MASTER THESIS

Selective Encryption Methods for Securing Multi-Resolution Smart Meter Data

submitted in partial fulfillment of the requirements of the degree

Master of Science in Engineering

at the department of Information Technology and Systems Management

at Salzburg University of Applied Sciences

submitted by:

Adnan Srna, BSc

Head of Department:	FH-Prof. DI Dr. Gerhard Jöchtl
Supervisor:	FH-Prof. DI Mag. Dr. Dominik Engel

Bowling Green, Ohio, July 17th, 2013

Declaration of Academic Honesty

By my signature I, Adnan SRNA, born on June 18^{th}, 1987 in Sarajevo, pledge that the present thesis is entirely the result of my own work. Any adopted insights are cited and the sources are always indicated with their origin.

Salzburg, Austria, July 17^{th}, 2013

Matriculation number

Adnan SRNA

Details

Surname and Name, Title:	Adnan SRNA, BSc
Institution:	Salzburg University of Applied Sciences
Department:	Master Informationstechnik und System-Management
Title of the Thesis:	Selective Encryption Methods for Securing Multi-Resolution Smart Meter Data
Keywords:	Smart Grid, Smart Meter, Security, Privacy, Selective Encryption, Wavelets
Advisor:	FH-Prof. DI Mag. Dr. Dominik Engel

Abstract

New directives in electric consumption metering enable utility companies to continuously monitor the condition of the electric grid. The benefits of the permanent feedback are lower energy production surplus, faster recoveries from blackouts and tailor-made energy rates. The drawback of this system is that every household is equipped with a digital metering device, a smart meter, which captures the electric consumption on a 15-minute basis. This individual smart meter data reveals user behavior non-intrusively and threatens users' privacy. As these smart meters are embedded systems with low power hardware components, security needs to be resource conserving. This thesis proposes a low resource consuming approach to prevent eavesdropping on smart meter data and defines an acceptable way to access these data without violating privacy. With the use of wavelet multi-resolution analysis a trade-off between security and performance is evaluated. It is shown that by means of wavelets and selective encryption the computational effort can be reduced by approximately 30%. For smart meters this implies that their hardware components and their prices can be reduced.

Those who desire to give up freedom in order to gain security will not have, nor do they deserve, either one.

Benjamin Franklin

Acknowledgments

There are several individuals who contributed to this master’s thesis to whom I am grateful. First of all I wish to express my gratitude to the Marshall Plan Foundation for their sponsorship and faith in my research. Without the funding of the Marshall Plan Foundation, I wouldn’t have been able to conduct my research project and the master‘s thesis in Ohio, America.

This work described in this paper was conducted at the “Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control”. The financial support by the Austrian Federal Ministry of Economy, Family and Youth and the Austrian National Foundation for Research, Technology and Development is gratefully acknowledged.

I also greatly appreciated the contribution of Dr. Dominik Engel while serving as advisor and judge. Much of the credit for the selection of this topic was instrumental by Dr. Engel. He made me aware of the needfulness for this area of study. It is fair to say that without his encouragement and support I would have simply chosen a different route, especially when the going became tough. He kept me going.

My thanks also go to Dr. Larry Hatch and his family. They were very helpful in the matter of making my stay in Ohio a wonderful experience for my life. And last but not least, I want to thank my girlfriend Birgit, for her unfailing support and encouragement during the months I was in the foreign exchange scholarship in America. I will ever remain grateful for being so patient with me throughout the period. Sometimes all it takes is someone believing in you. I thank you.

Contents

1 Introduction

When we talk about “Smart Grids” from a customer’s perspective, someone might tend to think about devices from a household that exchange data to fulfill the demands of a customer or to reduce energy consumption. For instance, a washing machine determines the cheapest daily energy rate from the Internet to do the laundry.

In fact, Smart Grids bear much more than simply gadgets that communicate with each others to reduce costs. From the perspective of the energy producer, the evolution of the worldwide energy networks also pose new challenges and security threats for producers as well as for customers. With energy generation systems like photovoltaics on housetops, the producer/consumer paradigm in the field of electricity shifts. Some of this so produced electricity may exceed the required amount of energy for warming a house. Therefore energy could be recycled in the power grid for other consumers. Energy producers need to comply with these changes and assert the consumption of the consumers more frequently than with traditional metering approaches (see Figure 1.1). Traditional consumption measuring was done by reading the electricity usage from analogue consumption meters or even through mathematical approximation once per year. This implies that firstly it was hardly possible for the consumer to economize and secondly the electricity producers had to allocate more electricity than the customers consumed. These issues could be addressed by using the so-called smart meters, digital metering devices that constantly communicate meter readings to the utility company. In October 2007, the federal government of the United States of America delegated the responsibility for coordinating, funding and encouraging the rollout of Smart Grids to the Department of Energy (DOE, 2007).

Almost two years later, in July 2009, the European Commission announced a directive, which requires that 80% of European households should be covered with smart meters by 2020 (European Comission, 2009). Thereupon the Austrian Legislator (BMWFJ, 2012) formulated that 95% of the consumption meters in Austrian households need to be replaced by smart meters by the end of 2019. By means of these smart meters, a much more detailed load profile from households can be achieved to provide a more precise energy feedback. In contrast to the benefits of this system, we cannot hide the fact that security is a must when it comes to customer data, especially if the data could reveal user behavior. There are numerous scientific reports exposing how much behaviors insight on behavior smart meters provide. E.g., the University of Applied Science in Münster disclosed in a technical report (Greveler et al., 2011) that by means of smart meter readings with a period of 2 seconds, it could be discovered that a user watched a movie on a conventional LCD-TV. Furthermore it was proved that meter readings in some cases were transferred to the data aggregator without any encryption at all. The keyword for preventing this hazard is granularity. In terms of smart meters,

the granularity means the sampling rate, which is used for the meter readings. The Austrian Legislator applied the law that smart meter sampling must be set to a 15 minute interval. Even with a sampling rate this low, insights on user behavior are provided. Therefore, there is a call to secure smart meter data.
The main arguments in favor of insufficient security are the costs for encryption. The encryption of the whole smart meter data requires vast computational efforts. With the solution proposed by this thesis, the costs for encryption can be reduced due to selective encryption. Furthermore, a customer should be able to decide which granularity is used for several parties involved by means of a multi-resolution representation of load profiles. The encryption of particular portions of the smart meter data will lead to decrease of the computational effort and the increase of protection of users' privacy. As a result, the benefits of Smart Grids will still be present and so will the privacy.

