Formation of Secure Wireless Ad-hoc Sensor Networks

By

Morten Pedersen

Than Kim Thong

Master Thesis in

Information and Communication Technology

Agder University College

Grimstad, 2004-06-01

I. Abstract

Looking into the wireless world today, the most of data/information is transmitted in plaintext over the ether. These sensitive data/information possibly route through several intermediate nodes to a destination. To secure the sensitive data/information, the various ad-hoc technologies like Bluetooth, WLAN and ZigBee have implemented different security mechanisms and routing protocols. But since most wireless ad-hoc networks demand battery driven devices they will have limit resources to provide feasible security.

The different technologies will be outlined regarding a set case scenario. We have a parking lot and want to secure cars against thievery. For this we conducted a research over suitable technologies and routing protocols. We concluded that using ZigBee would fill that role better than the other technologies examined. We also decided to use the build in routing mechanisms from ZigBee. We developed a solution outcast for the application level of our case and gave a foundation to build a test model on.

II. Preface

This thesis is part of the Master Degree in Information and Communication Technology at Agder University College, Faculty of Engineering and Science in Grimstad, Norway.

Our supervisors were Professor Vladimir Alexandrovich Oleshchuk and Assistant Professor Magne Arild Haglund both employed at Agder University College. We would like to thank them for their guidance and critics during the project period. We also want to thank them for their help defining this thesis. We would also like to thank Stein Bergsmark and Sissel Andreassen for their guidance during the writing of the thesis.

Grimstad, May 2004

Morten Pedersen Than Kim Thong

III. Table of Contents

IV. Table of Figures

V. List of Tables

VI. Abbreviations

AES - Advanced Encryption Standard

AM-ADDR – Active Member Address

AODV- Ad hoc On Demand Distance Vector

ARAN - Authenticated Routing for Ad hoc Networks

BD-ADDR – Bluetooth Device Address

CarList [] - List of all cars connected

CBC - Cipher Block Chaining Mode

CCM - Combination of CTR and CBC

CDMA/CA - Carrier Sense Multiple Access with Collision Avoidance

CN – Car node

CRC – Cyclic Redundancy Check

CTR - Counter Mode

DAC – Device Access Code

DES - Data Encryption Standard

EAP - extensible authentication protocol

EPC – Electronic Product Code

FFD - Full Function Device

FHSS – Frequency Hopping Spread Spectrum

GC – Group controller

GCID - Group Controller Identifier

GCList [] – List of all GC's connected

ISM – Industrial-Scientific-Medical band, 2.4 GHz

ISO – International Organization for Standardization

KList [] - List over all Shared Secret Key for cars it has stored

LNR – License number

OEM - Original Equipment Manufactures

PM-ADDR – Parked Member Address

RC – Row controller

RCID - Row Controller Identifier

RCList [] – List of all RC's connected

RDF - Reduced Function Device

RERR – Route error

RFID – Radio Frequency Identification

RREP – Route reply

RREQ - Route request

SAODV - Secure Ad hoc On Demand Distance Vector

SAR - Security - Aware Ad-Hoc Routing for Wireless Networks

SC – Super controller

SCID – Supper Controller ID

SCPointer - Routing information to reach the SC

SDR-Dynamic Source Routing Protocol

SIG - Special interest group (Bluetooth)

SNEP - Secure Network Encryption Protocol

SSP - Security Services Provider

TAODV - Trust Ad hoc On Demand Distance Vector

TempID - Temporary ID assigned by SC

TKIP - Temporal key integrity protocol

TS – Timestamp

UID – Unique identifier

U-NII – Unlicensed National Information Infrastructure, 5 GHz

µTESLA –

WAP - Wireless Access Point

WEP - Wired Equivalent Privacy

WLAN - Wireless Local-area Network

WPA – Wi-Fi Protected Access

K_{SSK} - Shared Secret Key

GCK_{SSK} - Shared Secret Key (Group Controller)

RCK_{SSK} - Shared Secret Key (Row Controller)

$TempID_{RC}$ - Temporary ID assigned by SC

VII. Thesis Definition

Formation of Secure Wireless Ad-hoc Sensor Networks

Consider a wireless ad-hoc sensor network that is built up like a tree structure where parent/child relation in the tree represents master/slave relation in the network hierarchy. At the top there will be a master (root) node that controls several slaves (children). Those nodes will be fixed. The slaves can in turn be masters to lesser nodes which will have none to few rights. They will join the network and register with their respective superior node, which will in turn inform their master node. For example a parking place for cars, where each car represents the lesser nodes. For a group of cars there is one controller node that answers to a super node that controls a section of the parking place. Those will then in return answer to an overall controller node that is responsible for the entire parking place.

The first part of the master thesis will be to define what security is required in general from an ad-hoc sensor network. Starting out from that definition we will analyze vulnerabilities and weaknesses in the proposed architecture. Based on the security analysis we will look at different ad-hoc sensor network technologies and discuss their ability to fit into the specific case. Taking the security analysis and the different technologies in consideration we will propose a possible solution for an architecture that will give sufficient security in the overall ad-hoc network. The solution should consist of what technology to use and any limitations as to what the system can or cannot do.

If there is time we will model and simulate the proposed solution and try to create a prototype model of the architecture.

1 Introduction

In the wireless ad-hoc environments, the mobile devices may enter and leave the ad-hoc network rapidly. Consider a situation where two communicating parties are out of the wireless range of each other, but still participating in same ad-hoc network. The information (packets) between source and destination then needed to set routed through one or several intermediate nodes. Many ad-hoc routing protocols have been proposed [9, 10] but none of the proposal protocols provide security.

1.1 Work Plan

In this master thesis we aim to find a wireless ad-hoc solution for the specified case in the next chapter. This report will start with a description of the case followed by a security analysis to point out weaknesses and possible attack points in wireless ad-hoc networks. This analysis will then be applied on our case to highlight weaknesses.

Following the security analysis we will take a look at different wireless ad-hoc technologies and try to find the most suitable technology to fit our case. After finishing the literature study we will create a proposed solution for the application level of our case scenario.

This solution will include messages to handle all of the possible scenarios for our application. As an optional task, the proposed solution should be used to develop a model description which should build the fundament for a test model.

Following the test model we set our goal to create a prototype using our proposed solution if there is sufficient time.

1.2 Work Progress

To start with our thesis was aimed at using Bluetooth to create a solution for our case. The working title was changed in February to include a technology study and determine if there are other technologies that may be better to use in our case. This means we used a month studying Bluetooth in detail which was not that needed anymore unless we concluded that Bluetooth would be our choice. We achieved to complete the technology and routing protocol study. We also completed a proposed solution and created a model description. Since everything after the proposed solution was optional and we were running out of time, the test model and prototype was dropped from the work list.

2 Case Definition

We will study a wireless ad-hoc sensor network in this master thesis. The basic architecture of this system will be a tree structure with parent/child relations representing master/slave relations in the network hierarchy. This means there will be some nodes that are fixed in the network and are not supposed to move. In our case we take a look at a parking lot. We assume there are several parking rows. For each of the rows there will be intersections for nodes controlling groups of nodes (cars). Figure 2.1 shows our envisioned system in a tree structure.

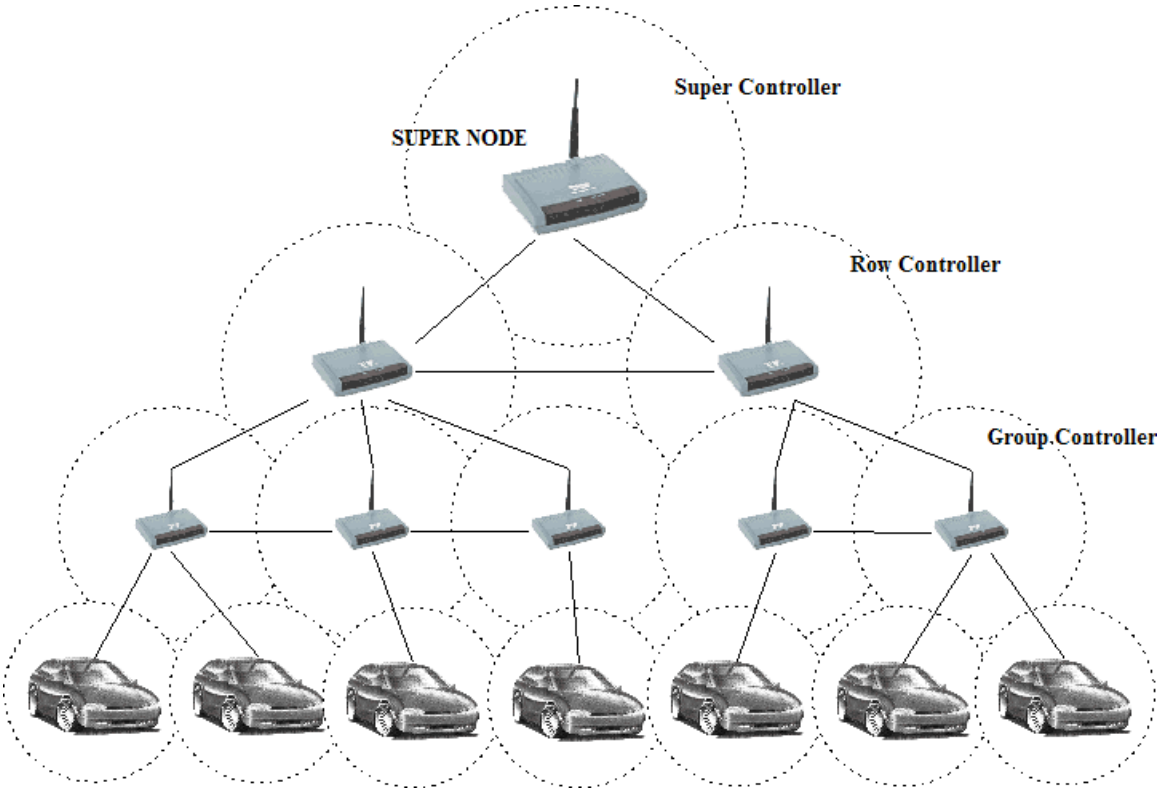

Figure 2.1: Case Setting

This system might be used for instance against car theft and for billing car owners for the time spend on the parking lot.

Regarding the placement of the nodes around the parking lot we want to let the different sections overlap each other to receive redundancy if some nodes should malfunction. For instance if a group controller malfunctions we do not want the cars to loose contact with the super controller.

2.1 Limitations

We limit our case to four levels in the hierarchy. In a real world implementation there may be more levels and connections between a car and a Row Controller or even a Super Controller.

2.2 Proposed System Solution

We take a look at our anti car theft system. A car (A) does arrive at our parking lot. The driver will have to find itself a parking space and park there. The driver then activates the system with e.g. using the car key to lock the car or simply when turning of the car. The transceiver node is then automatically activated. A will now register itself with its superior node the Group Controller (GC). The registered information could e.g. contain the vehicle registration number and eventually some more desired information. Now the GC will add a timestamp to that when A registered itself and pass this information on to its superior node, typical a Row Controller (RC). The RC will now add information about the GC to which A belongs and send this information on to the Super Controller (SC). The role of the SC can be defined as database consisting of all cars on the parking lot at the time being. Whether or not the SC stores the data locally or transfers it to a remote database which is in a secure location (away from the parking lot), it has to be secured against direct physical access and also have the highest level of security within the network. If the SC is compromised, the entire network will be in effect useless and should automatically trigger an alarm at the responsible authority. In larger cities it could be an advantage to keep the data from each parking lot located in one place instead of a database at each location. This would decrease the overall cost for each SC and rendering attackers unable to get direct physical access to the data stored.

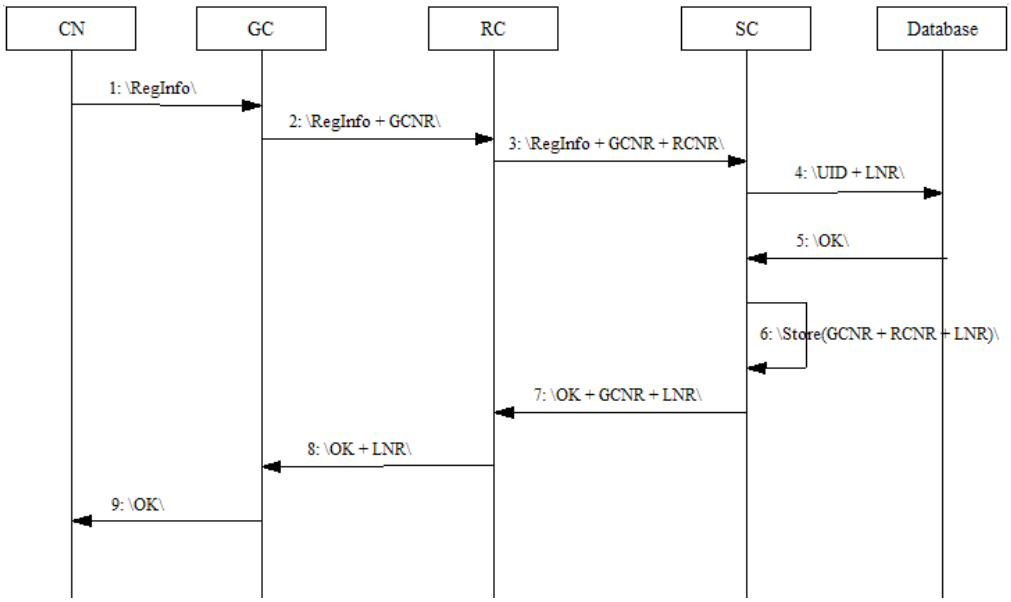

Figure 2.2: Message Flow

RegInfo does contain the UID and a Timestamp (TS) which is encrypted with cryptographic measures to ensure its authenticity and freshness.
The response message “OK” from the database is also encrypted. It will have to contain more information than only “OK”. A possible solution to this will be to add a cipher which is derived from the Secret Key used to encrypt the data and add this to the “Ok” and the Timestamp.

After the registration process is complete the GC wills in periodic intervals poll the CNs attached it. This makes sure to early detect any theft of a car attached to the system. If a registered CN does not reply to a poll the GC will send a message to the SC. The SC will then send a message to either the owner of the car using SMS or send to a central authority e.g. the company owning the parking lot.

2.3 Register Process

First we need to determine what kind of information will be transmitted in this part of our hierarchy. We need an identifier for each car and since the license plate does give us a unique number, we can use it as apart of our ID. However this number is very easily obtainable by any adversary and we will need to implement encryption for at least the authentication process. The registration process is the most vulnerable situation we will have in our system. If an attacker wants to gain access to the sensitive data sent over the network it is now. This means sending the unique identifier and timestamp will have to be encrypted. Using end-to-end cryptography, we will have the GC on a need to know basis. This means the GC will be told by the SC what nodes to poll. In essence this means the GC will function as a relaying device used to send information up and down the hierarchy and it will work as a device to poll its attached nodes to make sure they are there until they are properly signed off.

2.3.1 Car Node to Group Controller

Now we get to the registration process for a car. First we assume that a car comes into the range of a GC and when the car is turned off it registers itself with the system. Now this happens in a way where it sends its unique chip ID, which has to be strongly encrypted, and its license number to the GC it is connected to. To ensure protection against replay attacks and freshness we add a Timestamp to the message. Now we do not want a simple GC to be able to decrypt the message so all it does is to add its GC number and relay it on to the closest RC. The key used for encryption will be a Shared Secret Key which is only known to the Database/SC and the CN. This will lead us to the following message:

For CN to GC we will get:

$$MSG_{CN \to GC} = Encrypted(UniqueID + Timestamp) + LicenseNr$$

And from GC to RC we will get:

$$MSG_{GC \to RC} = MSG_{CN \to GC} + GCNr$$

2.3.2 Group Controller to Row Controller

Now since the chip ID number is treated confidential we don't want any part of our local system to be able to decrypt this. So the RC attaches its own RC number and relays the entire package to the SC.

$$MSG_{RC \to SC} = MSG_{GC \to RC} + RCNr$$

2.3.3 Row Controller to Super Controller

After receiving the message from a Row Controller it will store the path description related to the License Number in a database.

The SC now removes the location identifiers given by the RC + GC and sends the chip ID / License Number over an encrypted channel to a database containing a reference between License number and Shared Secret Key. The database looks up the License Number and retrieves the appropriate Shared Secret Key and uses it to decrypt the given information. If the Chip ID is verified successfully we can be sure that the car is what it claims to be. Now the SC will send an "OK" message down the same chain as the original message came from. For practical purposes we will only let the SC store the information containing the cars where they are parked. A RC on its part will only store its attached GC's and a GC will store what cars are connected to it.

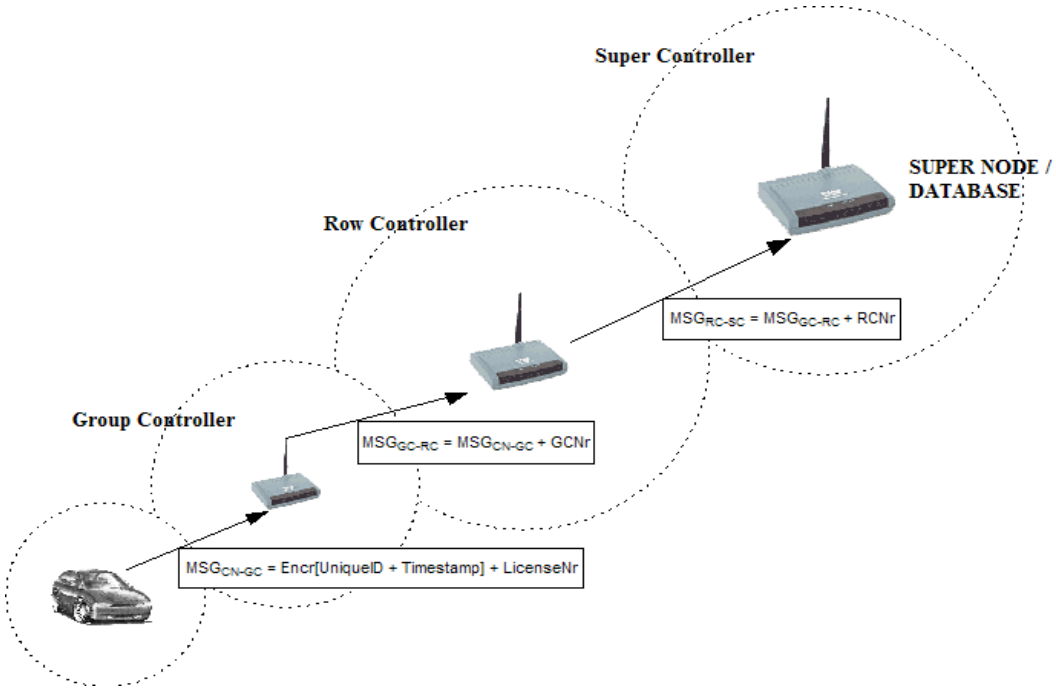

Figure 2.3: Register Process

Since every request from a car will be forwarded to the SC it will contain the full routing information to each car. In our envisioned scenario that will be around 4 nodes at least. This solution leaves us with a requirement of one powerful node and all other nodes don't have to store way too much data.

