Master Thesis
Computer Science
Thesis no: MCS-2007:08
March 2007

Decision Making for Finding an Adequate Security Level

- Providing trade-off between Performance and Security

Sergey Smirnov

Department of
Interaction and System Design
School of Engineering
Blekinge Institute of Technology
Box 520
SE – 372 25
Ronneby
Sweden

This thesis is submitted to the Department of Interaction and System Design, School of Engineering at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Sergey Smirnov
Address:
Lindblomsvägen 119
SE-37233 Ronneby
Sweden
E-mail: ssmirnow@msn.com

University advisor(s):
Henric Johnson
Department of Telecommunication Systems
School of Engineering

Department of
Interaction and System Design
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

Internet : www.bth.se/tek
Phone : +46 457 38 50 00
Fax : + 46 457 102 45

Abstract

The new opportunities that come with the Internet as a worldwide network bring the new threats and risks for private, institutional and corporate users. Therefore, it is important to integrate the security mechanisms into a network environment. Due to the significant increase in computers speed and features of applications, the people are not able any more to make quick and adequate decisions about which security mechanisms should be applied at the moment. In most cases they choose the strongest security level available. Along with the high security this approach brings additional costs and resources consumption and drastically reduces the performance of devices with limited resources. For such devices a trade-off between performance and security should be provided. Most of the time there are no risks and threats to devices since there are not under attacks, and the use of strong security wastes the available resources.

A user of computer networks and electronic devices (e.g. PCs, smartphones, PDAs) is faced with a wide range of different security mechanisms. These mechanisms differ in terms costs, complexity of used cryptographic algorithms, types of licence, processing speed, and required resources. The user has to make a decision on which security mechanism to apply. This decision is often based on user's preferences, device capabilities and available resources.

While a broad range of security mechanisms has been developed to secure devices and networks, too little attention is given to actual process of making a decision about the required security level with respect to the set of predefined requirements.

The main goal of this thesis is the developing of a practical decision making model for dynamic reasoning about an adequate security level providing trade-off between security and performance.

The thesis presents the methodology for security metrics identification, selection and quantification. The developed approach is not limited to a particular system or number of metrics. The scheme can be used to select and quantify security metrics for any decision making models and different systems under consideration.

This thesis analyses the range of decision making methods for their fitness to fulfil the main goal of this work. Three models are developed based on fuzzy reasoning, simple multi-attribute rating technique (SMART) and artificial neural networks (ANNs) for making decisions about an adequate security level. The models take into consideration the selected metrics (e.g. threat level, location, content, resources), and user's preferences and make a recommendation regarding security level. The models differ in terms number of security metrics used, user's intervention into decision making process, and number of security levels.

Finally, the thesis presents the results of the experiment that has been conducted to evaluate a performance of the adaptive approach for selecting an adequate security level. The motivation for this experiment is based on the fact that decision making process requires additional computations, which can lead to increased resources consumption and can make the use of adaptive approach impractical. The results show that with right software design and implementation the computations related to adaptive approach does not decrease the performance of mobile devices. Furthermore, the use of the adequate security level improves the resources utilization for memory and battery life. The improvements are feasible already for small data rates (~3.4 Mb). Thus, for the real life scenarios with the data rates of hundred megabytes, we can expect significant improvements in resources usage by using an adequate security level

Keywords: Adaptive security, adequate security level, trade-off between performance and security

Acknowledgements

I would like to express my deep gratitude and appreciation to my advisor Dr. Henric Johnson from Blekinge Institute of Technology, for his patience, advice and constructive criticism.

I also express my gratitude to Dr. Nedelko Grbic from Blekinge Institute of Technology, who provided valuable suggestions and interesting discussions.

I wish to thank Hamid Delalat from Blueguards AB for providing the equipment for experiments.

I would like to thank my parents, Gennadiy and Nina, my brother, Vitaliy, who encouraged and supported me through the studying years. I also would like to express my gratitude to all my friends for being around to support me.

Finally, I express my gratitude to my beloved wife Tatiana, for understanding and support.

Table of Contents

Figures Index

Index of Tables

1 Introduction

1.1 Background

Network communications play an essential role in the modern life. The Internet as a global network interconnecting millions of private, business, academic and government networks, provides new opportunities, countless resources and different services for private, institutional, and corporate users. However, among the advantages, the Internet brings new threats and security risks such as worms, viruses, denial of service (DOS) attacks etc. Therefore, it is important to integrate the security mechanisms into a network environment.

Information security is a fast-growing field. The number and types of attacks against corporate, private and government networks change dramatically from one year to another. Due to the significant increase in computers speed and features of applications, the people are not able any more to make quick and adequate decisions about which security mechanisms should be applied at the moment. As a consequence they usually choose the strongest security level available. Many consider that this approach is the only alternative, but on the other hand it brings additional costs and resources consumption. Furthermore, it dramatically decreases the performance of devices with limited resources such as pocket PCs, PDAs, and smartphones. The usage of strong security mechanisms under the insufficient availability of resources can serve as a base for denial of service (DOS) attacks, when the mobile device's resources become unavailable to the intended users. Thus, a trade-off between performance and security should be provided, with the aim to optimize the resources usage, while maintaining an adequate security level.

A user of the electronic devices and networks is often faced with a wide spectrum of different security mechanisms e.g. block ciphers, Virtual Private Networks (VPNs), Secure Socket Layer and Transport Secure Layer (SSL/TSL) protocols. They differ in terms of costs, complexity of used cryptographic algorithms, processing speed, required resources etc. The user must decide which security solution to choose in order to satisfy his personal preferences, device capabilities and companies' security polices (if the device is used within the corporate infrastructure).

While a broad range of security mechanisms has been developed to secure devices and networks [32, 33], too little attention is given to actual process of making a decision about the required security level with respect to a set of predefined requirements.

A decision making model can be developed within the concept of Always Best Security (ABS), which was first mentioned by Henric Johnson in [3]. The concept states that it is not necessary the strongest security level is the proper one to choose. Instead, the “best” level is the level that combines the criteria such as personal preferences, available resources and device capabilities to achieve the optimal trade-off between security and performance.

1.2 Problem statements

Decision making for finding an adequate network security level is a complex multi-criteria decision making process. The criteria can be numerical or defined in a linguistic form when a numerical assignment is hard or impossible. The problem statements, which are addressed in this work, are presented below.

1. The first problem is how we quantify security of the system under consideration with the aim to pass this data as input to a decision making process. In order to solve this problem

we need an approach for identification, measurement and quantification of criteria for decision making process in context of the IT security. It is a difficult and important task, since the criteria should be considered under quantitative and qualitative aspects. Once defined, the approach can be used for diverse decision making methods.

2. The second problem is the choice of the decision making method. There are plenty of methods existing in decision making area. Each method has its strengths and weaknesses. Therefore, we need to evaluate existing decision making methods for the purpose of their fitness to a practical decision making model in the network security field. By using this evaluation we can limit the scope of the work to the small number of suitable decision making approaches.

3. The last problem which we address in this thesis is a practical implementation of a decision making model on the mobile device with limited resources. The main question here is, whether the performance of the device is increasing by applying an adaptive approach for the selection of the network security level dynamically. The second question is to find out if the computations, which related to the decision making process, do not increase the usage of the resources and, therefore, decrease the overall performance of the mobile device. As it has been mentioned earlier, a trade-off between performance and security on such devices is a matter of importance.

1.3 Contributions

The focus of this thesis has been to develop a practical decision making model for adaptive selection of suitable security levels. In this context, this work provides the following contributions:

- A methodology for security metrics identification, selection and quantification has been proposed. This approach can be used for wide range of decision making methods to specify the input data for a decision making process. The example of using this methodology has been shown using a Pocket PC device. The proposed approach is not limited to a particular system, device or number of security metrics. The scheme can be used to select and quantify security metrics for any decision making model and any system under consideration.
- Three models based on fuzzy reasoning, simple multi-attribute rating technique (SMART) and artificial neural networks (ANNs) for making decisions about an adequate security level have been developed. The models are different regarding to number of metrics used, number of security levels and user's intervention into decision making process. The fuzzy reasoning approach is ready and easy to use even for users with no background knowledge in security area. However, the SMART and Neural Network approaches require further improvements.
- The performance of the adaptive approach for security level selection has been experimentally evaluated on a mobile device with limited resources. The results of the experiment show that with proper software design and implementation, the use of decision making tool for selecting an adequate security level does not decrease the performance of mobile devices due to additional computations. Furthermore, the adequate security level allows to achieve improvements in resources utilization even for a small data rates (~3.4 MB). For real life scenarios with the data rates of hundred megabytes, we can expect the significant improvements in resources usage by applying the adaptive approach for security level selection.

1.4 Research Methodology

The research methodology used in this work has been chosen with respect to the problem statements.

The conducted research has been based on data collection, data analysis i.e. interpretation of the results and findings, and data verification. The research has been logically organized moving from questions to answers, and conducted with the accuracy providing the replicable results.

The adaptive approach for dynamic security level selection has been empirical tested and verified on a mobile device with limited resources.

The information about the particular research methods will be also included in the individual chapters.

1.5 Related Work

This section presents the current situation of decision making in the network security area.

Much research has been done in the area of detecting attacks over a network such as Denial of Service Attacks (DOS) and Distributed Denial of Service Attacks (DDOS). For example in [14] a network security architecture was proposed that applies data mining technologies to analyse the data collected by an Intrusion Detection System. Once the DDOS is recognized the thresholds will be adjusted immediately in order to block the attack. The data mining is based on the classification rules. In order to obtain the rules the data miner has to be trained on the active data from the database marked manually as DDOS relative or DDOS non-relative. The theoretical framework to detect attacks based on the concepts of cost analysis and pricing under uncertainty found in economics and finance was developed in [15].

Some research has been conducted in order to apply Artificial Neural Networks to detect DDOS attacks [4, 5]. Artificial Neural Networks in the form of a Multi-Layer Perceptron (MLP) were used in [4] as classifier. The inputs of the MLP were metrics from different types of passive measurements that are available to a network administrator (i.e. packet capturing). The metrics were used to feed MLP, train it and evaluate its performance in terms of 'false positive' and 'true positive' rates. It was considered that the detections approach can classify the attacks with high true positive rates (75%) while keeping false positive low. Other related work topics are included in the individual chapters.

In [3] a practical decision making model based on the Analytical Hierarchy Process was developed by Henric Johnson. The model considers a set of factors such as subjective and objective aspects of the security in order to select an adequate authentication level.

Creighton et al in [30] designed an application called Context Aware and Adaptive Security Manager (CASM), which was used to determine the suitable security protocols for different categories of networks. The Analytical Hierarchy Process [16] was used for the decision making process.

1.6 Thesis Structure

The thesis is organized as follows. Section 2 starts with presenting an overview of the existing decision making methods (Section 2.1 and Section 2.2) and their fitness to the main goal of this work. Section 2.3 presents an approach for classification, selection and quantification of security metrics. Sections 2.4, 2.5, and 2.6 describes three models for decision making in the network security area based on Fuzzy Reasoning, Simple Multi-Attribute Rating technique and Artificial Neural Networks correspondingly. Section 3 presents the experimental evaluation of the adaptive approach for selecting an adequate security level. Finally, conclusions and future work are presented in Section 4.

2 Main Part

2.1 Classification of the Decision Making Methods

Decision making is a wide scientific field. There are thousands of articles and hundreds of books denoted to this subject. In this chapter we present a short classification of the existing classical and non-classical decision making methods with the respect of their fitness to the main goal of this master thesis - decision making in the IT security area.

