Security in Ad-hoc Networks — Survey and Implementation

Makoto Miyamoto

August 2003

Diplomarbeit

Ruhr-Universität Bochum

Chair for Communication Security
Prof. Dr.-Ing. Christof Paar

cv cryptovision gmbh
Benjamin Drisch and Dr. Thomas Zeggel

Erklärung

Hiermit versichere ich, dass ich meine Diplomarbeit selbst verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Ort, Datum

Unterschrift

Abstract

Some years ago the idea of a fully computerized world has been born, grown and has itself settled in the minds of researchers, scientists and businessmen around the world. They imagined that all microprocessors used in everyday products like cars, washing machines, coffee makers, wrist-watches, cell phones, PDAs (Personal Digital Assistants), notebooks, computers and so on would talk and interact with each other and form an information network for the user's benefit.

Although the full scope of such a vision is still futuristic, portions of it are already reality. Scientists and researchers around the world are working for nearly 10 years on ad-hoc networks and their commercial or military applications. Wireless LAN hot-spots are sprouting everwere. Bluetooth is connecting mobile phones with PDAs, digital cameras and computers. The interconnection of home entertainment equipment and home automation is becoming affordable for anyone.

Security is an important issue for deployment. The security holes which were discovered in the safety protocols of wireless LANs are so big that it makes no difference for an attacker whether they are turned on or not. The Bluetooth option of many mobile devices lies unused, because its difficult and complicated pairing process scares away many technical unskilled users. Routing in mobile ad-hoc networks is inevitable but still not solved in a satisfactory manner.

This document is twofold. The first part and focal point of this document is intended as a survey and introduction of existing techniques dealing with ad-hoc networks. It should guide a reader through the diverse layers and types of ad-hoc networks and emphasize security related problems. The glossary and list of abbreviations, which can be found in the Appendix define many of the new and ad-hoc specific acronyms found in this and many related papers. The second part is the description of an application called *simahnsai*, which has been designed and implemented during this diploma thesis. *simahnsai* is a secure instant messenger, which establishes a secret key between two instances and encrypts all messages between those two applications.

Contents

1. Introduction

Some years ago the idea of a fully computerized world has been born, grown and has itself settled in the minds of researchers, scientists and businessmen around the world. They imagined that all microprocessors used in everyday products like cars, washing machines, coffee makers, wrist-watches, cell phones, PDAs (Personal Digital Assistants), computers, notebooks, and so on would talk to each other and form an information network for the user's benefit. Because those networks are formed spontanously and are not ment to last for long, they are called ad-hoc networks.[1] The general idea of an ad-hoc network, is a device (node) which connects automatically to the network upon detection and provides its user all features and benefits it could handle. A pure ad-hoc network would consist entirely of such nodes.

1.1. Visions and Possibilities

There are plenty of possibilities for ad-hoc networks. Some of them are still visionary, some are fact. Here's a small collection of what could be and what is actually possible.

- One vision of an ad-hoc network would be in a disaster scenario, where infrastructural damage has ceased access to the telephone network. In this case the cell phones would connect to each other in an ad-hoc mode and relay a distress call to the nearest hospital.

- Other visions include everyday scenarios where "wireless" computers make life easier. The "Future Store" in Rheinberg [33] is a step in this direction. Regular visitors of the store can identify themselves on their shopping carts and the cart displays personalised shopping lists, advertisements and special offers to the user. This is not yet ad-hoc, because the visitor must interact with the cart, i. e., he must scan every bar code of his purchase. When Radio Frequency Identifications (RFIDs) have eventually replaced bar codes, the interaction of the user can be further minimized. The item's RFID will connect with the cart's computer automatically, when the appropriate item is placed in the cart. The payment of the purchased goods is currently done offline, using credit or debit cards. When credit cards and RFIDs merge, even online payment could be implemented.

[1] ad-hoc means "for this (only)" in latin

- In his 1995 sci-fi novel ”Diamond Age” [59] Neal Stephenson describes nanomachines (motes) which were released into the air and into body fluids to gather information. These motes were capable of forming together a huge information network. The DARPA (Defence Advanced Research Projects Agency) in corporation with UC Berkley currently researches smart dust in millimeter dimensions [41, 54], capable of forming a sensor grid in a hostile environment like a battlefield, or behind enemy lines.

- Wireless networks for notebooks, laptops and PDAs, like the free WLAN access points (hot-spots) in the ”Bermuda Dreieck Bochum” and in front of the Audimaximum of the ”Ruhr-Universität Bochum” [11] are becoming more and more popular. The access points are configured in infrastructure mode—they act as routers to the attached LAN for the wireless computers. The ad-hoc nature of such hot-spots is only marginal visible in missing patch cables.

1.2. Definition

One of the main aspects of ad-hoc networks is their diversity. Some can consist entirely out of mobile nodes, with no infrastructure what so ever; others may have limited access to the Internet. But all ad-hoc networks have in common that they are self-configuring and are forming an arbitrary topology, which changes randomly and unpredictably due to the mobility of their devices (nodes). The name MANET (Mobile Ad-hoc Network) has become customary for such networks. A MANET has the following characteristics.

diversity: In most MANETs the nodes are different, though homogeneous MANETs are also possible (especially sensor grids are more or less homogeneous).

mobility: The nodes of the MANET are more or less mobile, therefore changing the topology of the network. They also connect or disconnect at will (of the user).

energy constraint: Most devices would be battery driven to endorse the mobility of the node.

restricted computing power: Compared to a desktop PC, most mobile devices are restricted in their computing power. Their CPUs are less powerful, due to the energy constraints their batteries force upon them.

limited communication bandwidth and range: Due to power constraints and avoidance of overlapping and interfering channels the communication bandwidth and its range are normally restricted.

multi- vs. single-hop connections: One of the major problems in MANETs is routing in multi-hop networks.

security: Security is a major issue in MANETs because ad-hoc networks offer an attacker more working points, than an infrastructure network. Due to the open accessibility of ad-hoc networks, adversaries can easier control nodes, manipulate and redirect data.

Each communication infrastructure has some common security requirements. In common literature they are described by the acronym CIA which stands for confidentiality, integrity and authentication. These are the three major security issues in each communication:

confidentiality or privacy: The communication must not be accessible by others. This is normally solved by encrypting the communication.

integrity: The communication must not be undetectable altered with. This can be solved by using a MAC (Message Authentication Code).

authentication: The communication partners must be sure that the others are who they claim to be. This can be done using various different methods all of them using trust—to a single person or to an institution (trusted third party)—as a base.

When commercial applications entered the Internet, non-repudiation and availability became strong security issues.

non-repudiation: The communication partners must be unable to deny ownership of a message, once it was send.

availability: The communication platform must be resistance against Denial-of-Service attacks. New attacks like the sleep deprivation torture—where a device is detained from going into sleep mode (energy saving mode)—make MANETs more vulnerable to Denial-of-Service attacks, than traditional wired networks.

In a MANET additional security issues must be considered.

robustness of the communication: Not only resistance against normal packet loss, but especially resistance against mischievous disruption, like signal yamming. Spread spectrum techniques belong to the countermeasures. Availability is a part of robustness.

Together a new acronym can be formed: CRAIN consisting of the major key words confidentiality, robustness, authenticity, integrity and non-repudiation.

1.3. Motivation and Outline

Due to their economical potential and vast diversity ad-hoc networks are very interesting for researchers. Several industry projects and research groups around the world are working on different aspects of ad-hoc networking. Some are focusing on routing in multi-hop environments, others are working on adapting ad-hoc characteristics to existing technologies like WLAN, Bluetooth and IrDA.

The scope of this diploma thesis is twofold. First to write a survey of security related issues in ad-hoc networks, and second to implement a tool that should overcome security flaws which were found during the survey. The tool eventually became *simahnsai*. *simahnsai* can be used as a secure instant messenger, which fulfills confidentiality and integrity of the communication.

As mentioned this document tries to survey security related aspects of ad-hoc networks. This includes the wireless communication protocols WLAN (IEEE 802.11), Bluetooth and IrDA in Chapter 2 as well as multi-hop routing protocols in Chapter 3, while Chapter 4 reviews some existing papers and implementations of ad-hoc networks. Chapter 5 describes the implementation *simahnsai* and Chapter 6 summarizes and concludes this paper.

2. Fundamentals of MANETs (Wireless Communication Protocols)

Because ad-hoc networks are by definition mostly wireless, a way of communication must be found. Currently four major wireless protocols exists which are capable of serving as a fundamental basis for an ad-hoc network. These protocols can be divided into two different categories. WLAN (Wireless LAN) and HiperLAN (High performance radio LAN) are designed to extent an existing (traditional/wired) local network to mobile devices, whereas Bluetooth and IrDA (Infrared Data Association) were primarily designed as a cable replacement for computer peripherals, like wireless mice and keyboards. With rising popularity of wireless computing, IrDA was extended to include a network access. Bluetooth was designed from the beginning as a possible wireless network access replacement.

Table 2.1 shows that WLAN, HiperLAN and IrDA offer just a standard MAC (Media Access Control) interface for higher-layer applications and protocols, whereas Bluetooth supports all layers of the ISO/OSI model . This document focuses on WLAN and Bluetooth, as the two major wireless protocols on the market. WLAN being the market leader and Bluetooth gaining market shares especially with small mobile devices like PDAs, cell phones and digital cameras.

ISO/OSI reference	TCP/IP	WLAN	HiperLAN	IrDA	Bluetooth
Application	Application				x
Presentation					x
Session					x
Transport	Transport				x
Network	Internet				x
Data Link (MAC)	Network Access	x[1]	x	x	x
Physical		x	x	x	x

Table 2.1.: ISO/OSI reference model

[1] x means the the ISO/OSI layer is implemented in the appropriate protocol

IrDA and its three available protocols IrNET, IrLAN and IrCOMM are shortly introduced. HiperLAN has been left out because of its lack of commercial success.

2.1. WLAN (IEEE 802.11)

The IEEE (Institute of Electrical and Electronics Engineers, Inc.) standard 802.11 or WLAN was published 1999 as an extension of existing 802.x networks[2] for mobile devices, i.e., in warehouses and manufacturing. But due to their cost and time saving nature, WLANs are replacing more and more wired LAN (Local Area Network)s in enterprises and private households. A single WLAN card replaces the traditional network access card as well as all the necessary patch-cables. A temporary network, i.e., for testing purposes can easily be set up with WLAN. There's no need to wire every room in a house to achieve full Internet accessibility. And additions, moves, and changes within an organization require no changing of a wired infrastructure, when WLAN cards are used.

2.1.1. The IEEE Standards

802.11 specifies data rates of 1 and 2 Mbit/s via infrared or in the 2.4 GHz radio band using FHSS (Frequency Hopping Spread Spectrum) or DSSS (Direct Sequence Spread Spectrum)[3]. The IEEE supplement standard 802.11b (Wi-Fi (Wireless Fidelity)™) increases the data rates through different coding to 5.5 and 11 Mbit/s on the 2.4 GHz band using only DSSS. 802.11b is currently the industry standard. The coming supplements 802.11a (5 GHz band) and 802.11g (2.4 GHz band) will quintuple the data rates up to 54 Mbit/s resulting in net rates of about 2.5 to 3 MBytes/s.

[2] The following IEEE 802 standards exist at the time of writing:

- 802.1: LAN/MAN Bridging & Management
- 802.2: Logic Link Control
- 802.3: CSMA/CD (Carrier Sense Multiple Access/Collision Detection) Access Method (Ethernet)
- 802.5: Token Ring Access Method
- 802.10: LAN/MAN Security
- 802.11: Wireless LAN
- 802.12: Demand Priority Access Methods
- 802.15: Wireless Personal Area Networks (Bluetooth)
- 802.16: Broadband Wireless Metropolitan Area Networks (Wireless MAN)

[3] Spread spectrum techniques are used to reduce interference through overlapping transmitters.

2.1.2. Network Topology

Each WLAN component which is either a station or an access point, requires a transceiver and an antenna. Stations (STAs) are the nodes of the wireless network. An AP (Access Point) forms a bridge between wired and wireless LAN. When two or more stations recognize each other, they can form a so called BSS (Basic Service Set). A normal client/server relationship is formed, when several stations connect to an AP which acts as bridge to the wired infrastructure. This is therefore called infrastructure mode. If no access point is available, the stations can connect to each other in an ad-hoc mode (peer-to-peer). This is also referred to as IBSS (Independent Basic Service Set). When several BSS overlap—in means of range and topology—it's called an ESS (Extended Service Set). The SSID (Service Set Identity) is the name of the WLAN.

2.1.3. Security

WLAN's security relies entirely on RC4. RC4 is used as PRNG (Pseudo Random Number Generator) in the 'Shared Key' authentication and WEP (Wireless Equivalent Privacy) algorithm. WEP was designed to supply a WLAN with the same level of security as a traditional (wired) LAN (see 2.1.3.1 for more details).

RC4 (Ron's Code 4 or Rivest's Cipher 4) is a byte oriented stream cipher with variable key-size, which was designed by Ron Rivest in 1987 for RSA Data Security[4]. It was kept as a trade secret until 1994 when the code eventually leaked out. Because of its trademarked status an open alternative called Arcfour was made. RC4 (incl. Arcfour) is todays most widely used stream cipher in software implementations (it is used in SSL (Secure Socket Layer) to encrypt web traffic on a session basis). One of the reasons for its widely use is its simplicity. For more details on RC4 see Appendix A.

2.1.3.1. WEP (Wireless Equivalent Privacy)

The WEP (Wireless Equivalent Privacy) algorithm is the cornerstone of WLAN security. It was designed to be reasonably strong, self-synchronizing, efficient and exportable to other countries, regarding to U.S. Department of Commerce regulations. The IEEE standard 802.11 regards the implementation and use of WEP as an option for WLAN devices.

The encryption is done in the following manner (compare Figure 2.1):

- the ICV (Integrity Check Value)—a 32 bit checksum using CRC-32 (Cyclic Redundancy Checksum)—is calculated over the plaintext. The number (N) of octets of the plaintext is not specified in the standard.

[4] now RSA Security Inc.