Figure 2.3 shows a typical registration process. This includes only the shortest path scenario, but it may as well include several Group Controllers and/or Row Controllers.

3 Security Aspects of Wireless Networks

In this chapter we will look at typical weaknesses and attacks on wireless ad-hoc networks. Looking at physical security we can easily see that the fixed controller nodes need to be kept safe, which means at places not easily accessible by everyone. Since the nodes however need to be on the parking lot and have to be accessible for maintenance they will also be accessible for a possible attacker. So this is certainly a point to consider although not of relevance for this report so we will not discuss it any further.

3.1 Threats

Use of wireless links allows attackers to attack the ad-hoc network from every where within the wireless transmission range. These attacks are categorized as active and passive attacks. Active attacks could range from deleting, altering messages to injecting erroneous messages and impersonate a node. These active attacks can again be divided into the following subclasses [1]:

- **Masquerade** – occurs when a node pretends to be a different node.
- **Replay** – old information have been collected and replayed to perform unwanted access.
- **Modification of messages** – the genuine messages or part of them could have been modified under transmission.
- **Denial of service** – attackers could prevent legitimate users of a service from using that service.

Passive attacks are eavesdropping on wireless transmission. The attacker tries to access information that is being transmitted over the air. These passive attacks can be divided into two subclasses [1]:

- **Release of message contents** – the attacker could catch the messages or data which have been transferred wirelessly.
- **Traffic analysis** – the attacker could analyses messages traffic to discover the location and identity of communicating nodes.

The main categories of attacks on wireless networks are [1]:

- **Interruption of service –** the resources of the system are destroyed or become unavailable.
- **Modification –** attack on the integrity of the system. The attackers get access to the network, and modified the transferred messages.

- **Fabrication –** attack on the authenticity of the network. The attacker may insert record in a file.
- **Interception –** attack on the confidentiality of the network such as wiretapping or eavesdropping to capture data in the network.
- **Jamming –** the attacker flood the frequency band to interfere the legitimate traffic, such that such that the legitimate traffic could not reach the destination.
- **Client to client attacks –** Wireless network users need to defend against each other (internal users). Not just from an outsider.
- **Attacks against encryption –** weak encryption lead to an attacker can break the encryption.
- **Misconfiguration –** misconfigured access points allow unauthorized user easily access the networks.
- **Brute force attacks against passwords of access points –** the use of try/fails method, the attackers could discover single/simple password or key. The attackers try to guess the passwords or keys.
- **Insertion attacks –** this type of attack is based on deploying a new wireless network without following security procedure.

3.2 Security Services

Security services are properties that could be used to enhance the security.

- **Authentication** - this service is to ensure that the message is from an authentic-source, ensure that each communicating party is the entity that it claims to be. There are three different variations of authentication:
    - ***Entity Authentication*** – the identity of a communicating party are known. The source or destination of the data is known.
    - ***Message Authentication*** – the property that a given message was sent by the claimed sender.
    - ***Data Origin Authentication*** – this property implies both the entity authentication and the message authentication.
- **Confidentiality** – no one other than the sender and the intended recipient can read the message. The sender should encrypt the messages before transmit wirelessly.
- **Nonrepudiation** – this service prevents the sending or receiving party from denying the sent or received message. This means that when a message is received, the sender can confirm that the message was received by the assumed receiver.
- **Access control** – this service ensure that only the authorized users can use the system and deny unauthorized users from using the system.

- **Integrity** – this service ensure that the messages are sent properly without duplication, modification, reordering or replay. This mean the data received is guaranteed to be identical to the data that was sent.

4 Threat Analysis

After looking at typical attacks against wireless ad-hoc networks, we will now take a closer look at which of those attacks are of special interest in our case.

4.1 Physical Threats

Since our system can't be defined a true ad-hoc sensor network as it implements a stationary part, we get a slightly different threat scenario as a true ad-hoc network. Our super-nodes which build the core network will have to be protected against physical attacks. And even if an attacker gains direct access to them, they will have to be protected against being misused to bring down the network or falsify the information being sent to the controller node. This means we will need to use passwords to restrict access to the nodes. Here it is of importance that the passwords are made very difficult to guess and in a manner that brute force attacks will take very long time to accomplish. The physical protection will have to be implemented for each individual real life scenario and is hardly of any interest to this master thesis.

4.2 Wireless Threats

The securities we need to look at are protecting the node from tampering and protect the network from the common attack methods useable against wireless networks.
First of all we will want to protect it against masquerade attacks, so we can be assured that the one we are communicating with is indeed the one it claims to be.
Since it is quite impossible to eliminate the dilemma of hostile nodes eaves dropping our transmissions, we need to ensure that the critical data such as Unique ID's and Shared Secret Keys are kept safe. Resulting out of eavesdropping and unwanted collecting of packets send over the ether we will need to implement protection against replay attacks. This comes to mind in case a car leaves a parking lot in a legit way, although an attacker has successfully intercepted that packet, we don't want to enable the attacker to replay that packet next day and steal the car. Furthermore if an attacker attempts to perform a modification attack we will need to Cyclic Redundancy Check (CRC) check our packets that are sent over the ether. Other serious attacks on our network as Denial of Service and Jamming attacks have to be handled by the wireless technologies we are going to look at in chapter 5.

Since eavesdropping is fairly easy in a wireless environment we need to ensure that our cryptographic methods will sustain attempts to break them.

5 Wireless ad-hoc technologies

Before we start looking at the different technologies available on the market, we will describe some functionality that they need to fulfill. First we want the system to have some sort of redundancy which would seem natural to implement into the case scenario. This will imply that each node needs to be able to hold routing tables and relay information to the next node on its way up the chain. This is to make the system more robust to device failure and to negate the effect if a node should be compromised. Secondly the technology has to support cryptography to enable secure communication and routing between the nodes. Of further interest are physical limitations of the different technologies. Since we don't want the nodes in the cars to be easily visible and accessible, the technology has to have the ability to send and receive through a metal barrier. Also the transmission range has to be approximately 10 meters to fulfill the requirements set by the case. Another point of interest is the use ability in different countries due to legal limitations. Furthermore we acknowledge that our system isn't going to be a true ad-hoc network as it contains too many fixed parts which build our core network.

We choose to take a closer look at the following technologies: RFID, Bluetooth, W-LAN and ZigBee. This is only a small selection of technologies in existence, however this are those we considered to make most sense to examine, as they are present different areas of use and have different abilities. There are a few technologies which might suit our case, which are not taken into consideration in this report as they simply are not completed yet and cannot be bought to make a product out of it. SmartDust was one of them and it looks promising, however the homepage of it isn't updated anymore and it looks discontinued.

5.1 Radio Frequency Identification (RFID)

5.1.1 RFID Overview

"Radio frequency identification, or RFID, is a generic term for technologies that use radio waves to automatically identify individual items. [2]" Since this expression is very general and doesn't say much more than that each technology used to identify an item is in fact RFID we will discuss the typical RFID system. A typical RFID system consists of a tag and a reader. The tag will in most cases store a unique number used as identifier and some information regarding the item it is identifying. The existence of very many different RFID standards makes giving a general overview over this technology very difficult. The two largest standards are a series of standards proposed by ISO and Electronic Product Code (EPC).

5.1.2 RFID Tags

The tag itself consists of an antenna and a microchip. There are active tags that have a power supply on their own for computing instructions. However, the transceiver antenna still is without power. The other type is passive tags without any power supply. They will draw their power for computing and sending the response back from the power the reader emits. A passive tag will have a transmission range of less than 0.3 meters whereas an active (with power supply to the transceiver) tag can be read at up to 90 meters range. Which frequency is being used also affects read range. See table 5-1 for more information.

5.1.3 RFID Readers

To get any use out of RFID there is a need for a reader to read the information stored on the tags. The abilities of a reader depend on what it is designed to do. This includes how many tags it can read at one time and how much data it can read from those tags in a second. As an example from [21] there are readers that can read from 100 tags in one second and can manage up to 2000 tags in reading range.

5.1.4 RFID Frequency bands

For RFID there are specified four different frequency bands. Each of the different frequencies has their benefits and drawbacks. However which band is best suitable is decided by the application and the environmental challenges where it is going to be installed.

Frequency Band	Frequency Range	Range
Low Frequency	~125 KHz	< 0.3m
High Frequency	13.56 MHz	~0.9m
Ultra High Frequency	850-900 MHz	3m-6m
Microwave	2.45 GHz	-

Table 5-1 Source: www.rfidjournal.com

Lower frequency means better ability to penetrate non-metallic materials and will have less power consumption. Higher frequencies will have higher rate of data transfer and greater range, but will consume more power and will bounce of materials.

5.1.5 RFID Security

Security in RFID is realized in a way that the transceiver can encrypt the data stored on a RFID tag. This will render an attacker limited to reading encrypted information from the tag although he will not be able to decrypt it without knowing the proper key. Since the

transmission range is very short, it should be fairly hard to eavesdrop anything unnoticed, however the technological aspects of RFID makes it even harder as in most cases the sender sends at a much higher power than the reply signal from the tag will be able to. This is due to the fact that the tag uses the emitted power from the sender to power its own reply signal making it just as much as a whisper compared to the originating signal.

5.1.6 RFID Cost

A typical RFID tag will cost from 50 cents and up to $50. Adding a sophisticated sensor can boost the price to over $100. The readers which are used to program and read the RFID tags will typically cost $1000. Expectations have it that chips will drop below 5 cents the next year.

5.2 Bluetooth (802.15.1)

5.2.1 Bluetooth Overview

Bluetooth technology was initially conceived by Ericsson in 1994. As the idea grew, the special interest group (SIG) was formed to create Bluetooth standard. In the beginning the SIG consisted of five companies: Ericsson, IBM, Intel, Nokia and Toshiba. Later, four other companies: Microsoft, 3Com, Lucent and Motorola joined the SIG to form the Bluetooth Promoter Group.

"Bluetooth is an open standard specification for a radio frequency (RF)-based, short-range connectivity technology that promises to change the face of computing and wireless communication. It is designed to be an inexpensive, wireless networking system for all classes of portable devices, such as laptops, PDAs (personal digital assistants), and mobile phones. It also will enable wireless connections for desktop computers, making connections between monitors, printers, keyboards and the CPU cable-free. [4]"

5.2.2 Bluetooth technology

Bluetooth uses 2.4 GHz frequency radio band and transmits using a fast Frequency-Hopping Spread Spectrum (FHSS). Bluetooth provides ad hoc networking, and implements peer-to-peer communication without the need of base stations or administration. Every Bluetooth device has a unique 48-bit Bluetooth Device Address (BD-ADDR). This address is used when establish a connection or generate an access code. To establish a connection, two or more Bluetooth devices have to be within the range up to 10 meter.

Elements in a Bluetooth system:

- An RF portion, radio that transmits and receives data using 2.4 GHz frequency band.
- A Baseband module to enable wireless communication between two devices. Before a master unit transmits data to a slave unit, the Baseband convert the digital data into radio signal and sends to the slave unit. At the slave unit, the Baseband have to convert these radio signals data into digital data such that the host application can process.
- Link Manager, manages data transmissions
- An interface to the host device (such as a PDA)

Bluetooth modules can be used independent of the host, embedded or integrated the Bluetooth baseband module with the host.

In the independent option the RF portion can be implemented as a module or as a single chip and be used independently or with the Baseband module. In this option the lower-layer are supported in the Baseband modules, while upper-layer are supported by the host processor.

In the embedded option the RF and Baseband are embedded in a single chip. In this option the lower-layer and upper-layer reside in the same chip and will freeing the host processor from the protocol processing. The advances of the embedded option are that the design is less complex, use less power and lower cost in production.

5.2.3 Bluetooth Topology

Bluetooth provides both point-to-point and point-to-multipoint connections. When two or more Bluetooth devices are within the wireless transmission range they form a small group and is called a *piconet* (see Figure 5.1). *Piconet* consists of one master and up to seven active slaves. The device that initiates a connection becomes the master and the others are slaves and response to the master. In an environment where the several piconets overlap each other and a device in one piconet can communicate with another device in second piconet. These overlap-piconets are called *scatternet* (see Figure 5.1).

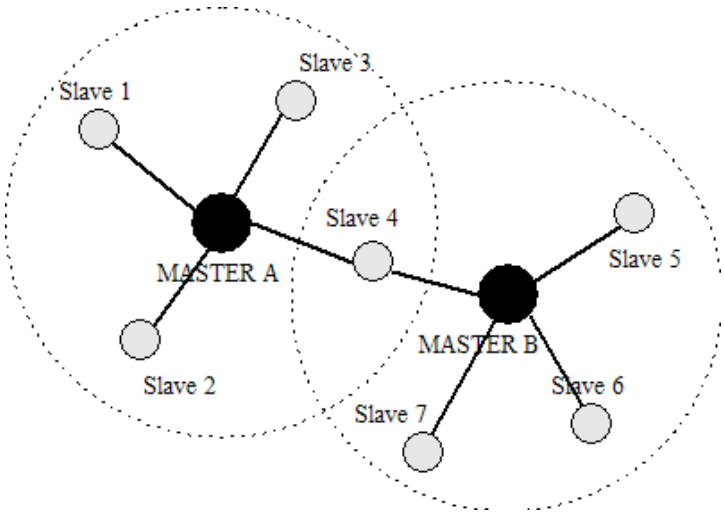

Figure 5.1: Piconet and Scatternet (source: [3])

Since Bluetooth provides ad hoc networking, and the movements of the devices, devices may enter or leave the piconet/scatternet rapidly. To save power, devices may enter one of the states or modes described bellow.

Bluetooth Major States:

- **Standby:** In this state, only the unit's native clock is running and requires low power. Hence only seven (at most) units can be active at a time, the units may enter the standby-state to free up the capacity on the piconet.
- **Connection:** In this connection states, active connections are established by two or more units and data are exchanging. In the connection state, the unit may be in one of the following modes:
    - ✓ **Active:** In this mode, the unit is active and participating the piconet. The active units are assigned an Active Member Address (AM-ADDR).
    - ✓ **Hold:** In this mode, the units only support SCO packets (voice) and not ACL-packets (data). This mode reduces the power consumption and the unit may enter paging, inquiry scan states.
    - ✓ **Sniff:** This reduced-power mode support both SCO and ACL packets.
    - ✓ **Park:** In this mode the unit releases its AM-ADDR and assigned a Parked Member Address (PM-ADDR). The unit are not participating the piconet, but stay synchronized with its channel, listen to broadcast.

Bluetooth Paging States: In this state the master repeatedly transmits the DAC (Device Access Code) of the slaves, while the slaves scan for their DAC. This is to locate and establish a connection between master and slave.

Bluetooth Inquiry States: Similar to Paging state, but in this state the master are looking for potential slaves which the master does not know the DAC. DAC are needed to establish a connection between master and slave.

5.2.4 Bluetooth Security

To secure the data sends between devices, the payloads of packets are encrypted using stream cipher E0 [19]. The E0 stream cipher consists of the payload key generator; the key stream generator and the encryption/decryption part (see Figure 5.2). When encryption is required, the master and slave must agree which encryption modes to use.

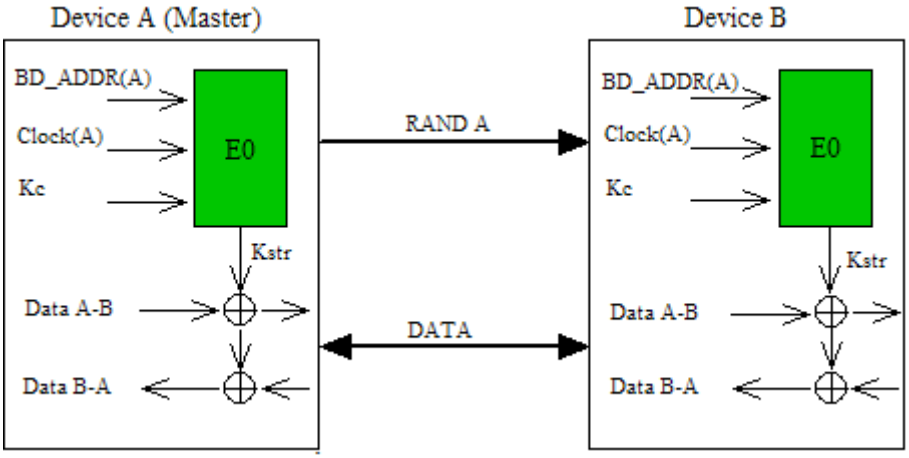

Figure 5.2: Bluetooth Security (source: [3])

Security Modes:

- **Security Mode 1-** the insecure mode, nothing is encrypted in this mode. Every device within the range can communicate.
- **Security Mode 2-** In this mode the addressed traffic is encrypted, while the broadcast messages are not.
- **Security Mode 3-** In this mode both authentication and encryption are enabled. All traffic is encrypted with the master key.

Bluetooth devices are authenticated using challenge-response algorithm. BD_ADDR, private authentication key, private encryption key and RAND are used in the challenge-response process. A successful authentication is based on the fact that both participants share the same key. First, the verifier sends the claimant a random number to be authenticated. Then, both participants use the authentication function E1 with the random number, the claimants BD_ADDR and the current link key to get a response. The claimant sends the response to the verifier, who then makes sure the responses match.

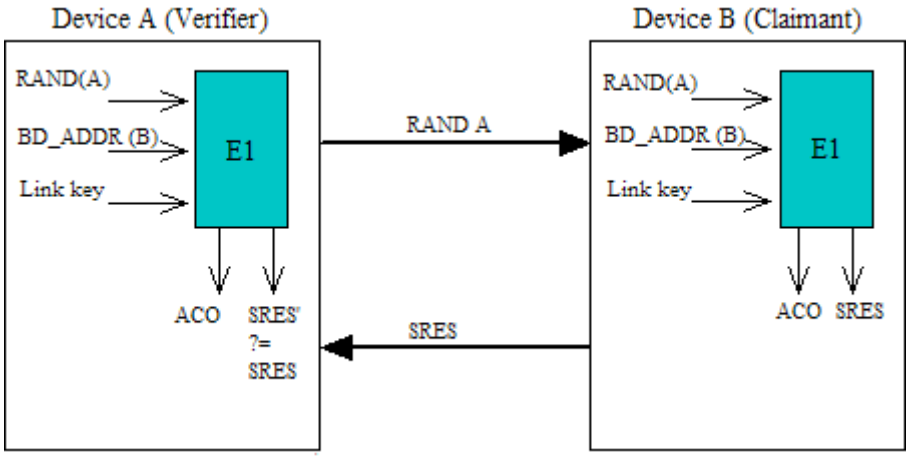

Figure 5.3: Bluetooth Authentication (source: [3])

The Bluetooth authentication process is depicted in figure 5.3.

5.2.5 Bluetooth Cost

According to *Milorad Mitrovic* [20] the price for Bluetooth chips will reach $5 each in 2005.

5.3 W-LAN (802.11x)

The first wireless Ethernet standard, plain 802.11, was adopted and published by the IEEE in 1997. This standard provided several modes of operation and data rates up to only two megabits per second (Mbps). Later, the higher performance standards, 802.11b, 802.11a and 802.11g are adopted. The "b and g" version operated in the same frequency range as the original 802.11, the 2.4 GHz Industrial-Scientific-Medical (ISM) band, but the "a" version operated on the 5 GHz Unlicensed National Information Infrastructure (U-NII) band.