Decision Making Methods can be divided into single criterion and multiple criteria methods. Further we will consider only multiple criteria methods on the grounds that decision making in the network area is a multiple criteria decision problem. We can distinguish between methods with finite number of alternatives and infinite number of alternatives. The problems with finite number of alternatives are called multi-attribute decision making problems and the problems with infinite number of alternatives are called multi-objective decision making (Table 2.1.1).

Decision making methods		
Single Criterion methods i.e. - Linear programming - Nonlinear programming - Discrete optimization	*Multiple criteria methods*	
	Methods with finite number of alternatives	Methods with infinite number of alternatives

Table 2.1.1: Decision Making Methods Classification

Multi-Criteria Decision Making Methods (MCDM) are suitable when a decision-maker has to consider multiple and often conflicting objectives. There are numerous models that have been proposed to structure and solve MCDM problems:

1. *Multi-Attribute Utility Theory (MAUT)*

   The MAUT family methods [27] are based on linear additive or simple multiplicative models for aggregating single criterion evaluation. They are most appropriated for the analysis of discrete alternatives. The Simple Multi-Attribute Rating Techniques (SMART) [2] is the simplest form of the MAUT methods.

2. *Outranking Methods*

   The outranking methods require pairwise or global comparisons among alternatives. It is practical when the number of alternatives is not too large. The examples of outranking methods are the Analytical Hierarchy Process (AHP) [16], ELECTRE family methods [17], PROMETHEE [18] methods.

3. *Mathematical or compromise programming*

   The compromise programming is used when the variables are continuously distributed. It is suitable for the analysis of complex environmental and land use problems.

4. *Group Decision and Negotiations theory*

   Group Decision is an aggregation of different individual preferences on a given set of the alternatives to a single collective preference. Negotiations theory considers how group of

individuals should and could make collaborative decisions.

5. *Non-classical MCDA approaches*

They include a Fuzzy MCDM, Decision Rule Approach, and Verbal Decision Analysis [23]. A Fuzzy MCDM attempts to deal with uncertain and imprecise knowledge and possibly vague preferences.

There are three main types of solutions for MCDM problems:

- *Selection.* These methods lie on a selection of a small number of "good" alternatives (actions) in such way that a single alternative can be chosen.
- *Sorting.* Each alternative from the set A is assigned to one of the number of predefined categories (sorting or classification operation). The assignment should be based on the intrinsic measure of a criterion for an alternative and not on its comparison with other alternatives from A.
- *Ranking.* A preference ranking is established on the set of alternatives A. The preference ranking represents a priority list of the alternatives.

The typical steps of the MCDM include:

- Establish decision goal
- Identify the alternatives
- Identify the criteria
- Assign the criteria scores
- Standardizations
- Combine weights and priority scores by aggregation or preference ranking of the alternatives.
- Sensitivity analysis
- Final recommendation.

All MCDM methods have the main withdraw that they require a human intervention to the decision making process. During the step of assigning criteria scores the decision-maker subjectively makes the judgements about the values.

The second withdraw is related to the outranking methods, which are based on the pairwise comparison of actions. As a consequence, such methods need a lot of computations that increases power and energy consumptions especially at the mobile devices. Furthermore, with increasing number of alternatives the amount of pairwise comparisons grows drastically.

The mathematical programming, group and negotiations theory models aim to the specific areas of applications as described above and, therefore, they are excluded from the consideration.

Another approach for decision making is the emulation of a human thinking by applying a mathematical model. Unfortunately, much mathematical description looses the dynamic nature of the thought process [25]. However, two techniques have been successfully applied to model human reasoning: fuzzy expert systems and artificial neural networks. The techniques have found applications in different engineering fields including classification problems in the network security area [14].

Both fuzzy systems and neural networks can be used to solve the same problems, but the conditions, under which they should be used, differ.

Decision Making Model(s)	Advantages	Disadvantages
MAUT	Simple computations	Methods require human intervention to the decision making process.
Outranking methods	Measurement of qualitative criteria	- Complex computations. - Methods require human intervention to the decision making process.
Fuzzy systems	Training data is not needed	Presence of domain expert
Neural networks	Domain expert is not required	Presence of prior obtained training data.

Table 2.1.2: Comparison of Decision Making Models

The reasons to use Neural Networks include:

- There are enough data to form a training set of actual inputs and correct outputs corresponding to these inputs;
- There is no expert, who has an idea how outputs are related to inputs;
- We are not particularly interested how outputs are related to inputs, as long as the system works. In this case Neural Networks act as a "black" box mapping outputs to inputs.

Conditions under which a fuzzy expert system may be the best approach:

- We have a domain expert who has a knowledge how outputs are related to inputs;
- We do not have sufficient data to use as a training set;
- We are interested in the way in which outputs can be derived from inputs.

Table 2.1.2 summarizes advantages and drawbacks of the decision making models being considered.

2.2 Choice of the Decision Making Method

To choose the decision making methods that can be used in the IT security area, we have specified the following requirements that they should fulfil:

- Methods should not include complex calculations and require much computational power that leads to greater resources consumption at the time of the decision process;
- The results of the decision making process should be reliable;
- Methods in ideal case should require a minimum human intervention in the decision making process or do not require human intervention at all.

We have used the results of the comparison of different decision making methods found in [6,7] . In [6] the authors analysed three methods for multi-criteria decision: AHP and ELECTRE, which are based on pairwise comparison of attributes and Simple Multi-Attribute Rating Technique (SMART) that uses orders of magnitude of ratios. The AHP and the SMART methods generate cardinal information, the final scores of alternatives, whereas ELECTRE makes the alternatives in complete or incomplete order. The experiments showed that the final rank orders of the alternatives produced by ELECTRE and other two methods turned out to be very similar.

The findings of an experimental study of the comparison of two multi-criteria methods for project selection are presented in [7]. Student subjects were asked to solve a simple and a

complex problem, by either using AHP or the SMART method. The results showed there was no significant difference between AHP and SMART concerning the ability to elicit goals and preferences. For the complex problem, SMART was scored higher due to the increased complexity of using pairwise comparisons for complex problems in the AHP method, which made it less attractive than SMART. For the simple problem no significant difference was found between the methods in a problem classification. These findings show that the AHP is less effective in improving the decision makers understanding of the decision making problem. The AHP method requires a large number of pairwise comparisons and computations for the complex problems.

We have selected the following methods for further investigation in this work:

- *Fuzzy Approximate Reasoning*

  Fuzzy reasoning has been selected as the method that does not require heavy computations. Despite the fact that the method needs a domain expert to construct the model, we are interested in the way how the adequate security levels may relate to the input parameters.

- *SMART*

  The method has been chosen due to the same level of performance and less computational effort in comparison with the outranking methods.

- *Artificial Neural Networks (ANN)*

  Neural networks have been chosen as a technique, which has been used to solve decision making and classification problems in the security area. Furthermore, ANNs do not require human intervention into the decision making process.

2.3 Quantification of Security

Before starting the decision making process we need to select security metrics to measure the performance of the system with respect to our main goal of selecting an adequate security level.

To develop the approach for security metrics identification, first, we defined common requirements, which should be fulfilled by the metrics. Then, the approach was structured into well defined steps, moving from data collection and data analysis to the actual metrics selection and quantification.

The selected metrics should fulfil the following requirements [9]:

- Selected metrics must use data that can be obtained from existing processes
- Metrics must measure processes that already exist and are relatively stable
- Metrics must yield quantifiable information (percentages, averages and numbers)
- Only repeated processes should be considered for measurement
- Metrics must be useful for tracking performance and directing resources.

To identify and select the security metrics we propose the following steps:

1. ***System identification***

   This step includes the identification of system related information such as hardware, software, system interfaces, data, and users.

2. ***Security environment identification***

   This step should identify any known or presumed threats to the system. The following aspects are to be considered:

   a) *Physical aspects*, which consist of any assumptions that need to be made about the physical location of the system or attached peripheral devices in order for the system to work in a secure way.

   b) *Personell aspects* that include any assumptions to be made about users and administrators of the system in order for the system to function in a secure way.

   c) *Connectivity aspects* that include any assumptions, which need to be made concerning connections between the system and other IT systems.

3. ***Identify goals and objectives***
4. ***Select the specific security metrics***
5. ***Quantification of the selected metrics.***

The following sections describe the example of applying the proposed scheme for a mobile device.

2.3.1 System Identification

Nowadays the usage of the mobile devices has become an essential part of everyday life. The common usage of the mobile devices includes sending encrypted e-mails and attached documents, accessing corporate networks, making secure transactions on the Internet. These functions require different security mechanisms such as block ciphers, VPN connections, SSL/TSL protocols. However, additional security increases the resources consumption based on the following reasons:

- Security mechanisms increase the power and energy consumption due to extra computations
- Additional security decreases the data rates, because the extra traffic is required for security and synchronization, and increases the amount of overhead.
- The time for data transmissions grows due to increasing processing time

A typical mobile device e.g. pocket PC has the following characteristics:

- It has limited resources i.e. operative memory, CPU power and battery life
- Usually the mobile device connects to networks using a wireless connections

- Mobile devices contain sensitive information that requires the particular security mechanisms.

2.3.2 Security Environment

Pocket PC may be located in different places where the Internet access points are available. Internet connections might be both secure, such as in corporate buildings, employee home; and insecure i.e. in airports, trains with hot spots etc. The owner of the pocket PC acts both as a user and as an administrator of the device. Pocket PC connects to the corporate networks usually using a wireless connections such as 802.xx protocols or Bluetooth. Both confidential and non-confidential information is transmitted from and to the pocket PC.

2.3.3 Goals and Objectives

The main goal is to provide always adequate security level with respect to the trade-off between resources consumption and the strength of the security mechanism. Therefore, the selected metrics should affect the decision about the adequate security level.

2.3.4 Selecting and Quantifying Security Metrics

After the analysis of the collected information in previous steps we have selected the following metrics that can influence the decision about the adequate security level:

- *Location* of the mobile device

  The location is defined geographically as a position in a physical space. Geographical coordinates enable to specify every location on the earth by the three coordinates of a spherical coordinate system. However, for some people the location means e.g. an address including a postal code, a town, a street, and a house number. Furthermore, for particular individual words like “office”, “home” etc. can also give the exact information about the location of the place. To indicate the location of the mobile devices the Global Positioning System (GPS) could be used. The GPS is a world-wide radio-navigation system formed from a set of satellites.

  The location of the mobile device has the direct impact on the decision about the required security level. However, it is hard to quantify the location respectfully to the security level. For instance, accessing the corporate network using the LAN/WLAN connection within the company's building is considered much more secure than an airports wireless hotspot. Therefore, the second case requires higher security level provided on the mobile device.

  The location can be divided into two groups with the respect to security: trusted and untrusted. By *trusted* location we understand the places where the network connection is known to be secured with the high security mechanisms. For instance, the offices have in most cases protected networks; therefore there are no needs to apply additional security mechanisms on the mobile devices by themselves. The untrusted location on the other hand defines the places that are known to have insecure networks i.e. public hot spots.

- *Threat Level*

  By threat level we understand the level of malicious activities in the network, which does not depend on the device location and may be high in both trusted and untrusted environment.

The threat level can be determined from one of the Intrusion Detection Systems (IDS). These systems are usually rule-based. For example the rule below is a sample rule from the SNORT open source Intrusion Detection and Prevention System [28]:

alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg:"mountd access";)

To every alert we can assign the threat level, which can be measured for instance, in 3 levels: high, medium and low. Different number of levels may be used. The numbers and security strength of these levels are subjective and depend on the system under consideration.

