks. *International Journal of Network Security*, 2011.

[105] G. Lackner and S. Kraxberger. Location Privacy Enhancement for WLANs based on Virtual Network Interfaces. In *Submitted to the Privacy Enhancing Technologies Symposium PETS2011*, 2011.

[106] G. Lackner, S. Kraxberger, P. Teufl, and M. Eian. Location Aware Access Regulation for Wireless Computer Networks - A Comparative Survey. *UNDER REVIEW! International Journal of Information Security*, 2011.

[107] G. Lackner, M. Lamberger, P. Teufl, and U. Payer. WiFi Chipset Fingerprinting. In P. Horster, editor, *DACH Security 2006*, pages 41–53, Munich, 2006.

[108] G. Lackner, U. Payer, and P. Teufl. Combating Wireless LAN MAC-Layer Address Spoofing with Fingerprinting Methods. *International journal of network security*, Vol. 9:164–172, 2009.

[109] G. Lackner and P. Teufl. Location Privacy in Kabellosen Netzwerken. In *DACH2010*, page 0, 2010.

[110] G. Lackner and P. Teufl. IEEE 802.11 Chipset Fingerprinting by the Measurement of Timing Characteristics. In *In Proceedings of the Australasian Information Security Conference 2011, AISC11*, page 0, Jan. 2011.

[111] G. Lackner, P. Teufl, and R. Weinberger. Unterschätzes Risiko durch ultramobile Geräte. In O. C. Gesellschaft, editor, *7. Information Security Konferenz*, pages 43–59. Österreichische Computer Gesellschaft, 2009.

[112] G. Lackner, P. Teufl, and R. Weinberger. User Tracking based on Behavioral Fingerprints. In *In Proceedings of the Ninth International Conference on Cryptology And Network Security (CANS 2010)*, page 0, Kuala Lumpur, 2010.

[113] M. Lei, Z. Qi, X. Hong, and S. V. Vrbsky. Protecting Location Privacy with Dynamic Mac Address Exchanging in Wireless Networks. In *2007 IEEE Intelligence and Security Informatics*, pages 377–377. IEEE, May 2007.

[114] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, Nov. 2007.

[115] C. Luo. A Simple Encryption Scheme Based on WiMAX. In *2009 International Conference on E-Business and Information System Security*, pages 1–4. IEEE, May 2009.

[116] L. Maccari, M. Paoli, and R. Fantacci. Security Analysis of IEEE 802.16. In *2007 IEEE International Conference on Communications*, pages 1160–1165, Glasgow, June 2007. IEEE.

[117] I. Mantin. Analysis of the Stream Cipher RC4. Master's thesis, Weizmann Institute Of Science, Israel, 2001.

[118] W. Mao. *Modern Cryptography: Theory and Practice.* Prentice Hall, 2003.

[119] T. Martinetz and K. Schulten. A "neural gas" network learns topologies. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, pages 397–402. Elsevier, Amsterdam, 1991.

[120] MathWorks. MATLAB - The Language Of Technical Computing, 2010.

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

[122] A. Menezes. *Handbook of Applied Cryptography (Discrete Mathematics and Its Applications).* CRC Press, 1996.

[123] Metasploit. Penetration Testing — The Metasploit Project, 2010.

[124] N. Michalakis. *Location-aware Access Control for Pervasive Computing Environments.* Master thesis, Massachusetts Institute of Technology, 2003.

[125] Microsoft Inc. Windows Phone 7 — Devices, news, downloads., 2010.

[126] N. F. Mir. *Computer and Communication Networks.* Prentice Hall, 2006.

[127] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements and Performance.* Ganga-Jamuna Pr, 2001.

[128] G. Moore. Moore's Law and Intel Innovation, 1965.

[129] National Institute of Standards and Technology. Data encryption standard. FIPS Publication 46-2, December 1993.