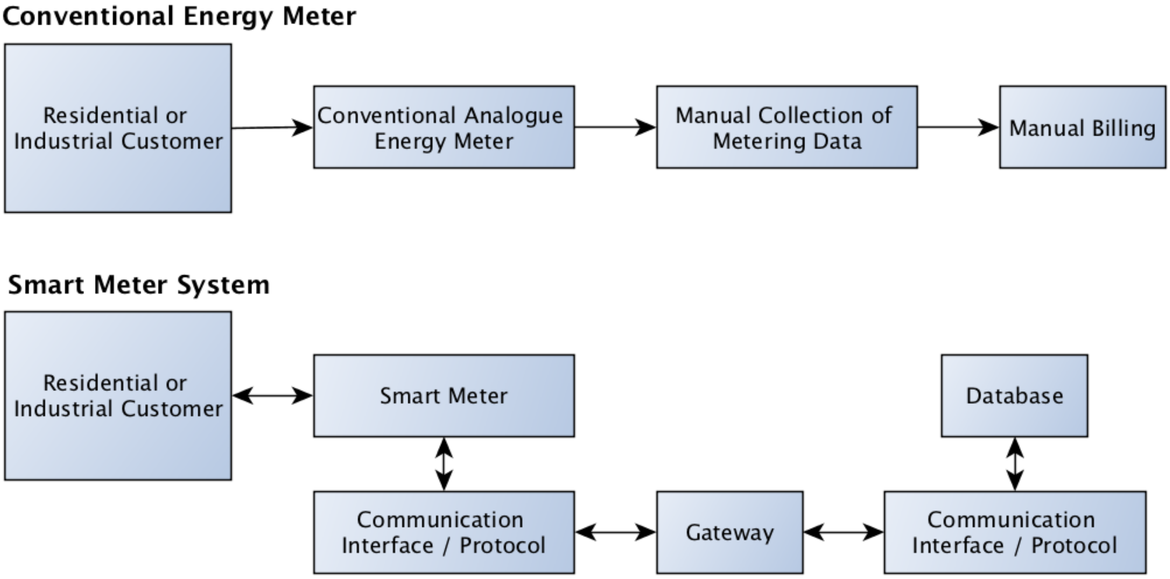

Conventional Energy Meter
Residential or Industrial Customer
Conventional Analogue Energy Meter
Manual Collection of Metering Data
Manual Billing
Smart Meter System
Residential or Industrial Customer
Smart Meter
Database
Communication Interface / Protocol
Gateway
Communication Interface / Protocol

Figure 1.1: Procedure of conventional energy metering and smart metering (Depuru et al., 2011)

1.1 Problem Formulation

The problem of this thesis was threefold:

1. To develop selective encryption methods for multi-resolution representation of load profiles from smart meter readings,
2. To find a trade-off between performance and security by reducing computational effort with selective encryption,
3. And to still provide a privacy-aware and secure access to load profiles from smart meters.

The major problem lies with the performance bottleneck and the hardware costs of embedded systems like smart meters. The more smart meters have to compute, the more powerful their hardware components like as processor and memory need to be. Thinking of one single device the costs would be affordable, but in the quantity of the about 114.8 million households in the United States of America (DOC, 2013) and the 3.6 million households in Austria (Statistics Austria, 2012), this would lead to millionfold increases for the hardware. Embedded systems like smart meters have their constraints in hardware performance: the encryption of a large amount of data, for instance the load profiles from meter readings, would require a vast computation capacity. Therefore the complexity of data aggregation, due to a large amount of meter readings, should not be negatively affected by the computational effort of encryption. This is the point where selective encryption approaches come into play: they reduce the hardware requirements for ciphering and still offer security to a certain degree. How well selective encryption fits, for securing the multi-resolution representation of smart meter readings, will be investigated in this thesis.

1.2 Applications

There are numerous applications that could benefit from the findings of this research. When it comes to smart meters, possible parties and scenarios that could profit are:

- Question of Costs
  - Firstly, the hardware costs of smart meters could be reduced whereas security still exists. This assumption comes from the fact that less computation effort is required by selective encryption methods. As a consequence, cheaper hardware for the smart meters could be used.
- Customer Acceptance
  - Utility companies could foster a sense of trust by guaranteeing that customer data is undisclosed.
  - Customers are provided with the ability to define on their own how their consumption profile is used by each participant.
  - Speaking of customer data, data privacy is a must. Privacy plays an essential role and users will be grateful when they know their privacy is not being violated. Failure to address consumer's privacy concerns will seriously threaten consumer acceptance.
  - Prohibiting unauthorized parties to spy on smart meter readings and thereby preventing the analysis of a consumer's behavior.

- The proposed solution could be adopted to encourage energy savings and introduce tailor-made energy rates. This is achieved by secured meter readings with a sampling period that provides much more details than with 15 minutes sampling.

1.3 Structure of the Thesis

The structure of this thesis is mainly based on four main parts. The first part provides an overall understanding of the problem covered in this thesis (see chapter 1 and 2). Thus, an introduction, a problem formulation, possible applications, objectives, research questions, methods for the solution, expected results, assumptions and target groups are formulated.

The second part (chapter 3, 4 and 5) gives theoretical background knowledge about the specific smart meter domain, selective encryption methods and the wavelet multi-resolution analysis.

In the third part of the thesis the methodology (chapter 6) and the implementation of different selective encryption methods with the corresponding designs are described (chapter 7).

Within the last part, the results and findings (chapter 8) are described. The final chapter 9, the discussion, concludes the research and gives perspectives.

2 Objectives for the Thesis

The aim of this thesis is to evaluate and implement selective encryption techniques with low computational effort, and as a result secure the privacy-aware data representation of smart meter data. The main objective is to discover a sensible trade-off in security where on the one hand the computational effort is manageable by embedded systems, and on the other hand the protection of the customer privacy in Smart Grid is given. It is necessary to determine if there are any applicable approaches from multimedia security that could be used for securing smart meter readings as well. E.g., in multimedia security, different resolutions of media contents are provided by means of DRM (Digital Rights Management). This is done by the partial encryption of several quality layers. As selective encryption secures not the entire data, but relevant parts, new types of attacks that are made possible due to this approach need to be investigated. The results of this research should demonstrate efficient solutions for securing multi-resolution smart meter data with low computational costs and ensure that authorized parties can access user data in an acceptable way.