- *Resources*

  Resources play an essential role for the mobile devices with limited resources. The following resources are limited: operating memory, battery life, connectivity speed, CPU speed. The usage of high security levels under the insufficient availability of resources may lead to Denial of Service attacks against the mobile users. The additional security mechanisms lead to increased power, energy consumption and CPU load. The first reason is, that security protocols increase the amount of overhead and delay between data transmission due to increasing processing time. Secondly, high level security protocols decrease the data rates, because in most cases additional traffic is required. As the example we can consider the measurement of the operating memory and battery life time as shown in Equations 1-2.

  $$Operating\ Memory\ Level\ (\%) = (\ Available\ Memory\ * 100\ )\ /\ Total\ Memory \quad (1)$$

  $$Remaining\ Battery\ Life\ (\%) = (Remaining\ Battery\ Life\ Time\ * 100)\ /\ Full\ Battery\ Life\ Time \quad (2)$$

- *Content of transmitted information*

  Content of the transmitted data has high impact on the decision about the IT security level. The data that has been classified as non-confidential does not require strong security in comparison with the high-confidential data. For content we can use the classification system of the United States Government that includes 4 levels [31]: *Top-Secret, Secret, Confidential* and *Unclassified* data. The information about the content can be saved in meta data along with the actual data to transmit.

Table 2.3.1 summarizes the security metrics quantification.

Metric	Values
Location	Trusted, untrusted
Threat level	High, medium and low
Resources	% from the total value
Content	Top secret, secret, confidential and unclassified

Table 2.3.1: Security Metrics Quantification

Diverse decision making methods require the input parameters in different format. Some make use of qualitative non-numerical values like high, medium etc. and the rest – numerical quantitative values. However, the qualitative values can be straightforward converted to the numerical values and vice verse. For example, the High qualitative value may correspond to 100 (maximum possible numerical value), Medium corresponds to 50 (one half from the maximum value), and Low is the value close to minimum e.g. 10.

2.4 Fuzzy Reasoning Approach

Fuzzy logic was developed by Zadeh in the middle of 1960s for representing uncertain and imprecise knowledge. It provides an approximate but effective way of describing the behaviour of complex systems that can not be easily analysed by using mathematical methods. In general a fuzzy logic system is a non-linear mapping of an input data vector into a scalar output. Fuzzy logic techniques have been successfully applied in a number of applications such as computer vision, decision making, different control systems and expert systems. In this chapter we propose a fuzzy reasoning model based on scalable monotonic chaining approach for finding an adequate security level.

2.4.1 Definitions

The concept of fuzzy logic is based on the following definitions:

Definition 2.4.1: Let X be a classical set of objects, called a universal set. A fuzzy subset A of X is defined by its characteristic function, whose values can be any number in interval [0,1].

Definition 2.4.2: The value A(x) is called the grade of membership of x in a fuzzy set F, and is often denoted by $\mu(x)$.

Definition 2.4.3: A linguistic variable is defined as a set <X,T(X),U,G,M>, where X is a name of the variable, T(X) – term set, U – an universal set, G – the syntax, which generates the terms of set T(X), M – the semantics, which generates the sense for each linguistic value X.

2.4.2 Fuzzy Expert Systems

Expert systems are designed to make available some of the skills of an expert to non-experts. Such systems attempt to emulate in some way an expert's thinking patterns. They provide an effective way of describing complex systems that can not be described precisely. The fuzzy expert systems can be divided into two classes:

- Fuzzy control systems;
- Fuzzy Reasoning Systems.

A simple fuzzy control system involves *fuzzification*, *fuzzy inference* and *deffuzification* processes. A fuzzification is a process of converting a crisp input value to a fuzzy value. The process of drawing conclusions from the existing data is called a fuzzy inference. This process uses rules to map linguistic fuzzy variables onto similar variables describing the output. The defuzzification process converts the fuzzy values into a final crisp value. The typical fuzzy control system is shown in Figure 2.4.1.

The application domain of the fuzzy control systems is well defined. They work quite well with input and output numerical values. In comparison, the domain of fuzzy reasoning systems is not well defined, they can deal with both numeric and non-numeric data [22].

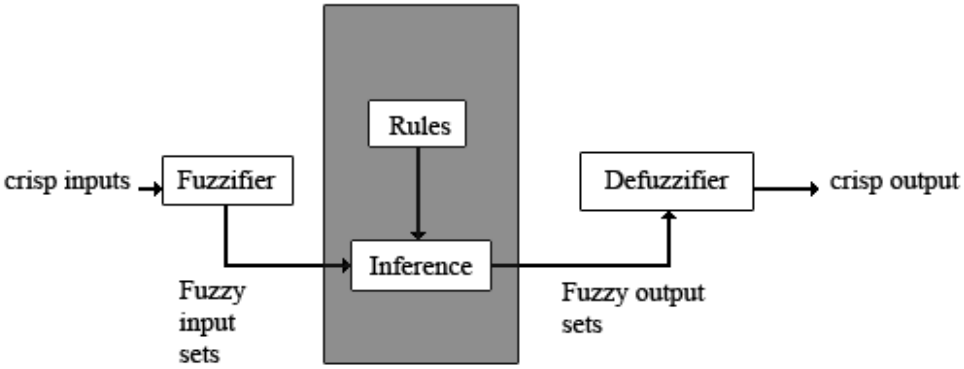

Figure 2.4.1: A simple Fuzzy Control System

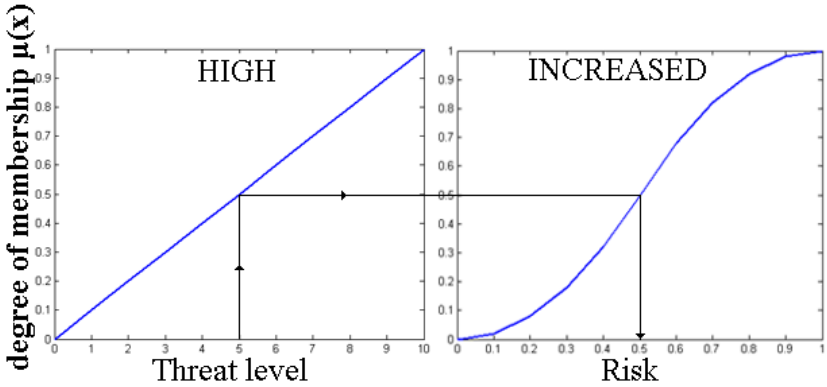

Figure 2.4.2: Two fuzzy sets: HIGH for threat level and INCREASED for risk

2.4.3 Monotonic fuzzy reasoning

Monotonic or proportional reasoning is when two fuzzy regions are related through a simple proportional inference function,

If x is Y then z is W (3)

that can be represented by the transfer function [22],

$z = f((x,Y),W)$ (4)

This fuzzy reasoning system is able to develop an expected value without going through the processes of fuzzification and defuzzification. The output value is estimated directly from the corresponding truth membership grade in the fuzzy regions. This method of fuzzy inference uses a method of implication called monotonic selection. For example, let us consider a simple computer system risk estimation model. The model consists of two fuzzy sets HIGH for the threat level (defined in range [0,10]) and INCREASED for the risk, which is defined in range [0,1] (Figure 2.4.2).

This model is based on the relationship between the current threat level and corresponding risk to the PC system. The relationship is expressed as a single conditional proposition,

If threat level is HIGH, then risk is INCREASED

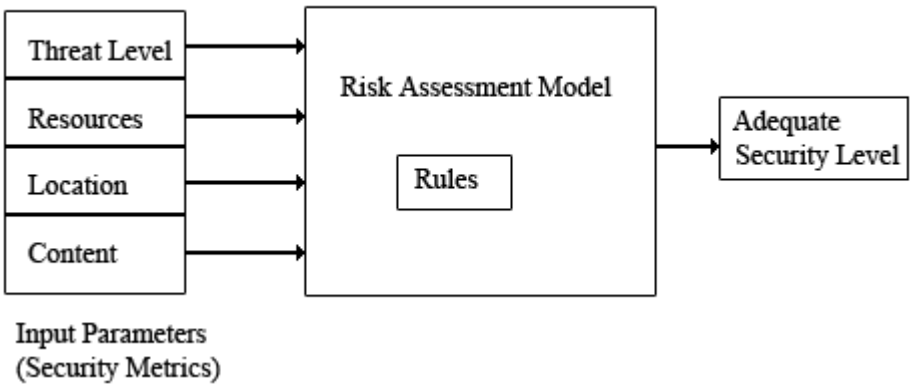

Figure 2.4.3: Fuzzy Reasoning Model for Finding an Adequate Security Level

A monotonic selection implication between fuzzy regions HIGH and INCREASED obeys the following algorithm:

- Find the membership $\mu_y[x]$ of an element x (Threat Level) in the domain of Y (HIGH);
- In fuzzy region W (INCREASED) of the membership which corresponds to $\mu_y[x]$, find the surface of the fuzzy manifold. By dropping the perpendicular line to a domain of the element z (risk) we obtain solution value. The value z is the solution to the implication function and can be expressed as,

  $z_w = f(\mu_y[x], D_w)$

Black arrow line on Figure 2.4.2 shows the example of the the implication process for the risk estimation model and the value of the threat value equal to 5. When threat level is 5, its membership value in the fuzzy set HIGH is 0.5. This truth function value is used with the INCREASED fuzzy set to find a value for the solution variable risk. We enter the fuzzy set at the 0.5 membership axis and move across until we encounter the surface of the INCREASED fuzzy set. A value for the solution variable risk is then obtained by selecting the domain value in INCREASED for this truth membership.

The fuzzy truth function can be generated from an arbitrary complex approximate expression that can be expressed in a general form as,

$$\text{if (x is Y)}\cdot\text{(k is U)}\cdot\text{(s is M)... then z is W} \qquad (5)$$

The operator · can be either the conjunctive (AND) or disjunctive (OR) operation. A general monotonic reasoning transfer function looks in this case as follows,

$$z = f(\Sigma(v_i, F_i), W) \qquad (6)$$

Where the Σ operation is the general aggregation operation.
In this case the monotonic reasoning acts as a proportional correlation function between two general fuzzy regions of fuzzy sets HIGH and INCREASED.

2.4.4 Fuzzy Reasoning Model based on a scalable monotonic chaining

For developing a fuzzy reasoning model we have used a special kind of the fuzzy reasoning called a scalable monotonic chaining that was first proposed by Earl Cox in 1994 [22].

Figure 2.4.3 shows the architecture of the model that evaluates the security metrics developed for the particular system and makes a recommendation on what security level to use. The chosen four security metrics represent just a few data points that can be used. For different systems this number can vary and different metrics can be developed and used.

As described in the previous chapter in this approach we do not create and then defuzzify the solution fuzzy set. Instead, we use monotonic chaining to map the variables specified in individual rules to an intermediate fuzzy risk variable. The result of this mapping represents the degree of a system risk for the particular security metric. Then the result of the monotonic chaining are summed to produce the total risk for the system. This scalar value, called *Total Risk,* is used to make an recommendation about the appropriate security level. The security level to be recommended is obtained from the Total Risk's degree of membership in a controlling fuzzy set (Figure 2.4.4).
We used four security metrics which were selected in Section 2.3.4. Each metric has an impact on the information system risk.