[130] M. Nauman, S. Khan, and X. Zhang. Apex: extending Android permission model and enforcement with user-defined runtime constraints. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 328–332. ACM, 2010.

[131] G. Netcat. The GNU Netcat – Official homepage, 2010.

[132] L. Ni and A. Patil. LANDMARC: indoor location sensing using active RFID. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).*, pages 407–415. IEEE Comput. Soc, 2003.

[133] NIST. FIPS PUB 186-3: Digital Signature Standard (DSS). *Federal Information Processing Standards Publication*, 2000.

[134] Nmap.org. Nmap - Free Security Scanner For Network Exploration &amp; Security Audits., 2010.

[135] M. Ongtang, S. McLaughlin, W. Enck, and P. McDaniel. Semantically Rich Application-Centric Security in Android. *2009 Annual Computer Security Applications Conference*, pages 340–349, Dec. 2009.

[136] C. Orthacker, P. Teufl, S. Kraxberger, G. Lackner, A. Marsalek, J. Leibetseder, and O. Prevenhueber. Android Security Permissions - Can we trust them ? In *Accepted but not yet published at MobiSEC 2011*, 2011.

[137] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara. Accurate GSM Indoor Localization. In *UbiComp 2005: Ubiquitous Computing*, pages 141–158. 2005.

[138] K. Pahlavan and J. Makela. Indoor geolocation science and technology. *IEEE Communications Magazine*, 40(2):112–118, 2002.

[139] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 user fingerprinting. In *In MobiCom 07: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, pages 99–110. In MobiCom 07: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, 2007.

[140] B. Parodi, H. Lenz, A. Szabo, J. Horn, J. Bamberger, and D. Obradovic. *Initialization and Online-Learning of RSS Maps for Indoor / Campus Localization*. IEEE, 2006.

[141] S. Pasanen. New Efficient RF Fingerprint-Based Security Solution for Bluetooth Secure Simple Pairing. *Security*, pages 1–8, 2010.

[142] U. Payer, M. Lamberger, and P. Teufl. Traffic Classification using Self-Organizing Maps. In *INC 2005 5. International Networking Conference Workshops, Samos Island, Greece*, 2005.

[143] D. Plummer. An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware, 1982.

[144] D. C. Plummer. RFC 862 - An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware, 1982.

[145] B. Potter, D. Till, and J. Sinclair. *An Introduction to Formal Specification and Z*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1996.

[146] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. *International Conference on Mobile Computing and Networking*, page 32, 2000.

[147] A. K. Qin and P. N. Suganthan. Robust growing neural gas algorithm with application in cluster analysis. *Neural Networks*, 17(8):1135, 2004.

[148] M. R. Quillian. Semantic memory. *Semantic Information Processing*, pages 227–270, 1968.

[149] K. Rasmussen. Realization of RF Distance Bounding. In *Proceedings of the 19th USENIX Security Symposium, 13 pages. USENIX*, 2010.

[150] I. Ray, M. Kumar, and L. Yu. LRBAC: A Location-Aware Role-Based Access Control Model. In A. Bagchi and V. Atluri, editors, *Information Systems Security*, volume 4332 of *Lecture Notes in Computer Science*, pages 147–161. Springer Berlin / Heidelberg, 2006.

[151] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji. Detecting relay attacks with timing-based protocols. *ASIAN ACM Symposium on Information, Computer and Communications Security*, page 204, 2007.

[152] RIM. BlackBerry App World - Home, 2010.

[153] J. Rissanen. Stochastic Complexity in Statistical Inquiry Theory. page 177, 1989.

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

[155] F. Robinson. 802.11i and WPA Up Close. *Network Computing*, 2004.

[156] RSA Laboratories. RSA Laboratories - PKCS #1: RSA Cryptography Standard, 2010.

[157] N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location Claims. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering WiSE'03*, pages 1–10, 2003.

[158] N. Sastry and D. Wagner. Security considerations for IEEE 802.15.4 networks. In *Proceedings of the 2004 ACM workshop on Wireless security - WiSe '04*, page 32, New York, New York, USA, 2004. ACM Press.