Wireless Local-area Network (WLAN) is a wireless communication technology using radio waves to transmit and receive data over the air. Based on WLAN non-cables characteristic and the fall in cost of WLAN products over the last years, WLAN’s are used widely within a building or campus.

5.3.1 WLAN Standards

WLAN standards includes 802.11, 802.11a, 802.11b and 802.11g

Standard	Data Rate	Radio band	Modulation Scheme	Security
802.11	Up to 2Mbps	2.4 GHz	FHSS[22] or DSSS[22]	WEP[24] & WPA[25]
802.11a	Up to 54Mbps	5 GHz	OFDM [23]	WEP[23] & WPA[24]
802.11b	Up to 11Mbps	2.4 GHz	DSSS with CCK	WEP[23] & WPA[24]
802.11g	Up to 54Mbps	2.4 GHz	OFDM above 20Mbps, DSSS with CCK below 20Mbps	WEP[23] & WPA[24]

Table 5-2: WLAN Standards

5.3.2 WLAN Configuration

In a WLAN configuration consists of a transceiver Wireless Access Point (WAP) connects to fixed network using standard Ethernet (IEEE 802.3). End user access the WAP using WLAN adapter such as PCIMCIA cards (see Figure 5.4).

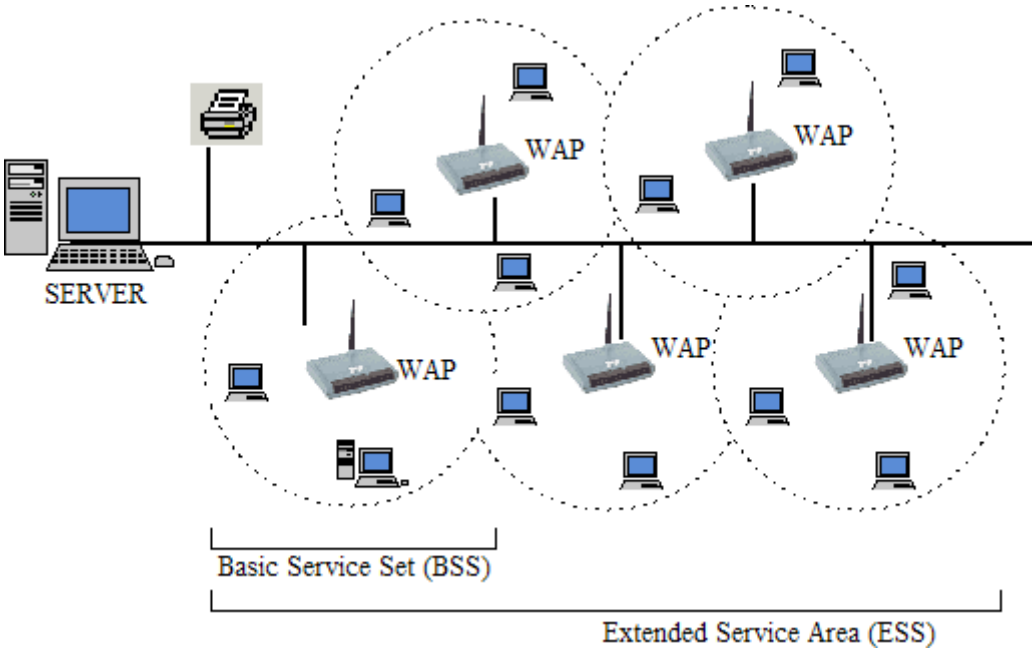

Figure 5.4: WLAN Configuration (source: www.wlana.org)

- **Peer-to-peer -** The simplest configuration of WLAN is a peer-to-peer. When two wireless adapters are within the wireless transmission range of each other, they can communicate directly.
- **Peer-to-Multipoint** - When two or more wireless adapters are within the wireless transmission range of each other, they can communicate through a centralized administrator. This means that the adapters can not communicate with each other directly, but through a hub.
- **WAP as repeater -** Access points can be used as a repeater to extend the wireless transmission range. Several repeaters (BSS) will then form an Extended Service Area (ESS).
- **Linking WLAN -** WLAN can be linked to other WLAN's, WAN or the internet.

5.3.3 WLAN Range and throughput

A single wireless access point can support several simultaneous users at a range of 90-100 meters in free space and much less via obstructions and typical data rates in WLAN range from 1 to 54 Mbps dependent on which standard is used (see Table 5-2).

5.3.4 WLAN Security

WLAN (802.11 standards) uses the encryption called WEP (Wired Equivalent Privacy) and using SSID and MAC-address table to secure the access to the APs. WEP uses the MAC-address to limit the access to the network. But the MAC-address can be sniffed and stolen. This is one of the security weaknesses in WLAN. To improve the security WLAN also uses WPA (Wi-Fi Protected Access) upon WEP. The encryption schemes are improved with the use of temporal key integrity protocol (TKIP) and hashing algorithm. To solve the stolen MAC-address problem, WPA uses user authentication through extensible authentication protocol (EAP) to ensure that only authorized users can access the network. EAP is built on a more secure public-key encryption system.

5.3.5 WLAN Cost

Wireless Access Point (WAP) range in cost from ~$100 to ~$1150 and wireless adapters (PC-cards) from ~$40 to ~$120 (prices from www.psdata.no 12.03.2004).

5.4 ZigBee (802.15.4)

The goals of the ZigBee product solution is to deliver a low cost and low power technology that will take its place in the short range and low throughput sector. It is aimed at transmitting text and nothing else. It builds upon the open standard 802.15.4 as its radio source. The ZigBee alliance developing ZigBee consists of more than 70 companies (as of May11, 2004). The goal of the association is to develop wireless network tools based on a global open standard. The main companies leading the ZigBee alliance are Honeywell, Invensys, Mitsubishi Electric, Motorola, Philips and Samsung.

5.4.1 IEEE 802.15.4

The standard describes a simple packet data protocol for lightweight wireless networks. Being an IEEE standard it is standardized by the LAN/MAN standards committee from the IEEE consortium. It uses three different frequency bands. Those bands have different areas of operation.

Frequency	Channels	Transmission Rate	Area of use
2.4 GHz	16 Channels	250kbps	Worldwide
868.3 MHz	1 Channel	20 kbps	Europe
902-928 MHz	10 Channels	40 kbps	America

Table 5-3: Frequency Bands

To access the channels 802.15.4 uses Carrier Sense Multiple Access with Collision Avoidance (CDMA/CA). It further features multilevel security. A further design goal of the standard is to minimize battery consumption hence delivering very long lifetime of batteries. The estimated transmission range is said to be ~30 meters. The protocol implements functionality up to the Link Layers Control. Maximum number of nodes attached to a network can be up to 2^{64} (18446744073709551616).

5.4.2 ZigBee Device Types

IEEE 802.15.4 implements three different types of nodes. First there is the Network Coordinator type which is the most powerful of the nodes. It possesses enough memory and computing power to keep track of the entire network knowledge. The Full Function Device (FFD) is capable of working as a router and as a connectivity device towards the real internet for instance. The last of the device types is the Reduced Function Device (RDF). It only has limited functionality and works perfectly as a slave node in the network topology.

5.4.3 ZigBee Topology

The devices in ZigBee may operate in either star topology or peer-to-peer topology. In the star topology, the FFDs may become a central controller (PAN coordinator) and form its own network. The central controller acts as a hub and allows either FFDs or RDFs to connect it. The communication between two devices may route through the central controller (see Figure 5.5).

In the peer-to-peer topology the devices may communicate directly with each other within the transmission range independent of the central controller. A peer-to-peer network can be ad hoc, self-organizing and self-healing. It may also allow multiple hops to route messages from any device to any other device on the network

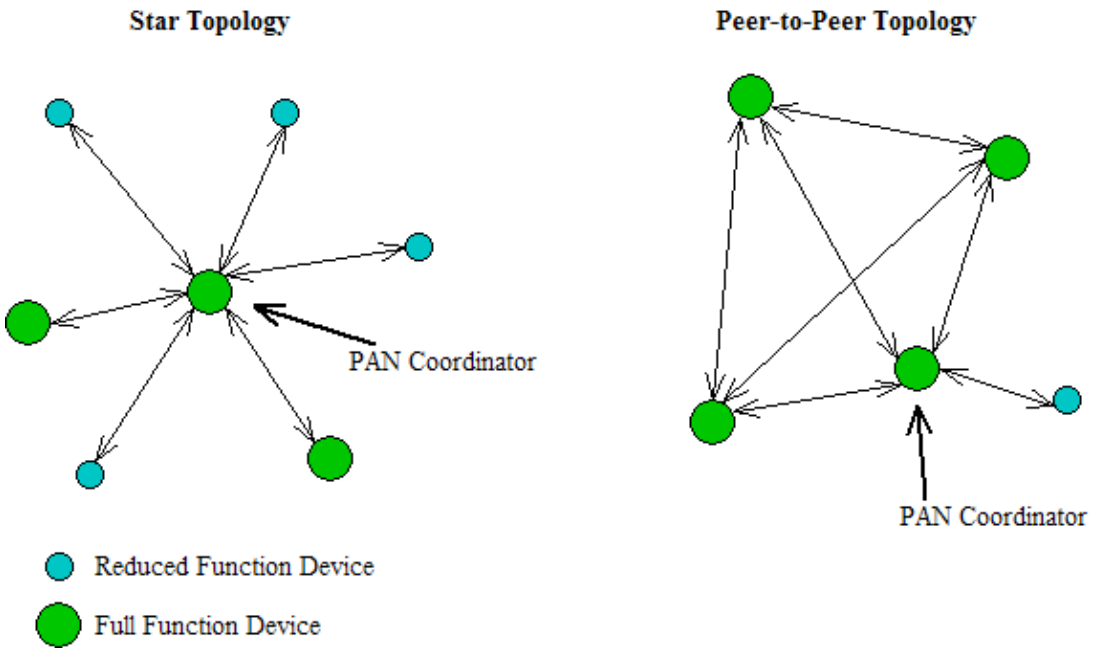

Figure 5.5: ZigBee Topology (source: [32])

5.4.4 ZigBee Routing

ZigBee uses a simplified AODV (described in section 6.2) to route the packets. Routing nodes in ZigBee are characterized as RN+ and RN-. Both the routing nodes (RN) know how to relay the packets. The different between RN+ and RN- is that the RN+ possesses a full routing table and forwards data if there is routing entry, otherwise initiates route discovery. While the RN- forwards the packets by using cluster tree routing.

5.4.5 ZigBee Latency

Since there are different modes of operation in a ZigBee environment there are different latency times. When a new node enters the area of a ZigBee controller it will take approximately 30ms to get enumerated. A sleeping node will have a wakeup time of about 15ms. The average transmission delay will be about 15ms.

5.4.6 ZigBee Security

ZigBee provides confidentiality, integrity and authenticity by using MAC layer security. The MAC layer security uses the Advanced Encryption Standard (AES) [7] as the core cryptographic algorithm and is based on three operation modes:

1. Counter Mode (CTR) – Encryption using AES
2. Cipher Block Chaining Mode (CBC) – Integrity using AES
3. CCM Mode – Combination of CTR and CBC

MAC layer security is used to secure MAC command, Beacon, acknowledgement frames and messages that transmitted over a single hop. However for multi-hop messages, the security is provides by upper layer (Network Layer). To apply integrity to a frame, the MAC header and payload are used to create a Message Integrity Code and the MIC is added to the frame (MIC, see Figure 5.6). However to apply confidentiality to a frame, the Frame Count and Sequence Count are used to create a nonce. This nonce again is used to encrypt the payload and ensures the freshness to protect against replay.

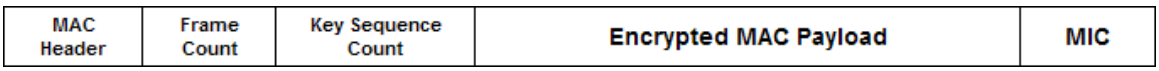

Figure 5.6 ZigBee Mac Frame

The Network layer also uses the Advanced Encryption Standard (AES) as the core cryptographic algorithm. The multi-hop messages are protect by the Network layer. When the Network layer transmits or receives a frame, it uses the Security Services Provider (SSP) to process the frame. SSP applies security to outgoing frames and removes security from incoming frames.

5.4.7 Advanced Encryption Standard

Until recently, the Data Encryption Standard (DES) [26] and 3DES was the most widely used algorithm for symmetrical encryption. But today's preferred choice is Advanced Encryption Standard (AES). Rinjdael algorithm has been chosen to be the new Advanced Encryption

Standard (AES). Rinjdael algorithm [7] is a symmetric block cipher [28] using 128-bit block size. With the cipher key length of 128 (default), 192 or 256-bit key, the AES is designed to be more secure than DES.

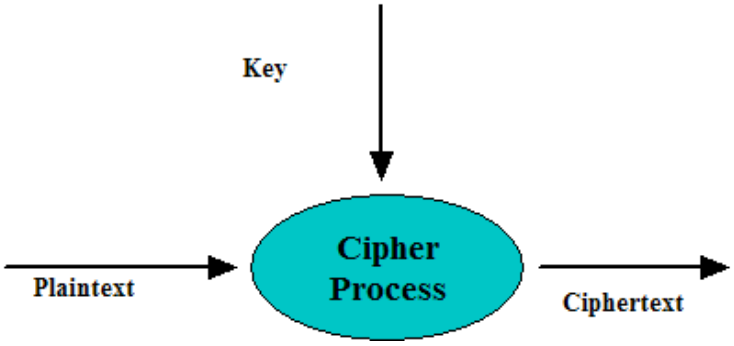

Figure 5.7: Cipher Process overview

The cipher process consists of the following transformation functions:

1. **AddRoundKey ()**, read AES [7] for details.
2. **SubBytes ()**, read AES [7] for details.
3. **ShiftRows ()**, read AES [7] for details.
4. **MixColumns ()**, read AES [7] for details.

In the pseudo code bellow shows the process where these transformations are applied to produce cipher text.

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
byte state[4,Nb]
state = in
AddRoundKey(state, w[0, Nb-1])
for round = 1 step 1 to Nr–1
SubBytes(state)
ShiftRows(state)
MixColumns(state)
AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
end for
SubBytes(state)
ShiftRows(state)
AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
out = state
end
```

Table 5-4: Cipher Pseudo Code (source [7])

The inverse cipher process consists of the following transformation functions:

1. **AddRoundKey ()**, read AES [7] for details.
2. **InverseShiftRows ()**, read AES [7] for details.
3. **InverseSubBytes ()**, read AES [7] for details.
4. **InverseMixColumns ()**, read AES [7] for details.

In the pseudo code bellow shows the process where these transformations are applied to produce plain text.

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
byte state[4,Nb]
state = in
AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
for round = Nr-1 step -1 down to 1
InvShiftRows(state)
InvSubBytes(state)
AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
InvMixColumns(state)
end for
InvShiftRows(state)
InvSubBytes(state)
AddRoundKey(state, w[0, Nb-1])
out = state
end
```

Table 5-5: Inverse Cipher Pseudo Code (source [7])

5.4.8 ZigBee Cost

According to [8] the price of a ZigBee chipset is going to be approximately $2.50 if bought by Original Equipment Manufactures (OEM's).

5.5 Summary

After looking at four different technologies which we considered being able to solve our case, we now need to make a choice as of which of them to use and implement in our solution. The first technology we looked at was RFID. As shown this is a technology where its shortcomings outweighs its pro's for our case. First of all its transmission range for passive chips is way to short with 0.9 meter. Furthermore plays in that if there is the slightest obstacle between sender and receiver range decreases or does make it impossible to send at all. Even the active chips with an increased range (9 meters) will not suffice as they also loose a lot of transmission range due to obstacles. But our main concern is the inability of the RFID chips to compute instructions. They will not be able to route information to another active node. Why did we take RFID into consideration at all? Our main reason was to couple a RFID tag on each car and put a reader as GC onto the parking lot. And then use an alternative technology as WLAN to relay the signals to a SC. However seeing the costs of either technology it would become a quite costly implementation and would leave us with a series of technical difficulties. Another problem with using WLAN will be the high power consumption. It will require each WLAN node to be supplied with a power cord. This would strive against the very idea to keep the whole solution wireless as we might as well use a wired technology instead.

This means we need to find another solution. Since the original thought was to use Bluetooth for our case it was very natural to take a closer look at it. In a first glance it looks like a technology perfectly suited to meet our demands. Its computational powers are sufficient, if not even vast, for the limited amount of data we are going to transmit in our scenario. Bluetooth's built in security features however have been openly criticized for being weak and that if you need a secure solution you will have to implement it yourself at application layer or remake the used routing protocol. This would lead to a lot of extra work but is certainly doable. The transmission range is 10 meter and 100 meter. The 10 meter solution is a tad short for our scenario and when using the high range solution it consumes too much power. Another shortcoming of Bluetooth is its high latency when it comes to registering new devices to the network. This may take up to 1 second or in worst case even more. Now given a driver is very fast in leaving his parking space he might be out of range before the necessary sign off data actually is sent. That is a problem which is very hard to overcome with any application changes. Bluetooth networks can consist of maximum 255 nodes in one network. So on larger parking lots there would be several networks that would need to be interconnected to resolve this problem.

Our conclusion is, after looking at the different technologies we deemed suitable for our case, that ZigBee offers the best solution. It is designed to be used in a sensor network that has static components to it. Therefore it has very limited power consumption and thus long lifetime on its devices. Furthermore its transmission range is ideal for a parking lot with 30 meters range and if needed the range can be increased to 100 meters for the RC and SC's. Its latency is short enough to ensure that processes will be complete before a node leaves the sending range. With 2^{64} maximum nodes per network any parking lot should be covered, if not all of them over the world. The total cost of an implementation will be fairly cheap too with around 2$ per node. As a common note on all of the technologies described they are all useable in any part of the world. They all offer a global frequency band. Table 5-6 gives a short overview over the key differences in the technologies.

	RFID	Bluetooth	W-LAN	ZigBee
Battery Lifetime	Infinite**	7 days	N/A*	1 Month to years
Security	N/A**	3 Security Modes	WEP/WPA	AES
Range	~0.3-9m	~10m	~100m	~30-100m
Latency	N/A**	~50ms		~30-45ms
Throughput	N/A**	2Mbit	1-54Mbit	20-250Kbps
Network Size	~2000**	8 active/248 passive nodes	10 per access point	2^{64}
Frequency Bands	~125Khz 13.56Mhz 850-900Mhz 2.45Ghz	2.4Ghz	2.4Ghz 5Ghz	868.3Mhz 902-928Mhz 2.4Ghz
Area of use	Identification	Small Application Devices	Internet	Sensor Networks
Cost***	0.50-100$	~5$	~100$	~2$

Table 5-6: Technology Comparison

* Since W-LAN is typically used as network technology for computers it consumes so much energy that it is not useable on normal batteries.

** Information regarding RFID is pretty much impossible to determine as there is a vast number of different standards which all will have different specs. A passive RFID chip will in effect have an infinite lifetime as it posses no power source at all.