1) *Threat Level*

   The high threat level being obtained from the IDS indicates that the risk of revealing the data to the third part or loosing this data is increasing.

2) *Content*

   Sending the data over the insecure networks increases the risk for the sensitive information being eavesdropped. Therefore, the content that has been classified as high-confidential has to be secured with the strong security mechanisms. On the other hand, the non-confidential data requires low security measures or no measures at all.

3) *Resources*

   With increasing resources' consumption, the risk to the system of being put out of the service (so called Denial of Service attacks) is big. The reason is, that the usage of the strong security mechanisms under the insufficient availability of resources can lead to hanging up the system, which will no be able to go on-line or perform any other task. Therefore, the high level security should be only used if the required resources are available.

4) *Location*

   Moving from the trusted location to the untrusted environment increases the risk to the system. Respectfully, the security level should be adjusted.

The model consists of four fuzzy sets for the security metrics: HIGH for the threat level, HIGH for the resources consumption, UNTRUSTED for the location and CONFIDENTIAL for the content.

Additionally, four similar fuzzy sets INCREASED are used for the intermediate risk. The monotonic reasoning function for this model is expressed as,

$$Z = \sum_{i=1}^{N} (v_i, F_i) W \qquad (7)$$

where

N – number of metrics used in the model, for this model N = 4
Z – summarized risk value

The following conditional rules have been defined for the model:

[Rule1] If *Threat Level* is HIGH then risk is INCREASED

[Rule 2] If *Resources Consumption is* HIGH then risk is INCREASED

[Rule 3] If *Location* is UNTRUSTED then risk is INCREASED

[Rule 4] If Contend is CONFIDENTIAL then risk is INCREASED

The fuzzy sets for the metrics are modelled as the linear proportional surface or as a straight increasing line. The concept of INCREASED is modelled as a growth S-function (S-curve). S-function is a popular method of fuzzy representation used in the control engineering environment [22]. The examples of the systems and events that used the S-curves include:

- the mean-time-between-failure (MTBF) of a hard disk drive
- the risk associated with morbidity underwriting
- a project risk assessment.

A growth S-function set moves from zero membership at its extreme left-hand side to complete membership at its right-hand side. The membership function is pivoted around its 50% membership point, called the inflexion point. An S-function is defined using three parameters:

- its zero membership value (α)
- its complete membership value (γ)
- inflexion (or crossover) point (β), in which the domain value is 50% true.

The value of the function for the domain point x is given as,

$$S(x;\alpha,\beta,\gamma) = \begin{bmatrix} 0 & \rightarrow x \le \alpha \\ 2((x-\alpha)/(\gamma-\alpha))^2 & \rightarrow \alpha \le x \le \beta \\ 1-2((x-\gamma)/(\gamma-\alpha))^2 & \rightarrow \beta \le x \le \gamma \\ 1 & \rightarrow x \ge \gamma \end{bmatrix} \qquad (8)$$

By using the S-function we ensure that the concept INCREASED risk reflects the underlying characteristics of the risk value. We take as the absolutely INCREASED risk those values of the threat level whose membership values are above the 0.95. For the threat level with membership less then 0.05 the INCRESED risk is close to minimum.

Network security level	*Total Risk's degree of membership*
Very high	[0.8 , 1]
High	[0.6 , 0.8)
Medium	[0.4 , 0.6)
Low	[0.2 , 0.4)
Very Low	[0 , 0.2)

Table 2.4.1: Network Security Levels and corresponding Risk Values

After we have got the final risk value we need a way to map this risk to the security level. At this point we have to make two design choices:

- How many security levels to use?
- How to define the boundaries between the levels?

In our model we used the fixed number of the security levels with fixed boundaries. Five levels were defined based on the Total Risk's degree of membership in the fuzzy set FINAL RISK as shown in Table 2.4.1. The FINAL RISK fuzzy set was modelled using the S-function.

In the proposed model the security metrics have equal preference. It means that there is no metric which is more important than the other metrics. However, the model can be further improved by adding the preference weights to the metrics. In this case the metrics can be ordered according to their importance to the system and to the user.

2.4.5 Summary

We have proposed an approach for finding an adequate security level based on a fuzzy reasoning model known as scalable monotonic chaining. The values of the following security metrics have been used during the adaptive decision making process: *Threat Level*, *Location*, *Content* and *Resources*. Nevertheless, the number of metrics can vary depending on the system under consideration and the requirements to the decision making model.

The proposed approach can be used as a framework for developing the customized models for different systems and various numbers of security levels. In our model, we have used a fixed set of five security levels.

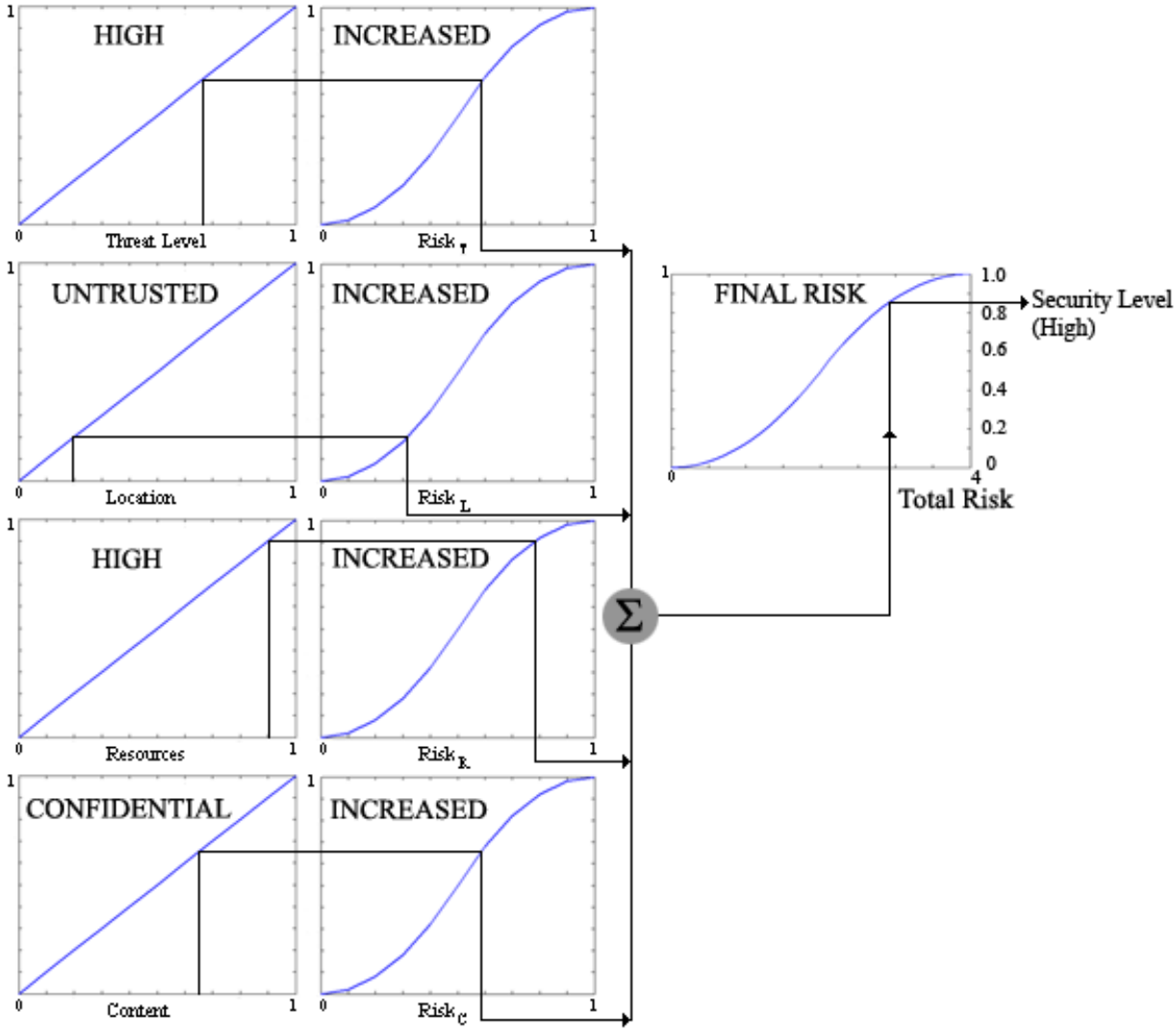

Figure 2.4.4: The Monotonic Chaining Scheme for Making the Decision about an adequate Security Level

2.5 Decision Making for finding an adequate network security level using Simple Multi-Attribute Rating Technique

The Simple Multi-Attribute Rating Technique (SMART) was proposed by Edwards in 1971 [1] and since then due to its simplicity the SMART has been widely used mainly in business and social sciences. The method does not require much computational effort and shows good performance [6, 7]. This makes it a suitable candidate for using in IT security area. However, no research has been done with the aim of adjusting the SMART approach for this purpose. The focus of this chapter is in presenting the decision making model based on Simple Multi-Attribute Rating Technique that can help to reason about a network security level in a dynamic environment.

2.5.1 SMART

The SMART is a simplest form of Multi-Attribute Utility Theory (MAUT) methods, which are based on linear additive or simple multiplicative models for aggregating single criterion evaluation. The main attraction of this approach is its simplicity in comparison with the other decision making methods.

The main steps of the SMART include:

1. Establish decision goal
2. Identify the alternatives

   The alternative courses of action should be defined.
3. Identify the criteria, which are relevant to the decision problem

   A criterion is used to measure performance of the alternatives in relation to decision goal.
4. For each criterion assign the criteria scores to measure the performance of the alternatives on that attribute
5. Determine a weight for each criterion

   How important the attribute is to the overall decision goal
6. For each alternative take a weighted average of the scores assigned to that alternative.

   It shows how well the particular alternative performs over the criteria
7. Make a provisional decision
8. Perform a sensitivity analysis to see how robust the decision is to changes in the weights.

Criteria serve as a performance measure for the application of the SMART; therefore their selection is a critical part for any decision process. To select the criteria the procedure of constructing a *value tree* can be used. We start by addressing the criteria which represent the general concerns, then decompose them to the level where they can be assessed [8]. We can judge the constructed value tree for accuracy and usefulness by using the following five criteria, which were suggested by Keeney and Raiffa [13]:

- *Completeness*

  The tree is complete when all criteria that are of the concern have been included.

- *Operationality*

  The tree is operational when all the lowest-level criteria are specific enough to evaluate and compare them for the different options.

- *Decomposability*

  The performance of an alternative on one criterion can be judged independently of its performance on other criteria.

- *Redundancy*
- *Minimum size*

The formula for a weighted average in the SMART is defined as follows:

$$U_i = \sum_j w_j u_{ij} \qquad (9)$$

$$\sum_j w_j = 1 \qquad (10)$$

where

U_i *is aggregate utility for the ith alternative*
w_j *– is normalized weight of the jth criterion*
u_{ij} *– is normalized scores of the ith alternative on jth criterion*

The following formula may be used for normalization:

$$u_{ij} = a_{ij} / \sum a_{ij} \qquad (11)$$

where

a_{ij} *- scores assigned to criteria*

The normalization allows handling of the different weighting scales. All the origin scores are converted to a common scale, which takes values between zero and one. There are two alternative approaches that can be used to measure the performance of the alternatives on each criterion direct rating and using value function. For our model we used direct rating. In this method a constant number of points that measure the relative importance of the objectives can be defined to the alternatives.

2.5.2 Decision Making Model based on SMART

This section presents our decision making model, which uses the security metrics selected in Section 2.3.4 in order to reason about the required security level. We followed the same steps that are described in method section 2.5.1 for the SMART.