The main goals can be declared as follows:

- Defining an appropriate selective encryption method for securing multi-resolution load profiles.
- Developing selective encryption methods for multi-resolution smart meter data.
- Evaluating the implemented selective encryption methods with respect to computation time, hardware requirements and security.
- Finding an appropriate trade-off between security and performance.
- Providing a general methodology for smart meter designers to enable efficient and privacy-preserving metering.

2.1 Research Questions

The crucial question which needs to be answered, can be defined as follows:
Which selective encryption methods can be used for securing multi-resolution smart meter data with focus on performance and security?

The following side questions need to be clarified in addition:

- Which encryption methods are appropriate for selective encryption?
- Which performance improvements can be achieved through selective encryption methods in contrast to encrypting the whole data?

- How is the PSNR[1] affected by variation of the amount of encrypted data?
- How is the trade-off between security and performance determined?
- Which differences in respect to performance between the Raspberry Pi[2] and a PC can be identified due to partial encryption?

2.2 Methods for Solution

First of all, it is necessary to clarify the scope of smart meters, and in detail, the load profiles, which are gathered from meter readings. It is required to determine which personal information and behaviors of the users may be retrieved due to load profiles. Possible conclusions regarding the user behavior will be investigated and presented by means of consumption data from smart meter readings.

Secondly, the focus is drawn to cryptographic methods in general and the use of selective encryption, specifically. When it comes to partial encryption, the first approach will be the investigation of symmetric and asymmetric key algorithms for securing samples from load profiles. The best suitable encryption algorithm for securing multi-resolution representation of smart meter readings will be identified. This is done by a closer examination of various encryption methods and their algorithms for encrypting data. The performance requirements, computation times, and reliability of several encryption algorithms are investigated in respect to the amount of encryption. It needs to be examined whether or not the data has to be encrypted as a whole because of the fact that an attacker could easily exchange encrypted parts of the signal with interpolated data.

An alternative approach, used with selective encryption, is the use of one dimensional Haar-Wavelets as data representation format for smart meter data. The subbands of the wavelets will be encrypted with the most appropriate encryption algorithm (determined by the comparison of several encryption algorithms). The encryption amount within the subbands is increased step-wise and the PSNR is analyzed. If the PSNR gets lower, the encryption quality gets higher and therefore security increases. To stay unrecoverable for an attacker the amount of encrypted data is determined. The two approaches are compared by evaluating their performance and secrecy. It is of importance to define the relevant scope for the evaluation and to measure the performance results on both a PC and an embedded system. For the embedded system a Raspberry Pi is used as its hardware components like RAM and CPU architecture (ARM) are similar to the smart meter hardware components. The Raspberry Pi is chosen because it represents a close approximation to the limited hardware of a smart meter.

[1] Peak-Signal-to-Noise Ratio; in this context used to determine the similarity between encrypted and plain data.

[2] `http://www.raspberrypi.org/`, Accessed: 2.15.2013

The implementation of partial encryption of wavelets and the evaluation process will be carried out in the programming language Java. The results from a performance comparison between PC and Raspberry Pi will show the impact of selective encryption on computation time. Furthermore, the identified PSNR will serve as a quality degree and security trade-off to describe the similarity between plain data and encrypted data. The findings from the relation between the amount of encryption and the PSNR, as well as the hardware requirements, are analyzed.
Summing up, the research questions will be answered in consequence of the following procedures:

- Investigation of load profiles.
- Implementation of selective encryption methods.
- Evaluation of selective encryption methods.
- Identification of the trade-off between security and performance due to metrics.
- Validation of the results on the basis of common attacks against the selective encryption methods.

2.3 Expected Results

Through the proposed solutions, the following results are expected:

- Identified threats to privacy caused by undisclosed load profiles of smart meter readings.
- Implementation of methods for securing multi-resolution smart meter data.
- Documentation of results gathered from the implemented software solution for selective encryption of smart meter data.
- Detailed evaluation of methods for selective encryption by analyzing and comparing computation time, hardware requirements and reliability.
- Trade-off between security and performance due to partial encryption of multi-resolution smart meter data.

2.4 Assumptions and Limitations

The development process and the subsequent findings are subject to certain assumptions and limitations inherent in the research. This thesis was based upon the following assumptions and limitations:

- This research focuses solely on electric smart meters. The results also may or may not apply for smart meters from other domains.

- The implemented cryptographic algorithms, while having unique potential, have some limitations in security. The validity of the reliability of the used ciphers in this work depends on their implementation. Fault analyses of the evaluated encryption algorithms are not carried out.
- The selection of the deployed encryption method is based on observational research, on the results from state of the art implementations, and the corresponding science community.
- The traceability of Java programs by means of reverse engineering is not addressed. As Java may not use scarce resources effectively, it is well suited for the proof of concept: if performance increases are achieved due to selective encryption methods with Java, even more performance improvements can be achieved by using assembler language.
- Following the European Interoperability Framework from IDABC (2005), the storage format of meter readings from smart meters and the used protocols within this research meet the requirements for open technical interoperability, this is to say for European standards. A global interoperability of the applied data format is excluded. Other cases are feasible as the norms are generally applicable by European Member States or the respective countries.
- Some items, like the load profiles from smart meter readings, used in the research were not publicly available. The test items are, however, available to researchers under nondisclosure agreements from the cited origins.

2.5 Target Group

This research mainly aims to address professionals in the electrical industry like electrical producers and experts in the area of Smart Grid who are searching for efficient and cost-effective approaches to provide privacy within the Smart Grid domain. The provided solutions raise awareness that there are low-cost solutions for securing smart meter readings. Manufacturers of smart meter devices and energy producers may be interested as their smart meters could be created at lower costs.

3 Scientific and Theoretical Background

The following section of the research specifies the notations used within the thesis. The introduced technologies within this chapter are all necessary, based on the demanding nature of this research.

3.1 Glossary of Terms and Acronyms

The following terms and acronyms are used throughout this thesis:

AES The Advanced Encryption Standard is a symmetric key algorithm established by the U.S. National Institute of Standards and Technology.

AMI An Advanced Metering Infrastructure is described by systems that collect, communicate, and analyze meter readings from smart meters.