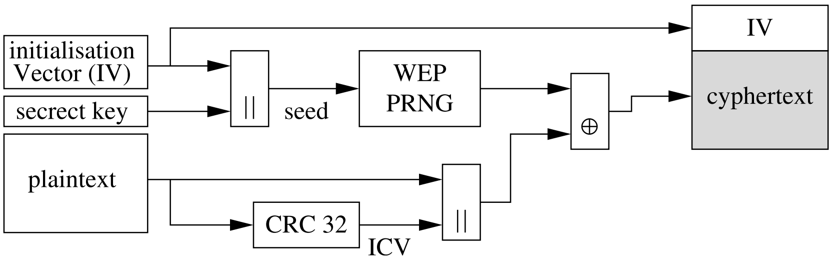


Figure 2.1.: WEP encryption algorithm

- plaintext and ICV are being concatenated resulting in a stream of $N + 4$ octets
- the 24 bit IV (Initial Vector) and the 40 (WEP40) or 104 (WEP128) bit secret key are being concatenated to a 64 or 128 bit seed for the WEP PRNG (RC4 PRNG), where the bits 0 to 23 of the IV correspond to the bits 0 to 23 of the PRNG seed
- the PRNG generates $N+4$ pseudorandom octets which are bytewise XORed with the plaintext and ICV stream resulting in the ciphertext.
- ciphertext and the plain IV are put together into the data frame which can now be send to the addressee (Figure 2.4)

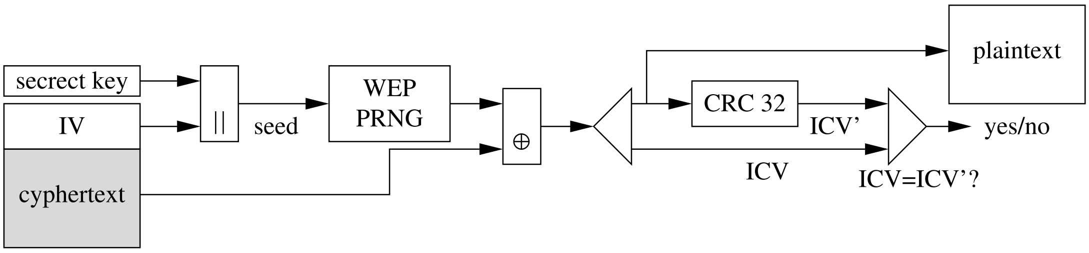


Figure 2.2.: WEP decryption algorithm

The decryption of the received data frame is done in the following manner (compare Figure 2.2):

- the secret key and the IV are concatenated to the seed of the WEP PRNG, which generates $N + 4$ pseudorandom octets
- the ciphertext and the generated octets are bytewise XORed resulting in the plaintext and the ICV

- the ICV is stripped off and the checksum ICV' over the plaintext is being calculated

- ICV and ICV' are being compared. If ICV and ICV' are not equal an error message is send to the MAC management layer.

2.1.3.2. Authentication

WLAN stations and access points can be run in two authentication modes. The protocol standard 'Open System' authentication accepts all clients without authentication. The client has to request an 'Open System' authentication at his access point. When the access point is configured to accept those requests (that is the default setting), the client gains full access to the network. The 'Shared Key' authentication mode is an optional challenge-response protocol where the shared secret key is verified in the following way (compare Figure 2.3):

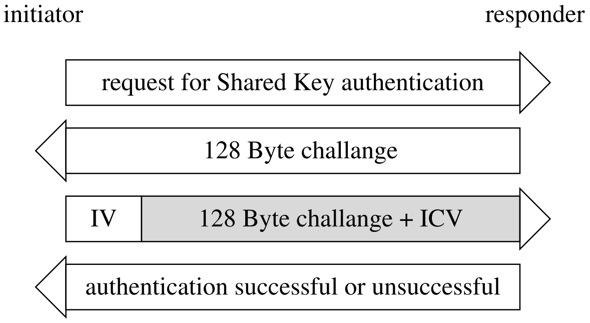

initiator
responder
request for Shared Key authentication
128 Byte challange
IV
128 Byte challange + ICV
authentication successful or unsuccessful

Figure 2.3.: 'Shared Key' authentication

1. the initiator sends a request for 'Shared Key' authentication to the responder

2. the responder generates a 128 octet pseudo random challenge using the PRNG of RC4 (Ron's Code 4 or Rivest's Cipher 4)/WEP. The chosen secret key and IV that act as seed for the PRNG are not specified by the standard

3. the initiator copies the challenge into the return-frame and encrypts the frame using the WEP encryption algorithm with his standard encryption key

4. the responder decrypts the return frame using the WEP decryption algorithm

- if the frame decrypts successfully he compares the received challenge with the one he sent in frame 2 and sends the initiator an authentication successful message if both are identical. If the check fails the initiator gets an authentication unsuccessful message.

Lucent Technologies has defined a proprietary authentication method called *Closed Network Access Control*. A network manager can either choose an open or a closed network. When open, the network accepts anyone, when closed only those clients are accepted who know the network ID (SSID).

2.1.3.3. Key management

The key management is vital to the security of WEP. Unfortunately, key management is not part of the standard. The standard only specifies that the secret keys should be transfered to the STA (Station)s or APs via an IEEE 802.11 independent path and that the IVs should be changed with every data frame (this is intended to work as salt, but we will see later on, that it does not work), resulting in a quasi code-book mode with a new key every frame, instead of one key for all frames. Every STA/AP can hold up to four system wide keys, but only one of these is used for encryption by each specific STA/AP. The ID number of the used encryption key is transmitted with the data frame (Figure 2.4) . An optional mode is a keyring with a single key for each communication partner the STA/AP may have. The number of those storageable keys is only limited by the amount of memory the STA/AP has.

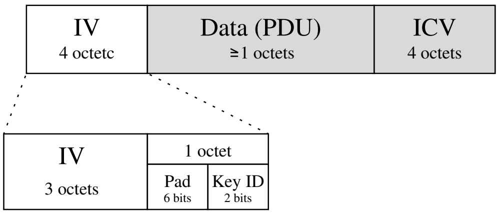


Figure 2.4.: Expanded WEP frame body

2.1.4. Security Problems

After WLAN became popular, several security holes became apparent. The authentication process of WLAN was flawed from the beginning and through a design error, a way was offered to break WEP. One reason, why those security holes remained undetected for a long time in the standards, can be found in the publishing policy of the IEEE. The IEEE offers its standards only to its members free of charge. Such that not many researchers felt obliged to pay money for a security review of the 802.11 standard. Only after WLAN became a commercial success, researchers focused on the 802.11 standards and found those security holes. The IEEE has learned from this and is currently allowing access to the 802.x standards free of charge, six months after they have been first published.

2.1.4.1. Flaws in WEP

WEP does not achieve the goals it was designed for. It has several flaws which undermine its intention to protect the wireless communication of stations and access points in a WLAN.

One major flaw of WEP is in its use of IVs. As mentioned earlier the IV is a 24 bit initialization vector which is appended to the secret key in order to form a family of 2^{24} keys. It would be fatal to reuse any key more than once, because a known-plaintext attack can be launched:

- assume $k_i = p_i \oplus c_i$ for $i = 1, 2, 3, \ldots, n$ where c_i represents the ciphertext bits, p_i represents the plaintext bits and k_i represents the keystream bits 1 to n
- if the plaintext or part of it is known, an adversary can simply recreate the keystream, as shown with the first 132 byte (128 byte challenge + 4 byte ICV) in the forged 'Shared Key' authentication on page 13.
- the intercepted keystream can now be used to decrypt any further communication which uses the same keystream.

It is no problem for the adversary to guess the plaintext in a typically installed TCP/IP (Transport Control Protocol/Internet Protocol)-infrastructure, because most of the TCP/IP headers are well known. Another way for the adversary to get a plaintext-ciphertext tuple is to simply send a known text, i.e., a large email to a member of the wireless network.

The probability of reusing a randomly chosen IV more than once—called collision—is over 50% when the adversary has intercepted $4823 \approx 2^{12}$ frames. This is due to the birthday paradox (see [62, page 4]). On a normal 11 Mbit/s AP the probability of a collision is over 99% (12430 frames) after only 2-3 seconds of normal traffic! If the stations just increment the IV every time they use one, they use the hole 24 bit spectrum, but unfortunately most stations reset the IV to 0 when they power up.

Using this technique the adversary can collect for every base key and for nearly every IV a code book containing the keystreams of the ”protected” WLAN. Assuming that an adversary needs 1500 octets[5] for every IV. A standard harddisk[6] is all he needs to decrypt the complete WLAN in realtime.

[5] a standard IP frame has normally less octets

[6] $2^{24} * 1500$ Bytes = 24 GB

Another problem of WEP is its lack of integrity, that means the receiver of a frame cannot be sure that the frame has been altered with. This is because WEP is linear. RC4 as well as CRC-32 are linear functions, such that: RC4$(k, x \oplus y) =$ RC4$(k, x) \oplus y$ and CRC-32$(x \oplus y) =$ CRC-32$(x) \oplus y$. An adversary can deliberately switch bits in a frame:

- he generates a pattern with 1s on the appropriate positions for the bits he wishes to either switch (XOR) or set (OR)
- he generates the CRC-32 checksum over the pattern and concatenates both together
- he XORs or ORs the frame with the pattern/checksum

Because the frames normally contain TCP/IP traffic, which uses a non-linear checksum the adversary can only guess the appropriate bit (50 % chance of success).

2.1.4.2. WEP Broken

Scott Fluhrer, Itsik Mantin and Adi Shamir found a weakness in the key scheduling algorithm of RC4 [31] in August 2001. They found out that RC4 is vulnerable to a large number of weak keys (about nine thousand out of 16 million possible IVs for WEP128), where the knowledge of some key bits suffices to determine many output bits with a non-negligible probability (every weak IV has a five percent chance of exposing a corresponding key byte). They pointed out that WEP with its use of IV may be highly vulnerable to this kind of attack. Some time later Adam Stubblefield, John Ioannidis and Aviel D. Rubin published a paper where they claimed a successful attack against WEP with full shared key recovery [60] using the revealed weakness in RC4. Stubblefield, Ioannidis and Rubin did not publish their tool, so several OpenSource proof-of-concept tools were written and published, two of them being AirSnort [2] and WEPCrack [7]. Both tools can be used by so-called wardrivers [6] to hack and break into a WLAN, or by an administrator to check, whether his safety precautions are working or not.

Dominik Blunk and Alain Girardet have implemented a proof-of-concept from Tim Newsham [48] in their diploma thesis [22]. Their tool WepAttack combines the features of a password cracker and a WLAN-sniffer. The WLAN-sniffer gathers WLAN frames which are attacked by the password cracker part. The password cracker tries several (hundred-) thousands of passwords which are derived from dictionary words [10]. If the decrypted ICV (see 2.1.3.1) is positively checked against the catched ICV, the password has been found. Very weak passwords (for example names) can be found with WepAttack in fractions of a second.

2.1.4.3. Authentication holes

The 'Open System' authentication represents—despite its name—no authentication whatsoever. The proprietary *Closed Network Access Control* from Lucent is flawed, because the needed SSID in the "closed network" authentication is transmitted in the clear on several management frames. So a simple "sniffer" is everything an adversary needs to get the SSID for accessing the network. The 'Shared Key' authentication can be broken too. [14] shows that an adversary can simply forge an authentication based on intercepted frames:

1. the adversary eavesdrops a 'Shared Key' authentication session.
2. he extracts the cleartext challenge from step 2 of the authentication protocol (see 2.1.3.2) and the ciphertext, IV and the keyID of step 3
   - he generates the ICV (simple CRC-32) and concatenates it to the challenge
   - he recreates the used keystream by XORing the challenge and the ICV with the ciphertext
3. he requests a 'Shared Key' authentication at the same AP he's eavesdropping
   - he generates the ICV for the received challenge and XORs the recreated keystream with his challenge and the ICV
   - he sets the keyID and IV[7] appropriately to the intercepted values before sending his frame
4. he is authenticated

2.1.5. WEP2

The 802.11i working group is currently working on WEP2 which is a fix of WEP. In May 2001 during a meeting of the full working group, a paper introduced by Bernard Aboba [12], showed that WEP2 is still vulnerable to many of the above mentioned flaws and also new ones (introduced through Kerberos).

- WEP2 does increase the IV key space to 128 bits, but it fails to prevent IV replay exploits and still permits IV key reuse; although IV key reuse is more improbable due to the extended key size.
- That same IV replay weakness of WEP, combined with a faked MAC address, also permits an attacker to forge authentication in WEP2.

[7] the adversary must not be uneasy about using the same IV again, see 2.1.4.1 for more details

- Known plaintext exploits—where the intruder knows or can guess part or all of the data payload or encrypted header contents and uses that information, and the IV key and CRC32 to crack the encryption itself—work as well with WEP2 as they did with WEP.

- The inclusion of mandatory Kerberos V support merely opens WEP2 to new dictionary-based attacks. (Aboba estimates that up to 10% of Kerberos-"protected" user passwords can be cracked within 24 hours, using an inexpensive network of PCs running parallel DES cracking techniques.)

- Because reassociate and disassociate messages are not secure, WEP2 is vulnerable to DoS attacks where an attacker floods the WLAN with those messages to disrupt connections.

The working group has neglected to begin from scratch with a new encryption and authentication algorithm, instead they choosed to fix WEP2. Until the release of this diploma thesis, WEP2 was still not finalized.

2.1.6. Conclusion

WEP—planed as a security feature for wireless 802.11 LANs—is a failure. Even worse, through its pretended security it is a danger to those unaware of its security holes and lulls them in security while as good as none is given. Through its unfortunate design, WEP can be broken with relatively low effort. A passive attack—the adversary only gathers data frames for this attack—with WepAttack can determine the key within seconds, if the user was careless enough or too lazy to choose a strong password. And even if the user chooses a strong password, AirSnort or WEPCrack can break a WEP key within hours or days, depending on the amount of network traffic. Regular changing of the password is therefore mandatory for the operation of a WLAN.

The authentication protocol of 802.11 is non existent. The *Open System* authentication is not worth its name and the *Shared Key* authentication is flawed. Sending both, cleartext and ciphertext of an XOR encrypted authentication system over the insecure network, is comparable to engaging a blind doorman. The optional use of Kerberos in WEP2 offers far more security, if strong passwords are used. But the additionally security is bought off with the necessity of a Kerberos server.

A WLAN must therefore be considered and dealt with like any insecure connection to the Internet. A safe way to use a WLAN is the use of a secure tunnel, like ssh, SSL, IPsec or VPN and appropriate authentication mechanisms.

2.2. Bluetooth, IEEE 802.15 and WPAN

In 1994 Ericson Mobile Computing investigated the feasibility of a low-power, low-cost radio interface for its cell phones and accessories. On May 1998 the Bluetooth[8] SIG (Special Interest Group) was formed. Its five founding members were Ericson, Nokia, Toshiba, IBM and Intel[9].