1. Decision Goal

The main goal is the choice of adequate security level.

2. Alternatives

For this decision making model we defined two alternatives:

- Lightweight authentication protocol that provides only authentication functionality (low security level).
- Strong protocol, which provides encryption and authentication (high security level).

3. Criteria

We chose the following subset of criteria that describes the performance of the security protocols mentioned above:

- Confidentiality (or Secrecy)
- Authentication
- Integrity
- Non-repudiation
- Efficiency
  The criterion includes computational costs, memory usage and bandwidth.
- Robustness
  It includes the aspects such as handling packet loss and packet reordering.

The value tree of the model is presented in Figure 2.5.1. This value tree is characterized as complete, operational and decomposable.

4. Assigning Criteria Scores

To assign the scores to the criteria we used the verbal statements which correspond to the particular scores (Table 2.5.1 and Table 2.5.2).

5. Weighting

We used the direct rating method to determine the weights for criteria. Security attribute has the highest importance and consequently the highest weight (Table 2.5.3). Robustness has the lowest priority. The choice of the weights was based on our assumption that the security attributes are the most important for users of the system under consideration. For normalization Formula 11 was used.

6. Weighting Average

We calculated the weighted average using Formula 9.

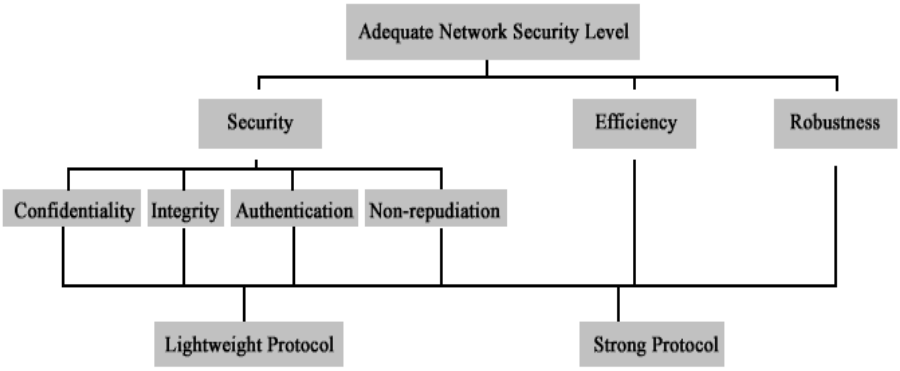

Figure 2.5.1: Value tree for decision making model in network security area

Verbal statement	*Scores*
High	100
Medium	50
Low	10

Table 2.5.1: Scores scale

Alternatives	*Security*	*Efficiency*	*Robustness*
Lightweight protocol	Low	High	Medium
Strong Protocol	High	Low	Medium

Table 2.5.2: Criteria of the alternatives and their scores

Criteria	*Weights*	*Normalized Weights*
Security	100	0.5
Efficiency	70	0.35
Robustness	30	0.15

Table 2.5.3: Weights of the criteria

Alternatives	*Weighted Average*
Lightweight protocol	47,5
Strong Protocol	61

Table 2.5.4: Weighted Averages (benefits) of alternatives

The results show that strong protocol has the maximum weighting average and, therefore, generally is preferable to lightweight protocol (Table 2.5.4).

2.5.3 Combing the decision making model with security metrics

According to the definition of criteria, they are used to measure the performance of alternatives with the respect to the decision goal. Therefore, we can not use the security metrics from the Section 2.3.4 as the criteria for the security protocols, because they describe the system being considered and not the security levels. We proposed a new scheme to connect the security metrics to the network security levels or protocols.

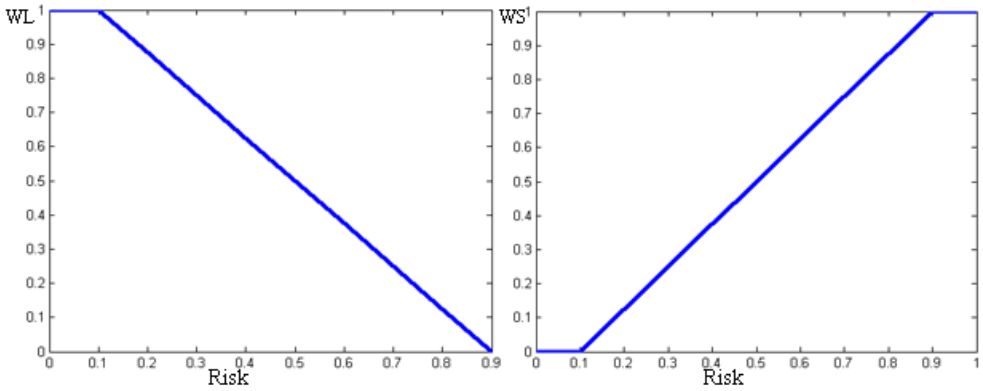

Figure 2.5.2: Weights Functions: WL - for Lightweight Protocol, WS - for Strong Protocol

A weight, which is a function of risk, is added to each alternative,

$W_i = f_i(R)$ (12)

where

W_i *- is weight for i-th alternative*
f_i *- is the weight function of the i-th alternative*
R *- is the risk for system e.g. calculated using the fuzzy reasoning approach from Section 2.4.3.*

The functions are constructed in such way that with increasing risk the weight corresponding to the strong security protocol is growing and the weight of the lightweight protocol is reducing. In opposite, with reducing risk, the weight of the lightweight protocol is increasing, and the weight of the strong protocol is decreasing. With such design we can calculate the weighted average of the alternatives taking the security metrics into consideration. The examples of the weight function are presented in Figure 2.5.2.

Sensitivity analysis is used to examine how robust the choice of an alternative is to changes in the figures used in the analysis [8]. For example, if we concern about the assigned criteria weights, we could perform the sensitivity analysis to study what would happen if these weights were changed.

We performed the sensitivity analysis to examine how robust the choice of alternatives is to changes in risk value (Figure 2.5.3).

The analysis shows that the Lightweight Protocol will be chosen when the risk level is lower than 0.45, correspondingly, for the risk higher than 0.45 – strong protocol is recommended.

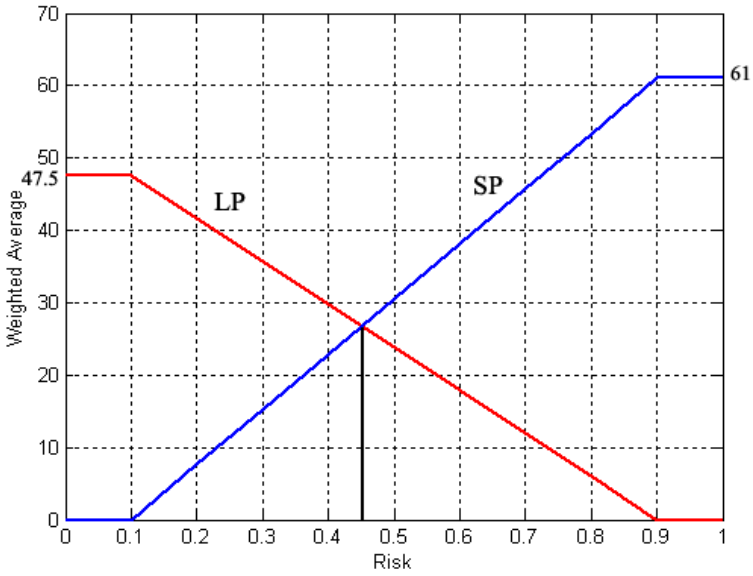

Figure 2.5.3: Sensitivity Analysis for risk value: SP – Strong protocol; LP – Lightweight protocol

2.5.4 Summary

We have developed a decision making model based on Simple Multi-Attribute Rating Technique method. The model uses both the criteria of alternatives and the security metrics collected from the information system under consideration. The security metrics are connected with the SMART approach by using additional weight factors, which depend on the information system risk. The particular steps of the decision model depend on the person or group of persons making a decision (decision makers). The subjectivity is presented in ranking criteria in order of importance and assigning scores and weights to criteria.

This subjectivity factor can be hardly removed from the model. We developed the model based on the assumption that security factors are the most important. However, an additional research is needed to study how the model reacts to different orders of importance and assigned scores and weights. In spite of presented subjectivity, sensitivity analysis shows that the proposed model adequate reacts to the changes in the metrics of the information system.

2.6 Classification of Security Metrics using Artificial Neural Networks

In this section we present the results of security metrics classification using Self-Organizing (SOM) and Learning Vector Quantization (LVQ) neural networks.

2.6.1 Artificial Neural Networks

Artificial Neural Networks (ANN) are composed of many simple processing units – neurons, that operate in parallel. These networks are inspired by biological nervous systems. Artificial neural networks find applications in many fields such as pattern recognition, identification, classification, signal processing, vision and control systems.

Figure 2.6.1 shows the model of a neuron, which forms the basis for designing artificial neural networks [29].

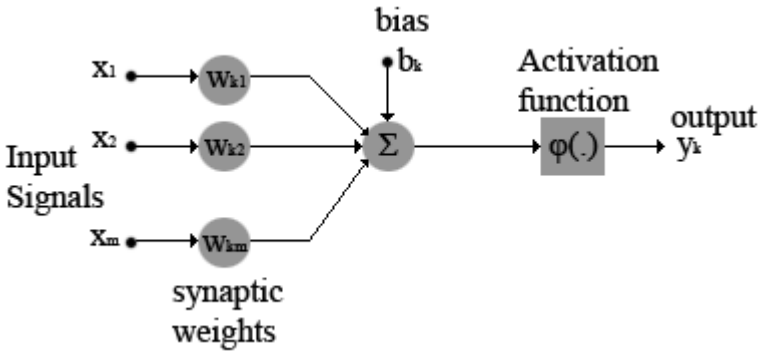

Figure 2.6.1: Non-linear model of a neuron

In mathematical terms we can describe a neuron using the following pair of equations:

$$U_k = \sum_{j=1}^{m} w_{kj} x_j \qquad (13)$$

and

$$y_k = \varphi(u_k + b_k) \qquad (14)$$

where $x_1,...,x_m$ are the input signals; $w_{k1},...,w_{km}$ are the synaptic weights of neuron *k;* u_k is the linear combiner output due to the input signals; b_k is the bias; φ(.) is the activation function; and the y_k is the output signal of the neuron.

Neural Networks have an important property: they are able to learn from input data to perform a particular function by adjusting the values of connections between neurons.

A neural network can be defined as follows [29]:

"Neural Network is a machine that is designed to model the way in which the brain performs a particular task or function of interest..."