AMR Automated Meter Readings define a technology to automatically collect meter readings.

Brute-Force is an attack where the systematic enumeration of all possibilities is performed until the solution is found.

Ciphers are algorithms used to encrypt and decrypt data.

Ciphertext is the encrypted form of a plaintext.

CP The Critical Point serves as a borderline, as it defines the minimum required amount for encrypted data using selective encryption methods.

Cryptosystems are collections of algorithms to perform the encryption and decryption of data.

CSV The Character-Seperated Value file format uses a textual structure and a delimiter character to separate each data set.

DRM Digital Rights Management defines an access control technology to limit the use of digital content.

DWT The Discrete Wavelet Transform is a form of signal representation that provides a time-frequency representation of a signal.

EU The European Union describes a political union of the 27 European member states.

GF Galois Fields define a finite number of elements used in abstract algebra.

GSM The Global System for Mobile Communications is the de facto standard for mobile communication technology.

IV In cryptography the Initialization Vector serves as a pseudorandom during initialization of the cryptosystem.

LCD A Liquid-Crystal Display is a visual display used in a wide field of applications.

MRA The Multi-Resolution Analysis is a commonly used DWT.

NILM Non-Intrusive appliance Load Monitor is the analysis of energy consumption within a household to deduce what appliances are used.

Payload is the content of a data transmission minus all attached headers and metadata.

Plaintext is a readable message, usually consisting of alphabetic and numeric characters.

Privacy is the condition of being secluded from the view of others.

PSNR Peak Signal-to-Noise Ratio describes the quality of a signal by comparing the original signal with the altered signal.

Key is a password used for encryption and decryption.

Receiver is a party that obtains a message.

RSA The Rivest-Shamir-Adleman algorithm defines an asymmetric key encryption algorithm.

Sender is a party that transmits a message.

Smart Grid is referred as an electrical grid equipped with information and communication technology to improve the efficiency of electricity distribution.

Smart Meter describes a digital metering device that collects and transmits consumption data, like electricity, gas, or water to the utility company.

Wavelets are wave-like oscillations used for signal processing.

3.2 Smart Meter

“Smart Meters are electronic measurement devices used by utilities to communicate information for billing customers and operating their electric systems.” (Edison Electric Institute, 2011, p. 7)

Technically understood, a smart meter (as illustrated in Figure 3.1) describes a physical technology to replace a traditional electric, gas, or water meter. The term “smart” is often coined as these computerized devices are comparable to embedded systems, mostly equipped with a micro controller (μC), to provide remote data collection on energy usage. With conventional meters, the determination of energy consumption per household was read and recorded manually once per year. Due to reading the real-time energy usage and periodically communicating (e.g. 15-minute, hourly, daily) this information to the utility company, smart meters provide constant feedback about the power demand. Information about voltage, phase angle, and frequency helps energy producers to monitor and improve the electric grid remotely (Depuru et al., 2011) and (Doris and Peterson, 2011). The benefits of smart meter technology are displayed in Table 3.1.

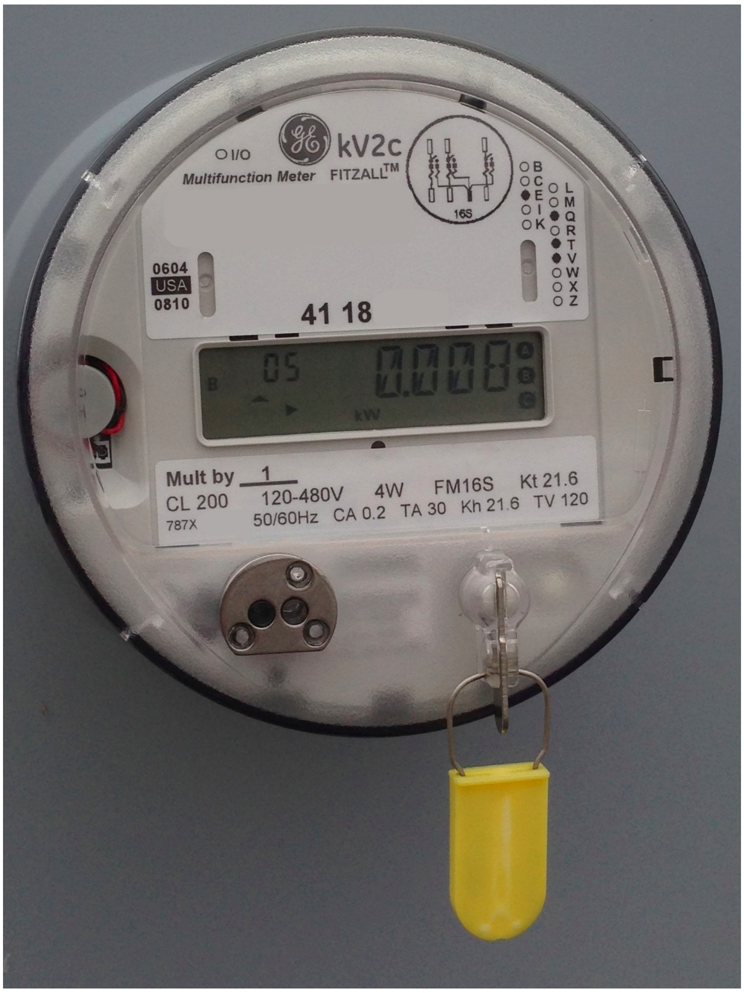


Figure 3.1: American smart meter kV2c model from General Electric

In Molina-Markham et al. (2012), the generic anatomy of a smart meter is specified by the following components:

- A load sensor for reading the power consumption,
- An analog front end for preprocessing and converting measurements from the load sensor,
- A micro controller (μC) unit for collecting, storing, securing, and transmitting metering data as well as driving an LCD screen,
- A flash memory for the storage of the load profiles,
- An LCD screen for manual meter-reading and diagnostics,
- And last but not least, a communication module that may be integrated or separately driven.