Originally developed to replace IrDA as a short range telecommunication protocol, it became obvious that IrDA was no match for Bluetooth. Because WLAN was developed around the same time and uses the same frequencies as Bluetooth, the media pushed WLAN as main competitor for Bluetooth although both are aiming at different markets. WLAN aimed at extending an existing LAN with wireless capabilities, while Bluetooth was designed as a person-based wireless interface, used within the persons operating space (POS). The acronym PAN (Personal Area Network) became apparent. The IEEE began working on the 802.15 standard, which is in fact a PHY and MAC layer description of the current Bluetooth standard (see figure 2.5 on the following page). It was done to make it possible to integrate Bluetooth into existing 802.X networks. The IEEE also created the trademarked acronym Wireless Personal Area Network™ (WPAN) for devices which uses the 802.15 standard.

The current version 1.1 of Bluetooth will be followed by version 1.2 in autumn 2003, which will be downward compatible to its predecessor. The major changes will be the adaption of Adaptive Frequency Hopping to increase the compatibility of Bluetooth with other 2.4 GHz devices.

2.2.1. Protocol Stack

The Bluetooth protocol stack is depicted in Figure 2.5. It shows the standard ISO/OSI reference model with the appropriate TCP/IP layers and the corresponding Bluetooth protocol stack. Bluetooth supports all layers of the ISO/OSI reference model, as stated earlier in Section 2.1. The radio and parts of the Link Controller/Baseband layer correspond to the physical layer of the ISO/OSI model, whereas the rest of the LC/Baseband and the Link Manager as well as the Link Level Control and Adaption Protocol belong to the Data Link layer of the ISO/OSI model. RFCOMM, SDP and Audio as well as PPP over Bluetooth

[8] The name Bluetooth and its logo are inspired by the danish king Harald Blåtand who united and Christianized Denmark and parts of Norway in the 10th century. Though Blåtand is literally translated to *bluetooth* the viking king had no blue teeth. *Blå* referred to his dark skin and hair while *tan* means great man. The Bluetooth logo which was introduced May 17th 2000 symbolizes the old Scandinavian runes H and B which stand of course for Harald Blåtand.

[9] The five founding members wanted Bluetooth to unite the world of computing and telecommunication as their viking eponym united Denmark and Norway. Currently over two thousand members of the SIG are developing new bluetooth enabled devices, standards, specifications and technologies to fulfill the vision.

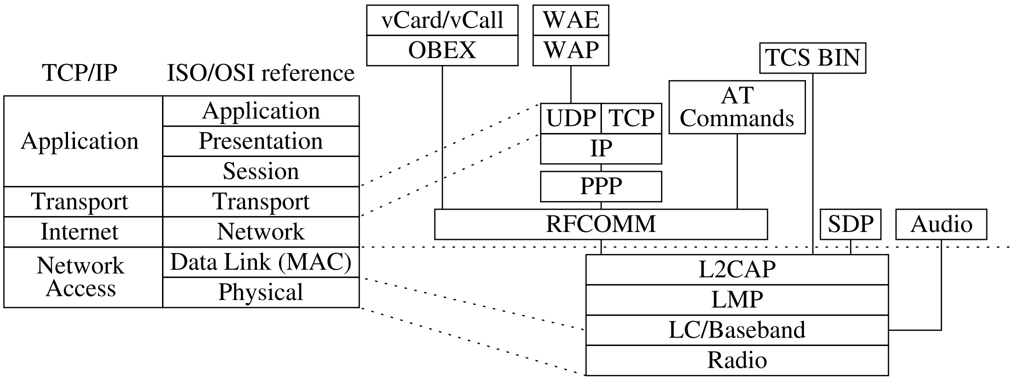


Figure 2.5.: Bluetooth protocol stack

and IP belong to the Network layer and the UDP/TCP layer obviously refers to the Transport layer of the reference model. The higher layer protocols like WAP and OBEX correspond to the Application layer.

2.2.1.1. Radio, LC (Link Controller) and Baseband

The Radio protocol of Bluetooth [21, Part A] uses the same free ISM band (2.4 GHz) as WLAN does. The Radio Specification of the Bluetooth Core Specification defines the frequency-band for each region/country (France and Spain offer only a limited set of frequencies) as well as the power output of the transceiver.

The LC (Link Controller) specification [21, Part B] covers baseband protocols like modulation, duplexing, channels, packet transmission and link-level security. It uses the frequency hop spread spectrum technique with up to 1.600 hops per second on 79 channels (in Spain and France only 23) to avoid collisions and interference in the ISM band. This results in time-slots of 625μs with a data rate of 1Mb/s. For full duplex transmission a TDD (Time-Division Duplex) scheme is used. Data is transmitted in packets; after each packet the channel is switched referring to the hopping sequence, which is deterministically derived from the masters unique BD (Bluetooth device) address. A packet normally covers one time-slot (625μs), but can be extended up to five slots. Bluetooth supports either one asynchronous data channel (max. 723.2 kb/s forth and 57.6 kb/s back), three synchronous voice channels (each with 64 kb/s in both directions), or one channel with both asynchronous data and synchronous voice. Synchronous voice channels use QoS (Quality of Service) to achieve nondisturbed communication.

Contrary to WLAN, Bluetooth knows only master-slave relationships. A master device can have up to seven active slaves and many more[10] inactive (parked) slaves. A master and his slaves form a so called piconet. A device can be in multiple piconets (because the hopping-sequence is derived from the masters

[10] up to 255 parked slaves and in special cases even more

BD_ADDR (Bluetooth Device Address) a device can only be master in one piconet, but slave in multiple piconets). This is called a scatter net. A device which is present in multiple piconets must time-share and synchronize to the master with which it is currently communicating. The slaves talk only to their master device. Because BDs are normally battery powered, four power saving modes as well as power control for the radio have been defined.

active: Only seven active slaves can connect to a piconet master.

sniff: A slave that is in sniff mode has entered power saving mode. That means the slave listens at reduced rates to the traffic in the piconet. It is the least efficient power saving mode.

hold: A device in hold mode is in the intermediate power saving mode. It listens less occasional to the piconet than in sniff mode, but more than in parked mode.

parked: During parking the slave is still synchronized to the piconet, but listens only occasionally for broadcast messages. Up to 255 parked slaves can participate in a piconet. Parked devices are in the most efficient power saving mode.

For link-layer security four different security items are used:

1. public unique address for each BD, called BD_ADDR (48 bit)
2. secret key used for authentication, called authentication or link key (128 bit)
3. secret key used for encryption, called encryption key (8-128 bit; bytewise)
4. random number, called RAND (128 bit)

The Bluetooth device address (BD_ADDR) is a worldwide unique 48 bit address compliant to the 48 bit IEEE address. It is used to identify the device and to generate the hopping sequence, the CAC (Channel Access Code) and DAC (Device Access Code), if the device is master of a piconet. The link key is generated during initialization of two BDs (during the pairing process) and remains under normal circumstances[11] static for the duration of their relationship. A new encryption key is derived from the link key and a random number everytime encryption is needed.

[11] applications may require new link keys for safety reasons

2.2.1.2. LMP (Link Manager Protocol) and L2CAP (Logical Link Control and Adaption Protocol)

The LM (Link Manager) [21, Part C] is managing the links. Two kinds of links have been defined:

- A SCO (Synchronous Connection-Oriented) link, which is a point-to-point link between a master and a slave. The master reserves time-slots in regular intervals for this link.
- An ACL (Asynchronous Connection-Less) link, which is a point-to-multipoint link, between the master and all of his slaves. Slots which are not reserved for an SCO link may be used for an ACL link on a per-slot basis.

The LM provides link set-up and control. The LMP messages are filtered out and interpreted by the LM. These messages are not passed on to the higher layers.

The Logical Link Control and Adaption Protocol [21, Part D] supports multiplexing (interleaving) of higher layer protocols including packet segmentation and reassambly (SAR) as well as the conveying of QoS (Quality of Service) information. L2CAP resides together with the LMP in the Data Link layer of the ISO/OSI reference model. L2CAP is only defined for Baseband ACL links; SCO links (voice) are not supported.

2.2.1.3. Higher Level Protocols in Bluetooth

RFCOMM [21, Part F:1] is a serial cable emulation protocol, adapted from the ETSI (European Telecommunications Standards Institute) standard TS 07.10. Bluetooth supports only a subset of TS 07.10 and has adapted some parts of the protocol. The Bluetooth RFCOMM protocol supports up to 60 simultaneous connections between two BDs. The actual number of possible connections is implementation specific. RFCOMM is used over L2CAP.

OBEX (Object Exchange) [21, Part F:2] or better IrOBEX (Infrared Object Exchange) is part of the IrDA protocol. Bluetooth utilizes the OBEX protocol to exchange objects—like vCards—in the liking of the IrDA stack. Higher layer applications can either use Bluetooth OBEX or IrDA OBEX. OBEX runs on top of RFCOMM.

TCS (Telephony Control protocol Specification) protocol [21, Part F:3] defines the calling signals for establishing speech and data calls used, i.e., by headsets for mobile phones.

As Bluetooth devices share many characteristics with WAP (Wireless Application Protocol) devices, it is only consequential to combine them. PPP over Bluetooth [21, Part F:4] can therefore be used as communication bearer for the WAP and TCP/IP protocol.

SDP (Service Discovery Protocol) [21, Part E] is one of the most important protocols in Bluetooth. Contrary to the above mentioned protocols (RFCOMM, OBEX, TCS and PPP over Bluetooth) SDP is mandatory for a working BD. During an inquiry a BD uses the SDP to find out what services (service profiles) the other BDs offers.

2.2.2. Security

Because Bluetooth has a vast variety of possible applications, the security mechanisms of it must be as flexible as its applications. Furthermore, a transparent security infrastructure is in the mere interest of the designers, to heighten the usability for the device owners. Bluetooth is therefore applying a flexible security architecture, with three modes of security. Mode 1 (non-secure) has no security enabled. In mode 2 (service-level enforced security) security mechanisms will be enforced after a link has been established. In this mode very flexible and different access policies for the applications are allowed. Mode 3 (link level enforced security) requires security mechanism initiation before the link is established. The actual security mechanisms are not part of the Bluetooth specification [21], but sourced out into two whitepapers [19, 20] that are intended to guide implementors.

Birth	unit key
inquiry	non-discoverable limited discoverable general discoverable
Paging	non-connectable connectable
Master/Slave	non-pairable pairable
Pairing	initialization key
Authentication	authentication/link key
Encryption	encryption key

Table 2.2.: BD states

Table 2.2 shows the different states a BD can assume.

- A unit key is generated, when a BD is for the first time in operation. This unit key is almost never changed.

- During inquiry a BD searches for other devices using an inquiry hopping sequence. Whether the other devices answer the inquiry depends on their mode. In non-discoverable mode they do not answer, in limited discoverable mode they answer only to inquiries using a manually preset inquiry hopping sequence, whereas in general discoverable mode they answer to all inquiries.

- When a device is discovered, a connection can be established (called paging), but only when the device is in connectable mode. During paging the initiator—which becomes the master—uses the hopping sequence of the slave to establish the connection. When the connection is established, the master's hopping sequence is used.

- Before BDs can communicate they must pair. If a BD is in non-pairable mode, pairing cannot be accomplished. During pairing an initialization key is generated, which is used for authentication. This initialization key is derived out of the BD_ADDR, two random numbers from both participants and a shared secret PIN (Personal Identification Number). The PIN is either entered via keypad into both devices, or when one device has not have a keypad the fixed PIN of the device is used. When both devices lack a keypad, that means both devices have a fixed PIN, they cannot pair at all.

- There are three possible ways to generate an authentication or link key. The first two versions use the initialization key to encrypt the transmissions during the protocol.

  - The first link key generation protocol is requested when a device has a shortage of memory to store the key. The unit key of the requesting device is used as link key.
  - In the second link key generation protocol both devices generate a random number, which is combined with the devices unique device address (BD_ADDR). Both numbers are exchanged and then XORed to generate the link key.
  - The third version is only used in point-to-multipoint connections. It uses a so called master key, which is generated by the master and is transmitted using a previously established link key.

- For authentication a challenge-response protocol between the claimant and the verifier is used. The verifier sends the claimant a random number. The claimant uses the random number and his own BD_ADDR as well as the shared link key to generate the 32 bit answer for the verifier. The verifier makes the same calculations and checks whether the answer is correct or not. During this calculation a 96 bit authenticated cipher offset is generated, which is stored for future use during an encryption. For mutual authentication the roles of the verifier and the claimant are switched.

- Encryption is available as soon as at least one communication device has been authenticated. After both devices have agreed on the key length, the master sends his slave a random number which is used in combination with the link key and the cipher offset to generate the encryption key. For point-to-point encryption the authenticated cipher offset is used, for point-to-multipoint encryption the masters BD_ADDR is used as cipher offset. For the latter the master key is used instead of the link key.

The actual encryption is done using the so called stream cipher E0. For every data packet an initialization vector is used, which is calculated from the masters BD_ADDR and his clock.

There is no real end-to-end encryption, because encryption only concerns the lower layers. That means applications must provide a way for end-to-end encryption.

Security must be considered for every state. A BD which is in non-discoverable mode can not pair at all and is limited to its own resources. The limited discoverable mode is therefore a good choice for devices which carry sensitive data. The device answers only to those inquires with the right inquiry hopping sequence. Another security risk in discoverable mode is the possibility of tracking the device and its owner. Because the BD_ADDR is transmitted during an inquiry an adversary can track a BD by using a grid of devices under his control which continuously send out inquiries.

The major security risk is during pairing. If the devices pair in an insecure environment—that means when they can be eavesdropped—an adversary can try to calculate the link key of both devices, or launch a man-in-the-middle attack. The use of a PIN is critical, because humans tend to use weak PINs. [24, section 3.2] shows a good margin for user-selected PINs.

Another problem occurs when a device must use its unit key as link key, because all previously connected devices can eavesdrop the communication. A similar situation occurs when an adversary can get control over a slave in a point-to-multipoint piconet; he can eavesdrop all the communication within that piconet.

A theoretical weakness has been found in the used encryption algorithm. [30] has shown that the encryption algorithm E0 has in effect a limited key length of 73 respectively 84 bits even when used with 128 bits. This reduces the strength of the encryption algorithm, but has no practical effect in the cryptographic strength and therefore the security of Bluetooth communications.

2.2.3. Conclusion

The security system of Bluetooth is more thought through than that of 802.11. Bluetooth uses a flexible, adaptive security policy, which is highly configurable. Whether or not a Bluetooth device is ”visible” to other devices depends on the settings the user chooses. That’s also a drawback of Bluetooth, because as so often, the additional security has been bought off with less userfriendliness. The relatively complicated pairing process of Bluetooth devices—which is mandatory for an encrypted connection—scares away many technically not so educated users.

2.3. IrNET, IrLAN & IrCOMM

There are three different possible ways to use TCP/IP over IrDA (Infrared Data Association). See figure 2.6 for a general overview of the IrDA stack. IrNET connects the lower layers of the IrDA stack directly with ppp (Point-to-Point Protocol). IrNET is used by Windows 2000 to connect two PCs as a direct cable connection over IrDA.