*** Cost is a price based on market prices found on the internet and is a price per chip.

6 Secure Routing

In this chapter we will take a look at different routing solutions proposed for wireless ad-hoc networks. Seeing that there hasn't been too much focus on security around the standard protocols used for routing so far, we will not take a look at those. Considering our case, only secure protocols do interest us and thus we will try to determine which of the available routing protocols will fit our needs and eventually decide to implement it into our final solution.

6.1 Routing

Data packets can be transmitted directly between to nodes if they are within wireless transmission range, no routing is required then (see Figure 6.1, on the left). But in the real ad-hoc environments, where most cases these two nodes are not within the wireless transmission range and data packets need to route through one or several intermediate nodes before they reach the destination (see Figure 6.1, on the right).

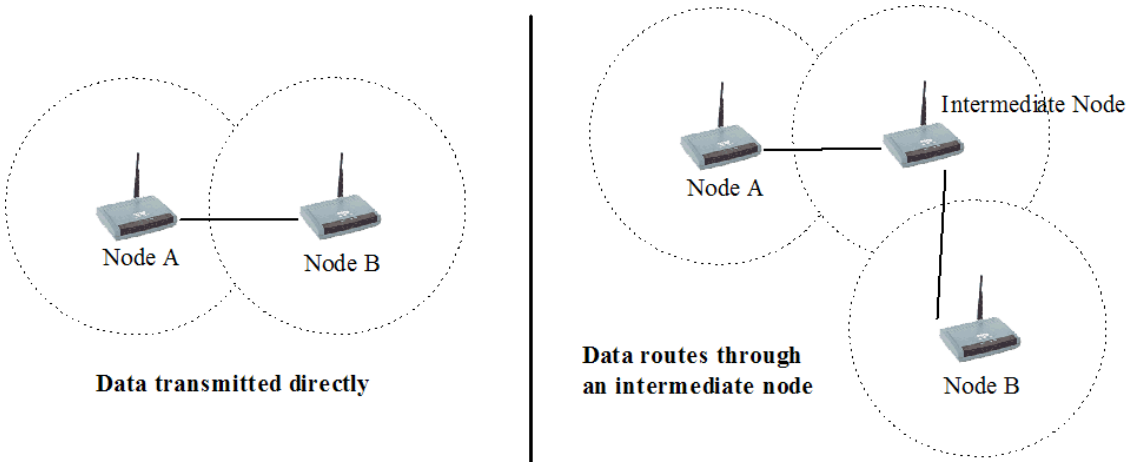

Figure 6.1: Data transmitted directly and through an intermediate node

6.2 Table Driven and On-Demand protocols

When node A attempts to communicate with node B, and find out that node B is not within its wireless transmission range and a direct communication is not possible. To route the data packets to the destination node, node A need to route the data packets through one or several intermediate node, which is within the range of both node A and node B. When the data packets arrives an intermediate node, the intermediate node determine whether or not it is the destination node. If not, it forwards these packets to its neighbor node or the destination node (the case where neighbor node is destination node).

Ad hoc wireless routing protocols can be divided into two categories, table driven (proactive) routing protocol and on-demand (reactive) routing protocol [9]. In proactive routing protocol, the nodes store routing information to all the nodes in the entire network in a routing table. These routing tables are updating as the network topology changes. While reactive routing protocols, the up-to-date route tables are not maintained. When a source attempts to send data packets to a destination node, it invokes a **route discovery** procedure (see subsection 6.3.1) to find a path to the destination. In the following subsections we outline the two most relevant protocols, **AODV**-Ad hoc On Demand Distance Vector (On-Demand) and **SDR**-Dynamic Source Routing Protocol (table-driven).

6.3 Ad hoc On Demand Distance Vector (AODV)

The Ad hoc On Demand Distance Vector (AODV) [9] is a routing protocol designed for use in wireless ad hoc mobile networks. In AODV the network is completely self-organizing and self-configuring and without the need of any network infrastructure or administration. AODV are characterized as On-demand protocol. When the need arise the AODV protocols invokes a **route discovery** procedure to find the path to destination. The found paths are maintains as long as they are needed by the sources. AODV is capable of both unicast and multicast routing. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes.

6.3.1 Broadcast RREQ

When a source node does not have a route for a required destination, AODV invokes a route discovery by broadcasting a route request (RREQ) packet to every node within its wireless transmission range (see Figure 6.2).

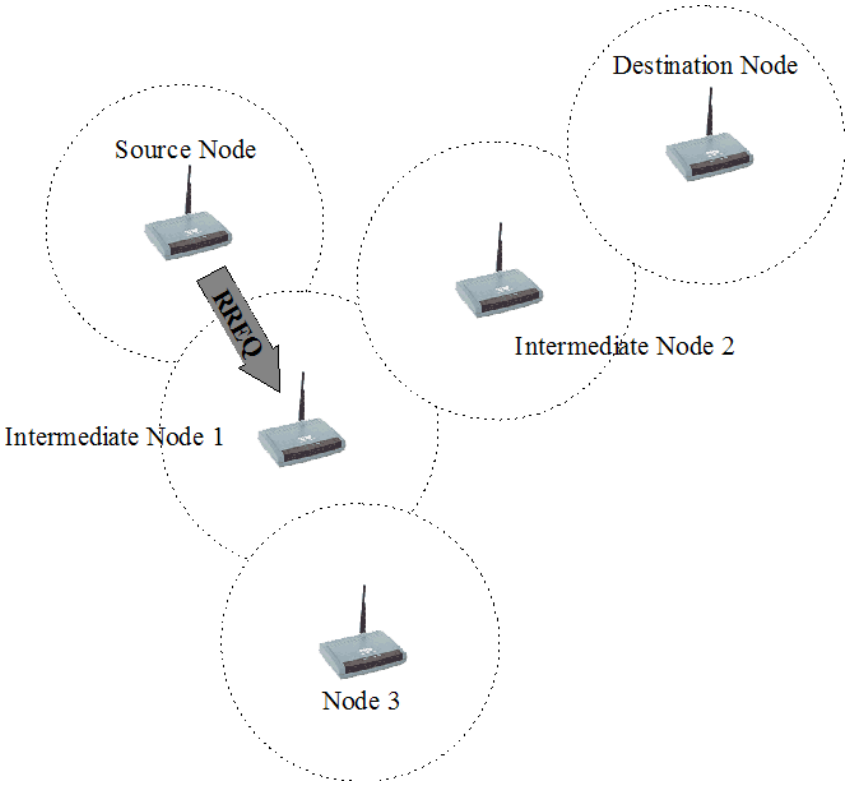

Figure 6.2: Source Node sends RREQ (source: [9])

The RREQ contains the source node's IP address, current sequence number, and broadcast ID; the RREQ also contains the most recent sequence number for the destination.

6.3.2 Intermediate Node Rebroadcasts RREQ

When receiving the RREQ packet, every node forward the RREQ packet to its neighbors if it self is not the destination. When forwarding the RREQ packet, the node has to update its route table to include a reverse point in the reverse path to the source (see Figure 6.3). This process will continue until a route to the destination node is found, or the route request process times out.

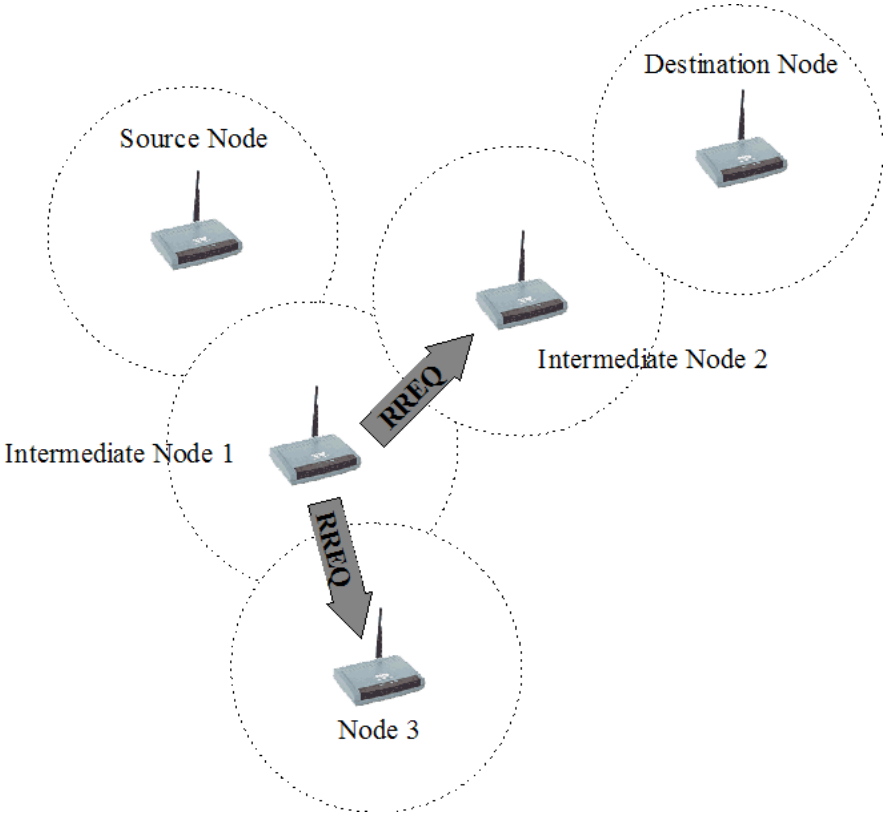

Figure 6.3: Intermediate Node 1 rebroadcasts RREQ (source: [9])

6.3.3 Intermediate Node Sends RREP

When a node is the destination node, or has a route to the destination node, it will respond by sending a route reply (RREP) to the source node (see Figure 6.4). Intermediate nodes update their route information about the source and destination nodes.

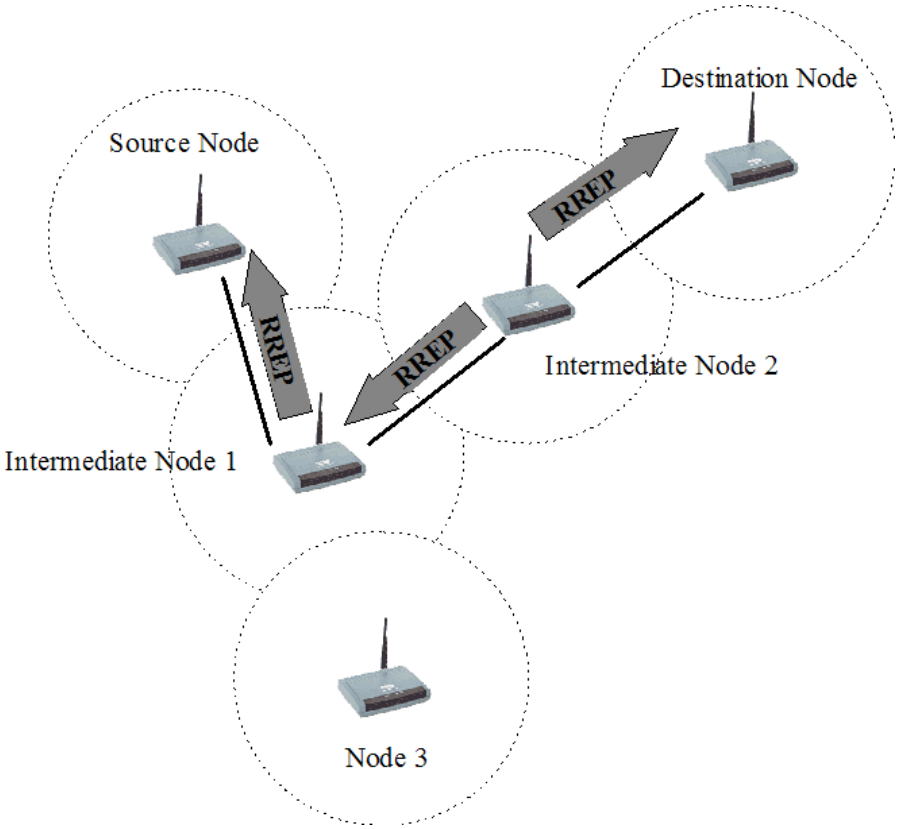

Figure 6.4: Intermediate Node 2 sends RREP (source: [9])

Upon receiving the RREP, the source node creates a new route and the source node can forward data to the destination node using this newly created route. If the RREP is not received within a certain time frame, the source node will retry the RREQ.

6.3.4 Broken Communication

A route will remain active as long as data continues to travel across the route. If a route becomes inactive for a defined period of time, the route will be deleted. Each time a packet is sent across a route, the timer is reset. When a communication is broken, a route error (RERR) will be send to any neighbors that had been using the node as the next hop (see Figure 6.5). Each node deletes the invalid route from its route table after receiving the RERR. If a route to the destination is still required, the source node will re-invoke the **route discovery** process.

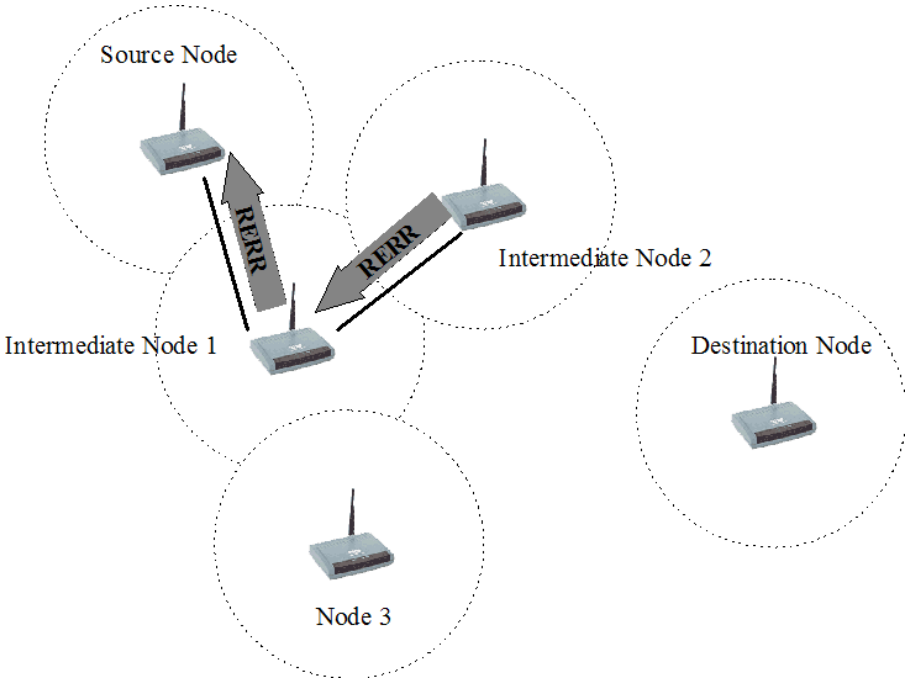

Figure 6.5: Destination Node has been separated from the network (source: [9])

To avoid the route loops and to guarantee the freshness of route information, AODV uses the sequence numbers. Each route maintains a sequence number, with higher sequence numbers indicating “fresher” routes. The sequence numbers are incremented when the RREQ, RREP and RERR are broadcasted. When multiple routes are available to a destination node, the route with the greatest sequence number is used. Packets with lower sequence numbers are ignored and dropped.

6.4 Dynamic Source Routing Protocol

Similar to AODV [9] the Dynamic Source Routing Protocol (DSR) [10] is also a routing protocol designed for use in wireless ad hoc mobile networks. DSR is composed by two mechanisms, **Route Discovery** and **Route Maintenance**. The purposes of these mechanisms are to allow nodes to discover and maintain the source route to destination in the ad hoc network. DSR does not use periodic routing advertisement, link status sensing or neighbor detection packets. As the nodes move and the communication pattern change, DSR packet overhead automatically (on-demand) scales to only that needed to track the routes currently in use. The nodes learn and cache multiple routes to any destination for future use. If a route fails to destination, a node may try another route in the cache. This reduces the need to perform a new **Route Discovery** each time a route in use breaks.

6.4.1 Route Discovery

When a source node **S** attempts to send a packet to destination node **D** it search for the suitable route in the cache. If the suitable route is found, it places the *source route* and *sequence of hop* in the header of the packet and the packet should follow this route all the way to destination. When a source node **S** attempts to send a packet to destination node **D** and does not have a route for the required destination, it performs a **Route Discovery**. Similar to AODV, the source node **S** broadcast (local) RREQ-packet to all the nodes within the wireless transmission range (see Figure 6.2). Upon receiving the RREQ-packet and if the node is the target, it sends RREP-packet back to the initiator of the Route Discovery (similar to AODV, see Figure 6.4). The initiator then caches the route which is used to send data-packets to the destination. The node saves the original data-packets in a *Send Buffer* while the node initiating a **Route Discovery**. The node sends the data-packets when a route is available. To prevent the RREQ-packet loops in a circle, the node drop the RREQ-packet if it has recently received the same RREQ-packet with the same request id and it finds it self in the route-list.

6.4.2 Route Maintenance

When the packet follow the source route to the destination, the packet is responsible for confirming that the packet is received by the next hop along the route to destination. For example source node **S** is responsible for receipt of the packet at **INT_1**, node **INT_1** is responsible for receipt of the packet at **INT_2**, node **INT_2** is responsible for receipt of the packet at **INT_3**, and so forth along the route to the destination (see Figure 6.6). If the packet is retransmitted by Max_Hop (maximum hop count) and no receipt is received, the node returns RERR (Route Error) to the source node **S** (original sender of the packet). The source node **S** then removes this broken route from its route cache and **S** try to send the packet through another route from its route cache. If no other route are available the source node **S** need to initiate a new **Route Discovery**.