[159] K. Scarfone and J. Padgette. Guide to bluetooth security, 2008.

[160] B. Schneier. We are all security consumers. *IEEE Security & Privacy Magazine*, 1(1):104–104, Jan. 2003.

[161] a. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev, and C. Glezer. Google Android: A Comprehensive Security Assessment. *IEEE Security & Privacy Magazine*, 8(2):35–44, Mar. 2010.

[162] Y. Shaked and A. Wool. Cracking the bluetooth pin. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 39–50. ACM, 2005.

[163] B. Sieka. Active fingerprinting of 802.11 devices by timing analysis. In *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, volume 1, pages 15–19, 2006.

[164] W. Simpson. RFC1994: PPP Challenge Handshake Authentication Protocol (CHAP), 1996.

[165] D. Singelee. *Study and Design of a Security Architecture for Wireless Personal Area Networks*. Phd thesis, KU Leuven, 2008.

[166] D. Singelee and B. Preneel. Location verification using secure distance bounding protocols. *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.*, pages 834–840, 2005.

[167] D. Singelee and B. Preneel. Location privacy in wireless personal area networks. In *WiSe '06 Proceedings of the 5th ACM workshop on Wireless security*, 2006.

[168] E. Sithirasenan, S. Zafar, and V. Muthukkumarasamy. Formal Verification of the IEEE 802.11i WLAN Security Protocol. In *Australian Software Engineering Conference (ASWEC'06)*, pages 181–190. IEEE, 2006.

[169] Skyhook Wireless Inc. Skyhook Wireless http://www.skyhookwireless.com/, 2010.

[170] D. Slamanig, G. Lackner, C. Stingl, and U. Payer. Schutz der Privatsphäre in einem webbasierten Multiuser-System. In *DACH Security 2007*, Apr. 2007.

[171] N. P. Smart. *Cryptography: an introduction*. McGraw-Hill, 2003.

[172] D. Stanley, J. Walker, and B. Aboba. Extensible Authentication Protocol (EAP) Method Requirements for Wireless LAN, 2005.

[173] Tcpdump. Tcpdump/Libpcap public repository, 2010.

[174] P. Teufl, S. Kraxberger, C. Orthacker, G. Lackner, A. Marsalek, J. Leibetseder, and O. Prevenhueber. Android Market Analysis with Activation Patterns. In *Accepted but not yet published at MobiSEC 2011*, 2011.

[175] P. Teufl and G. Lackner. RDF Data Analysis with Activation Patterns. In K. T. Und Hermann Maurer, editor, *Proceedings of the 10th International Conference on Knowledge Management and Knowledge Technologies iKNOW 2010 Graz Austria*, Journal of Computer Science, page 0, 2010.

[176] P. Teufl, G. Lackner, and U. Payer. From NLP (Natural Language Processing) to MLP (Machine Language Processing). In I. Kotenko and V. Skormin, editors, *Proceedings of the Mathematical Methods Models and Architectures for Computer Networks Security Conference 2010 MMMACNS 2010 St Petersburg Russia*, volume 6258 of *Lecture Notes in Computer Science*, pages 256–269. Springer Berlin Heidelberg, 2010.

[177] P. Teufl, U. Payer, and R. Fellner. Event correlation on the basis of activation patterns. In *Proceedings of the 18th Euromicro Conference on Parallel Distributed and NetworkBased Processing PDP 2010*, pages 631–640, 2010.

[178] P. Teufl, U. Payer, P. Parycek, A. Macintosh, and E. Tambouris. Automated Analysis of e-Participation Data by Utilizing Associative Networks, Spreading Activation and Unsupervised Learning. *ePart 09 Proceedings of the 1st International Conference on Electronic Participation*, 5694:139–150, 2009.