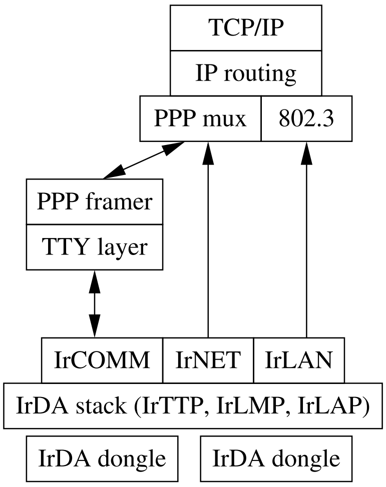


Figure 2.6.: IrDA-stack

Due to its nature IrCOMM (Serial and Prallel Port Emulation over IR (Wire Replacement)) uses the serial TTY-layer to access the ppp stack. IrCOMM is therefore performance wise slower than IrNET. Its serial line emulation is the reason why the IrCOMM protocol is used most commonly when ppp is carried over IrDA. Data-enabled mobile phones with IrDA use IrCOMMs pseudo serial port for communication.

IrLAN (Infrared LAN Access Extensions for Link Management Protocol) is the official protocol of the IrDA group for transporting TCP/IP over IrDA. IrLAN is basically an Ethernet (IEEE 802.3) emulation over an IrDA socket.

All three methods have in common that—due to the nature of infrared light—only direct connections using a line-of-sight can be established. That means that the user of an IrDA device has to explicitly point his/her device to another IrDA device to establish a connection. The slogan ”just point and click” has been a common allegory for that.

Another drawback of IrDA is its missing capability to establish true point-to-multipoint connections. Although IrLAP (Serial Infrared Link Access Protocol) supports such connections, its upper layers do not. IrLAN and IrNET devices which support point-to-multipoint connections use several IrDA devices to emulate such an behavior.

Security issues in IrDA include the possiblity of denial of service attacks by shielded infrared emitters. The three IrDA protocols do not include further security precautions, like encryption of links and authentication of IrDA devices.

2.4. Summary

For the implementation part of this diploma thesis, a decision had to be made between WLAN and Bluetooth, because the application should communicate with another instance of it on another PC. The use of IrDA has very early been denied, because of the need for a direct visible link between two devices and the absent support of multiple connections. WLAN with its numerous security holes would be advantageous for the demonstration of the necessity of high-level end-to-end encryption. But due to financial aspects, two Bluetooth USB dongles have been chosen.

3. Robust Routing

Robust routing is one of the major concerns in todays research regarding multihop mobile ad-hoc networks. Routing in those networks differs from the traditional, infrastructural routing, because of the highly mobile component. If a node moves the routing information must be updated to ensure communication. Since no routing infrastructure exists (there are no dedicated routers), each node has to become its own router. This burdens additionally computational effort to the mobile devices. In combination with the energy constraints this makes routing an interesting field of research.

What follows is a compilation of routing protocols proposed for mobile ad-hoc networks which is divided into four parts. The first two parts describe table driven and source-initiated on-demand routing protocols. The third part focuses on secure routing protocols, which ensure the authenticity of routing messages and part four presents cooperation based routing schemes.

As mentioned earlier routing is a major part in todays research on ad-hoc networks. It has therefore been dedicated a chapter in this diploma thesis, although the application simahnsai requires no routing, because of its purpose and design.

3.1. Table Driven Routing Protocols

In a traditional, infrastructural network, routes rarely change. The traditional routing protocols are therefore table-driven. Due to the mobile component in ad-hoc networks, table-driven routing protocols create an overhead to keep the tables up-to-date.

In DSDV (Destination-Sequenced Distance-Vector) [51] each node in the network maintains one table with all possible routing destinations, the number of hops to the destination and a sequence number, assigned by the destination. The sequence numbers enable the node to distinguish between new and obsolete routes. To keep the tables up-to-date, periodical full dumps are made, which contain all the necessary information for table generation. Incremental updates are made between full dumps to reduce the overhead.

CGSR (Clusterhead Gateway Switch Routing) [27] uses DSDV as underlying routing protocol and inherits therefore some of its drawbacks. A clusterhead (dedicated node) controls a group of ad-hoc nodes and serves as kind of routing server to his nodes (the cluster). Because of the clustering, routes are calculated using gateways (nodes which are members of multiple clusters).

3.2. Source-Initiated On-Demand Routing Protocols

Source-initiated on-demand routing protocols are the alternative to table-driven routing protocols. The nodes do not keep their tables up-to date. Instead they create routes only when they are needed. When a node requires a route to a destination, it initiates a route discovery process. Once the route has been established it will be kept alive using a route maintenance process. Source-initiated on-demand routing is ideal for changing network topologies.

AODV (Ad-hoc On-demand Distance Vector) [52] builds on the above mentioned DSDV. But instead of periodical update messages it uses a source-initiated approach. When a source node seeks a route to a destination node, it sends out a RREQ (Route Request) packet to its neighbors. They hand off the packet to their neighbors and so on until the packet receives the destination or a node with an up-to-date route to the destination. During forwarding the RREQ packet the intermediate nodes remember the neighbor of whom they received the packet, therefore generating a reverse path. Identical RREQ packets received by other neighbors (slower, other routes) are discarded. The destination or an intermediate node with up-to-date route to the destination will send out an RREP (Route Reply) packet along the reverse path. Nodes which receive such an RREP packet will setup a route forward entry in their routing tables which points to the node from which the RREP packet came.

Like AODV (Ad-hoc On-demand Distance Vector), DSR (Dynamic Source Routing) [40] is a source routing protocol. Each node has a route cache, which contains the nodes source routes. If a node wants to initialize a route, it checks whether the route already exists in its cache. If no route exists, it sends out a route request packet to its neighbors. The route request packets contain the destinations address, the address of the source and a unique identification number. Each intermediate node checks if it knows a route to the destination. If not, it adds its own address to the route record of the route request packet and hands it to another neighbor. If the packet has already been received by the node, that means its own address is in the route record (loop), or the packet has been received from another node (alternative, slower route), the packet is discarded. If the destination node or an intermediate node which has an up-to-date route to the destination receives the route request packet, it sends back a route reply packet which contains the route record. If symmetric links are supported the reverse path is used. If not, the destination will initiate a route request which contains the route record as additional payload. If a node recognizes a fatal error in its data link layer (a hop does not respond any more), it will generate a route error packet. If such a packet is received, the corresponding node of the error packet will be removed from the route cache and all routes will be truncated at the corresponding hop.

TORA (Temporally Ordered Routing Algorithm) [50] uses a three-dimensional graph called DAG (Direct Acyclic Graph) and a synchronized clock for route calculation. Each adjacent (one-hop) node gets a height metric; the source gets the highest, the destination the lowest metric. Each link between the nodes gets a direction (upstream or downstream), therefore a shortest route can be calculated using only the steepest downstream links. This also creates multiple routes for a source-destination pair. If a link breaks (route erasure) a new route can be calculated going upstream until an existing route to the destination is found. TORA is therefore qualified for highly mobile networks. One drawback of TORA is its requirement for a synchronized clock.

ABR (Associativity-Based Routing) [61] uses a new metric called degree of association stability, or associativity tick. A route is selected not for its shortest path, but for its likelihood not to change. Each node periodically sends out a beacon signal. For receiving such a beacon signal a node increments the associativity tick for the corresponding node. A high associativity tick indicates a stable connection over time and space, while a low associativity tick indicates a higher mobility of the node. If a node moves out of proximity its associativity tick will be reseted. For route discovery a source sends out a BQ (Broadcast Query) packet. Each intermediate node adds his address and his associativity ticks to the query, while erasing all associativity ticks of his predecessor except those ticks regarding himself and his upstream. The destination can therefore choose of all arriving BQ packets the most stable route. A reply packet is send along this route and all participating nodes mark their route valid. Route reconstruction may include partial route discovery, invalid route erasure, route update or a new route discovery. When a route is no longer needed a route deletion packet is broadcasted through the network, such that all nodes update their routing tables.

SSR (Signal Stability-Based Adaptive Routing) [29] favours routes with high signal strength. SSR can be divided into two cooperating protocols DRP (Dynamic Routing Protocol) and SRP (Static Routing Protocol). DRP maintains the signal stability and the routing table, where the signal strength of its neighboring nodes is recorded either as a weak or a strong channel. All packets are received by the DRP, which hands the packets after updating its tables to the SRP. SRP checks the routing table if a route to the destination is known. If no route is known, a route search is initiated. Such route requests are propagated throughout the network, but are only forwarded if the request was received over a strong channel and has not been replayed. The destination chooses the first arriving request packet and sends a route reply packet back the reverse path. The routing tables along the path are updated accordingly. If a route request does not reply within a timeout period, a flag will be set, that weak channels are also acceptable.

3.3. Secure Routing

Each of the above mentioned routing protocols is successful in dealing with the mobile component of MANETs, but they all do not consider security issues.

- All of the above mentioned protocols assume that all nodes are benign. But what can happen, if a node is in control of an adversary?
- The protocols assume that all nodes collaborate in the protocol. What should be done, if a node is selfish and does not (re-)transmit packets from other nodes, but only his own? How should one distinguish between a selfish node and a node with low battery reserves?

A malicious node can cause havoc in an unprotected network. It can alter routing information and therefore create loops, cause DoS (Denial of Service)-attacks by misleading routing packets and fabricate totally bogus route requests to drain the energy of mobile nodes. If a lot of malicious nodes work together, they can partition a MANET by positioning themselves in a way that they control the routing to or from a part of the network.

3.3.1. ARAN (Authenticated Routing for Ad-hoc Networks)

The ARAN (Authenticated Routing for Ad-hoc Networks) developers [55] propose a scheme that detects and protects against malicious actions of adversaries using authentication, message integrity and non-repudiation to secure routing information in a so-called *managed-open environment*. The authors differ three environments for MANETs. The *open*, *managed-open* and *managed-hostile* environment. The *open* environment exists entirely out of mobile nodes, there is no trusted third party or another way to pre-exchange initialization parameters between nodes. The *managed-open* environment allows this through a trusted third party (CA), or through manual exchange. Nodes in a *managed-hostile* environment are deployed from a single common source, like a military facility. The exchange of the parameters can therefore be done before deployment. The distinguishing difference of the *managed-hostile* and the *managed-open* environment is the physical threat of take-over and capture of these nodes. A routing protocol for the *managed-hostile* environment must therefore make sure that a node is not compromised or exposed through the protocol. ARAN satisfies only the requirements for the *managed-open* environment, because it requires the existence of a trusted third party. The *open* environment does not fulfill that and because ARAN exposes the network topology, it is not suited for the *managed-hostile* environment either.

ARAN uses cryptographic certificates to ensure security during route discovery. Each intermediate node involved in the route discovery protocol first verifies its predecessors signature and then signs the route discovery packet with its own certificate before forwarding the packet to its neighbor. This does not ensure a shortest path to the destination. To obtain a shortest path, an additional protocol may be used. The obtruded extra work for each route discovery is therefore immense.

3.3.2. SRP (Secure Routing Protocol)

[49] describes a way to secure several already existing routing protocols by adding the SRP (Secure Routing Protocol) or respectively altering the existing protocol to include the characteristics of SRP. Like ARAN, SRP requires an existing trust relationship (or security association) between source and target of a route discovery.[1] But unlike ARAN only source and target are using a shared secret key[2] to calculate a MAC. All intermediate nodes only forward the route discovery packets and add their identifier (e.g., IP-address) to the packet. The target receives those packets and replies one or several of those route discovery packets to the source over the reverse path. Source and target gain therefore a diverse network topology. The MAC prevents a malicious node to alter the packets or forge a fake of one. Because of the simple design, SRP can be added to a number of different (insecure) routing protocols like DSR (see 3.2 on page 25), ABR (see 3.2 on page 26) and IERP [35] of the Zone Routing Protocol (ZRP) [34].

3.3.3. Ariadne

Ariadne [37] is a secure routing protocol based on symmetric cryptography. Its rudimentary design is based on DSR (see 3.2). The authentication mechanism for broadcast messages like route request in Ariadne can be either of three schemes. Firstly digital signatures, secondly shared secrets between each pair of nodes and the final scheme is based on TESLA [53] which uses one-way key chains in combination of a loose synchronized clock. Every member of the ad-hoc network would generate a one-way key chain using a one-way hashfunction H on a random chosen key K_N, such that $K_i = H^{N-i}[K_N]$. Each member publishes his keys K_i in pre-defined, fixed interval t in reverse order of the generation. Using the loosely synchronized clock, the known maximum time synchronization error Δ and a pessimistic upper bound of the end-to-end network delay, a sender can calculate a key K_i, which should not been published at the time the packet receives its destination. The sender then calculates a MAC using the K_i and attaches it to the packet for the destination.

The receiver has the ability to determine which keys a sender may have already published, by the knowledge of T_0, the time the first key has been published, the time interval t and the maximum time synchronization error Δ. Upon reception

[1] The actual kind of security association has not been specified by the authors.

[2] i.e., established through the elliptic curve diffie hellman key exchange—see

of the packet, the receiver checks whether the key used to authenticate the packet is still not published, by checking that the difference between T_0 and $t' - \Delta$ has not exceeded i time intervals. If that is the case, he buffers the packet until publishing of the key. If the check fails, the receiver discards the packet, because an adversary might have forged the packet.

Ariadne requires the pre-deployment of authentic keys, either through a trusted key distribution center (KDC), a public-key infrastructure or through pre-loading of keys into the nodes, i.e., through a common administrative entity (military). The authentic keys are used to generate MACs for end-to-end integrity. Ariadne does not support confidentiality or secrecy, because the authors negate the necessity for this in routing protocols. In their opinion privacy is only relevant for higher layer protocols.

3.4. Cooperation Based Routing

Another way to ensure routing in a multihop environment are cooperation based routing schemes. Those schemes ”buy” the cooperation of the involved nodes using a reward, or they ”punish” misbehaving nodes.

3.4.1. Nuglets

Levente Buttyán and Jean-Pierre Hubbaux have introduced a new scheme for routing protocols, to ensure the cooperation of all nodes within an ad-hoc network [26]. Their scheme works with a virtual currency called *nuglets*. The *nuglets* are in fact a tamper resistant counter that is increased, if a packet is forwarded and decreased by the number of hops a packet would travel to its destination. Selfish behavior of a node—forwarding none or very few packets—results in starvation, because the *nuglet* counter would drop to zero and the node could not send any packets for its own. On the other hand would a greedy node—which would forward any packet and collect as many nuglets as possible—be draining its batteries very fast. A moderate attitude, where every node forwards many packets but keeps an eye on its energy reserves would therefore suffice to keep the network alive and working.