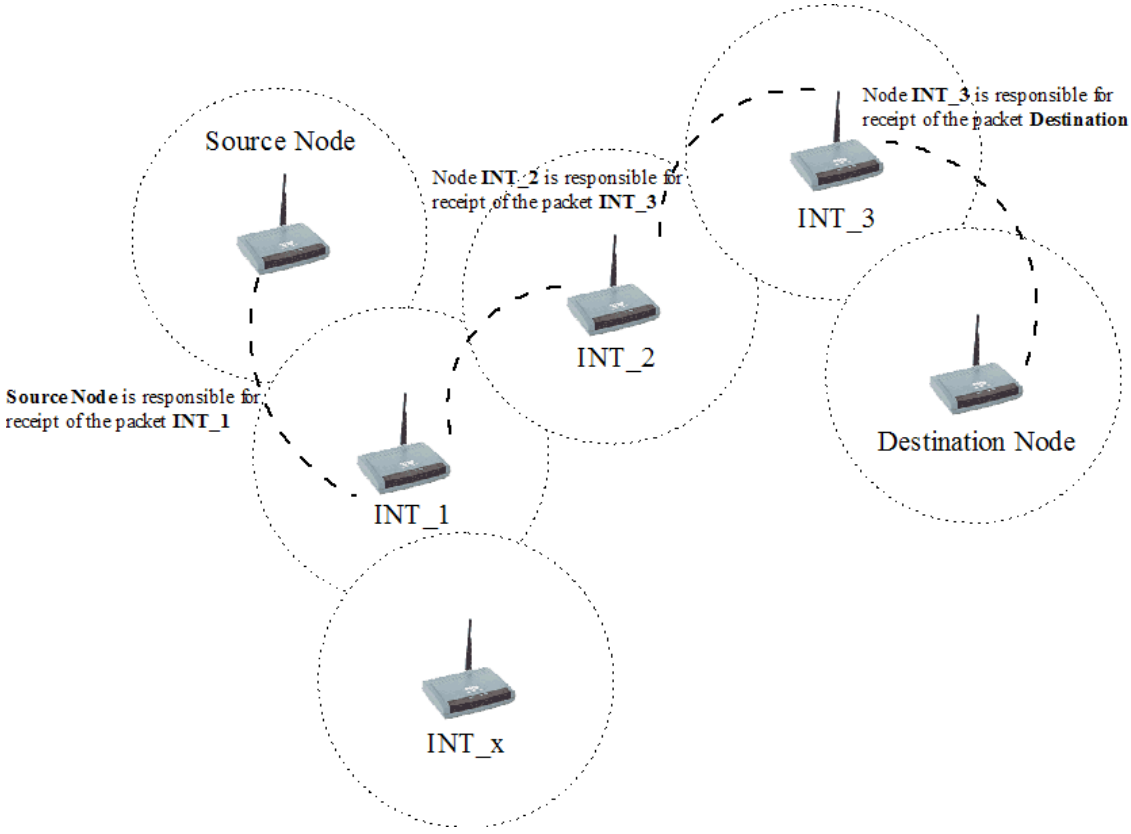

Figure 6.6: Nodes responsible for receipt at the next hop (source: [10])

6.4.3 DSR vs. AODV

The following table compares the two routing protocols SDR and AODV:

DSR vs. AODV		
Properties	DSR	AODV
Loop Free, Distributed, Reactive	Yes	Yes
Multicast capability	No	Yes
Multiple route	Yes	No
Routes maintained in	Route cache	Route table
Routing table format	Full path	Next hop
Routing metric	Shortest paths	Freshest & Shortest
Route checking	Passive ACKs	'Hello' messages
Utilizes route cache/table expiration timers	No	Yes
Unidirectional link support	Yes	No
Periodic broadcasts	No	Yes
CPU/Memory usage	High	Low
Scalability	Poor	Excellent
Rate of propagation	Fast	Slower
Ability to handle frequent topology change	Good	Fair

Table 6-1: DSR vs. AODV

6.5 Trust Ad hoc On Demand Distance Vector (AODV)

Trusted AODV (TAODV) reuses AODV [9] (Ad hoc On-demand Distance Vector) and employs trust functionalities to protect routing information (see Figure 6.7) [11].

6.5.1 TAODV Framework

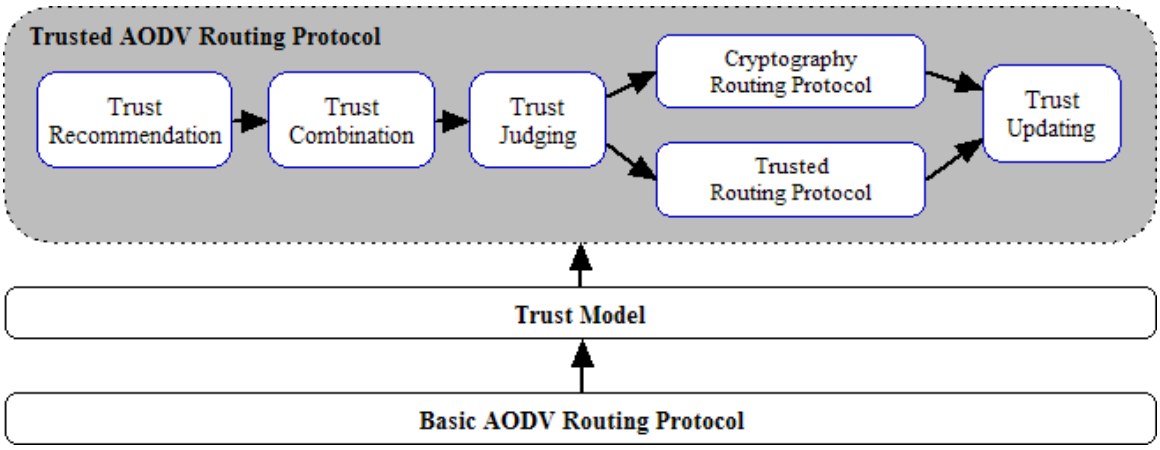

Figure 6.7: Framework of the Trusted AODV (TAODV) (source: [11])

The TAODV protocol employs trust functionalities to protect routing information by using an opinion metric [x, y, z]. The metric [x, y, z] represents *Belief, Disbelief and Uncertain* respectively. Initially each node's opinion metric is [0, 0, 1], which mean that the nodes do not trust or distrust. The uncertainty among them are high (*Uncertain*=1), but after a period of time the components in the opinion metric will change according to successful or failed communications. This means that after a successful communication the first component (correspond to *Belief*) in the opinion metric will increase, otherwise a failed communication will cause the second component (correspond to *Disbelief*) to rise and the third components will then decrease correspond to sum of these three components must be 1(see Figure 6.8 and 6.9). Opinion in TAODV is 3-dimentional metric and is defined as follow:

Definition (opinion): *Let* $\omega \frac{A}{B} = (b \frac{A}{B} + d \frac{A}{B} + u \frac{A}{B})$ *denote any node A's opinion about any node B's trustworthiness in a network, where the first, second and third component correspond to belief, disbelief and uncertainty, respectively and these components satisfy:*

$$b \frac{A}{B} + d \frac{A}{B} + u \frac{A}{B} = 1$$

Suppose node A invokes a **route discovery** to find a route to B. But in the first place A is not sure whether it should believe or disbelieve any other nodes. A will then use a secure scheme to perform the **route discovery**.

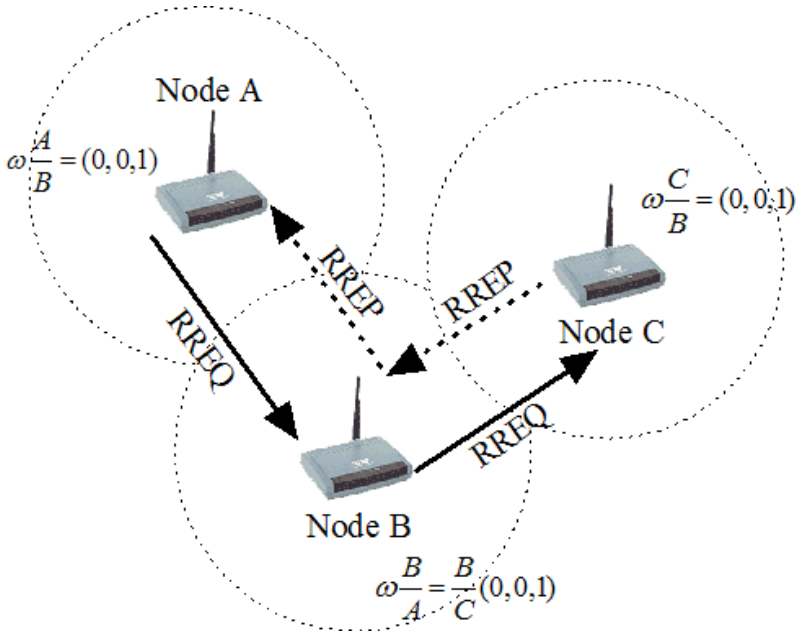

Figure 6.8: Initialization for TOADV (source: [11])

Initially A is not sure whether it should believe or disbelieve any other nodes.

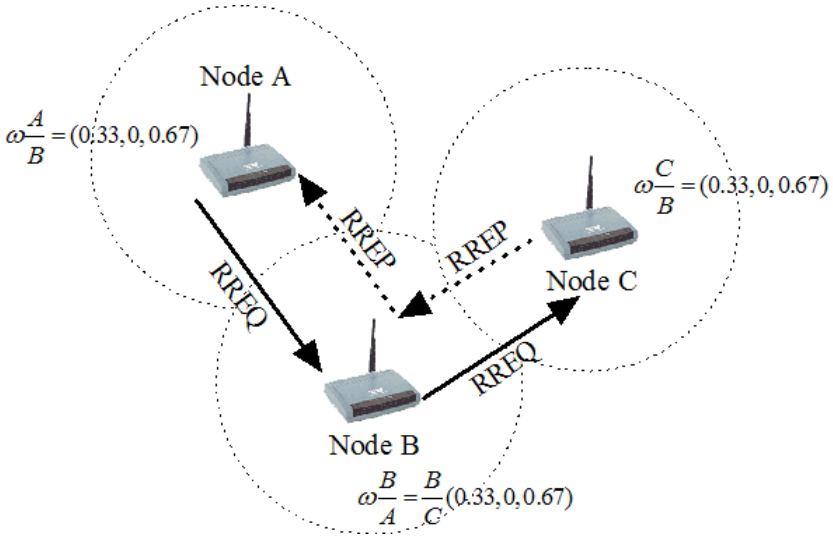

Figure 6.9: TAODV after a period of time (source: [11])

A will update its opinions toward the other nodes after some successful or failed communications.

6.5.2 Trust Model

Trust model is composed by Trust Recommendation, Trust Combination, Trust Judging and Trust Updating and their task as follow:

- **Trust Recommendation** – The Trust Recommendation protocol is used when a node want to know trustworthiness toward another nodes.
- **Trust Combination** – Several nodes operate with different trustworthiness about other nodes. To get a relative trustworthiness about a node, TAODV uses the Trust Combination operation to combine different opinions to get relative one.
- **Trust Judging** - The Trust Judging contains rules what the node should do according to opinion. The rules as follow:

    1. If A's opinion toward a node B with the *Belief*-component of opinion is larger than 0.5, A will trust B and continue to perform routing related to B.
    2. If A's opinion toward a node B with the *Disbelief*-component of opinion is larger than 0.5, A will not trust B and will refuse to perform routing related to B.
    3. If A's opinion toward a node B with the *Uncertain*-component of opinion is larger than 0.5, A will request B's digital signature whenever A has interaction with B.
    4. If A's opinion toward a node B with all the components (*Belief, Disbelief, Uncertain)* of opinion are smaller than or equal 0.5, A will trust B and continue to perform routing related to B.
    5. If node B has no route entry in node A's routing table, A's opinion about B is initialized as [0, 0, 1].

- **Trust Updating** – The Trust Updating contains several policies, which describe how and when a node needs to update the trust opinion. The policies as follow:

    1. Each time a node A has performed a successful communication with another node B, B's successful events in A's routing table will be increase by 1.
    2. Each time a node A has performed a failed communication with another node B, B's failed events in A's routing table will be increase by 1.
    3. Each time when the field of the successful or failed events changes, the corresponding values of opinion will be recalculated.

4. If node B's route entry has been deleted from node A's route table because of expiry, or there is no B's route entry from the beginning, the A's opinion to B will be set to [0, 0, 1].

6.5.3 Modified Routing Table with Trust Information

TAODV add three new fields into each node's original routing table: *positive events, negative events* and *opinion* (see Figure 6.10).

Destination IP	Destination Seq	....	HopCount	....	LifeTime	**Positiv Events**	**Negative Events**	**Opinion**

Figure 6.10: Modified Routing Table

The new fields, positive/negative events correspond to successful/failed communication times between two nodes. Opinion means this node's belief towards another node's trustworthiness.

6.6 Secure Ad hoc On Demand Distance Vector (SAODV)

SAODV [12] also uses AODV and add additional functionalities to secure the AODV messages. Hence no security are implemented in AODV, malicious nodes can perform different attacks against AODV. SAODV has implemented to mechanisms to secure the AODV messages, *hash chains* [15] and *digital signatures* [14]. Digital signatures are used to authenticate the non-mutable fields of the messages, and hash chains are used to secure the hop count information (see Figure 6.11 and 6.12).

RREG Message Format:

Type	J	R	G	Reserved	Hop Count (mutable)
RREQ ID					
Destination IP Address					
Destination Sequence Number					
Originator IP Address					
Originator Sequence Number					

RREP Message Format:

Type	R	A	Reserved	Prefix Sz	Hop Count (mutable)
Destination IP Address					
Destination Sequence Number					
Originator IP Address					
Lifetime					

Figure 6.11: RREQ and RREP Message Format

Type	Length	Hash Function	Max Hop Count (mutable)
Top Hash			
Signature			
Hash			

Figure 6.12: RREQ and RREP Signature Extension Format (source: [12])

Every time a node broadcast a RREQ or RREP, first it generates a seed (a random number) and applies a one way hashing to this seed with Max_Hop_Count times, where the Max_Hop_Count equal the TimeToLive from IP-header. This is done to ensure that the Hop_Count has not been modified by an attacker. To protect the integrity of the non-mutable fields in RREQ and RREP, SAODV uses *Digital Signatures* to sign all the non-mutable fields and the seed using security system like IPsec.

6.7 Security - Aware Ad-Hoc Routing for Wireless Networks

Similar to TAODV [11], Security – Aware ad-hoc routing (SAR) [13] uses AODV [9] and employs additional functionalities to ensure that data is routed through a secure route and secure the information in the routing protocol messages. Messages route through nodes are based on quality of security. This means that messages routes only through secure nodes and not through insecure nodes even if the path is shorter (see Figure 6.13). If one or more routes that satisfy the required security, SAR will find the shortest route. And if several nodes with the same length and satisfies security requirements, SAR will find routes that are optimal. On the other hand if the ad-hoc network does not have a path with nodes that satisfy security requirements, SAR may fail to find a route even if the network is connected.

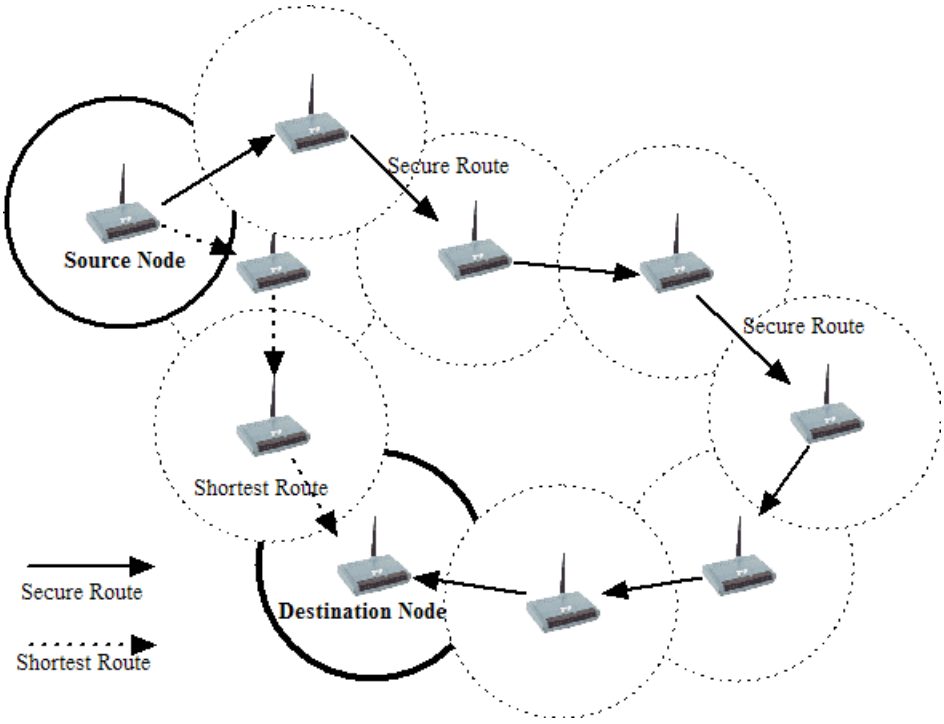

Figure 6.13: Secure Route and Shortest Route (source: [13])

6.7.1 Trust Hierarchy

Nodes are organized like tree hierarchy and associated a number with each privilege level. These numbers represent the trust level (security, importance and capability) of the mobile nodes and their paths (see Figure 6.14).

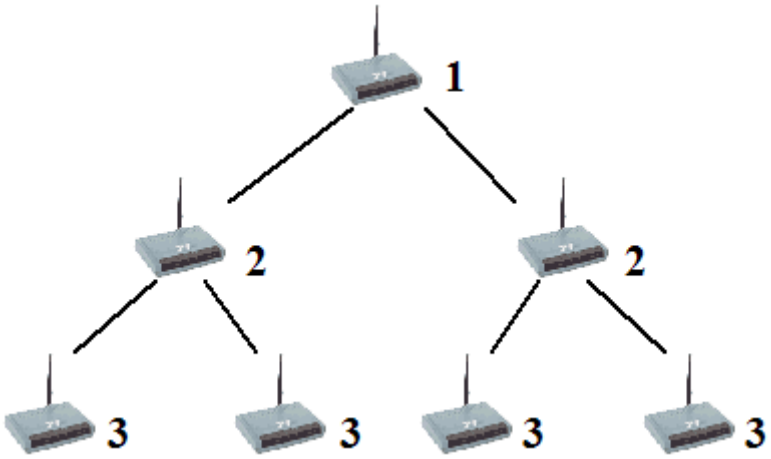

Figure 6.14: Trust Hierarchy (source: [13])

These trust level should be immutable. A node with a lower trust level should not be able to change its trust level, or change the trust level of the other RREQ request it forwards. To provide this guarantee, many techniques can be employed.

6.7.2 Route Discovery and Changes to RREQ and RREP

In Security-aware AODV, the RREQ have two additional fields. The first is RQ_SEC_REQUIREMENT (see Figure 6.15) that indicates the required security level in the trust hierarchy for the route the sender wishes to discover. This field is fixed and is only set once by the sender. When the RREQ message arrive a node, the protocol uses this field to check if the node satisfies the security requirement or not. If the node satisfies the security requirement, then protocol forward the RREQ packet to it neighbors. Otherwise, if the security requirement is not satisfied the packet is dropped even if the network is connected. The second additional field is RQ_SEC_GUARANTEE. This field indicates the maximum level of security afforded by the paths discovered. This field is useful in the case where route discovery discovers a route that is more secure than the sender asked for. It is also useful for the security aware applications to get more detailed information about the quality of security for the paths discovered.

RREG Message Format:

Type	...	...	...	Reserved	Hop Count
RREQ ID					
Destination IP Address					
Destination Sequence Number					
Originator IP Address					
Originator Sequence Number					
RQ_SEC_REQUIREMENT					
RQ_SEC_GUARANTEE					

RREP Message Format:

Type	...	...	Reserved	...	Hop Count
Destination IP Address					
Destination Sequence Number					
Originator IP Address					
Lifetime					
RP_SEC_GUARANTEE					

Figure 6.15: Changes to RREQ and RREP (source: [13])

The arrival of a RREQ packet at the destination indicates the presence of a path from the sender to the receiver that satisfies the security requirement specified by the sender. The destination node sends the RREP packet as in AODV, but with additional information indicating the maximum security available over the path. This information is suitably protected so that only nodes that belong to a particular trust level can process these packets. The value of the RQ SEC GUARANTEE field in the RREQ packet is copied to RP SEC GUARANTEE field in the RREP packet. When the RREP packets arrives at an intermediate node in the reverse path, intermediate nodes that are allowed to participate, update their routing tables as in AODV and also record the new RP SEC GUARANTEE value. This value indicates the maximum security available on the cached forward path. When a trusted intermediate node answers a RREQ query using cached information, this value is compared to the security requirement in the RREQ packet and only when the forward path can guarantee enough security is the cached path information sent back in the RREP.