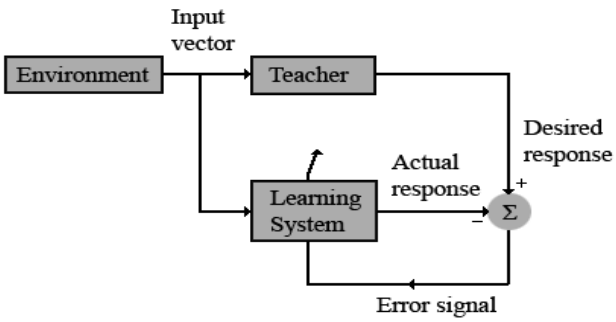

Figure 2.6.2: Block diagram for supervised learning

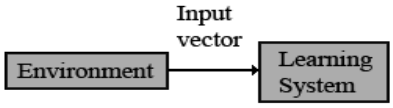

Figure 2.6.3: Block diagram for unsupervised learning

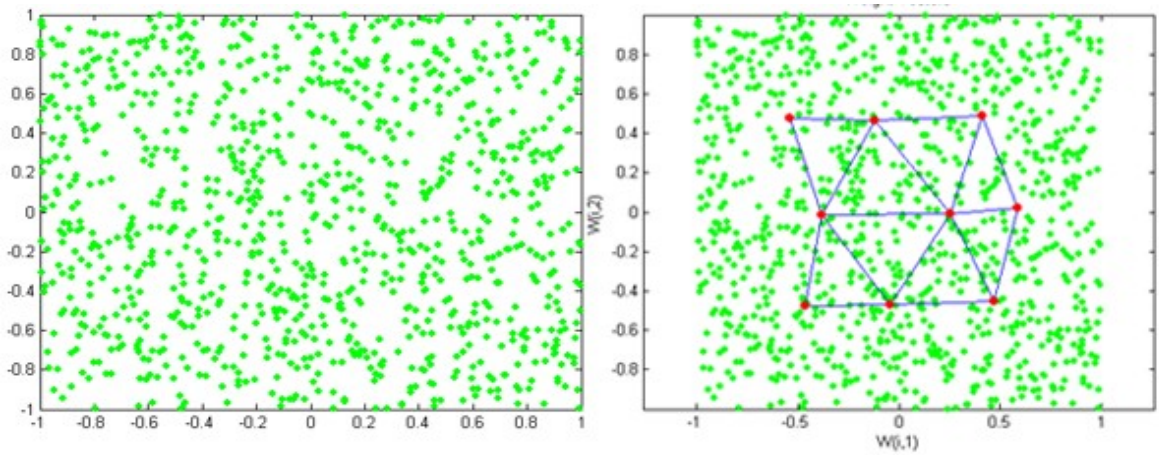

Figure 2.6.4: Classification of input vectors using Self-Organizing Maps

There are two learning approaches to train neural networks:

- *Learning with a teacher or supervised learning*

  A teacher has knowledge of the environment. This knowledge is represented by a set of input-output samples. However, the environment is unknown to the neural network. The input vector describes the state of the environment (Fig 2.6.2)

- *Learning without a teacher* or unsupervised learning

  In unsupervised learning there is no external teacher. Rather, the free parameters of the network are optimized with the respect to a task independent measure of the quality of the representation (Fig. 2.6.3).

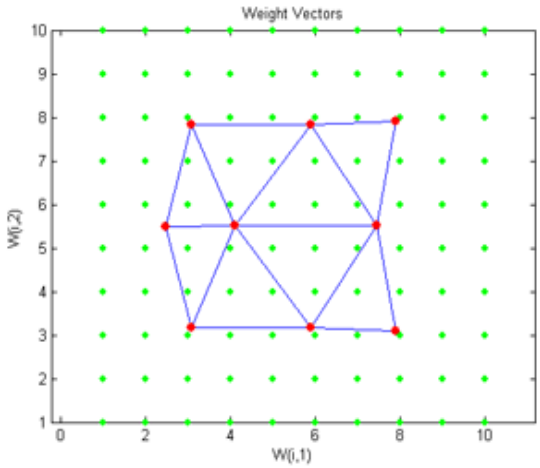

Figure 2.6.5: Security Metrics Classification

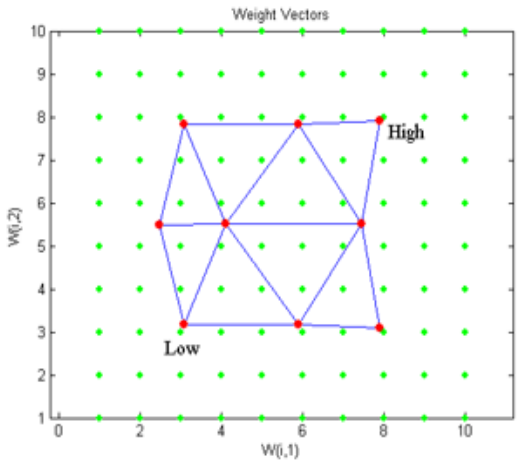

Figure 2.6.6: Assigning Security Levels to border Classes

2.6.2 Self-Organizing Maps

The problems of categorisation of the input vectors on the basis of how similar is one input with the other can be solved by using self-organizing maps.

Self-organizing maps are neural networks that can learn to detect regularities and correlations in the input data and adapt their future response to that input accordingly. Self-organizing maps learn to recognize groups of similar input vectors in such a way that neurons physically close together in the neuron layer respond to similar input vectors.

Fig. 2.6.4 shows an example of input vectors classification using a two dimensional map of 9 (3x3) neurons. The green dots represent the input vectors. The red dots on the right figure are the weight vectors, which represent the feature map. The map is trained with 100 epochs. After the training has been finished, the map is very evenly spread across the input space.

2.6.3 Classification of the Security Metrics

The previously discussed decision making approaches (Section 2.4 and Section 2.5) have the main withdraw that they require a human intervention into the decision making process. In case of fuzzy reasoning, the person acts as a expert with deep knowledge of the domain area. We needed this knowledge in order to construct the relationships between security metrics, information system risk and the corresponding network security levels. In approach based on the SMART method the decision maker judges about criteria scores.

Since the neural networks can be trained without a teacher (Section 2.6.1) to perform a particular function. We decided to apply the unsupervised learning scheme to classify the input space of the security metrics that have been chosen in Section 2.3.4. We used a self-organizing map to perform this task.

To test the abilities of self-organizing maps we started with the classification of the two-element input vectors. The two element vector presents the input that consists of two security metrics. We

did not consider particular metrics, but assumed that values of the metrics change in range [1,10].

The two-dimensional map of 3 by 3 neurons is used. The map is then trained for 100 epochs. The classification results are shown in Figure 2.6.5. As we can see a two dimensional self organizing map has learned the topology of its input space. The map classified the input data into 9 classes. However, now we need to assign the network security level to each class. We decided to use 3 levels at the beginning: *high, medium* and *low*.

As we can see from the figure 2.6.6, it is straightforward to assign two border classes to high and low network level correspondingly.

However, with the rest of the classes it is not so obvious. At this point we decided to use another neural networks method – Learning vector quantization (LVQ) to classify the classes in a supervised manner.

LVQ networks learn to classify input vectors into target classes chosen by user, what makes it suitable to classify the obtained 9 classes into 3 network security levels.

An LVQ network has a first competitive layer and a second linear layer. The competitive layer learns to classify input vectors into subclasses. The linear layer transforms the subclasses into target classifications defined by the user. The competitive and linear layers have one neuron per class. The number of neurons in competitive layer is always larger than the number in the linear layer.

The architecture of the used network is shown in Figure 2.6.7. The network has two inputs, six neurons in the competitive layer and 3 neurons in the linear layer. Therefore, the network is able to transform the competitive layer subclasses into 3 target defined network security levels classifications. Figure 2.6.8 shows the vectors of the training data and the weights after the training. The network was trained for 10 epochs.

Table 2.6.1 presents the results of the 9 classes and assigned network security levels and Figure 2.6.8 visualizes the classes assigning.

Class Number	Wx	Wy	Network security level
1	7.9209	3.1001	Medium
2	5.9179	3.1643	Medium
3	3.0801	3.159	Low
4	7.4723	5.5013	High
5	4.1114	5.5004	Medium
6	2.4799	5.4996	Low
7	7.9206	7.8997	High
8	5.9184	7.8358	High
9	3.0802	7.8399	Medium

Table 2.6.1: Assigning network security levels to classes

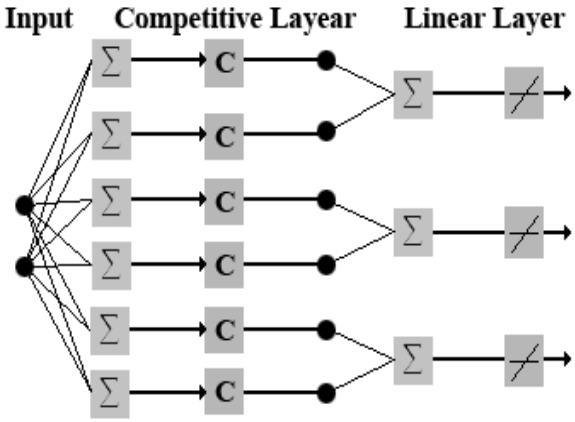

Figure 2.6.7: LVQ Network Architecture

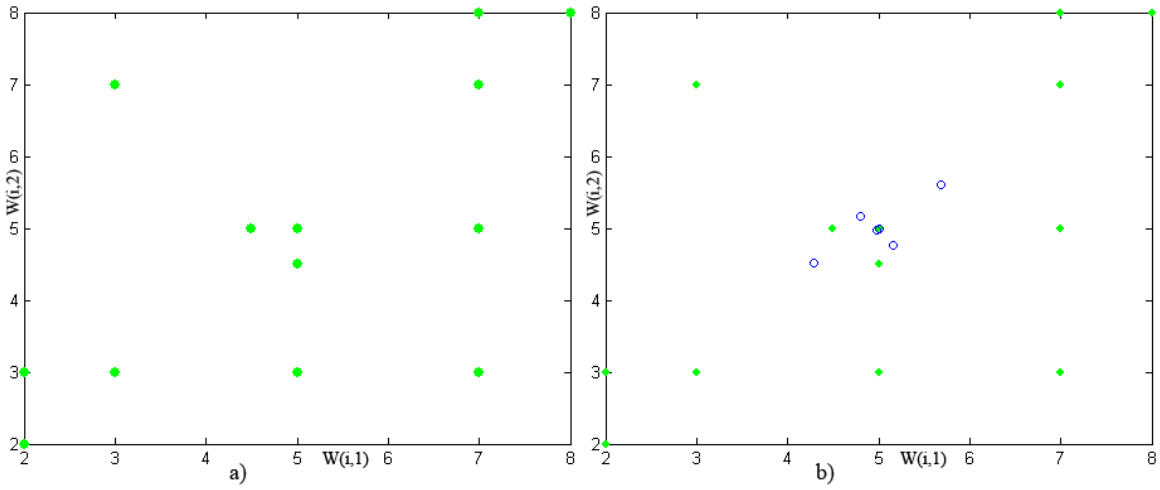

Figure 2.6.8: Training data for LVQ (a) and Weights after training (b)

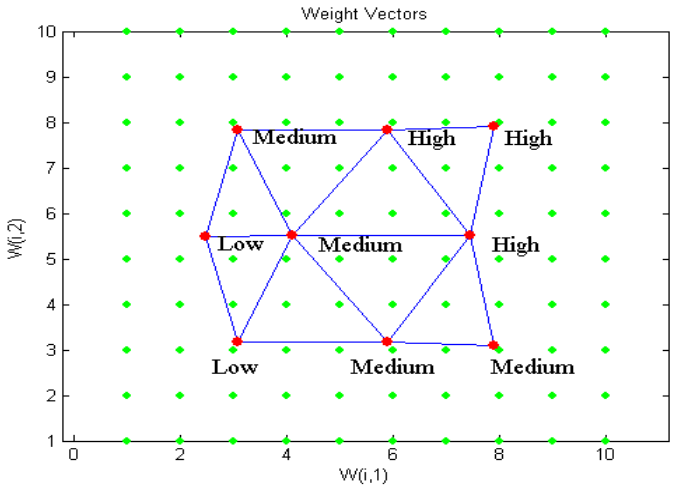

Figure 2.6.9: Assigning security levels to classes using LVQ

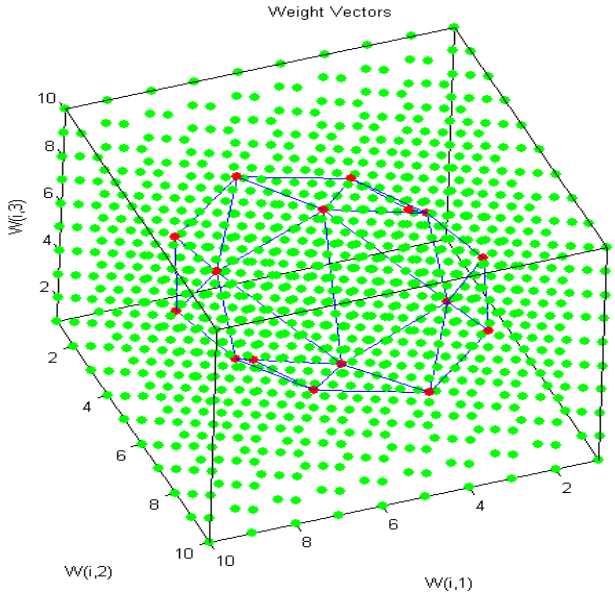

Figure 2.6.10: Classification of three Metrics into sixteen Classes

As we can see the self-organizing neural networks in cooperation with the learning vector quantization are able to classify security metrics values into different security levels. The model is applicable not only for two metrics as has been shown, but any number of the security metrics. However, the difficulty with increasing number of metrics is in producing the accurate training data for the learning vector quantization. For example, Figure 2.6.10 presents the classification of three security metrics.