The use of an tamper resistant security module as proposed by the authors is costly, because it presumes a special kind of hardware. The requirement of such a security module limits this protocol only to those nodes.

3.4.2. Mitigating Routing Misbehavior

The authors of this scheme [46] propose two mechanisms. A watchdog for identification of misbehaving nodes and a pathrater for the finding of a path avoiding the misbehaving nodes. The authors show that even with a high amount of misbehaving nodes, an acceptable throughput of the network can be achieved.

3.4.3. CONFIDANT

The authors of CONFIDANT (Cooperation of Nodes - Fairness in Distributed Ad-hoc Networks) [25] propose a scheme where the detection, avoidance and isolation of misbehaving nodes ensures network cooperation. Each node implements a neighborhood monitor, to identify abnormal routing behavior and a path manager which maintains path rankings and performs sanctions against misbehaving nodes, like dropping of route request packets which come from such a misbehaving node. The neighborhood monitor requires the node to work in promiscuous mode.

3.5. Summary

Routing protocols for ad-hoc networks exist in many favours. But most of them only concentrate on routing and neglect the idea of adversaries, malicious users and attackers. Because every node in an ad-hoc network is also its own router, a malicious node can run havoc in an unsecured network.

The need for a secure and robust routing mechanism for ad-hoc networks is therefore undisputable. The presented schemes burden additional computational effort to the nodes of a MANET. ARAN with its extensive use of certificates requires much computational power in the route discovery process. Small nodes like smart dust or PDAs may be too overburden with that. SRP and Ariadne on the other hand prefer MACs to ensure the integrity of routing messages. This disburdens the intermediate nodes in the route, but requires either the pre-deployment of $N - 1$ symmetric keys for every node in a network with N participants, or a working public key infrastructure to ensure the authentication of the MAC keys. The cooperation based schemes require either a tamper resistant module, which is very difficult to realise [13] or in the case of CONFIDANT the implicit use of promiscuous mode in every node.

4. Available Papers, Implementations, Applications and Projects

There already exist several papers on security related issues in ad-hoc networks, some of them where already mentioned in the preceding chapters. What follows is an example of some papers, which include further reading material.

Section two of this chapter introduces some existing implementations, applications and projects dealing with ad-hoc networks of different research groups around the world.

4.1. Papers

Although this paper [32] from 1993 focuses mainly on software design, the authors Forman and Zahorjan characterize most of the problems found in ad-hoc networks. Regarding wireless communication they deal with disconnection, low bandwidth, high bandwidth variability, heterogeneous networks and security risks. They have a special focus on the problems of mobility (address migration and location dependent information) and portability (low power, risks of data, small user interface and small storage capacity).

Kärpijoki [44] gives a general overview about security issues in wireless ad-hoc networks. He focuses on the security requirements for different areas of application and introduces several aspects of network related security issues, like routing, key management, availability and access control.

The Terminodes designer have published this paper [38] on the MobiHOC'01 symposium. It provides an overview of security problems for MANETs, distinguishing the threats for basic and for security mechanisms. The authors show by example, that the security requirements are quite different for various MANETs. Nevertheless, they summarize some commonalities most MANETs share. The basic threats for MANETs include tampering and stealing of nodes, eavesdropping and interference of the wireless communication and missing cooperation or selfishness of nodes. The vulnerability of the security mechanisms are more or less equal to those in traditional (wired/static) networks, like maliciously placed public keys, compromised keys and hostile takeover of a (distributed) trusted server. The suggested solutions for the basic network mechanisms include tamper resistance of the nodes (or parts of it), motivated routing and service enforcement through ”nuglets”. Regarding the threats of the security mechanisms, the authors focus their attention to the fundamental key exchange. Their proposal is a PGP-like public-key infrastructure, where the users issue their own keys, instead of relying on a public certificate directory as PGP does.

The document [45] of Law, Hartel and Etalle is a fine and up-to-date literature review. The authors review several papers using the TCP/IP model as rough guideline. The network layer of the TCP/IP model deals mostly with routing, which is a problem in ad-hoc networks because of the changing topology. Papers [23, 39, 63] on different routing protocols are mentioned. Regarding the transport layer two papers [15, 36] are mentioned which try to adapt TCP for ad-hoc networks. Furthermore the paper reviews some proposals for homogeneous secure ad-hoc networks, like Security Protocols for Sensor Networks (SPINS [54]) and peeblenet [16]. Other reviews include threshold cryptography [65, 18], policy based security mechanisms [57, 58, 17] and intrusion detection in ad-hoc networks [64]. The authors also dwell on the subject of heterogeneous nodes and summarize that it’s of fundamental importance to secure the transit/routing of information within an ad-hoc network and give some ideas how this could be done.

4.2. Implementations, Applications and Projects

The Wireless Communications Technologies Group of the NIST (National Institute for Standards in Technology) [3] is working on several projects. They distinguish between sensor networks and mobile ad-hoc networks. The main differences between those two kinds of ad-hoc networks lies in the nature of their nodes. Sensor networks consists out of homogeneous nodes, which are normally small in size (like ”smart dust”), because they lack a user-interface. These nodes are commonly distributed by a single developer. The characteristic point of MANET nodes is their heterogeneity. The working group has gathered several links dealing with ad-hoc networks under [1].

WINGs (Wireless Internet Gateways) [9] was a project funded in 1996 by the DARPA / ITO at the University of California, Santa Cruz (UCSC) and Rooftop Communications Corporation (Rooftop) of Mountain View, California. Its goal was the development of two kinds of WINGs including the necessary protocols. Long-range WINGs, which are transportable and reside in vehicles, tents, or on roof tops, to be used to establish dynamic backbones, and short-range WINGs, which are low power and can be hand held, serve as the access points for mobile users. The innovative WING protocols developed in this project include channel access protocols, link control protocols, and routing protocols. The WINGs project was completed in 2000. Several of the results of WINGs found their way into Nokia wireless routers, after Nokia acquired Rooftop in 1999.

Terminodes [5] are a cross between a terminal and a (mobile) node. The terminodes are a development of the NCCR MICS (National Center of Competence in Research - Mobile Information and Communication System). The Center's goal is to study fundamental and applied questions raised by new generation mobile communication and information services, based on self-organization. This includes questions ranging from fundamental mathematical issues (statistical physics based analysis, information and communication theory) to networking, signal processing, security, distributed systems, software architecture, and economics.

The Wi-Fi (Wireless Fidelity) Alliance [8] is a nonprofit international association formed in 1999 to certify interoperability of wireless Local Area Network products based on IEEE 802.11 specification. The goal of the alliance is the distribution of Wi-Fi certified products.

The importance of MANETs is underlined through the fact that an IETF MANET working group exists, which deals with routing in MANETs [4]. There are currently four routing protocols supported, as Experimental RFCs. These are: AODV, DSR, OLSR (Optimized Link State Routing Protocol) and TBRPF (Topology Dissemination Based on Reverse-Path Forwarding). There exist some experiences with implementations of these protocols. The goal of this working group is to standardize IP routing protocol functionality suitable for wireless routing application within both static and dynamic topologies.

5. Implementation

The goal of this diploma thesis was a survey of security related issues in mobile ad-hoc networks as well as an implementation of an ad-hoc communications tool. The tool should be using the cv act library from the cv cryptovision gmbh for the underlying cryptographic security. The cv act library is a static Microsoft Windows library, which offers many cryptographic directives, including symmetric and asymmetric cryptosystems, digital signatures, key exchange protocols, hash functions and (pseudo) random number generators.

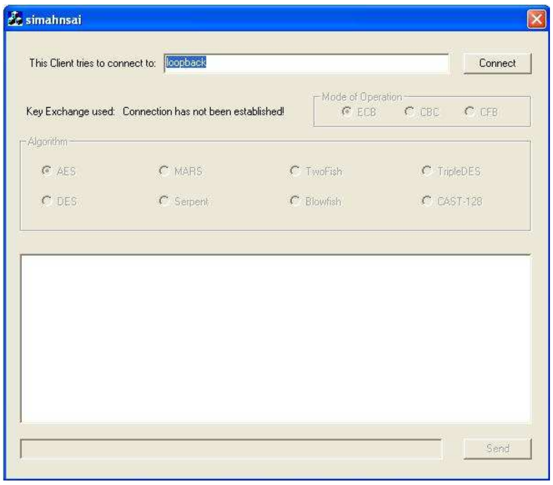
The resulting tool *simahnsai* is a client-server based application, that can be used as a secure instant messenger. It utilizes the existing Windows sockets for communication and the cv act library for encryption. Due to the time restriction of the diploma thesis it was not possible to include all possible features in *simahnsai* as it was intended.

5.1. Functionality and Usage of *simahnsai*

simahnsai offers the possiblity to exchange small messages in a secure (confident) way. The application guarantees confidentiality and integrity. Confidentiality is achieved through the encryption of the messages. The secret key verifies that the message can not be altered undetect by an entity which does not know the key. Unfortunatly a man-in-the-middle attack is still possible as long as unauthenticated keys are used in the key exchange protocol. The attacker hereby impersonates user B for user A and vice versa. Only signed keys can prevent this kind of attack.

Figure 5.1 shows *simahnsai* at the start. The tool is initially in client mode and asks for a server to connect to. This is contrary to the ad-hoc mode, the tool was intended to use. A manual interaction of the user is not ”ad-hoc”. But as described on page 40 the lower layers of the application have not yet been written. When no server can be found, the tool is switching to server-mode.

The server listens for connections requests on its port and after accepting a connection, an elliptic curve Diffie-Hellman key exchange (ECDH) (see Annex B.2.1 for a detailed description) is automatically initiated. After the key has been successfully negotiated, client and server are in the state as pictured in Figure 5.2. The server may then change the symmetric cryptosystem and mode of operation. Both server and client can exchange small messages which will be encrypted using the chosen symmetric cryptosystem and mode of operation (see



simahnsai

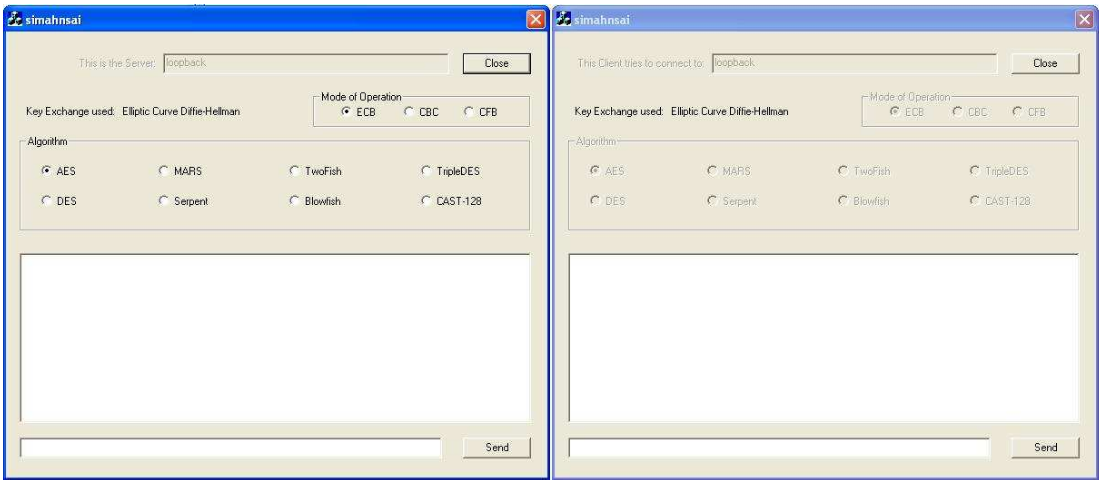
This Client tries to connect to: loopback

Key Exchange used: Connection has not been established!

Mode of Operation: ● ECB ○ CBC ○ CFB

Algorithm: ● AES ○ MARS ○ TwoFish ○ TripleDES ○ DES ○ Serpent ○ Blowfish ○ CAST-128

Figure 5.1.: simahnsai after Start



simahnsai

This is the Server: loopback

Key Exchange used: Elliptic Curve Diffie-Hellman

Mode of Operation: ● ECB ○ CBC ○ CFB

Algorithm: ● AES ○ MARS ○ TwoFish ○ TripleDES ○ DES ○ Serpent ○ Blowfish ○ CAST-128

simahnsai

This Client tries to connect to: loopback

Key Exchange used: Elliptic Curve Diffie-Hellman

Mode of Operation: ● ECB ○ CBC ○ CFB

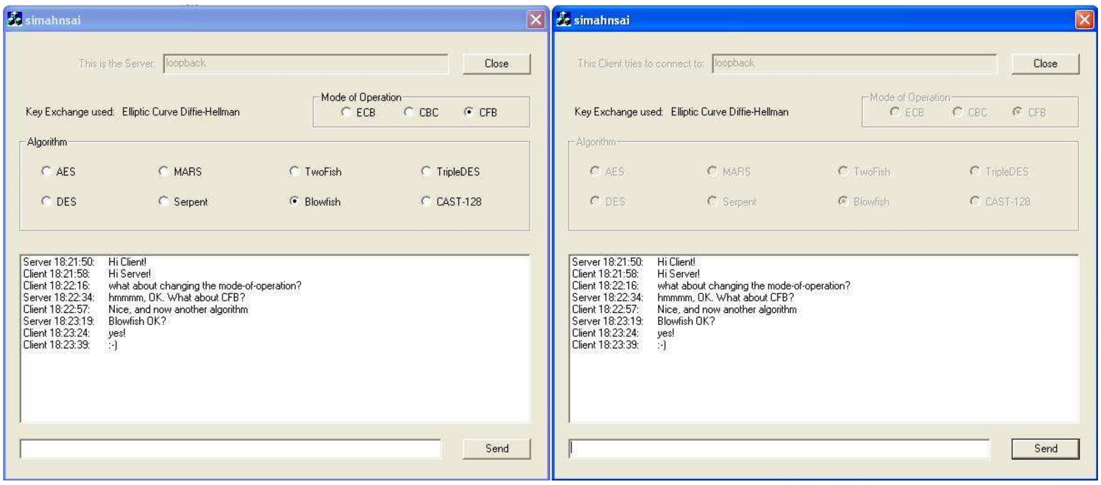
Algorithm: ● AES ○ MARS ○ TwoFish ○ TripleDES ○ DES ○ Serpent ○ Blowfish ○ CAST-128

Figure 5.2.: Server and Client after Connection and Key Establishment

Figure 5.3 for an example). Actually standard settings for algorithm and mode of operation would suffice, i.e., AES and CBC. But due to the show-character of *simahnsai* for the cv act library the selection has been implemented.