6.8 Secure Network Encryption Protocol (SNEP)

The main goal of the Secure Network Encryption Protocol (SNEP) is to secure the peer-to-peer wireless connection. SNEP provides confidentiality, data integrity and data freshness.
Facts:

- 8 Byte Overhead per message
- Keeps state at each end points, which makes it unnecessary to transmit the counter
- Semantic Security (randomization)

Properties:

- Semantic Security: Counter value is incremented each time a message is sent. This means that each message is encrypted differently. Before each message a random bit string will be sent to make it impossible to guess if a bit is 1 or 0.
- Communicating parties share a counter, which is used as an Initialization Vector (IV)
- Counter is not sent with the message
- Counter value is never repeated
- Counter value in MAC prevents replay attacks
- Data authentication: Uses a MAC (Message Authentication Code) to verify origination of the message.
- Replay Protection: Counter value included in the MAC prevents this.
- Weak freshness: The receiver will know that a message has been sent after the previous message has been received correctly if it verifies correctly.
- Low communication overhead: Since the counter state is kept on each end point it doesn't need to be send hence reducing overhead.

The base station shares a master key with all nodes. Nodes-to-nodes keys can be negotiated with help of the base station. Initially two devices A and B will share a master key $K_{A,B}$, which is used as input into the RC5 function [33] to generate the encryption key ($K_{encryption}$), the MAC key (K_{MAC}) and the random number generator key (K_{RAND}). The key scheduling process is depicted in the figure below. Later, these keys will be use in the encryption process, the authentication process and to generate the MAC.

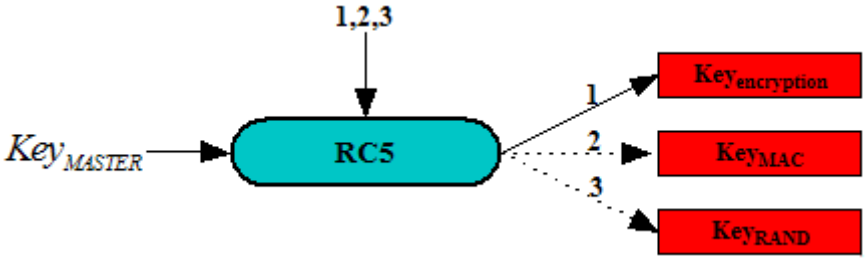

Figure 6.16: SNEP Key scheduling

Notations:

Encrypted data: $E = \{D\}_{(K_{encryption}, C)}$

MAC: $M = MAC(K_{MAC}, C|E)$

SNEP provides two types of messages from A to B:

With encryption:

$$\mathbf{\{Msg\}_{<Key\_encrytion,\ Counter>},\ MAC(K_{MAC},\ Counter\ |\ \{Msg\}_{<Key\_encrytion,\ Counter>})}$$

Without encryption:

$$\mathbf{Msg,\ MAC(K_{MAC},\ Msg)}$$

Where 'D' is the data, 'C' is the Counter (Initialization Vector), '$K_{encryption}$' is the encryption key and 'K_{MAC}' is the MAC key.

6.8.1 SNEP Encryption

SNEP encrypts the messages using the $Key_{encryption}$ and a Counter. The Counter value is incremented each time a message is sent. This means that each message is encrypted differently. Before each message a random bit string will be sent to make it impossible to guess if a bit is 1 or 0. The RC5 function takes Counter and $Key_{encryption}$ as inputs. The outputs are then XOR with the plain text to get cipher text. The processes are depicted in the figure below.

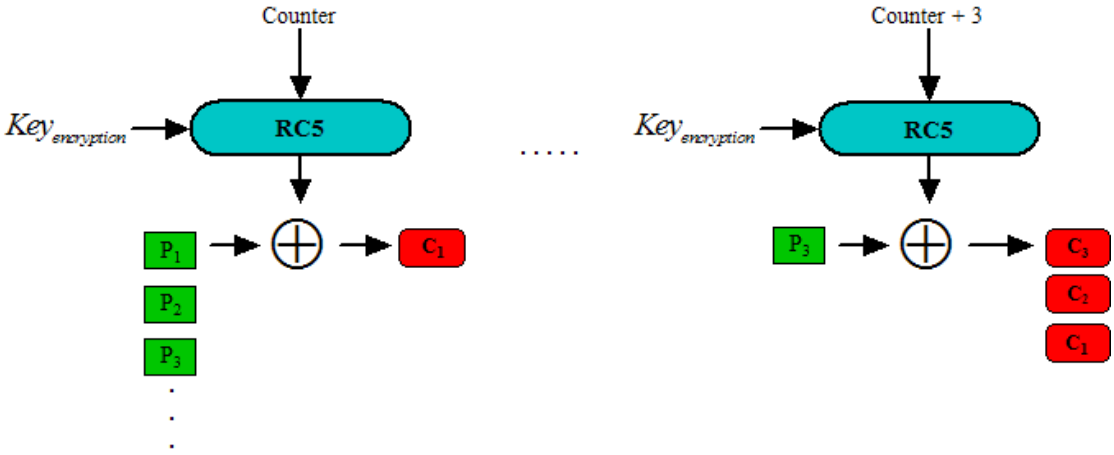

Figure 6.17: SNEP Encryption

6.8.2 SNEP MAC

Data authentication uses a MAC (Message Authentication Code) to verify origination of the message. SNEP MAC uses Cipher Block Chaining (CBC) where every block of inputs are affects the outputs. The processes are depicted in the figure below.

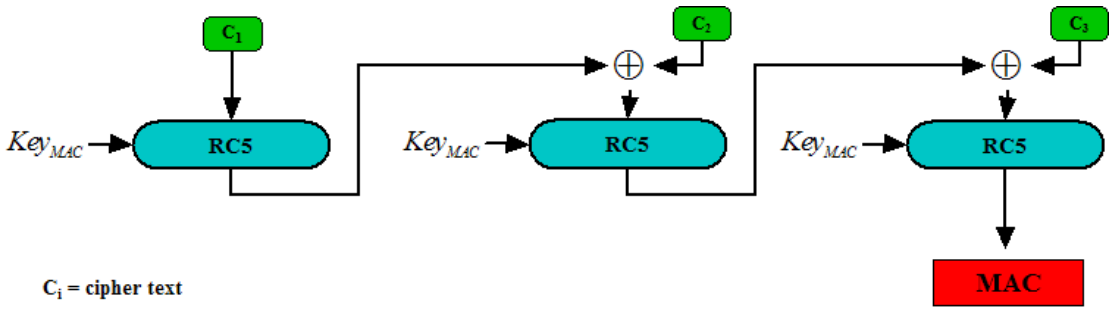

Figure 6.18: SNEP MAC generation

6.8.3 SNEP Authentication

SNEP provides Authentication either with or without encryption. For the Authentication with encryption, the messages are encrypted and the Counter is included in the MAC. The Base stations will keep the current Counter for every node.

The processes are depicted in the figure below.

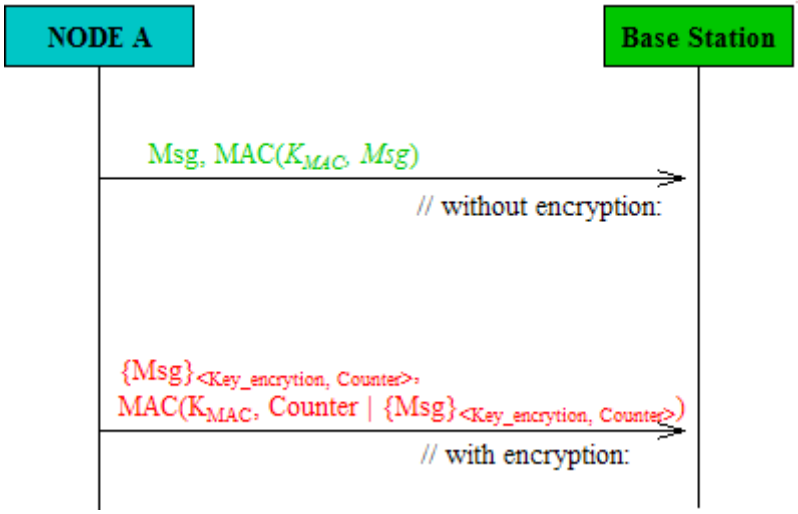

Figure 6.19: SNEP Authentication

For more detailed information and reference see [16].

6.9 µTESLA

µTESLA is a lightweight implementation of the TESLA protocol. The original protocol has a packet overhead of 24 bytes. A standard size message in a sensor network is about 30 byte. Furthermore the key chain will not fit into the memory of a node. µTESLA implements asymmetric authentication through a delayed disclosure of symmetric keys. This is done to reduce the computational demands of asymmetric keys. To do this it requires the nodes to be loosely time synchronized, a maximum synchronization error and a time schedule for disclosure of the keys. The base station will first compute a MAC on the packet it wants to send to a node. The node on its side will receive the message and verify that the corresponding MAC key hasn't been disclosed yet by using the clock and maximum synchronization error. The base station will distribute the verification key to a given time. Now the node will use the verification key to verify the correctness of the stored packet.

6.9.1 Sender Setup

The first thing a sender does is to generate a sequence of secret keys. Those one way key-chains of length n are generated by randomly choosing K_n as the last key. The remaining values are calculated using a one way function F (e.g. MD5 [29]). $K_j = F(K_{j+1})$. Since a one way function is used one can only compute forward, which means you can compute K0 if you have K_{j+1} given. This renders one however unable to compute K_{j+1} if you have K_0.

6.9.2 Broadcasting

Each key of the one-way key chain is assigned to a uniform divided time interval. This means in time interval *i,* K_i is being used to compute the message authentication code (MAC). Key disclosure is delayed a few time intervals. This amount of time is mostly chosen to be greater than a normal roundtrip time between the sender and its receivers.

6.9.3 Bootstrapping a new receiver

Since we are using a one-way key chain all keys are self-authenticating. All we need to start with is one authenticated key (K_i) and can thus verify a new key (K_{i+1}) with using $K_i = F(K_{i+1})$. The requirements to bootstrap µTESLA are one authenticated key, the nodes need to be loosely time synchronized and the receiver has to have knowledge about the key disclosure time. To resolve the last two problems a mechanism that offers strong freshness and point-to-point authentication can be used. The mechanism works as follows that the receiver R sends a nonce N_R in a request message to sender S. S then returns its current time T_S a key K_i which was used in the past interval *I,* the starting time of that interval T_i, the duration T_{int} and the disclosure delay δ.

$$M \rightarrow S : N_M$$

$$S \rightarrow M : T_S \mid K_i \mid T_i \mid T_{\text{int}} \mid \delta \mid MAC(K_{MS} \mid N_M \mid T_S \mid K_i \mid T_i \mid T_{\text{int}} \mid \delta)$$

6.10 ARAN (Authenticated Routing for Ad hoc Networks)

According to [17] ARAN is a fairly simple routing protocol. It consists of a preliminary certification and a route instantiation process. To discover routing paths is achieved by broadcasting a route discovery message from a source node which is replied to unicast by the destination node.

To achieve certification the protocol requires a trusted certificate server. All nodes have to know its public key. The way this key is distributed is not specified by the protocol and will require and individual solution. After certification is with a certificate server is completed a node A with receive a certificate message from the server which includes the IP address of A, the public key of A, a timestamp of when the certificate was created and a time when the certificate expires. Nodes now will use those certificates to authenticate themselves to other nodes during message exchanges.

6.11 Summary

In chapter 5.5 we concluded that ZigBee is the most promising solution. However the intention was to look at other secure routing protocols independent of technology choice. As there might be routing protocols better suited for our case than those deployed with standard packages.

As outlined in the subsections above, the most of the secure protocols are based on AODV or DSR. AODV, however, doesn't provide any security at all. To secure the packets under transit, TAODV, SAODV and SAR employ security functionalities upon AODV. These security functionalities again are based on one of the cryptographic schemes. Since ZigBee uses AODV and AES. AES takes use of Rinjdael algorithm which is known to be the most secure algorithm today. At this point we do not see the intention to take other considerations than the use of AODV and AES as it is implemented in ZigBee/802.15.4.

7 Proposed Solution

In this chapter we will outline our proposed solution to the case scenario. After choosing technology and routing protocol in the previous to chapters, we will now outline a solution for the application part of the case. This chapter will build the foundation for the following chapter where we will attempt to design a working model and eventually if there is time create a test model and prototype. Our model will contain sufficient information to build a framework for a prototype.

7.1 Wireless Ad-hoc network technology

As lined out in chapter 5 we will use ZigBee for the implementation of our solution to the case scenario.

7.2 Routing Protocol

As we choose 802.15.4 as our wireless Ad-hoc network technology, it seems to us to be a logical step to choose ZigBee which is build upon 802.15.4 with the purpose of delivering an ad-hoc wireless sensor network technology. We are going to describe the solution around ZigBee and use different solutions as far as possible if we feel ZigBee will not be able to cope with a setting on its own. Our solution presented in this chapter describes the application level of it. Reliable transmission and transmission security is handled by ZigBee. We do however implement an extra layer of security using encryption where we handle sensitive information. ZigBee will also take care of route discovery and synchronization of timestamps.

7.3 Nodes

In our solution we will have two different types of nodes. The first type is the car node (CN) which is the ad-hoc part of the wireless network. The second types are the so called controller nodes performing operational and routing tasks. Those will be static implementations and are the trusted part of the network (Figure 7.1).

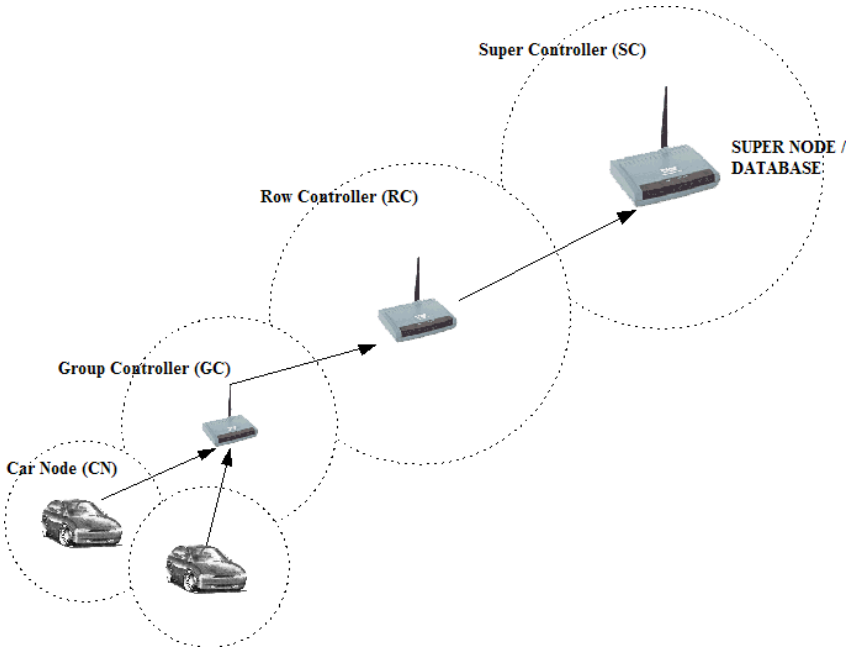

Figure 7.1: Nodes

7.3.1 Car Node

The chip installed in the car will be the initiator of any communication with the system. Its first action will be to authenticate with the closest Group Controller. Since we will apply end-to-end authentication the CN will not communicate with the GC except for polling responses and use it as a means of getting the information to its proper destination. The CN will contain following fields needed for authentication with the system.

Name	Description
UID	Unique Identifier
K_{SSK}	Shared Secret Key
LNR	License Number
TS	Timestamp

Table 7-1 Car Node Data Fields

Given this table we get the following formula for the first message to be sent by the CN to the SC/Database. Note that this message is not sent directly, but going via Row and Group Controllers before it reaches the Super Controller (See Figure 7.1).

$$MSG_{CN \to SC} = K_{SSK}(UID + TS) + LNR$$

UID and TS will be encrypted by the CN's secret key (K_{SSK}). This is done to ensure that UID cannot be read by intruders and TS is added to ensure freshness of the data to prevent replay attacks.

7.3.2 Controller Nodes

The tasks of the Controller Nodes are to perform routing operations and ensure reliable data transfer from the CNs towards the Super Controller (SC). They will authenticate themselves using a secret key encryption method. They will inherit a key which is also known to the SC and will use this to authenticate each other.
Each node will store some basic routing information of which other nodes it is connected to.

Name	Description
GCID	Group Controller Identifier
GCK_{SSK}	Shared Secret Key (Group Controller)
$TempID_{GC}$	Temporary ID assigned by SC
RCList[]	List of all RCs connected to the GC
GCList[]	List of all GCs connected to the GC
CarList[]	List of all cars connected to the GC
SCPointer	Routing information to reach the SC
TS	Timestamp

Table 7-2 Group Controller Data Fields

Name	Description
RCID	Row Controller Identifier
RCK_{SSK}	Shared Secret Key (Row Controller)
$TempID_{RC}$	Temporary ID assigned by SC
RCList[]	List of all RCs connected to the RC
GCList[]	List of all GCs connected to the RC
SCPointer	Routing information to reach the SC
TS	Timestamp

Table 7-3 Row Controller Data Fields

Note there is no direct link to the SC from a GC node as we want them to go through the nearest RC. Only if the RCList and GCList are empty a GC node will attempt to communicate directly with the SC if it detects in range.

A RC will have to keep track of other RCs it is connected to and other GCs it is supervising. It will store a shortest path (through other RCs) to the SC to reduce dataflow over the net to increase battery lifetime.

Name	Description
SCID	Supper Controller ID
KeyList[] (database mode only)	List over all Shared Secret Key for cars it has stored
RCList[]	List over all attached RC's
GCList[]	List over all attached GC's
CarList[]	List over all cars attached to the system
Timestamp	Timestamp for synchronization purposes

Table 7-4 Super Controller Data Fields

Taking a closer look at the tables stored on the SC we will see that KList [] is in fact the table which can be implemented in an external database if desired. If the system is to be implemented on a single parking lot it will be of no use to create an external database server to store car information. However if the system is to be implemented on several different parking lots with a common customer base it will be more efficient to use a common database server to check and store parking info. This will leave us with less computational power needed at the SCs, which in its turn will let us install smaller nodes which cost less. We will also get an increased term of security as an external database server can be put in a physically better protected place than what may be possible at parking lots. This has one weakness though as if the server gets compromised or "hacked" an attacker will gain access to information enough to bring down the entire system. In the scenario where all information is stored on SCs we will still have this problem however as we need to secure each node as good as possible to maintain a feasible security.

After reading chapter 7 so far a natural question emerges that it might be better to combine GC and RC to one type of node as their area of use overlaps very much. Yes in the first place it may seem so. However, we give RCs a stronger transmitter so that they can send up to 100 meters. A GC will only have a transmitting range of about 10 – 15 meter. Since the GC is doing the most of data transmitting it will also have the highest power consumption. Hence to increase the lifetime of our nodes it will use less power to transmit over short range, so we lay the task of sending for long range to the RCs.