2.6.4 Summary

In this chapter we have shown that the neural networks are able to classify successfully the input values of the security metrics into the number of classes. The number of classes depends on the structure of the network used for classification. The number of neurons in linear layer of the network defines the number of the classifiable target classes.

This model does not require the human intervention into the decision making process. Furthermore, once the neural network has been trained, no additional complex computations are required for security metrics classification into different security metrics. However, for training the LVQ networks we need to produce the accurate training data in order to assign network security levels to classes.

Neural networks work especially well with the non-linear input data that makes them suitable not only to classify the input vectors with predefined range, but also any inputs.

3 Experiment

The users of mobile devices such as PDAs, smartphones, and Pocket PCs are often faced with the wide range of the security mechanisms. These mechanisms differ with respect to multiple factors such as costs, complexity of used cryptographic algorithms, types of licence, processing speed, and required resources.

Due to complexity of making a decision about the required level of the security solution users in most cases choose the strongest security level available, for instance, the cryptographic algorithm with the longest encryption/decryption key. However, for devices with limited resources the "always highest security" approach brings additional resources consumption.

To help users to improve the decision making process, different models can be used (e.g. Section 2.4, 2.5, 2.6). The usage of the adaptive approach for security level selection can decrease the resources consumption due to trade-off between security and performance. However, the important question is whether the decision making process brings additional resources usage by itself due to additional computations related to the reasoning process.

The objective of the experiment, presented in this section, was to evaluate the adaptive approach for security level selection on a mobile device with limited resources.

3.1 Assumptions and Experimental Setup

The performance of the adaptive approach was evaluated using a controlled experiment. The goal of the experiment was to evaluate the performance of the adaptive approach with respect to efficiency of using mobile device's resources in context of applying different security mechanisms.

The following issues have been considered and some assumptions have been made before conducting the experiment:

- Accessibility of data

  Previously in Section 2.3.4 we have defined four security metrics that may be collected on the information system. However, not all metrics are trivial to gather. For example the *Resources* i.e. CPU level can be measured directly. The *Threat Level* could be determined from an IDS i.e. SNORT [28]. Therefore, an additional interface for data transmission is required. The *Location* can be obtained from GPS software, which in turn requires an interface and a method to transform the obtained coordinates to decision making approach suitable values.

- How often should the data be gathered?

  This question is important due to following issues. In case if the data for decision making is collected rarely, then if some security related event has occurred between two data collection points, we can miss the time when a security level has to be changed. As a consequence we can not adequate react to changing in security environment. On the other hand if we will collect the information very often we can add additional resources consumption to the system. However, the degree of influence of data collection frequency should be evaluated during the experiment.

  Another approach to data collection is event driven method. The method implies that the data will be collected as soon as a security related event has occurred. For example, if the location

of the system has been changed, the data collection and the decision making process should be initialized.

The following assumptions have been made to make the experiment less time and money consuming:

- We decided to use simulation to produce the input data for the decision making module instead of actual data collection
- We have chosen a fuzzy reasoning approach described in Section 2.4 for the decision making module. This choice was based on the following:

  a) The model can be easily adapted for using different numbers of security levels

  b) Different number of security metrics can be used without much changes to the decision making process
- For the experiment we have considered the following case, which is based on the real life usage scenario:

  There are several files with different content both confidential and non-confidential located on the mobile device. These files are automatically backed up by sending them via e-mail to the corporate mail server after some predefined interval. Before transferring the files, they should be encrypted to avoid the eavesdropping. The decision process is initialized prior to each data transmission in order to decide which data encryption cipher to use.

3.2 Hypotheses

The following hypotheses were formulated for our experiment:

Null hypothesis (H_0): The adaptive approach decrease the performance of the mobile device due to increasing resources consumption related to decision making process.

Alternative hypothesis (H_1): The usage adaptive approach does not decrease the performance of the device.

3.3 Instrumentation

We developed three modules for the experiment: data collection module, decision making module and security module (Figure 3.3.1).

The data collection module is responsible for gathering input data for the decision making process. In our case, the module uses the random pre-generated values of security metrics i.e. *Threat Level, Location, Content*, and *Resources.*

The decision making module provides a recommendation about the required security level based on the input information.

The security module performs files encryption using the recommended encryption cypher.

We used Visual C++ and Microsoft Visual Studio Professional as the software development environment.

To conduct the experiment we used HP iPAQ pocket PC that is run by Windows Mobile Pocket PC 2005 operating system. The pocket PC has the following characteristics:

- Intel PXA270 processor (520 Mhz)

– 256 MB total memory (192 MB ROM 64 MB RAM)

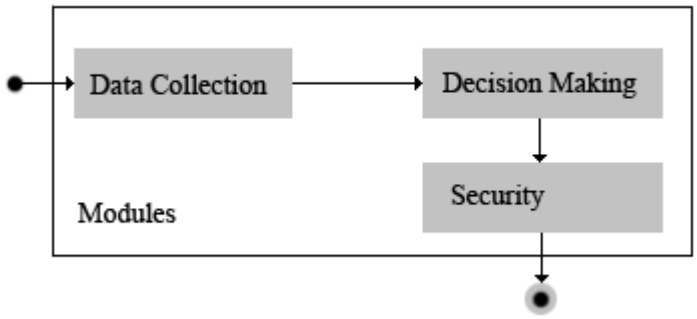

Figure 3.3.1: System Model

The following metrics were measured to evaluate the performance of the approaches under consideration:

- Remaining Battery life
- Memory load

The metrics are defined using the following equations:

$$\textit{Memory Load (\%)} = (\textit{Available Memory} * 100) / \textit{Total Memory} \quad (15)$$

$$\textit{Remaining Battery Life (\%)} = (\textit{Remaining Battery Life Time} * 100) / \textit{Full Battery Life Time} \quad (16)$$

We have chosen three block ciphers for data encryption (Table 3.3.1). The security level was assigned to each block ciphers based on its characteristics.

Block cipher	Key length (bits)	Block size (bits)	Rounds	Security level
RC2	128	64	18	Low
AES 128	128	128	10	Medium
AES 256	256	128	14	High

Table 3.3.1: Block ciphers

Several text files were created for the experiment with the average size 3.4 MB.

3.4 Threats to validity

The threads to the experiment's validity are presented below:

- *External validity*

  There is a threat that randomly created input data for the metrics is not corresponded to the real world situations. However, the values of metrics presented in used data set include variable values from minimum to maximum, which cover the most real world cases. Another threat is the size of the file being used for the experiment; usually the data rates are greater.

- *Conclusion validity*

  We tried to minimize this thread using robust statistical techniques and measures.

3.5 Experiment operation and Results

Two test configurations of the experiment's software were used in order to evaluate the performance of the adaptive approach for security level selection:

- In the first configuration we used only the highest security level for data encryption without the decision making and data collection. The highest security level corresponds in our case to AES 256 block cipher.
- The second configuration includes the data collection and the decision making module to select the block cipher dynamically based on the input information i.e. security metrics.

A sample set of security metrics has been created for the data collection module. In both configurations the encryption event occurred every 20 seconds. Only one file was encrypted at each time.

The results of the first set of experiments showed that the performance of the decision making module depends in a great extent on the data types used for calculations. The usage of double precision numbers in calculations decreases greatly the performance of the device due to heavy run-time computations. Figure 3.6.1 shows the dependence of battery life time from the experiment's time. This experiment ran for 7 hours. The red line corresponds to the first experiment with the highest security level (AES 256) used, whereas, the blue line shows the battery life for the second experiment with the adaptive approach for the security level selection.

As we can see, the performance of the decision tool suffers from the usage of the heavy run-time computations, and with this application design it is hard adequately to evaluate the performance of the adaptive approach.

For the second set of the experiments we changed the decision making module for using lookup tables. Then the experiment ran two times, first, using only AES-256 cipher for encryption and, second, using the adaptive approach for cipher selection. The experiment lasted for 6 hours.

Table 3.5.1 shows the average memory load and the corresponding standard deviation for two approaches.

Approach	Average Memory Load (%)	Sample standard deviation (%)
Approach with the highest security	31.31	0.8
Adaptive approach	30.5	0.541

Table 3.5.1: The average memory load and the corresponding standard deviation in percentage for two approaches.

In order to measure the influence of the approaches on the remaining battery life, we ran the experiments ten times for each approach. We measured the remaining battery life after 100 rounds of data encryption, with the 20 seconds interval (Table 3.5.2). Table 3.5.3 shows the corresponding average values and standard deviations.