Stakeholder	**Benefits**
Utility Company	• Constant access and data to manage energy usage and reduce peak demands • Automated data collection systems reduce costs for meter readings, disconnection and maintenance • Improved outage restoration by means of remote problem solving on smart meters • More accurate and timely billing • Early detection of meter tampering and theft
Customer	• Faster power restoration without scheduled maintenance due to remote access • Increased rate flexibility which fits consumption patterns • Financial savings by knowing patterns of energy use
External Stakeholders	• Improved environmental benefits due to fewer peak loads and avoiding the use of less efficient, more polluting power plants

Table 3.1: Benefits of smart meters regarding stake holders (Depuru et al., 2011) and (Doris and Peterson, 2011) and (Edison Electric Institute, 2011)

3.3 The Creation of Load Profiles

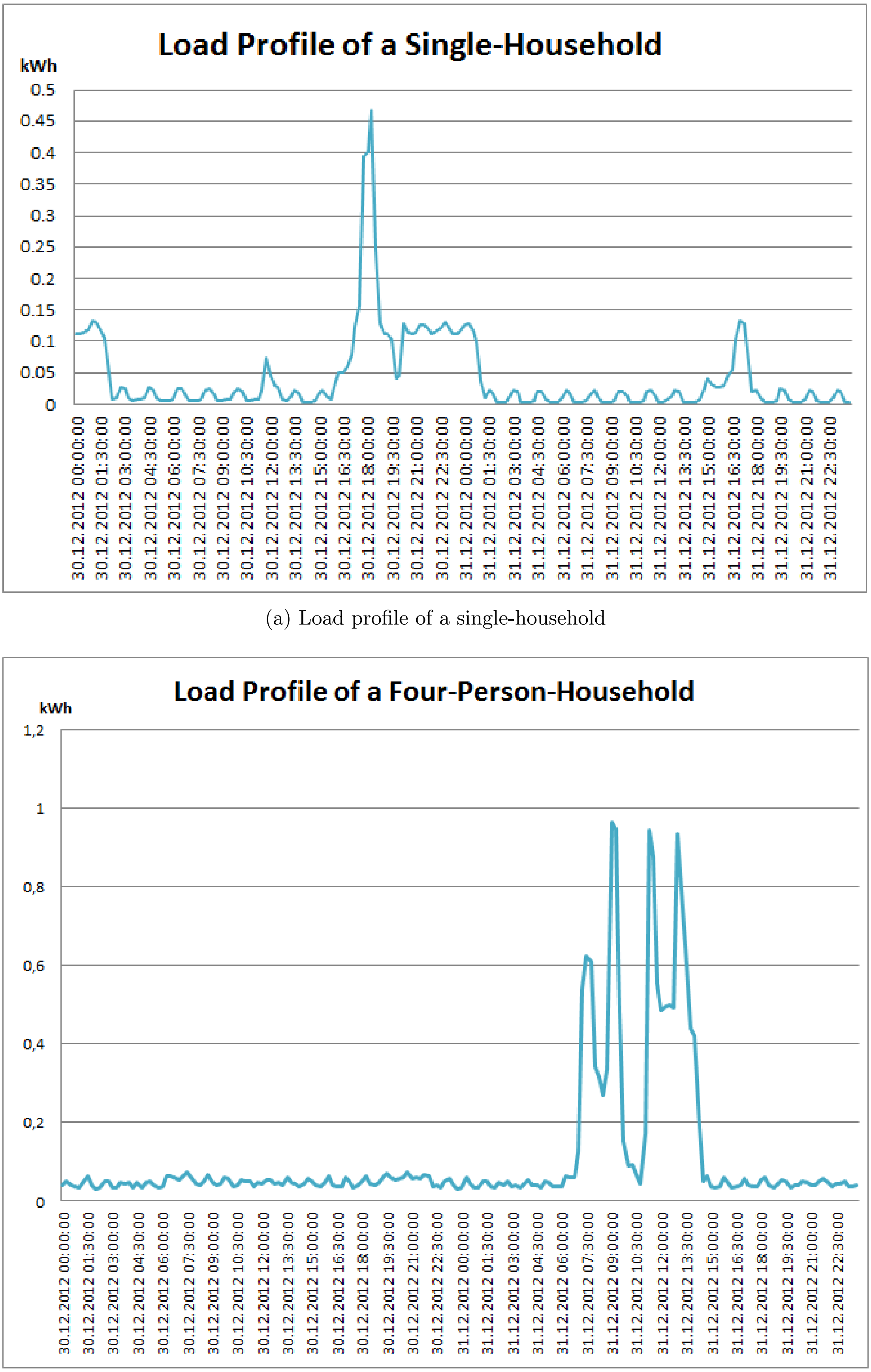
Load profiles are created by meter readings from smart meters. They play an essential role in the Smart Grid domain as they provide an in-depth feedback about the electricity demand and the condition of the grid. By means of these profiles, not only the consumption from a household is recorded, but user behavior may be non-intrusively deduced as well. Depending on the granularity of the metering data, detailed usage statistics can be generated and households can be monitored unknown to the involved parties (Lu et al., 2012). E.g., non-intrusive load monitoring (NILM) as described in Lisovich et al. (2010), can be performed to gain deep insights on user behavior without the awareness of the users. Without any doubt, the technical capabilities made possible by load profiles raise privacy concerns.

A load profile is simply described as a graph where the energy consumption over time is displayed. The used resolution varies by country between 1 and 60 minutes, whereas recent smart meters are capable of collecting 30-60 data points per second (Lu et al., 2012). The storage of the meter readings may vary regarding particular specifications and smart meters. However, the load profiles need to be stored in a comprehensive way to enable interoperability. The used format for the readings within this research and the used smart meter is a character-seperated value (CSV) file format. Listing 3.1 provides an excerpt of meter readings from a single-person household between 6 and 8 a.m. on a 15-minute basis. This data is derived from the EasyMeter Q3D[3] model, a smart meter from the German manufacturer EasyMeter GmbH. Each line of the meter readings within the CSV file represents the exact value of one consumption measure with its date and time information. With this information, a load profile can be created.

Figure 3.2 illustrates two load profiles generated from metering data on a 15-minute basis right before New Year's Day. The first profile (Figure 3.2a) displays a single-household whereas the second profile (Figure 3.2b) represents a four-person household. Both charts provide the ability to identify particular activity patterns from the power consumption. E.g., both households probably were not celebrating New Year's Eve at their home.

It might be obvious that load profiles could provide insights on someone's daily activities. Encryption of metering data can be performed to prevent the hazard of eavesdropping.

[3] `http://www.easymeter.com/fileadmin/bilder/downloads/100125_Q3D_Produktblatt.pdf`, Accessed: 3.19.2013



(a) Load profile of a single-household

(b) Load profile of a four-person household

Figure 3.2: Load Profiles from two different households right before New Year's Day.