7.3.3 External Database

We envision using an external database connected to the SC via Internet to identify a car. The database will contain the secret key which is programmed into the CN. The key is used to establish secure end-to-end communication. It will receive the initial message from the CN, which contains License Number and the Unique ID. Using the License Number it will determine which secret key to use for decryption. If the decryption is successful the database will know that the CN is authentic. It will now return a message to the CN encrypted with the shared secret key.

$$\mathrm{Re}\,g\,\mathrm{Re}\,sp = K_{SSK}("OK" + LicenseNr + TimeStamp)$$

7.3.4 Data fields

Down below is a presentation of the lists presented in the last sub chapters.
The KeyList contains a reference from the LNr of each registered user to its UID and Shared Secret Key. This table is used for looking up the Secret Key and verifying the authenticity of a registering car.

KeyList	Data fields			
	LNR	UID	K_{SSK}	Location

Table 7-5 KeyList

The CarList contains a list of all cars registered at a parking lot and when they are last polled or if a poll has failed. The SC will include a path to where to reach the car.

CarList	Data fields			
	LNR	Path(SC only)	Last_Poll	Failed_Poll

Table 7-6: CarList

The RC/GCLists contain an overview of which nodes a Controller is in range too. It will also contain information about if it managed to authenticate them with the SC.

RC/GCList	Data fields		
	TempID	Auth	Last_Auth

Table 7-7: RC/GCList

The SCPointer will contain information about which nodes to contact to reach the SC. We envision a pointer in the form: GCNode(a) → GCNode(b) → RCNode(c) → SC

7.4 Message Information

7.4.1 CN Registration

When a car is getting parked near to a GC it will first attach itself towards the network using standard ZigBee protocols and authentication methods. After this it will start its work to authenticate with the database. This may be the SC or an external server connected to via internet.

After authentication with the corresponding GC a CN sends its Register Request (RegReq) to the network and awaits reply from the SC or Database server (see Figure 7.2).

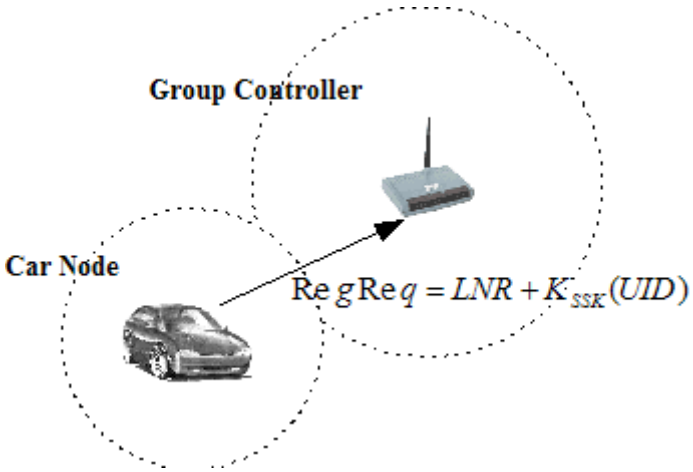

Figure 7.2: Message CN to GC

The GC will do nothing to the package except to add its own ID and store the LNR to its car list. This will subsequently be done by each GC or RC which is contact with the package before getting to the SC (see Figure 7.3).

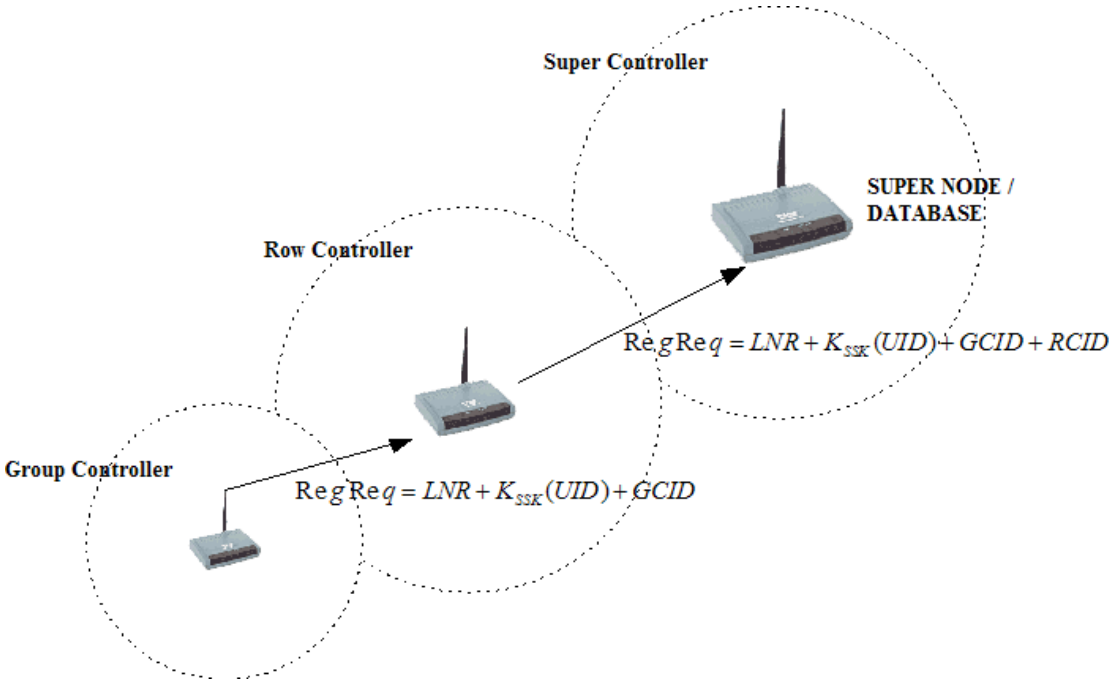

Figure 7.3: Message GC to SC via RC

SC will now store GCID and RCID with the LNR to its car list to get the link where to reach the CN. GCID and RCID are removed before sending the message either to a database server or checking the authenticity of the CN. LNR will be used to look up the proper decryption Key. The key will now be used to decrypt the message and the received UID will be checked with the database. If the UID matches the stored ID the server will respond with a Register Respond message:

$$Reg\,Resp = K_{SSK}("OK" + LicenseNr + TimeStamp)$$

If the check fails this will be sent:

$$Reg\,Resp = K_{SSK}("Failed" + LicenseNr + TimeStamp)$$

If the registration process fails another time we envision a solution where the database sends a message to the car user and informs him/her that the car is not secured.

7.4.2 CN Polling

Once a node is registered we need to make sure that a car stays on its place until it is properly signed off. This is achieved by simply polling the CN in a periodic interval. If no response is given a message will be sent to the SC/Database.

$$MSG = GCK_{SSK}(LNr + PollFailed) + GNr$$

7.4.3 CN Signoff

When a car wants to leave the parking lot we should let it do so. If we for example couple the signoff to when a car is started with its proper key the CN will send a Signoff message.

$$MSG = K_{SSK}(UID + Signoff) + LNr$$

After this message is received the Database will inform the SC that the car is leaving, if the decryption is successful. The SC will now send a message down the chain to each RC and GC which are listed in the routing table for the specific car and order them to remove the car from their CarList. If the decryption fails the car will get reported as stolen.

7.4.4 GC/RC Registration

When we add new GC/RCs to our system, we need them to register with the SC. This will happen in a similar way as with a CN. All GC/RCs will have a unique ID and a Shared Secret Key. It will encrypt its unique ID with the Shared Secret Key and send it to the SC. The SC on its part will decrypt the message and verify the ID. If it is successful a confirmation message will be sent back. If it fails one retry chance is given. If this fails too the SC will sent out an alert to the system administrator warning him that there might be a possible attack.

7.4.5 GC/RC Authentication

Since GCs and RCs can connect to several different nodes of each type we want to implement a solution for a node checking another's legitimacy. This is achieved by requesting the TempID from the node in question. Then encrypting it with its own K_{SSK} and sending it on the already authenticated path. If we call the other node for B and the node requesting the authentication for A, we get the following formula:

$$Auth = K_{SSK-A}(TempID_B) + TempID_A$$

7.5 Encryption

Initial, when the car is within the wireless transmission range of the GC it will start authentication procedure to verify the identity of the CN and the GC. At this state the authentication messages are encrypted using Advanced Encryption Standard (AES, see subsection 5.4.5), which is implemented in ZigBee.

Base on our solution described above, the messages which are sent to the Super Controller (SC)/Database to verify the membership contains Unique Identifier (UID) and this UID must be kept secret. The CN contains also a Shared Secret Key (K_{SSK}) which is known only by the car and the SC/Database. To enhance the security at the authentication state the cars will encrypt the messages using Self-synchronizing stream ciphers [18]. Stream ciphers are secret-key cryptography where the same key is used to compute and verify the messages. In this case the messages are encrypted/decrypted using the UID and Self-synchronizing stream ciphers (see Figure 7.4 and 7.5).

Self- synchronizing stream ciphers generates a key stream as a function of the key (UID) and a fixed number of previous cipher text digits. The encryption processes are depicted as follow:

Encryption:

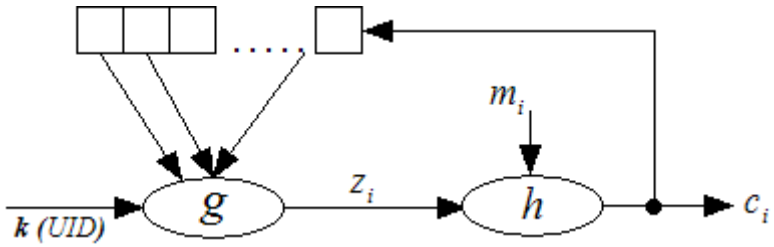

Figure 7.4: Stream Ciphers Encryption

k **=** key (UID)

g = pseudorandom number generator which generates the key stream

$\boldsymbol{Z_i}$ = key stream

$\boldsymbol{m_i}$ = plaintext

h = XOR- function which XOR the key stream C_i with plaintext m_i

$\boldsymbol{C_i}$ = cipher text

The pseudorandom number generator ***g*** takes ***k (UID)*** and a fixed number of previous cipher text digits as inputs, and generates a key stream Z_i as output. Next the function ***h*** combines (XOR) the key stream Z_i with the plaintext m_i to produce the cipher text C_i.

Decryption:

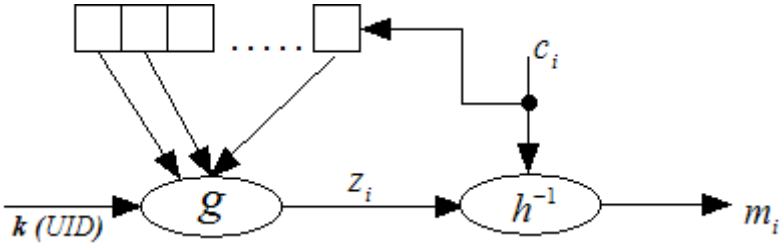

Figure 7.5: Stream Ciphers Decryption

k **=** key (UID)

g = pseudorandom number generator which generates the key stream

Z_i = key stream

m_i = plaintext

C_i = cipher text

h^{-1} = XOR- function which XOR the key stream C_i with plaintext m_i, an inverse process to decrypt the cipher text C_i.

To decrypt the cipher text C_i, the XOR-process is applied to cipher text C_i and the key stream to get the plaintext.

8 Model Description

In this chapter we will present a model of our proposed solution. As a modeling tool we choose to use SPIN [31]. We choose SPIN due to that we have worked with it earlier and it's a well suited tool to simulate protocols and distributed systems with.
We wish to test our proposed solution in this tool, to demonstrate its workability and stability.
Spin is a tool used for formal verification of distributed software systems. It has three modes of operation. It uses PROMELA, a high level language used to describe system descriptions, as syntax language.

8.1 Model

First we will take a look at the states of our Car Node. After that we will take a look at states of the RC and GC. Their function is quite similar to each other so it will suffice using one state machine for both. At last we will take a look at the SC, which will hold the registration processes and store the travel path to the cars. Message sequence charts are presented in chapter 7 and do not need to be repeated here.

8.1.1 State machine for Car Node

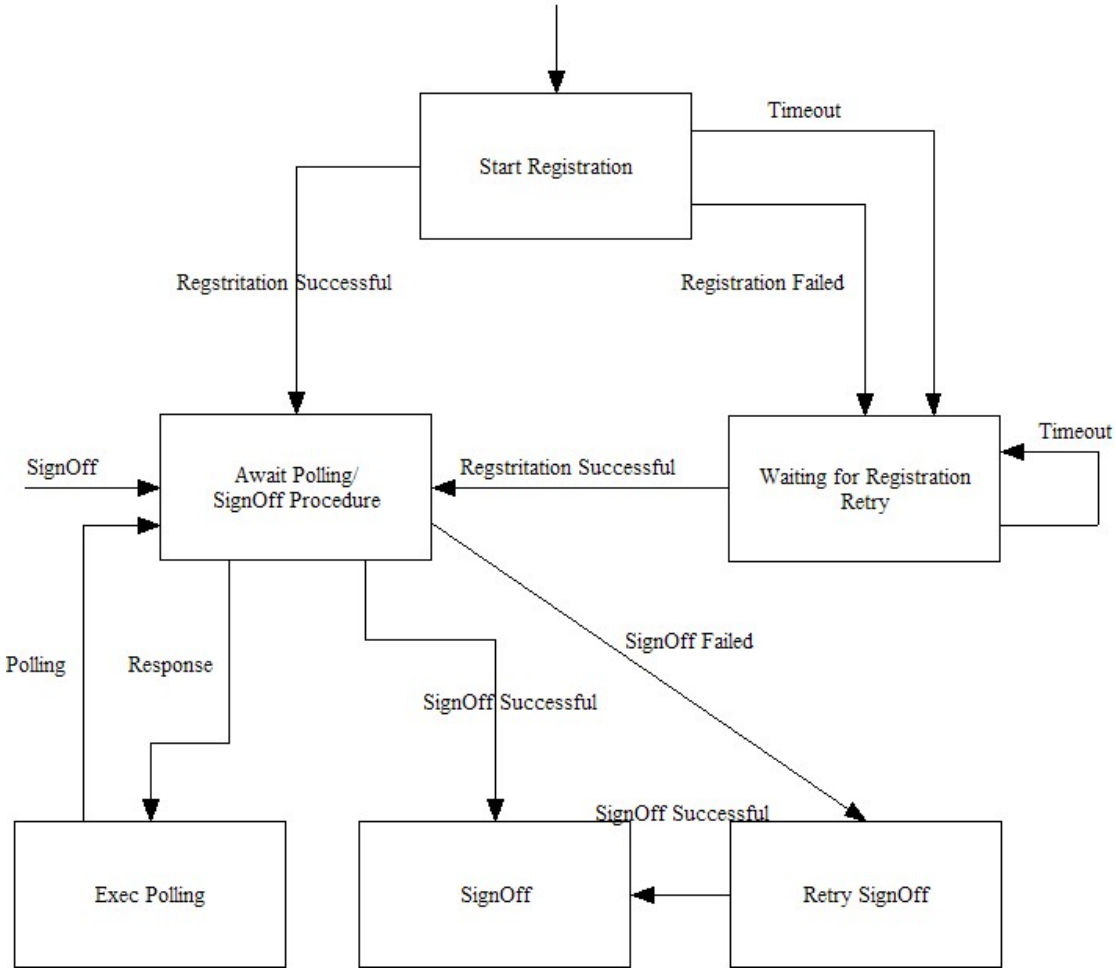

Figure 8.1: Car Node State machine

When we take a look at the state machine of our Car Node (Figure 8.1), we will see that upon external interaction (e.g. car is turned off) the registration process will start. It will start with the CN sending its registration information out on the ether to the closest GC. Now it will await a response from the network. In the cases registration fails or a timeout occurs it will try to re-register. Since its quite essential that the car gets registered even on a bad connection we set the limit to five retries. If registration is successful the CN will enter a passive mode where it will await the polling messages from the network. From this state also the Signoff procedure will be initiated. When the car wants to leave the CN will send a Signoff message to the GC. Incase this should fail one retry will be given. This is done since there will be a limit as of how many retries can be done until the car leaves the transmission range to its GC.

8.1.2 State machine for Group/Car Controller

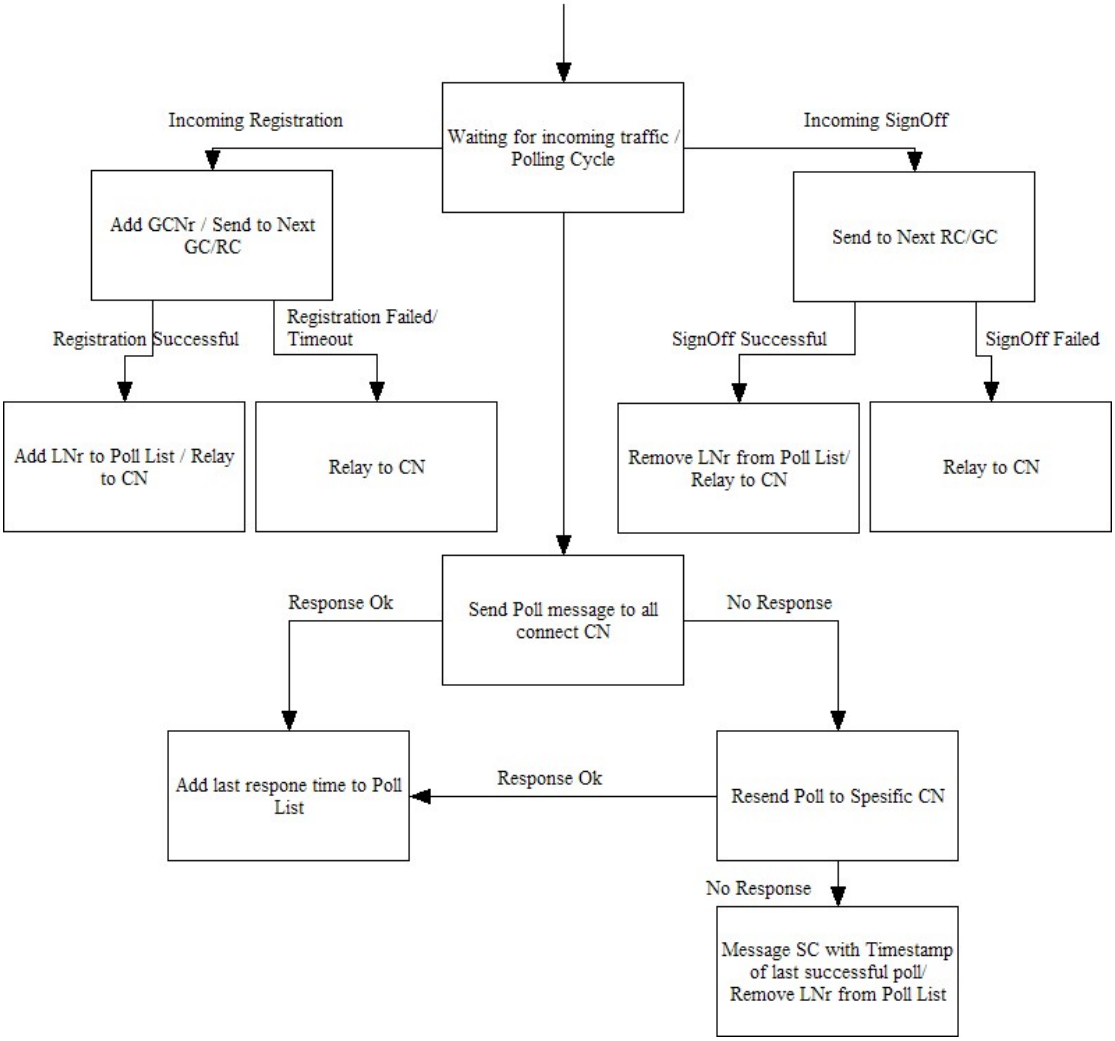

Figure 8.2: Group/Car Controller State machine

Moving on from the Car Node to the next link in the chain we get to the "work horses" of our application. The Row and Group Controllers will perform message relaying and the periodical polling. Looking at the state machine (Figure 8.2) we see that those nodes will be listening to incoming data traffic. There are three types of messages that they should be able to process. First there are the registration messages from the CN. Since the GC/RCs are not capable to decrypt those messages they do only add their GC/RC number to the end of the messages and send them further on to the next link in the chain towards the SC. When a registration is successful a GC will add the LNr to its poll list. This is about the only point where there is a slight difference between a GC and RC. The RC can in this case either store a GC and the License number (LNr) to which it has to relay packets to reach a specific car or it simply stores an LNr which is attached directly to it. The second type of incoming data is a Signoff messages from a CNs. This message will simply be relayed to next node up the chain towards and nothing will be done with it. The GC/RC awaits a response from the SC in order to relay

this message to the CN. If the Signoff was successful the CNs LNr will be deleted from the poll list. If it failed however it will remain in poll list and should it move out of sending range to the GC/RC it will be marked as stolen. This is where the last function of the GC/RC comes in. They will periodically poll all CNs they have stored in their Poll List. If a response is sent no action will be taken. If a node does not respond, a second poll to that specific node will be sent, if that one remains unanswered a notify message will be sent to the SC and the car marked stolen.