Approach	1	2	3	4	5	6	7	8	9	10
Approach with the highest security	95	94	95	96	95	95	96	94	95	95
Adaptive approach	96	95	96	96	96	95	96	96	96	95

Table 3.5.2: Remaining battery life (%) after 100 rounds of encryption

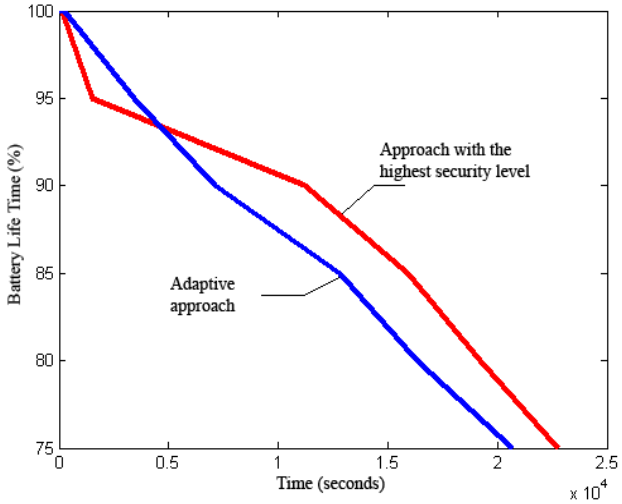

Figure 3.5.1: Remaining Battery Life Time versus the Experiment's Time (using double precision numbers)

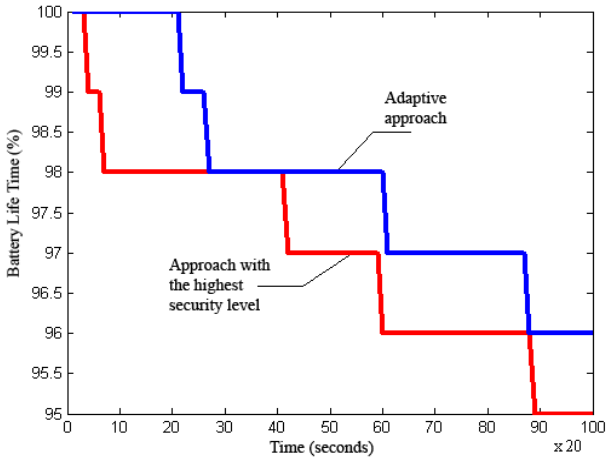

Figure 3.5.2: Remaining Battery Life Time versus the Experiment's Time (using lookup tables)

Approach	Average remaining battery life after 100 rounds of data encryption (%)	Sample standard deviation (%)
Approach with the highest security	95	0.663
Adaptive approach	95.7	0.484

Table 3.5.3: The average remaining battery life and the corresponding standard deviation after ten rounds of data encryption with 20 seconds time interval

According to results presented in Tables 3.5.1 and 3.5.3 the use of decision making for selecting a security level does not increase the resources consumption. Furthermore, the performance is slightly improved for memory usage and battery life.

To test our null hypothesis we performed a two sample t-test with the standard significance level $\alpha=0.05$ under the assumption that the sample data follows a normal distribution [34]. Based on the test's results we rejected the null hypothesis and conclude that the adaptive approach performs better than the approach with the highest security level with respect to resources utilization.

3.6 Summary

The presented experiment evaluates the performance of the adaptive approach for selection an adequate security level. For making a decision we used the fuzzy reasoning model presented in Section 2.4. We used a test software that includes three modules: data collection, decision making, and security.

The results of the experiment show that the use of decision making for security level selection does not increase resources consumption for devices with limited resources. Furthermore, the adaptive approach improves the usage of the resources such as memory and battery life.

For our experiment we have used data files with the average size of 3.4 Mb. In the real life scenarios the data rates usually can achieve hundreds megabytes. In this case we can expect significant improvements in resources usage by using the adaptive approach for security level selection.

4 Conclusions and Future Work

The usage of the strong security mechanisms decreases the performance of devices with limited resources. Furthermore, it could lead to Denial of Service attacks under the insufficient presence of resources.

In most cases, there are no risks and threats to the devices since they are not under attack. Thus, the use of strong security wastes the reserve of available resources. The trade-off between security and performance should be provided.

Under these circumstances, the process of making decision about the required or adequate security level, with respect to different criteria, plays an essential role to a user of devices.

The following section presents the results of the conducted research, which addresses the issues related to the decision making about an adequate security level.

4.1 Summary of the Master Thesis

Security quantification: To produce the input data for the decision making process about an adequate security level, a methodology for security metrics identification, selection and quantification has been developed. As an example we have shown how this approach can be applied to a mobile device i.e. Pocket PC. Following the methodology we selected four security metrics: *location, threat level, resources,* and *content*. Nevertheless, the proposed methodology is not limited to a particular system or number of metrics. The scheme can be used to select and quantify security metrics for any decision making models and different systems under consideration.

Decision making models: We have evaluated different decision making methods for their fitness to decision making in the security area. Three methods have been chosen for developing the decision making models to select an adequate network security level. The developed models are based on fuzzy reasoning, simple multi-attribute rating technique and artificial neural networks.

The proposed fuzzy reasoning model uses four security metrics (*location*, *threat level*, *resources* and *content*) identified for a mobile device and five security levels to choose between. This model can be adopted to take into consideration any number of security metrics and any number of security levels. The model is easy and ready to utilize even for users, which do not have a deep background in the security area.

The model, which is based on simple multi-attribute rating technique, uses both the criteria of alternatives and the security metrics collected from the information system under consideration. This design allows to consider the characteristics of different security levels along with security metrics. We have used two security level alternatives (strong and lightweight protocols) with the following three characteristics: security, efficiency, and robustness. The same set of four security metrics has been used including *location*, *threat level*, *resources,* and *content*. This model has a subjectivity factor, which is presented in ranking criteria in order of importance, and assigning scores and weights to criteria. This subjectivity factor can be hardly removed from the model. In spite of presented subjectivity, the results of the sensitivity analysis show that the proposed model reacts adequately to changes in security environment of a system.

The artificial neural networks have been successfully applied to classify the input values of the two security metrics into a number of classes. The main advantage of this model is that it does not require the human intervention into the reasoning process. The second advantage is that once

the neural network has been trained, no complex computations are required to make a decision about an adequate security level. Nevertheless, the complexity of the model increases with the growing number of security metrics and security levels to consider.

Empirical evaluation of an adaptive approach for security level selection: The motivation to conduct this experiment was based on the fact that decision making models require additional computations. These computations can lead to increased resources consumption that can make the use of adaptive approach impractical. Nevertheless, the results of the experiment show that with right software design and implementation the computations related to the adaptive approach do not decrease the performance of mobile devices. Furthermore, the use of an adequate security level improves the resources usage (e.g. memory load and battery life). For our experiment we used relatively small data rates (~3.4 MB). Thus, for the real life scenarios with data rates of hundreds megabytes, we can expect significant improvements in resources utilization by using the adequate security level.

4.2 Future Work

The research conducted in this master thesis could be further extended. The possible future work topics include:

- The model based on fuzzy reasoning model can be improved by adding the preference weights to the security metrics. It will allow to put the metrics in order of importance based on user's preferences and system requirements.
- An additional research is required to study how the SMART decision making model reacts to different orders of importance and assigned scores and weights. This research will help to minimize the effect of the subjectivity factor presented in the model.
- Additional research is required to adapt the neural network model to different number of security metrics and security levels.
- An experiment can be conducted to compare the performance of the proposed decision making models in real world environment instead of using simulation data.

References

[1] EDWARDS, W. Social Utilities, Engineering Economist, Summer Symposium Series 6, 1971

[2] EDWARDS, W. How to use multiattribute utility measurement for social decision-making, IEEE Transactions on Systems, Man, and Cybernetics, 7(5), 1977, pp.326-340

[3] JOHNSON H. Toward Adjustable Lightweight Authentication for Network Access Control, Blekinge Institute of Technology, School of Engineering, Doctoral Dissertation Series No. 2005:09, 2005

[4] SIATERLIS, C., MAGLARIS, V. Detecting incoming and outgoing DdoS attacks at the edge using a single set of network characteristics, Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005), 2005

[5] XIANG, Y., ZHOU, W., Mark-Aided Distributed Filtering by Using Neural Network for DDoS Defense, IEEE Globecom, 2005

[6] LOOTSMA, F.A., SCHUIJT, H. The Multiplicative AHP,SMART and ELECTRE in a Common Context, Journal of Multi-Criteria Decision Analysis, vol. 6, 185-196, 1997

[7] Yap C.S., Raman K.S., Leong C.M., Methods for Information System Project Selection: An Experimental Study of AHP and Smart, IEEE, 1992

[8] GOODWIN, P. Decision analysis for management judgment -3rd ed. John Wiley & Son, Ltd , 2004, ISBN 0-470-86108-8

[9] SWANSON, M., BARTOL, N., SABATO, J. HASH, J., AND GRAFFO, L. Security metrics guide for information technology systems, NIST Special Publication 800-55, 2003.

[10] STONEBURNER, G., GOGUEN A., AND FERINGA A. Risk management guide for information technology systems, NIST Special Publication 800-30, 2002

[11] Common methodology for information technology security evaluation, v.2.3., ISO/IEC 15408, 2005

[12] RYAN, P.Y.A., SCHNEIDER, S.A. The modelling and analysis of security protocols: the CSP approach. Addison-Wesley, 2001

[13] KEENEY R. L., RAIFFA, H. Decisions with multiple objectives: preferences and value tradeoffs, Wiley, New York, 1976

[14] NEN-FU HUANG, CHIA-NAN KAO, HSIEN-WEI HUN, GIN-YUAN JAI, CHIA-LIN LI. Apply data Mining to defense in-depth network security, Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05), 2005

[15] UPADHYAYA, S., CHINCHANI, R., KWIAT, K.. An analytical framework for reasoning about intrusions, Reliable Distributed Systems. Proceedings. 20th IEEE Symposium , 2001 pp.99-108

[16] SAATY T.L. The analytic hierarchy process. McGraw-Hill, New York, 1980

[17] BENAYOUN, R., ROY, B., SUSSMAN, B., ELECTRE: Une méthode pour guider le choix en présence de points de vue multiples. Note de travail 49, SEMA-METRA International, Direction Scientifique, 1966.

[18] BRANS, J.P. L'ingénièrie de la décision; Elaboration d'instruments d'aide à la décision. La méthode PROMETHEE. In R. Nadeau and M. Landry, editors, *L'aide à la décision: Nature, Instruments et Perspectives d'Avenir,* pages 183–213, Québec, Canada, 1982. Presses de l'Université Laval.

[19] BHARAT B.MADAN et al., A method for modeling and quantifying the security attributes of intrusion tolerant systems, Performance Evaluation 56 (2004) 167–186

[20] ORTALO R. Experimenting with Quantitative Evaluation Tools for Monitoring Operational Security, IEEE

Transactions on Software Engineering, Vol.25, No.5, 1999

[21] National Institute of Standards and Technology (NIST), http://www.nist.gov/

[22] COX, E. The Fuzzy Systems Handbook: a practitioner's guide to building, using, and maintaining fuzzy systems. Academic Press, Inc., 1994

[23] Edited By FIGUEIRA, J., GRECO, S., EHRGOTT, M. Multiple Criteria Decision Analysis. State of the Art Survey. Springer Science + Business Media, Inc., Boston, 2005

[24] ROJAS, R. Neural Networks: a systematic introduction. Springer-Verlag, Berlin Heidelberg New York, 1996

[25] SILER, W., BUCKLEY, J.J., Fuzzy expert systems and fuzzy reasoning, John Wiley & Sons, Inc., 2005

[26] ANDERSON JR, Rules of the Mind. Lawrence Erlbaum, Mahwah, New Jersey, 1993

[27] R.L. KEENEY and H. RAIFFA. *Decision with Multiple Objectives: Preference and Value Tradeoffs.* Cambridge University Press, New York, 1993.

[28] SNORT, the open source detection and prevention system. URL: http://www.snort.org

[29] HAYKIN, S., Neural Networks: a comprehensive foundation - 2nd ed., Prentice-Hall, 1999

[30] CREIGHTON T.R., Context aware and adaptive security for wireless networks, Blacksburg, Virginia, November 2004

[31] The United States Government executive order 12392 "Classified National Security Information", http://www.whitehouse.gov/news/releases/2003/03/20030325-11.html

[32] WILLIAM, S., Network Security Essentials - 2nd ed., Prentice-Hall, 2003

[33] ANDERSON, R. J., Security Engineering: A Guide to Building Dependable Distributed Systems, John Wiley & Sons, Inc., 2001

[34] MASON, R.L., GUNST, R.F., HESS, J.L. Statistical Design and Analysis of Experiments. With Applications to Engineering and Science, 2nd ed., John Wiley & Sons, 2003