Listing 3.1: Meter readings from a single-household between 6 and 8 a.m. with 15 minute intervals

```
...
25.09.2012 06:00:00;2110.949929
25.09.2012 06:15:00;2110.952393
25.09.2012 06:30:00;2110.954871
25.09.2012 06:45:00;2110.957339
25.09.2012 07:00:00;2110.959789
25.09.2012 07:15:00;2110.962233
25.09.2012 07:30:00;2110.964684
25.09.2012 07:45:00;2110.967141
25.09.2012 08:00:00;2110.969594
25.09.2012 08:15:00;2110.972055
...
```

3.4 Selective Encryption of Metering Data

The proposed selective encryption methods within this work serve the protection of load profiles with low-resource computation devices. A conventional smart meter, like the EasyMeter Q3D, stores its metering data in the format outlined by Listing 3.1. This data representation allows the creation of load profiles with a granularity of 15 minutes. A user may want to keep the utility company from gaining insight on activities within its household but still wants to be billed accurately or receive a detailed feedback on its consumption. A multi-resolution representation of the metering data addresses this issue by only providing access to specific granulations of the meter readings. This means that the granularity of the measurement data regulates the level of details within the load profiles and therefore prevents unauthorized parties to violate user privacy. If there is a need for high-resolution metering data (like for power saving applications or a more detailed feedback for users), this access could be granted optionally.

However, the idea behind selective encryption methods is not to provide conditional access, but to make it substantially difficult for an attacker to retrieve a user's smart meter data. This is performed by encrypting a sufficient amount of the data to keep it safe.

There are two approaches for selective encryption of metering data that are analyzed in this research. The first discussed technique uses the existing CSV file format by encrypting a specific amount of the smart meter data while the remaining parts are left in plaintext. E.g., the 15-minute timing load profile from Listing 3.1 will be transformed to a load profile with a specific amount of encrypted readings. This means that only the date of the first hour (see Listing 3.2), fourth hour and so on is left unencrypted.

Listing 3.2: Selective encryption of smart meter data

```
...
25.09.2012 06:00:00;2110.949929
?HX?fda2d09e30a89064e027b25c41fe4f4ea6151af
?HX?9fe08bf4703ac152fa532939c9673a14a6151af
?HX?df4c6afaaaeaa277830cfc8000fe304ea6151af
25.09.2012 07:00:00;2110.959789
?HX?d989de767a4c7937396c52e602f3e37aa6151af
?HX?51dcaa2761c1a15af7a892ca2ad5f3d8a6151af
?HX?5249548bd3112a9f82ff3bf0665aa66ca6151af
25.09.2012 08:00:00;2110.969594
?HX?6c4e6cbb47291998d22a3f30e042e033a6151af
...
```

The second method, suggested by Engel (2011) and Engel (2013), operates with the multi-resolution analysis (MRA) of wavelets. By means of discrete wavelet transformation (DWT) (as described in section 5.2.1), a hierarchy of resolutions can be generated. Therefore, a wavelet transformation is applied to the original metering data to create a low frequency and a high frequency band. Within the low frequency subband, the wavelet analysis step is carried out recursively up to a maximum level to provide a multi-resolution representation of the load profile. Each resolution of the load profile is encrypted separately with a different key to enable end-user control to access different granularities of the smart meter data (Engel, 2013).

In this thesis the MRA described by Engel (2011) and Engel (2013), is adapted to develop selective encryption methods for multi-resolution smart meter data and to find a trade-off between performance and security. The trade-off is defined by recursively increasing the number of encrypted subbands during MRA.

4 Encryption Methods

This chapter describes the scientific and theoretical background of cryptography and the encryption of data from an information theoretical point. Traditional methods for data encryption as well as modern cryptographic approaches are outlined with their commonly used protocols. An evaluation of recent encryption methods is provided to justify the chosen encryption algorithm for securing the multi-resolution representation of the smart meter data.

4.1 Introduction to Cryptography

Strictly speaking, encryption begins with the origin of language writing back in 2000 B.C.E. The Egyptians used hieroglyphs for communicating among a selected category of people, usually upper nobility. In ancient Greece, secret writing was established by the so-called *scythe*[4] that became famous for its military purpose by the Spartans. From ancient Rome, the famous Caesar cipher arose. The first attempts for cryptography are certainly rooted in historical tradition. Without these early attempts for secure communication, it is disputable if there would be individual or corporate privacy nowadays (Vaudenay, 2005) and (Paar and Pelzl, 2010).
According to Paar and Pelzl (2010), modern cryptography may be separated (see also Figure 4.1) in the following areas:

Cryptology combines the studies of the two branches cryptanalysis and cryptography.

Cryptanalysis describes the discipline of analyzing and breaking cryptosystems for either improving the cryptosystem or finding vulnerabilities to gain access to undisclosed information.

Cryptography derivates from the two Latin words *crypta* and *graph* (in plain English, secret and writing) and is therefore referred as the science of secret writing. It is serving the cause of hiding confidential information (also called plaintext) within a message. Cryptography can be implemented by three different approaches **symmetric Ciphers**, **Asymmetric- (or Public-Key) Ciphers** and **Protocols**.