5.2. Details of the Implementation

The following is a description of the main classes used in *simahnsai*. The Tables 5.1, 5.2 and 5.3 show the member functions of those classes.



This is the Server: loopback

Key Exchange used: Elliptic Curve Diffie-Hellman

Mode of Operation: ○ ECB ○ CBC ● CFB

Algorithm: ○ AES ○ MARS ○ TwoFish ○ TripleDES ○ DES ○ Serpent ● Blowfish ○ CAST-128

```
Server 18:21:50:   Hi Client!
Client 18:21:58:   Hi Server!
Client 18:22:16:   what about changing the mode-of-operation?
Server 18:22:34:   hmmmm, OK. What about CFB?
Client 18:22:57:   Nice, and now another algorithm
Server 18:23:19:   Blowfish OK?
Client 18:23:24:   yes!
Client 18:23:39:   :-)
```

This Client tries to connect to: loopback

Key Exchange used: Elliptic Curve Diffie-Hellman

Mode of Operation: ○ ECB ○ CBC ● CFB

Algorithm: ○ AES ○ MARS ○ TwoFish ○ TripleDES ○ DES ○ Serpent ● Blowfish ○ CAST-128

```
Server 18:21:50:   Hi Client!
Client 18:21:58:   Hi Server!
Client 18:22:16:   what about changing the mode-of-operation?
Server 18:22:34:   hmmmm, OK. What about CFB?
Client 18:22:57:   Nice, and now another algorithm
Server 18:23:19:   Blowfish OK?
Client 18:23:24:   yes!
Client 18:23:39:   :-)
```

Figure 5.3.: Communication Between two simahnsai Instances

5.2.1. Class CryptoStuff

The class CryptoStuff contains all functions and cryptographic directives of the cv act library, which are used in *simahnsai*. They are all publicly accessible from the base class simahnsaiDlg.

CryptoStuff
BlobFlush
BlobFill
GenerateKeyPair
GenerateSecret
SymmetricEncrypt
SymmetricDecrypt

Table 5.1.: CryptoStuff member functions

BlobFlush converts a Blob (Binary Large Object) object into a `strstream`[1] object and converts existing `NULL`s into a two character escape sequence.

BlobFill converts an `istrstream` object into a Blob object and reconverts the escape-sequences into `NULL`s.

GenerateKeyPair generates randomly—seeded from system information—the public and private key parts for the ECDH. The key parts are exported into Blobs.

[1] `strstream` and `istrstream` are in the `strstrea.h`

GenerateSecret generates the secret session key from the private key part and the public key part from the other participant in the ECDH. Blobs are used to store the keys.

SymmetricEncrypt encrypts the Blob which contains the message text into another Blob.

SymmetricDecrypt decrypts the Blob with the encrypted message text.

5.2.2. Class MySocket

For every connection a MySocket object is generated. It contains four protected member functions which handle the connection details and a public function which binds the object to the calling parent object.

MySocket
OnConnect
OnAccept
OnReceive
OnClose
SetParent

Table 5.2.: MySocket Member Function

SetParent binds the object to the parent.

OnConnect is called, when the a client socket tries to connect to a server socket. If the other socket replies, the OnConnect function of the parent object is called. If the other socket does not exist, a dialogue box asks the user whether to start a new server or try another address.

OnAccept is called, when the server receives a connection request from a client socket.

OnReceive is called, when a socket receives a packet from the connected other socket.

OnClose is called, when a connection breaks or is terminated.

For all four functions asserts: If no error occurs, the corresponding parent function is called.

simahnsaiDlg
OnAccept OnReceive SendSystemData ...

Table 5.3.: Some simahnsaiDlg Member Function

5.2.3. Class simahnsaiDlg

simahnsaiDlg is the main class of the application. It contains the dialogue as well as most of the connection protocol.

OnAccept is called by the socket child function of the server socket. Its main purpose is to accept a connection and, if a connection is already established, reject incoming other connection requests.
OnAccept calls GenerateKeyPair and SendSystemData to send the public key part of the server to the client socket.

OnReceive is called by the socket child function of the client and server socket. In client mode, the first packet received from the server is its public key part. The client then generates its own private and public key parts and calculates the secret key. The client key part is send to the server. Other packets from the server could be system settings or encrypted messages. System settings—like encryption algorithm and mode of operation—are escaped with three leading line feeds. All other packets are handled as encrypted messages.
The first packet in server mode is the public key part of the client. The server generates the secret key and encrypted messages can be send back and forth between server and client.

SendSystemData is called by OnAccept, OnRset, OnRmoset and OnBsend, the function that is called when the send button is pressed. OnRset and OnRmoset are the functions which are called, when the radio buttons for the algorithms and mode of operation are pressed.
SendSystemData checks which event trigged its call and branches to the appropriate function part. If a radio button was pressed, a `char*` buffer is created with three leading line feeds. If the send button was pressed, the message string is first converted into a Blob and then encrypted using the SymmetricEncrypt function. The resulting Blob is converted in a strstream object (using BlobFlush) which is itself converted into a `char*` buffer which is send to the other socket.

5.3. Changes of *simahnsai* During the Implementation

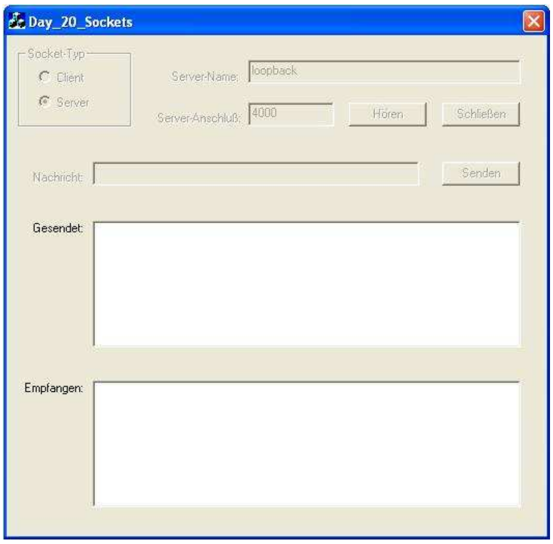
5.3.1. Bluetooth

Because of the pairing process, Bluetooth's ad-hoc mode is involved with interaction of the user. During the implementation this has been a problem. It was not possilbe to open an unprotected PPP-connection between the two bought Bluetooth dongles, because of the fixed security precautions in their Windows drivers.

5.3.2. Blobs

One of the main problems during the development process of *simahnsai* was the sending of the Blobs. A simple conversion of the key Blobs into strings using the integrated `.str()` function did not work, because a key Blob consist of at least two strings. The `.str()` function cuts off the second string after the final `NULL` character of the first string. Using escape sequences was the only way to solve this. BlobFlush copies the Blob bytewise into a temporary `byte`-array, exchanging every `NULL` character with two escape characters. The `byte`-array is then written into the output `strstream`. BlobFill is reversing the process.

5.3.3. GUI Design



Socket-Typ: ○ Client ● Server

Server-Name: loopback

Server-Anschluß: 4000

Figure 5.4.: Design Prototype

The design of the GUI (Graphic User Interface) has changed slightly during the development process. The prototype resembled Figure 5.4. The selection fields for the symmetric algorithms and the mode of operation have been added to demonstrate the possibilities of the cv act library.

5.4. Conclusion and Outlook

simahnsai is by far not complete; currently only the communication part is finished. The lower layers have not yet been written. Several improvements can and should be added, including signed key exchange, saving of preferences, point-to-multipoint connections, several other small features and especially the lower layers, which should communicate with the operation system internals as Figure 5.5 shows.

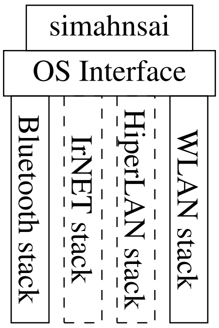


Figure 5.5.: simahnsai in the Overall Picture

Further development may include the adaption of *simahnsai* for other operating systems like PalmOS and Linux. An adaption for other operating systems must include the (re-)programming of the cryptographical primitives, because the cv act library is only available for Windows PCs and PocketPC.

6. Summary and Conclusion

This thesis introduced ad-hoc networks to the reader. It showed what is fact and what is fiction in ad-hoc network technology. The idea of a fully interconnected world has not yet become reality, although it seems not very far away.

Some major issues remain, though. IEEE 802.11 is currently the main choice as a communication bearer for ad-hoc networks. Its explained security problems must therefore be dealt with. Bluetooth with its flexible and adaptive security policy burdens the user with a complicated pairing process. And as experience shows, security with less usability tends to be turned off. The same applies to IrDA, because of it needs for a line-of-sight to work properly. Most routing protocols do not deal with security issues at all. Only a hand full of protocols are designed from the ground up to deal with adversaries and misbehaving nodes.

As this small summary shows, many security related issues remain unanswered in ad-hoc network technologies. Combined with its highly constraint devices, this makes ad-hoc networks a very interesting field of research for the near future. This thesis only showed some extends of the security problems a developer might occur when working on an ad-hoc network. During the development of *simahnsai*, it became clear that a secure end-to-end connection realised on higher layers, is the only way to overcome todays security problems with ad-hoc networks. Although *simahnsai* still lacks several features, its overall design allows a secure way of communication. The extension of *simahnsai* to a complete and secure messenger will be continued in the part time of the author.

A. RC4 aka arcfour

The following pseudo code (and figures A.1 and A.2) describe RC4 aka arcfour[1]:

- RC4 consists of two parts: a key scheduling algorithm which initializes an 8-bit×256 S-Box (Substitution Box) and an output generator.
- The key scheduling algorithm permutates the 256 possible values of the S-Box using the variable length key.
- 1st it initializes the S-Box `S` with its identity values.

  ```
  for (i=0; i<256; i++)
      S[i]=i;
  ```

  Then another 256 S-Box `S2` is filled with the secret key `K`, if the key is smaller than $256 * 8 = 2048$ bits the key will be repeated:

  ```
  for (i=0; i<256; i++)
      S2[i]=K[i%keylen];
  ```

  The counter `j` will be initialized to `0`. `j` is randomly changed (using the secret key) and the corresponding values of the S-Box `S` will be swapped permuting it:

  ```
  j=0;
  for (i=0; i<256; i++)
  {
      j=(j+S[i]+S2[i])%256;
      temp=S[i];
      S[i]=S[j];
      S[j]=temp;
  }
  ```

  For safety measures zero key `K` and `S2`:

  ```
  for (i=0; i<256; i++)
  {
  ```

[1] code has been taken from: [42]

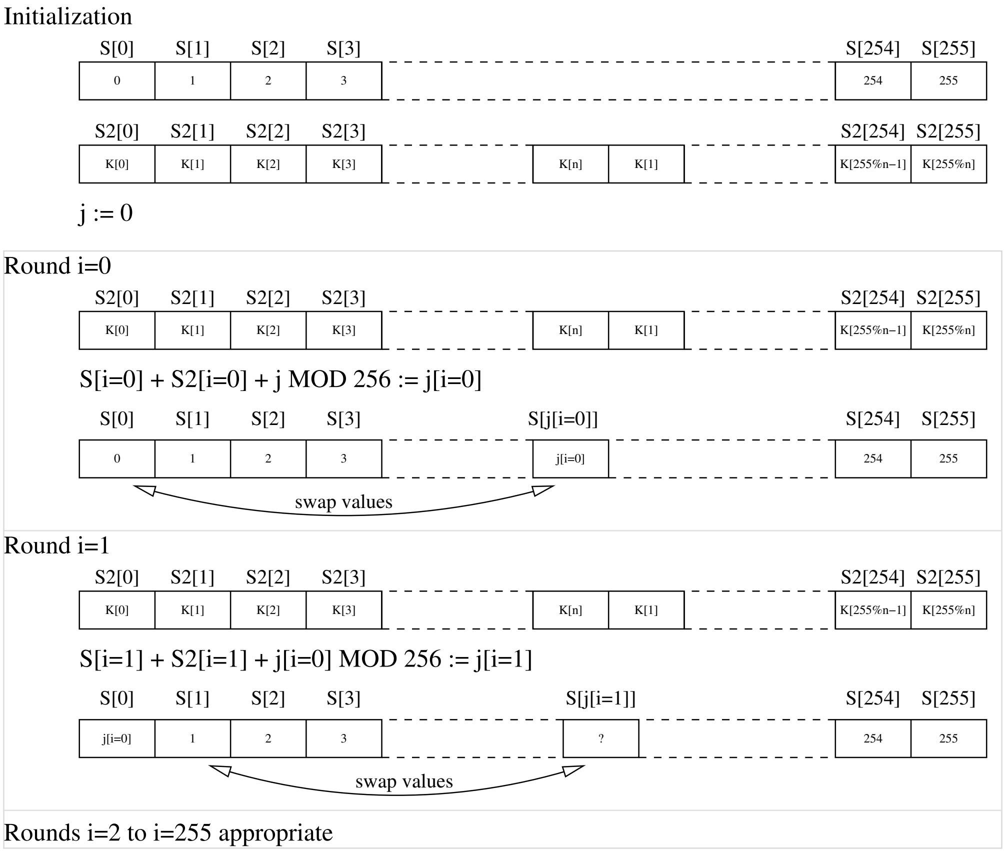


Figure A.1.: RC4/arcfour Key Scheduling Algorithm

```
        S2[i]=0;
        K[i]=0;
    }
```

initialize i and j to zero:

```
    j=0;
    i=0;
```

Using this the S-Box `S` has an incredibly amount of $256! * 256^2 \approx 2^{1700}$ possible states!

- The Output generator generates the keystream O, which is bytewise XORed with the plaintext to encrypt it.

```
    i=(i+1)%256;
    j=(j+S[i])%256;
    temp=S[i];
    S[i]=S[j];
    S[j]=temp;
    t=(S[i]+S[j])%256;
    O=S[t];
```

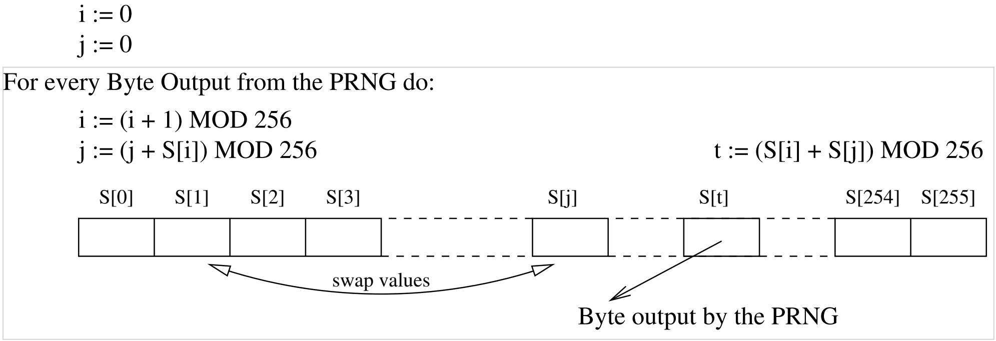


Figure A.2.: RC4/arcfour Stream Output generator

The S-Box will be changed slowly during use. `i` guarantees that every element of `S` will be changed, while `j` guarantees a random change.