Each GC and RC will also have a registration process similar to the CN (Figure 8.3).

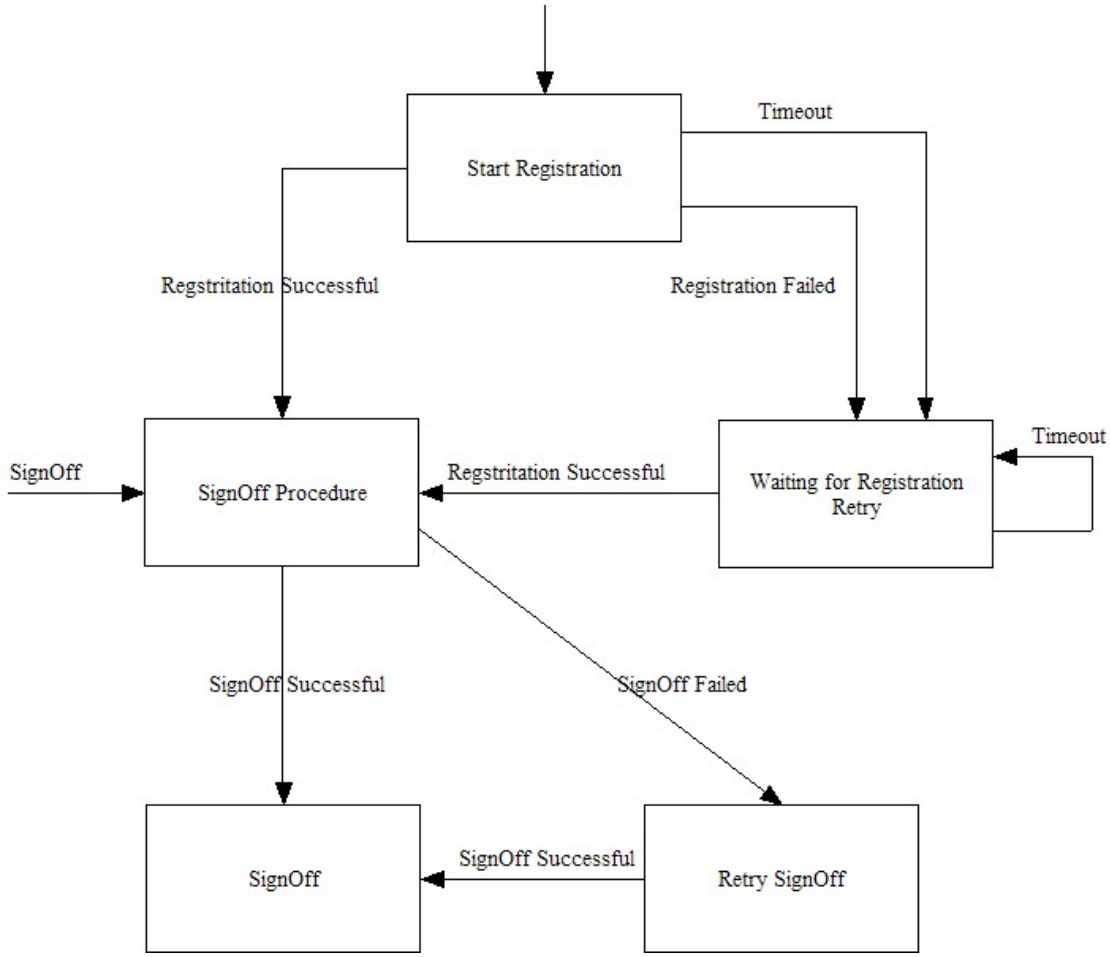

Figure 8.3: GC/RC Registration/SignOff

8.1.3 State machine for Super Controller

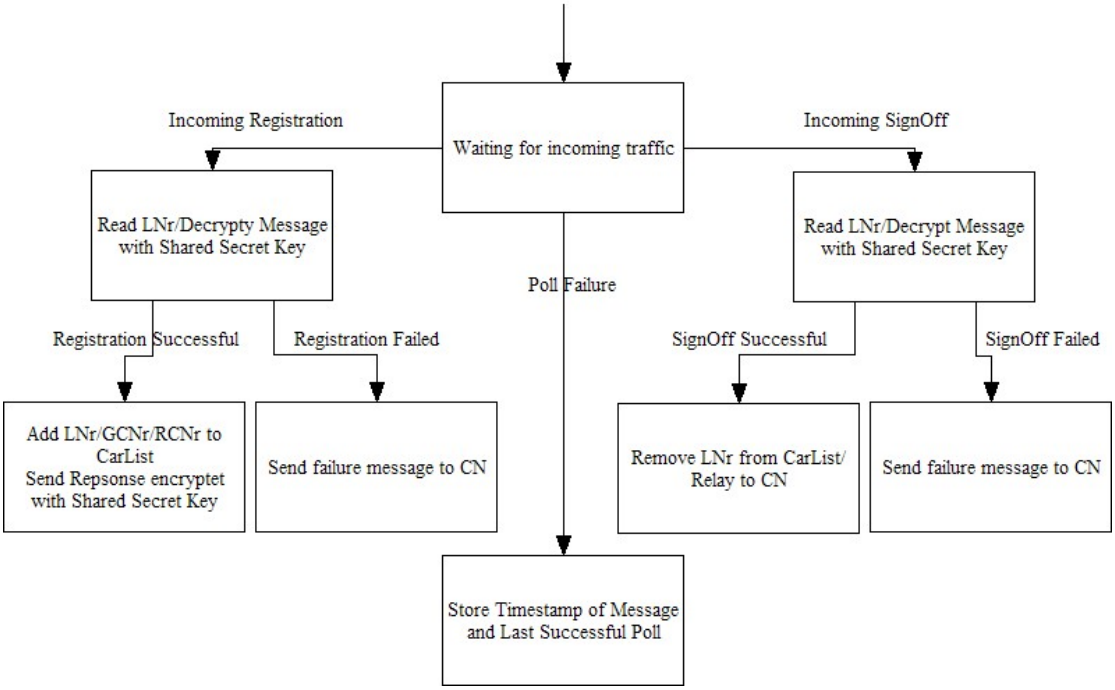

Figure 8.4: Super Controller State machine

Same as the GC/RC nodes the SC will be waiting for incoming data from the network. We define it to handle three types of data. First, will be an incoming registration from a CN relayed by GCs and RCs. It will take the LNr which is sent in plaintext and look it up in its database over all registered users of the network. In case the message decodes successfully, it will send a encrypted message back to the CN. It will also instruct all controller nodes on the way down to the CN to add the car to their list. If the message has been corrupted or is simply a fake it will only send a failure message back and ask the node to retry the registration process. Second, a Signoff message will be processed in the same way as the registration message. It will decrypt and the appropriate message will be sent. The last state is a response to a poll failure event. If a CN does not respond, a poll failure message will be sent to the CS. It will store the time of the last successful poll and when the CN was missing for the first time. Optional a system could be implemented that a SMS or any other form for messaging system will contact the owner and inform her/him of the loss of its car. It can also be sent to a central authority. Figure 8.4 illustrates the state machine described.

8.2 Message

Message flows have been presented in Chapter 7 and will not be repeated here.

9 Discussion

Since we have no prototype to test our proposed solution we can only discuss matters at hand based on theory and the model we developed.
In our general case description we choose to implement the network overview in one central node. This has the benefit that overall computational power is concentrated in one node and hence reduces power consumption throughout the network. This node may also be centrally placed so that it can be connected to the public power grid and only uses a battery as backup. This has some weaknesses though. First if the master node is compromised the entire network can be brought down. Second if the master node fails the network will also go down.

For the time being ZigBee looks like a good solution for wireless sensor networks, but there are other technologies which also can accomplish the same task, like Bluetooth. Those would however require more work as they are not designed to work as sensors. Since they are designed to fulfill different roles they have a set of different abilities. WLAN and Bluetooth have access to more bandwidth as they are designed to transmit more data than ZigBee. Bluetooth on its side again is designed to connect devices to each other such as Mobile Phones and PDAs. This again gives it less transmission range. WLAN on its side is designed to send great amount of data over fairly large distances. Since increased transmission range and bandwidth increases battery consumption, we can see pretty easily that Bluetooth and WLAN are bound to use more than a short-range and low bandwidth solution. This is why we also considered RFID as it may be an everlasting solution due to the nature of passive chips. It has short range and low bandwidth. However, as shown in chapter 5 the range proved to be too limited. To keep the number of nodes as low as possible, we needed a reasonable transmission range. RFID was inadequate in any way doing this. Bluetooth was scratching the lower edge of the limitations we set. It might have been possible to rewrite the protocol stack of Bluetooth and use stronger transceivers, which however only would lead to fairly increased power usage. WLAN's range was more than sufficient; however we don't need 11+ Mbps/s data rates for sending some sensor information. Taking into consideration that WLAN is not designed to be used on battery driven devices its power consumption is fairly much for our case. An implementation would bring physical difficulties as it would require power cords to each transceiver node placed on the parking lot.
There are also some technologies not discussed in this thesis, which might work as a solution. Those technologies were not taken into consideration as they are on a research project basis (e.g. SmartDust). ZigBee is today's only commercial solution designed for sensor networks.
In the course of the study, we looked at several different routing protocols. However, we decided to use the routing protocol that follows ZigBee. This decision was made after

considering the level of security we wanted. Since ZigBee is, as stated before, designed to operate in a semi ad-hoc scenario, it has a routing protocol designed for exactly doing this task for us. This will ease the work when creating a prototype as the entire protocol stack of ZigBee does support our application. Taking a brief look on Bluetooth, we came to the conclusion that the security in Bluetooth is considered fairly weak. Here we would have to rewrite the protocol stack to make it work in our scenario. With ZigBee we get a developer environment, which will let us build our solution directly upon the ZigBee protocol stack. It will take care of re-transmission and general transmission security.

Based on the theoretical part we anticipate it to perform well in a real life scenario. However, since we don't have a prototype, so we do not know how it performs in a life like scenario. We also have to add that it's a fairly new technology and there may be weakness we are not aware of and that has not yet been discussed.

As we try to achieve a secure system which will take care of the sensitive data sent over the network, we analyzed several different routing protocols. Among them we analyzed some protocols which don't implement any security at all. They serve as a foundation for a couple secure protocols though. We found none of them to be a better choice than the routing protocol already implemented in ZigBee. The Rinjdael algorithm which AES uses is said to be one of the more secure cryptology methods in use today. And as ZigBee uses AES we conclude that there is, as far as we can see, no need to rework ZigBee's security architecture, as the solutions are not likely to be any more secure. Regardless of this we do not know the real strength of the ZigBee security architecture, as it hasn't been tested thoroughly. We didn't find any information on analysis made in thought of breaching the security of ZigBee.

Out of this reason we implemented a Shared Secret Key system to make the system more secure and since we needed a way to authenticate a car, independent of location and what technology used. The main purpose of this solution is to achieve authentication, as we need to know who is registering with us and that would require too much data capacity if the system is to be implemented at different locations. This however has the positive side effect that we get an addition layer of security.

Bottom line is even if ZigBee should show to be insecure our system would only partially be compromised. We send some data plaintext but it is not key information which can bring the network down. Doing excessive research on the Internet we did find very little information about weaknesses of the encryption protocols used and we guess this has yet to come as it is a new technology and not quite widespread yet. When its popularity increases we will naturally also see an increase in attacks against such systems and with this learn in how far ZigBee is truly secure.

10 Future Work

Since we didn't get to create a prototype and to test our proposed solution more thoroughly there is more work to be done. The first instance would be to create the SPIN [31] model and test it on logical errors. After that, it would be natural to create a prototype of our solution before any eventual product can be sold. Furthermore since ZigBee is such a new technology a study regarding its transmission security should be conducted.

11 Conclusion

After studying several different technologies we got to the conclusion that there are many out there which can do the job, but require very much work to get it working. Also they will require a lot more maintenance due to shorter lifetime of the devices. ZigBee offers a complete package, which will give us what we need to create a product solution. As stated in chapter 6.11 we concluded that the security architecture implemented in ZigBee is secure enough. There are as noted in the discussion unknown factor which yet remain to be discovered. In that case the ZigBee protocol stack is very small and easy to get an overview over, so it can be modified to fix upcoming problems. Seeing the great number of members in the ZigBee alliance, it sure is a technology with future as so many companies decide to support it.

Considering our solution in a market situation, would our solution have any chance of surviving? Our system would only be truly effective if your car would be secured at any location. This would imply that each parking lot in a country would have this system installed. Seeing there are many different operators in this market, this will be a task next to impossible.

References

[1] P.Nicopolitidis, M.S. Obaidat, G.I. Papadimitriou and A.S. Pomportsis. “Wireless Networks”, Wiley 2003

[2] RFID journal: http://www.rfidjournal.com/article/articleview/207#anchor#008 (28.04.2004)

[3] Robert Morrow. “Bluetooth Operation and Use”, McGraw-Hill Professional 2002.

[4] D.M. Bakker and Diana McMichael Gilster. ”Bluetooth End to End”, John Wiley & Sons Inc 2002

[5] Marjaana Träskbäck, “An overview of Bluetooth Security” http://www.cs.hut.fi/Opinnot/Tik-86.174/Bluetooth_Security.pdf (26.04.2004)

[6] Jon Adams (Motorola), “What you should know about the ZigBee alliance” (24.09.2003) http://www.zigbee.org/resources/documents/Adams-Heile_SensorsExpo_AnaheimSept03_V1_000.ppt (23.04.2004)

[7] Joan Daemen, Vincent Rijmen, “Note on naming” (03.09.1999) http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael-ammended.pdf (26.04.2004)

[8] ZigBee Prices, http://news.zdnet.co.uk/business/0,39020645,2133331,00.htm (23.04.2004)

[9] AODV, http://moment.cs.ucsb.edu/AODV/aodv.html (26.04.2004)

[10] David B. Johnson David A. Maltz Josh Broch, “The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks (DSR)“ http://www.monarch.cs.rice.edu/monarch-papers/dsr-chapter00.ps (26.04.2004)

[11] Xiaoqi Li, Michael R. Lyu, and Jiangchuan Liu, “A Trust Model Based Routing Protocol for Secure Ad Hoc Networks (TOADV)” http://www.cse.cuhk.edu.hk/~lyu/paper_pdf/Aero04_TAODV.pdf (26.04.2004)

[12] Manel Guerrero Zapata and N. Asokan, “Securing Ad hoc Routing Protocols (SAODV)” http://lambda.cs.yale.edu/cs425/doc/zapata.pdf (26.04.2004)

[13] Seung Yi, Prasad Naldurg and Robin Kravets, “Security-Aware Ad-Hoc Routing for Wireless Networks (SAR)” http://www.cs.uiuc.edu/Dienst/Repository/2.0/Body/ncstrl.uiuc_cs/UIUCDCS-R-2001-2241/pdf (26.04.2004)

[14] Digests and digital signatures: http://www.busan.edu/~nic/networking/puis/ch06_05.htm (26.04.2004)

[15] B. Preneel, “Analysis and Design of Cryptographic Hash Functions”, Ph.D. Thesis, Katholieke University Leuven, 1993.

[16] Adrian Perrig and Robert Szewczyk and Victor Wen and David E. Culler and J. D. Tygar, “Mobile Computing and Networking” ,”SPINS: Security Protocols for Sensor Networks”, pages 189-199, 2001

[17] Kimaya Sanzgiri and Elizabeth M. Belding-Royer. “Authenticated Routing for Ad Hoc Networks (ARAN)”

[18] A. Menezes, P. van Oorschot, and S. Vanstone, “Handbook of Applied Cryptography” Chapter 6 - Self-synchronizing stream ciphers, CRC Press, 1996.

[19] Stream Cipher, http://www.rsasecurity.com/rsalabs/faq/2-1-5.html

[20] Bluetooth chip price http://www.cas.mcmaster.ca/~wmfarmer/SE-4C03-02/projects/student_work/mitrovm.html

[21] RFID: http://rfidjournal.com/article/articleview/720/2/4/

[22] Sorin M. SCHWARTZ, Frequency Hopping Spread Spectrum (FHSS) vs. Direct Sequence Spread Spectrum (DSSS) in the Broadband Wireless Access and WLAN Arenas, ver.6, 2001

[23] Part 11: wireless MAC and PHY specifications: High speed physical layer in the 5 GHz band,” P802.11a/D6.0, May 1999.

[24] Part 11: wireless MAC and PHY specifications: P802.11/D10, Jan 1999.

[25] WPA : http://www.wi-fi.org/OpenSection/protected_access.asp

[26] DES : FIPS PUB 46-2,1988, http://www.itl.nist.gov/fipspubs/fip46-2.htm

[27] Stream Cipher, http://www.rsasecurity.com/rsalabs/faq/2-1-5.html

[28] Block Cipher, http://www.rsasecurity.com/rsalabs/faq/2-1-4.html

[29] R.L. Rivest, “RFC 1321: The MD5 [29] Message-Digest Algorithm”, Internet Activities Board, 1992

[30] National Institute of Standards and Technology (NIST), Announcement of Weakness in the Secure Hash Standard, 1994.

[31] Spin: http://spinroot.com/spin/whatispin.html#A

[32] Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Std 802.15.4™-2003

[33] Ronald L. Rivest, “The RC5 encryption algorithm”, 1997