[4] A cylinder with a vellum wrapped around that contains the secret message

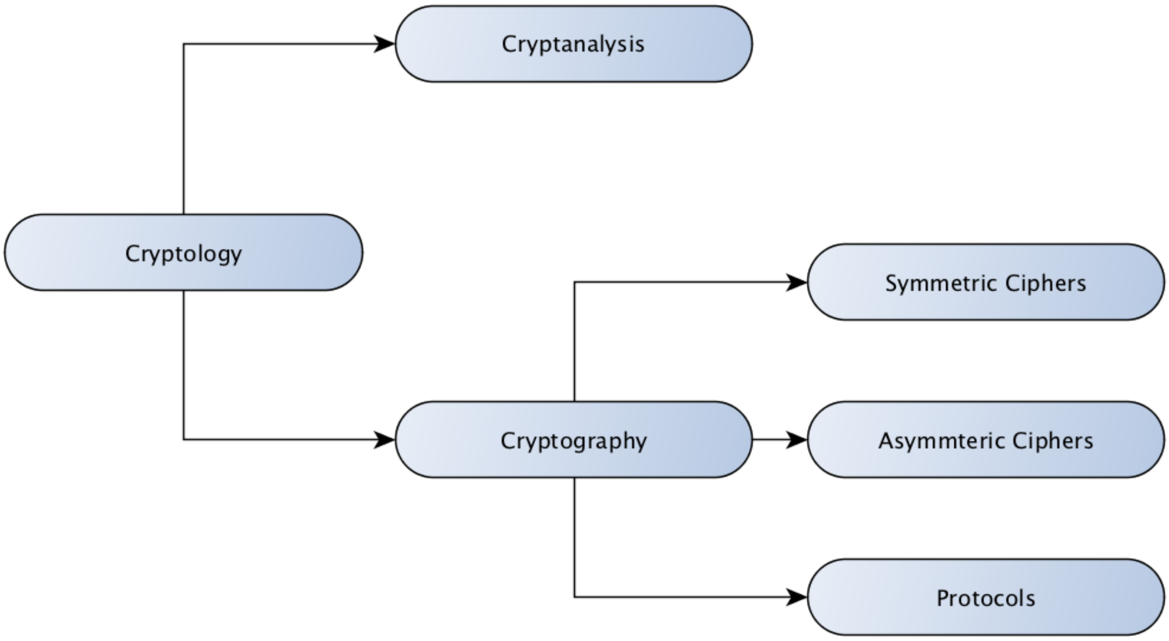


Figure 4.1: Areas of expertise in cryptology (Paar and Pelzl, 2010, p. 3)

According to Knudsen and Robshaw (2011), cryptography should provide some of the following:

Confidentiality: Even if public communication channels are used, information about the content of a secured communication should not be gained.

Data integrity: An unauthorized tempering of content from secured communications should not be possible even if an adversary has access to the communication channel.

Data origin authentication: Unwarranted modification and/or misrepresentation of the true origin of some communication should be averted.

The basic algorithm for cryptography (and so to speak, for all encryption methods) is outlined in equation 4.1.

$$Plaintext = Decrypted(Encrypted(Plaintext)) \tag{4.1}$$

In this thesis, the focus is mostly drawn to the cryptography portion of cryptology, and within this scope, confidentiality is of the most interest. Cryptanalysis is discussed at the final stage to provide an overview of possible attack scenarios for the used techniques. However, an in-depth fault analysis on cryptosystems is not provided, as it would go far beyond the scope of this work.

4.2 Entropy

In 1948, a famous and classical measure of uncertainty in information theory was defined by (Shannon, 1948). Shannon suggested that the entropy $H(X)$, the average amount of information of a discrete random variable X, can be defined by (Bonneau, 2012):

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \tag{4.2}$$

on the following terms:

- The variable X consists of a finite sample space $x_1, x_2, x_3, \ldots, x_n$,
- A probability distribution $p(x_i)$ that attains values $x_i \geq X$,
- As well as $\sum_{i=1}^{n} p(x_i) = 1$.

4.3 Galois Fields

A Galois field (GF) is a mathematical concept that is often used in cryptography (e.g with AES which is described in section 4.4.4) and coding theory. Galois fields consist of a finite set of elements in which we can add, subtract, multiply, and invert. The underlying assumption of these fields is that for every prime integer *prim* and every natural number n, solely one field with $prim^n$ elements exists. The *order* or also called *cardinality* of a Galois field is defined by the number of elements it implies. Therefore, a field of the order o only exists if $o = prim^n$. Working with bytes (8 bits), this means that a Galois field can be described as $GF(2^8) = GF(256) = \{0, 1, 2, 3, \ldots, 255\}$ with $prim = 2$ and $n = 8$ (Paar and Pelzl, 2010).

4.3.1 Arithmetics in Galois Fields $GF(2^n)$

All 2^n members of a $GF(2^n)$ can be represented as polynomials from a degree up to n-1. With this representation, each element $E \in GF(2^8)$ is viewed as $E(x) = e_7 \cdot x^7 + e_6 \cdot x^6 + \ldots + e_1 \cdot x + e_0$, with $e_i \in GF(2) = \{0, 1\}$. Thus arithmetic operations like addition and subtraction can be performed by standard polynomial addition and subtraction (Paar and Pelzl, 2010) and (Knudsen and Robshaw, 2011).

According to this, the summation SUM and difference $DIFF$ for the different elements $E_a(x)$, $E_b(x) \in GF(2^n)$ is computed by:

$$
\begin{aligned}
SUM(x) = E_a(x) + E_b(x) = \sum_{i=0}^{n-1} polyop_i \cdot x^i, \qquad polyop \equiv e_{ai} + e_{bi} \mod 2 \\
DIFF(x) = E_a(x) - E_b(x) = \sum_{i=0}^{n-1} polyop_i \cdot x^i, \quad polyop \equiv e_{ai} - e_{bi} \equiv e_{ai} + e_{bi} \mod 2
\end{aligned}
\tag{4.3}
$$

With the two elements as 8-bit vectors $E_a = (e_{a0}, e_{a1}, \ldots, e_{a7})$ and $E_b = (e_{b0}, e_{b1}, \ldots, e_{b7})$. As addition and subtraction make no difference to the modulo 2 operation, the same result is achieved for both of the mathematical procedures.
For the multiplication of two elements $E_a(x)$, $E_b(x) \in GF(2^n)$, the *irreducible polynomial* p_{ir} of degree n needs to be defined (NIST, 2001):

$$p_{ir}(x) = x^n + e_{n-1} \cdot x^{n-1} + e_{n-2} \cdot x^{n-2} + \ldots + e_2 \cdot x^2 + e_1 \cdot x + e_0 \tag{4.4}$$