B. Key Exchange

The main purpose of key exchange algorithms is to establish a secret key over an insecure channel between two or more participants. This key can then be used to establish a secure tunnel over the insecure channel. There exist several key agreement schemes for traditional/infrastructural networks, where a server-client structure is available. Most of these schemes can be adopted to ad-hoc networks, when the existence of a public key infrastructure or the availability of a trusted third party is given.

B.1. RSA key exchange

Because of its use in the SSL-layer of the two most commonly used Internet browsers (Microsoft Internet explorer and Netscape Navigator) it is the most commonly used key exchange algorithm. The RSA key exchange is derived from the RSA encryption algorithm [56].

The RSA key exchange is a strictly server-client based protocol. The server sends the client after a request his certificate and his public key. After verifying the certificate, the client generates a random key, encrypts it with the server's public key and sends it to the server. The server decrypts the packet and uses the key to open the secure channel.

The server has no impact in generating the key. So if the client is compromised, it could choose a key—instead of randomly generating one—and the communication would not be secure at all.

A public key infrastructure is required, because the server must authenticate itself to the client. The client must have the opportunity to check whether the certificate of the server is correct and if the certificate has been revoked. That means the client must have an online-connection to the CA. In a pure ad-hoc network such an Internet connection cannot be guaranteed. The RSA key exchange is therefore not suited for pure ad-hoc networks.

B.2. Diffie-Hellman key exchange

The Diffie-Hellman key exchange was the first protocol for key agreement. Actually it was introduced in a paper of Diffie and Hellman that defined public key cryptography for the first time in 1976 [28]. Its security is based upon the difficulty of computing logarithms over finite fields[1].

The algorithm is a two way handshake protocol. First the two participants (A and B) agree on a cyclic group $\mathbb{F}_p^*$ and on g which is a generator of $\mathbb{F}_p^*$, such that for every a in $\mathbb{F}_p^*$ there exists an x with $1 \leq x \leq p-1$ with $a = g^x \mod p$. x is called the discrete logarithm of a to the basis g. p and g do not need to be secret, so A and B may agree on them over the insecure channel.

Then A chooses a random integer x and calculates $X = g^x \mod p$ while B chooses a random integer y and calculates $Y = g^y \mod p$. A and B exchange their calculated values and keep their random integer secret. A then computes $k = Y^x \mod p$ and B computes $k = X^y \mod p$. Both A and B now have the same k because $k = Y^x = g^{yx} = g^{xy} = X^y \mod p$.

The above described DH scheme neither supports key authentication nor key verification and is therefore vulnerable to man-in-the-middle attacks.

B.2.1. Elliptic Curve Diffie-Hellman key exchange

Since 1985 when Koblitz and Miller [43, 47] supposed the use of elliptic curves as a group for the discrete logarithm problem, the ECC (Elliptic Curve Crpytography) has become a strong competitor for established asymmetric cryptosystems, like RSA and ElGamal.

The ECDH is similar to the DH key exchange except that it utilizes elliptic curves as a cyclic subgroup.

A and B choose an elliptic curve $E(\mathbb{F}_q)$ defined over the finite field $\mathbb{F}_q = GF(q)$. For a cryptographical application q would be choosen as $q = p$ with p prime or $q = 2^m$, $m \in \mathbb{N}$. G is generator of a subgroup in the pointgroup of $E(\mathbb{F}_q)$ where the order n of G is a large prime with $nG = \mathcal{O}$. G is called basepoint. The ECDL problem is now defined in this subgroup.

A (resp. B) then chooses a private random a (resp. b) of order of magnitude p. A then calculates $K_a = aG$ (respectively $K_b = bG$) which is public and sends it to the other participant. A computes $K = aK_b = abG = baG = bK_a$.

The secret K is a point on the elliptic curve $E_p(a, b)$ and consists therefore out of two numbers. To generate a key for a symmetric cryptosystem, one can for example use the x coordinate.

[1] discrete logarithm problem

List of Figures

List of Tables

Index

Glossary

ARAN *Authenticated Routing for Ad-hoc Networks* Routing Protocoll for Adhoc Network, which uses certificates for routing messages to permit non-repudiation, authentication and message integrity. *Page XII*

Arcfour Fully compatible, open alternative to RC4. *Page 7*

Baseband description for the cumulative protocols in the LC. *Page 16*

BD_ADDR *Bluetooth Device Address* 48 bit address, unique for each Bluetooth device *Page XI*

Harald Blåtand Viking king in the 10th century, first to unite and christianize Denmark and parts of Norway. Blåtand literaly translates to blue tooth. *Page 15*

Bluetooth Non-profit international organization[2] favouring a wireless low-cost, low-power, short-range interface, named after Harald Blåtand. *Page 5*

ciphertext encrypted text/data. *Page 8*

claimant The person/device who/which want's to be authenticated by the verifier. *Page 20*

E0 Stream cipher used by Bluetooth. *Page 21*

Ethernet IEEE standard 802.3, is a LAN. *Page 6*

IEEE *Institute of Electrical and Electronics Engineers, Inc.* Non-profit, technical professional association[3] of more than 377,000 individual members in 150 countries. *Page XI*

Inquiry Hopping Sequence A sequence of 32 (16) frequencies, derived out of the lower 24 bits of its BD_ADDR. *Page 19*

[2] *http://www.bluetooth.com*

[3] *http://www.ieee.org*

IrDA *Infrared Data Association* is an International Organization[4] that creates and promotes interoperable, low cost infrared data interconnection standards that support a walk-up, point-to-point user model. The Infrared Data Association standards support a broad range of appliances, computing and communications devices. *Page 15*

L2CAP *Logical Link Control and Adaption Protocol* Bluetooth protocol for packet segmentation and reassambly, support for higherlevel (RFCOMM) multiplexing. *Page XII*

LAN *Local Area Network* Standard Network with fixed Infrastructure, today mainly based on Ethernet (IEEE 802.3). *Page XI*

LC *Link Controller* carries out Baseband protocols and other low-level link routines. *Page XI*

managed-open environment second of the three defined environment in ARAN. Nodes are capable to exchange initialisation parameters (certificates) before beginning communication. *Page 27*

mote Term used by Neal Stephenson to describe smart dust nanomachines *Page 2*

node Devices within a wireless network. *Page 2*

Open System authentication Standard WLAN authentication method. No authentication token is needed. *Page 9*

peeblenet Network of small independent sensors *Page 32*

plaintext unencrypted or unencyphered text/data. *Page 8*

PRNG *Pseudo Random Number Generator* A mathmatical formula or a program written for, and used in, cryptography, probability and statistics applications when large quantities of random digits are needed. *Page XII*

PRNG seed The seed of a PRNG is the input from which the output is generated. *Page 8*

RC4 *Ron's Code 4 or Rivest's Cipher 4* Fast and efficient stream cipher from RSA Security Inc.[5] *Page XI*

RFCOMM is a simple transport protocol, which provides emulation of RS232 serial ports over the Bluetooth L2CAP protocol. *Page 18*

Shared Key Authentication WEP-based authentication method in WLAN. *Page 9*

[4] *http://www.irda.org*

[5] *http://www.rsasecurity.com*

smart dust Nanomachines (state of the art: micromachines) capable of forming a sensorgrid. *Page 2*

verifier The person/device who/which authenticates the claimant. *Page 20*

WEP *Wireless Equivalent Privacy* Security mechanism for WLAN. Badly desgined by the IEEE and therefor broken. *Page XII*

WLAN *Wireless LAN* Wireless LAN aka IEEE 802.11. *Page XII*

List of Acronyms

ABR .... Associativity-Based Routing Source-initiated on-demand routing protocol for Ad-hoc Networks. *Page 26*

ACL........... Asynchronous Connection-Less Bluetooth link. *Page 18*

AODV......... Ad-hoc On-demand Distance Vector Source-initiated on-demand routing protocol for Ad-hoc Networks. *Page 25*

AP .... Access Point WLAN-bridge to an infrastructe LAN. *Page 7*

ARAN......... Authenticated Routing for Ad-hoc Networks Routing Protocoll for Ad-hoc Network, which uses certificates for routing messages to permit non-repudiation, authentication and message integrity. *Page 27*

Baseband...... description for the cumulative protocols in the LC. *Page XVII*

BD .... Bluetooth device Abbreviation of Bluetooth device. *Page 16*

BD_ADDR ... Bluetooth Device Address 48 bit address, unique for each Bluetooth device *Page 17*

Harald Blåtand Viking king in the 10th century, first to unite and christianize Denmark and parts of Norway. Blåtand literaly translates to blue tooth. *Page XIV*

Blob........... Binary Large Object A universal data object used in the cv act library. A Blob is like std::vector<unsigned char> of the C++ standard library with the difference that Blob overwrites the used memory area with all zeros. *Page 36*

Bluetooth...... Non-profit international organization[6] favouring a wireless low-cost, low-power, short-range interface, named after Harald Blåtand. *Page XIV*

BQ .... Broadcast Query Route discovery packet in ABR. *Page 26*

[6] *http://www.bluetooth.com*

BSS .... Basic Service Set WLAN network consiting out of an AP and some STAs. *Page 7*

CAC .... Channel Access Code Channel Identifier of a Bluetooth piconet. *Page 17*

CGSR .... Clusterhead Gateway Switch Routing Table-driven routing protocol for Ad-hoc Networks. *Page 24*

CIA .... Confidentiality Integrity Authenticity Keywords for a secure communication. *Page 3*

CONFIDANT . Cooperation of Nodes - Fairness in Distributed Ad-hoc Networks Secure Routing Protocol. Nodes watch each other and punish misbehavior. *Page 30*

CRC-32 .... Cyclic Redundancy Checksum a common technique for detecting data transmission errors. *Page 7*

CSMA/CD .... Carrier Sense Multiple Access/Collision Detection A set of rules determining how network devices respond when two devices attempt to use a data channel simultaneously (called a collision). Standard Ethernet networks use CSMA/CD. This standard enables devices to detect a collision. After detecting a collision, a device waits a random delay time and then attempts to re-transmit the message. If the device detects a collision again, it waits twice as long to try to re-transmit the message. This is known as exponential back off. *Page 6*

DAC .... Device Access Code Address for a Bluetooth device within a piconet. *Page 17*

DAG .... Direct Acyclic Graph Three dimensional graph used by TORA to calculate the shortest route. *Page 26*

DARPA .... Defence Advanced Research Projects Agency US Military research agency. *Page 2*

DoS .... Denial of Service Attack aimed to stop services provided by a host.y *Page 27*

DRP .... Dynamic Routing Protocol Part of the SSR. *Page 26*

DSDV .... Destination-Sequenced Distance-Vector Table-driven routing protocol for Ad-hoc Networks. *Page 24*

DSR .... Dynamic Source Routing Source-initiated on-demand routing protocol for Ad-hoc Networks. *Page 25*

DSSS .... Direct Sequence Spread Spectrum is a transmission technology used in WLAN transmissions where a data signal at the sending station is combined with a higher data rate bit sequence, or chipping code, that divides the user data according to a spreading ratio. The chipping code is a redundant bit pattern for each bit that is transmitted. *Page 6*

ECC .... Elliptic Curve Crpytography A special group for the discrete logarithm problem. *Page V*

ESS .... Extended Service Set Overlaping BSS form an ESS. *Page 7*

ETSI .... European Telecommunications Standards Institute is a not for profit organization whose mission is to produce the telecommunications standards that will be used for decades to come throughout Europe and beyond. *Page 18*

FHSS .... Frequency Hopping Spread Spectrum is a transmission technology used in WLAN and Bluetooth transmissions where the data signal is modulated with a narrowband carrier signal that "hops" in a random but predictable sequence from frequency to frequency as a function of time over a wide band of frequencies *Page 6*

full duplex ..... Full-duplex data transmission means that data can be transmitted in both directions on a signal carrier at the same time. *Page XX*

GUI .... Graphic User Interface a user interface based on graphics (icons and pictures and menus) instead of text; uses a mouse as well as a keyboard as an input device *Page 40*

HiperLAN ..... High performance radio LAN European version for a fast radio LAN. *Page 5*

IBSS .... Independent Basic Service Set If no AP is available WLAN STAs can connect in an ad-hoc/peer-to-peer mode. *Page 7*

ICV .... Integrity Check Value CRC-32 checksum used by WLAN. *Page 7*

IEEE .... Institute of Electrical and Electronics Engineers, Inc. Non-profit, technical professional association[7] of more than 377,000 individual members in 150 countries. *Page 6*

[7] *http://www.ieee.org*

IrCOMM ...... Serial and Prallel Port Emulation over IR (Wire Replacement) A protocol, aimed to replace the wired infrastructure with infrared devices. *Page 22*

IrDA .... Infrared Data Association is an International Organization[8] that creates and promotes interoperable, low cost infrared data interconnection standards that support a walk-up, point-to-point user model. The Infrared Data Association standards support a broad range of appliances, computing and communications devices. *Page 5*

IrLAN .... Infrared LAN Access Extensions for Link Management Protocol IEEE 802.3 extension of the IrDA-stack. *Page 22*

IrLAP .... Serial Infrared Link Access Protocol Lower layer protocol of the IrDA-stack. *Page 22*

IrNET .... Wireless Network based on IrDA. *Page 22*

IrOBEX .... Infrared Object Exchange see IrDA and OBEX. *Page 18*

ISO .... International Standards Organization International Organisation for Standards in Telecommunication.[9] *Page XVII*

ISO/OSI reference model OSI communication reference model of the ISO Group *Page 5*

IV .... Initial Vector 24 bit used to widen the secret key in WLAN. *Page 8*

L2CAP .... Logical Link Control and Adaption Protocol Bluetooth protocol for packet segmentation and reassambly, support for higher-level (RFCOMM) multiplexing. *Page 18*

LAN........... Local Area Network Standard Network with fixed Infrastructure, today mainly based on Ethernet (IEEE 802.3). *Page 6*

LC .... Link Controller carries out Baseband protocols and other low-level link routines. *Page 16*

LM............ Link Manager used for link set-up and control in the Bluetooth-stack. *Page 18*

LMP .... Link Manager Protocol protocols used by the LM. *Page 18*

[8] *http://www.irda.org*

[9] *http://www.iso.org*

MAC .... Media Access Control Second layer in the ISO/OSI communication reference model. *Page 5*