[179] E. Tews, R.-P. Weinmann, and A. Pyshkin. Breaking 104 Bit WEP in less than 60 seconds. In *Proceedings of the 8th international conference on Information security applications*, pages 188–202. Lecture Notes in Computer Science, 2007.

[180] N. O. Tippenhauer, K. B. Rasmussen, C. Pöpper, and S. Čapkun. Attacks on public WLAN-based positioning systems. *Proceedings of the 7th international conference on Mobile systems, applications, and services - Mobisys '09*, page 29, 2009.

[181] A. Toninelli, R. Montanari, O. Lassila, and D. Khushraj. What's on Users' Minds? Toward a Usable Smart Phone Security Model. *IEEE Pervasive Computing*, 8(2):32–39, Apr. 2009.

[182] M. Toorani and A. A. Beheshti Shirazi. LPKI - a Lightweight Public Key Infrastructure for the mobile environments. In *2008 11th IEEE Singapore International Conference on Communication Systems*, pages 162–166. IEEE, Nov. 2008.

[183] W. Trappe and L. C. Washington. *Introduction to Cryptography with Coding Theory (2nd Edition)*. Prentice Hall, 2005.

[184] G. Tsatsaronis, M. Vazirgiannis, and I. Androutsopoulos. Word sense disambiguation with spreading activation networks generated from thesauri. In *International Joint Conference On Artificial Intelligence*, pages 1725–1730, 2007.

[185] T. Vennon and D. Stroop. Android Market: Threat Analysis of the Android Market, 2010.

[186] Vesanto, Himberg, Alhoniemi, and Parhankangas. SOM Toolbox for Matlab, Technical Report A57. Technical report, Helsinki University of Technology, 2000.

[187] B. Warneke, M. Last, B. Liebowitz, and K. Pister. Smart Dust: communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51, 2001.

[188] B. Waters and E. Felten. Proving the Location of Tamper-Resistant Devices. Technical report, Department of Computer Science Princeton University, 2003.

[189] B. Waters and E. Felten. Secure, private proofs of location, 2003.

[190] M. Weiser. The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review*, 3(3):3, 1999.

[191] L. Wenjing and R. Kui. Security, privacy, and accountability in wireless access networks. *IEEE Wireless Communications*, 16(4):80–87, Aug. 2009.

[192] F.-L. Wong and F. Stajano. Location Privacy in Bluetooth. In R. Molva, G. Tsudik, and D. Westhoff, editors, *Security and Privacy in Ad-hoc and Sensor Networks Second European Workshop, ESAS 2005, Visegrad, July 13-14*, volume 3813 of *Lecture Notes in Computer Science*, pages 176–188, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[193] A. Wool. A Note on the Fragility of the Michael Message Integrity Code. *IEEE Transactions on Wireless Communications*, 3(5):1459–1462, Sept. 2004.

[194] D. Wu, Y. Xu, and L. Ma. Research on RSS based Indoor Location Method. In *2009 Pacific-Asia Conference on Knowledge Engineering and Software Engineering*, pages 205–208. Ieee, Dec. 2009.

[195] S. Xu and C.-T. Huang. Attacks on PKM Protocols of IEEE 802.16 and Its Later Versions. *2006 3rd International Symposium on Wireless Communication Systems*, pages 185–189, Sept. 2006.

[196] M. Youssef and A. Agrawala. The Horus WLAN location determination system. *International Conference On Mobile Systems, Applications And Services*, page 205, 2005.

[197] K. Zeng, K. Govindan, and P. Mohapatra. Non-cryptographic authentication and identification in wireless networks [Security and Privacy in Emerging Wireless Networks. *IEEE Wireless Communications*, 17(5):56–62, Oct. 2010.

[198] Y. Zhou and Y. Fang. Security of IEEE 802.16 in Mesh Mode. In *MILCOM 2006*, pages 1–6. IEEE, Oct. 2006.

TU Graz
Graz University of Technology
Senat

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ………………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ………………………………………………..
date (signature)