MAC .... Message Authentication Code A MAC is a function that takes a variable length input and a key to produce a fixed-length output. It is used to verrify the integrety of a message. *Page 3*

MANET .... Mobile Ad-hoc Network A network consisting out of mobile devices, which connect to each other in an ad-hoc fashion, without fixed infrastructure. *Page 2*

NIST .... National Institute for Standards in Technology Standards organisation of the US government. *Page 32*

OBEX .... Object Exchange Originaly part of the IrDA-stack. Bluetooth has adapted OBEX to allow applications to exchange objects either over Bluetooth or IrDA. *Page 18*

OSI .... Open Systems Interconnect Model, to describe defined layers in a network operating system. *Page XVII*

PAN .... Personal Area Network Contrary to LAN *Page 15*

PDA .... Personal Digital Assistant Electronic filofax, normaly palm-sized. *Page v*

PGP .... Pretty Goog Privacy Software used to encrypt and digitaly signate email. *Page 32*

PHY .... PHYsical Layer First and lowest layer in the ISO/OSI communication reference model. *Page 15*

piconet .... One Bluetooth master and up to seven slaves form a piconet. *Page XV*

PIN .... Personal Identification Number Shared secret between two Bluetooth devices during the pairing process. The PIN can either be entered manualy or through an upper layer security application into both BDs. Or when only one device has a keypad, a fixed PIN of the padless device is used. *Page 20*

POS .... Personal Operating Space Physical space of the Bluetooth-user. *Page 15*

ppp .... Point-to-Point Protocol P provides a method for transmitting IP and IPX datagrams over serial point-to-point links, for example over a modem.*Page 22*

PRNG . . . . Pseudo Random Number Generator A mathmatical formula or a program written for, and used in, cryptography, probability and statistics applications when large quantities of random digits are needed. *Page 7*

QoS . . . . Quality of Service Scheme originaly developed for IPv6 to support multimedia applications through prefered routing and reserved time-slots. *Page 16*

RC4 . . . . Ron's Code 4 or Rivest's Cipher 4 Fast and efficient stream cipher from RSA Security Inc.[10] *Page 7*

RFCOMM . . . . . is a simple transport protocol, which provides emulation of RS232 serial ports over the Bluetooth L2CAP protocol. *Page XVII*

RFID . . . . Radio Frequency Identification Wireless equivalent to the barcode see *http://www.rfid.org/* or *http://www.rfidjournal.com/* for more details. *Page 1*

RREP . . . . Route Reply routing replay packet in AODV. *Page 25*

RREQ . . . . Route Request routing request packet in AODV. *Page 25*

SAR . . . . segmentation and reassambly Functionality of the L2CAP. *Page 18*

S-Box . . . . Substitution Box A substitution table. Each input gives a table corresponding output. *Page I*

SCO . . . . Synchronous Connection-Oriented Bluetooth link. *Page 18*

SDP . . . . Service Discovery Protocol is used to locate services provided by or available through a Bluetooth device. *Page 18*

SIG . . . . Special Interest Group Bluetooth consortium,[11] founded by Ericson, Nokia, Toshiba, IBM and Intel; currently over two thousand members. Defines and sets standards, specifications for Bluetooth. *Page 15*

SPINS . . . . Security Protocols for Sensor Networs Sensor net *Page 32*

SRP . . . . Static Routing Protocol part of the SSR *Page 26*

SRP . . . . Secure Routing Protocol Security extension for a multitude of existing routing protocols. *Page 28*

[10] *http://www.rsasecurity.com*

[11] *http://www.bluetooth.org*

SSID .... Service Set Identity Name of a WLAN. *Page 7*

SSL .... Secure Socket Layer A commonly-used protocol for managing the security of a message transmission on the Internet. *Page 7*

SSR .... Signal Stability-Based Adaptive Routing Source-initiated on-demand routing protocol for Ad-hoc Networks. *Page 26*

STA .... Station WLAN enabled computer/notebook/PDA. *Page 7*

TCP/IP .... Transport Control Protocol/Internet Protocol Standard Protocol for the Internet. *Page 11*

TCS .... Telephony Control protocol Specification This protocol defines the call control signalling for the establishing of speech and data calls between Bluetooth devices. *Page 18*

TDD .... Time-Division Duplex Packets use different time-slots on the same channel to avoid collision and to gain full duplex mode. *Page 16*

terminode ..... TERMINal and NODE Catchword created of Terminal and Node (*http://www.terminodes.org*). *Page 31*

TORA......... Temporally Ordered Routing Algorithm Source-initiated on-demand routing protocol for Ad-hoc Networks. *Page 25*

WAP .... Wireless Application Protocol is a specification for a set of communication protocols to standardize the way that wireless devices, such as cellular telephones and radio transceivers, can be used for Internet access, including e-mail, the World Wide Web, newsgroups, and Internet Relay Chat (IRC). *Page 18*

WEP .... Wireless Equivalent Privacy Security mechanism for WLAN. Badly desgined by the IEEE and therefor broken. *Page 7*

Wi-Fi.......... Wireless Fidelity The Wi-Fi Alliance[12] is a nonprofit international association formed in 1999 to certify interoperability of WLAN products based on IEEE 802.11 specification. *Page 6*

WLAN .... Wireless LAN Wireless LAN aka IEEE 802.11. *Page 5*

WPAN .... Wireless Personal Area Network IEEE trademarked acronym. *Page 15*

[12] *http://www.wi-fi.com*

XOR .... Exclusive Or Bitwise addition modulo 2 OR a Boolean operator that returns a value of TRUE only if just one of its operands is TRUE. *Page 8*

Bibliography

[1] Ad-hoc links. http://w3.antd.nist.gov/wctg/manet/adhoclinks.html.

[2] Airsnort. http://airsnort.sourceforge.net/.

[3] Manet. http://w3.antd.nist.gov/wctg/manet/.

[4] Manet working group of the ietf. http://www.ietf.org/html.charters/manet-charter.html.

[5] Terminodes. http://www.terminodes.org.

[6] Wardriving. http://www.wardriving.com/.

[7] Wepcrack. http://wepcrack.sourceforge.net/.

[8] Wi-fi. http://www.wi-fi.org.

[9] Wings for the internet. http://www.cse.ucsc.edu/research/ccrg/projects/wings.html.

[10] Wordlists. http://wordlists.security-on.net/.

[11] kabelab.de - drahtloser zugang ins internet. http://www.tmr.net/service/presse/pr240503.htm, http://www.tmr.net/service/presse/pr230503.htm, http://www.tmr.net/service/presse/pr220503.htm, May 2003.

[12] Bernard Aboba. Wep2 security analysis. http://www.drizzle.com/~aboba/IEEE/11-01-253r0-I-WEP2SecurityAnalysis.ppt, May 2001.

[13] R. Anderson and M. Kuhn. Tamper resistance—a cautionary note. In *Proc. 2nd USENIX Workshop on Elektronic Commerce*, 1996.

[14] W. A. Arbaugh, S. Shankar, and J. Y. C. Wan. Your 802.11 wireless network has no clothes. http://citeseer.nj.nec.com/arbaugh01your.html.

[15] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz. A comparison of mechanisms for improving tcp performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, 1997.

[16] S. Basagni, K. Herrin, E. Rosti, and D. Bruschi. Secure pebblenets. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, pages 156–163, 2001.

[17] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *17th IEEE Symposium on Security and Privacy*, number 96-17, Los Alamos, 28 1996. IEEE Computer Society Press.

[18] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J. Hubaux, and J. Le Boudec. Self-organization in mobile ad-hoc networks: the approach of terminodes. http://citeseer.nj.nec.com/blazevic01selforganization.html, 2001.

[19] Bluetooth Special Interest Group. *Bluetooth Security Architecture Version 1.0*, 15 1999.

[20] Bluetooth Special Interest Group. *Bluetooth Security Whitepaper V. 1.0*, 19 2002.

[21] Bluetooth Special Interest Group. *Bluetooth V1.1 Core Specifications*, 2002.

[22] D. Blunk, A. Girardet, and A. Prof. Dr. Steffen. Wlan war driving. Master's thesis, Züricher Hochschule Winterthur, October 2002.

[23] J. Broch, D. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.

[24] Bundesamt für Sicherheit in der Informationstechnik BSI. Bluetooth gefährdungen und sicherheitsmaßnahmen. *BSI Broschüre*, 2003.

[25] S. Buchegger and J.-Y. Le Boudec. Performance analysis of the confidant protocol (cooperation of nodes – fairness in distributed ad-hoc networks). In *Proccedings of the ACM Symposium on Mobile Ad-Hoc Networking and Computing (MobiHOC)*, Lusanne, Switzerland, June 2002.

[26] L. Buttyán and J. Hubaux. *Stimulating Cooperation in Self-Organizing Mobile Ad-Hoc Networks*, volume ACM/Kluwer Mobile Networks and Applications (MONET). Kluwer Academic Publishers, Mar. 2002.

[27] C.-C. Chiang. Routing in clustered multihop, mobile wireless networks with fading channel. In *Proc. IEEE SICON*, pages 197–211, April 1997.

[28] W. Diffie and M.E. Hellman. New directions in cryptography. In *IEEE Transactions on Information Theroy*, pages 644–654, Nov 1976.

[29] R. Dube. Signal stability based adaptive routing (ssa) for ad-hoc mobile networks. In *Proc. IEEE Pers. Commun.*, pages 36–45, Feb. 1997.

[30] S. R Fluher and S. Lucks. Analysis of the e0 encryption system. In *Lecture Notes in Computer Science 2256*. Springer-Verlag, 2001.

[31] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. *Lecture Notes in Computer Science*, 2259:1–24, 2001.

[32] G. H. Forman and J Zahorjan. The challenges of mobile computing. In *IEEE Computer Magazine*, volume 27, pages 38–47. University of Washington, 22 1994.

[33] Metro Group. Future store initiative. http://www.future-store.org, April 2003.

[34] Z.J. Haas and M. Perlman. The performance of query control schemes of he zone routing protocol. In *IEEE/ACM Transactions on Networking*, volume 9 of *4*, pages 427–438, Aug. 2001.

[35] Z.J. Haas, M. Perlman, and P. Samar. *The Interzone Routing Protocol (IERP) for Ad Hoc Networks.* IETF MANET Working Group, draft-ietf-manet-zone-ierp-01.txt, Jun. 2001.

[36] G. Holland and N. Vaidya. Analysis of tcp performance over mobile ad hoc networks. In *Proceedings of ACM/IEEE MobiCom'99*, Seatle, Washington, 1999.

[37] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *The 8th ACM International Conference on Mobile Computing and Networking*, September 2002. http://citeseer.nj.nec.com/hu02ariadne.html.

[38] J. Hubaux, L. Buttyán, and S. Čapkun. The quest for security in mobile ad hoc networks. In *Mobile Ad Hoc Networks Proceeding of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC 2001)*, Long Beach, CA 2001.

[39] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proceedings of ACM/IEEE MobiCom'99*, pages 195–206, 1999.

[40] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad-hoc wirelss networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer, 1996.

[41] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for ”smart dust”. In *Fifth Annual International Conference on Mobile Computing and Networking (MOBICOM’99)*, pages 271–278, http://citeseer.nj.nec.com/kahn99next.html 1999.

[42] K. Kaukonen. Internetdraft for arcfour. http://www.mozilla.org/projects/security/pki/nss/draft-kaukonen-cipher-arcfour-03.txt.

[43] N. Koblitz. Elliptic curve cryptosystems. In *Mathematics of Computation*, volume 48, pages 203–209, 1987.

[44] V. Kärpijoki. Security in ad hoc networks. In *Proceedings of the Helsinki University of Technology, Seminar on Network Security fall 2000*, 2000.

[45] Y. W. Law, P. Hartel, and S. Etalle. Security of ad hoc networks: A preliminary discussion. http://wwwhome.cs.utwente.nl/ ywlaw/pub/paper.pdf, 13 2002.

[46] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 255–265, 2000. http://citeseer.nj.nec.com/marti00mitigating.html.

[47] V. Miller. Uses of elliptic curves in cryptography.

[48] T. Newsham. Wep password cracker, 2002. http://www.lava.net/˜newsham/wlan/WEP_password_cracker.ppt.

[49] P. Papadimitratos and Z. J. Haas. Secure routing for mobile ad hoc networks. In *In SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, 01 2002.

[50] V.D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proc. INFOCOM ’97*, Apr. 1997.

[51] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Comp. Commun. Rev.*, pages 234–244, 1994.

[52] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. 2nd IEEE Wksp. Mobile comp. Sys. and Apps.*, pages 90–100, Feb. 1999.

[53] A. Perrig, R. Canetti, J.D. Tygar, and D. Song. Efficient authentication and signing of multicast streams over lossy channels. In *IEEE Symposium on Security and Privacy*, May 200.

[54] A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. Tygar. Spins: Security protocols for sensor networks. http://citeseer.nj.nec.com/perrig02spins.html, 2001.

[55] K. Sanzgiri, B. Dahill, B. N. Levine, and E. M. Shieldsnd Belding-Royer. A secure routing protocol for ad hoc networks. http://citeseer.nj.nec.com/551839.html.

[56] B. Schneier. *Applied Cryptography - Protocols, Algorithms and Source Code in C, 2nd Edition*. John Wiley and Sons, 1996.

[57] F. Stajano. The resurrecting duckling – what next? *Lecture Notes in Computer Science*, 2133:204–??, 2001.

[58] F. Stajano and R. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols, 7th International Workshop Proceedings*, pages 172–194, 1999.

[59] N. Stephenson. *The Diamond Age/Or, a Young Lady's Illustrated Primer*. Bantam Doubleday Dell Pub, 1 1995.

[60] A. Stubblefield, J. Ioannidis, and A. Rubin. Using the fluhrer, mantin, and shamir attack to break wep. http://citeseer.nj.nec.com/article/stubblefield01using.html, 2001.

[61] C.-K. Toh. A novel distributed routing protocol to support ad-hoc mobile computing. In *Proc. 1996 IEEE 15 Annual Int'l Phoenix Conf. Comp. and Commun.*, pages 480–486, Mar. 1996.

[62] J. R. Walker. Ieee p802.11 wireless lans unsafe at any key size; an analysis of the wep encapsulation. http://citeseer.nj.nec.com/558358.html, 27 2000.

[63] S. Yi, P. Naldurg, and R. Kravets. Security-aware ad hoc routing for wireless networks. In *Proceedings of the 2001 ACM International Symposium on Movile Ad Hoc Networking and Computing*, pages 299–302. ACM Press, 2001.

[64] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *Mobile Computing and Networking*, pages 275–283, 2000.

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