Now the polynomials can be multiplied MUL using the standard polynomial multiplication rule (Paar and Pelzl, 2010):

$$MUL = E_a(x) \cdot E_b(x) \mod p_{ir}(x) \tag{4.5}$$

Division is performed by multiplicative inverses. For the inverse E_a^{-1} in $GF(2^n)$ of a nonzero element $E_a \in GF(2^n)$, the following formula is applied (Paar and Pelzl, 2010):

$$E_a^{-1}(x) = \frac{1 \mod p_{ir}(x)}{E_a(x)} \tag{4.6}$$

4.4 Symmetric Key Algorithm

Algorithms that use an identical key for encrypting a plaintext and decrypting a coded message (Figure 4.2) are called symmetric key algorithms. In some cases, symmetric key algorithms also use similar functions for encryption and decryption (e.g. the Data Encryption Standard (DES) uses widely identical functions). The major drawback of these algorithms is that all parties who are involved in the secure communication need to have the key, as well as the particular function for en/decryption. The transmission of the key or the so-called shared secret needs to be carried out through a secure communication link to prevent the compromise of encrypted communications. Once the key is exchanged, further communications can be carried out using hopefully faster/more convenient public communication channels. As a drawback of the symmetric key algorithm it must be admitted, that there is the problem with the potentially large number of keys: as every communication group cg holds its own key,

there are $\frac{cg_k \cdot (cg_k - 1)}{2}$ possible key pairs k within a domain. Even for the relatively small community of 226,000 households in Salzburg (Statistics Austria, 2012), symmetric key algorithms would require approximately 25.5 million keys. The demanding key generation process and the even more demanding secure transmission of this amount could pose a crucial challenge.
symmetric key algorithms can be applied by:

- Stream ciphers (see section 4.4.1)
- Block ciphers (see section 4.4.3)

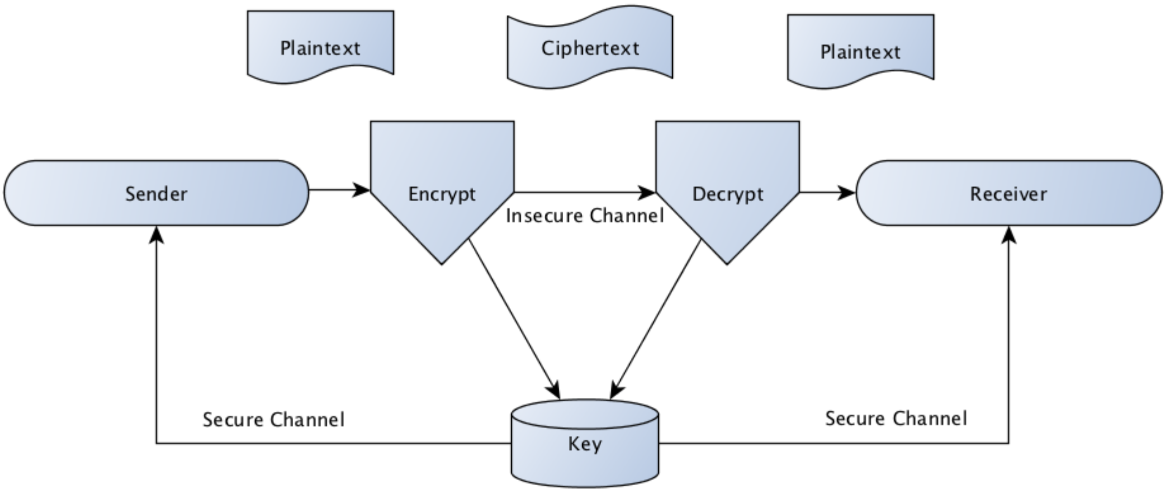


Figure 4.2: Basic proceeding of symmetric key algorithms

4.4.1 Stream Ciphers

Whenever characters from a plaintext are encrypted individually by symmetric key algorithms, stream ciphers are applied. A stream cipher takes one character from a plaintext at a time and adds a symbol from the pseudorandom key stream to the plaintext. Therefore, stream ciphers only work on a single bit or byte at a time. This is performed by a key stream generator that outputs a stream of $k_1, k_2, k_3, \ldots, k_n$ key bits and uses logical XOR-operations against a plaintext $P_1, P_2, P_3, \ldots, P_n$ to produce the stream of ciphertext bits $C_1, C_2, C_3, \ldots, C_n$ (see equation 4.7). To retrieve the plaintext from the ciphertext, the same key stream is applied against the ciphertext (see equation 4.8). Since $P_i \cdot k_i \cdot k_i = P_i$ this approach is valid. The reliability and security of this cryptographic system completely depends on the key stream generator. If the key stream $k_1, k_2, k_3, \ldots, k_n$ consists of a constant stream of zeros, the ciphertext will be exactly the same as the plaintext. The most secure key stream would be an endless stream of “real” random bits instead of pseudorandom key streams (Schneier, 1996).

$$C_i = P_i \cdot k_i \qquad \text{C} \ldots \text{ciphertext} \tag{4.7}$$

$$P_i = C_i \cdot k_i \qquad \text{P} \ldots \text{plaintext} \tag{4.8}$$

However, the used pseudorandom key streams and XOR-operations for encryption are mostly not complex operations which makes stream ciphers typically quite fast. As stream ciphers encrypt plaintexts on a character basis, the length of the message does not need to be known beforehand, making stream ciphers a well-fitting candidate for media streaming or encryption of wireless communications (Paar and Pelzl, 2010).
The most famous stream ciphers are the RC4 stream cipher, used in secure internet communication protocols like Secure Socket Layer (SSL) (Sen Gupta et al., 2013), and the A5/1 GSM stream cipher (Biham and Dunkelman, 2000), used for voice encryption in Global System for Mobile Communications (GSM).

4.4.2 RC4

Ron's Code 4 (RC4) is a famous and widely used variable key-size stream cipher in symmetric encryption, developed by Ronald Linn Rivest in 1987. RC4 is used in Secure Socket Layer (SSL), Wi-Fi security protocol WEP, etc. It works like a finite automaton and consists of a Key-Scheduling Algorithm (KSA) and a Pseudo-Random Generation Algorithm (PRGA). KSA (see Listing 4.1) uses a key k, usually of the length $40 \leq l \leq 128$ bits and the key array K to generate a pseudorandom permutation $S\{0, 1, 2, \ldots, 255\}$, also called S-Box. The S-Box is used as a nonlinear substitution operation to obscure the relationship between the key and the ciphertext. The PRGA (see Listing 4.2) uses the S-Box as the initial value of the internal state (Fontaine, 2011; Sen Gupta et al., 2013).

Listing 4.1: A pseudocode to initialize the S-Box

```
// l = chosen key-size (usually 40 <= l <= 256 bits)
for i = 0 to 255
    S[i] = i
end

j = 0
for i = 0 to 255
    j = (j + S[i] + K[i]) mod l
    Swap(S[i], S[j])
end
i = j = 0
```

Listing 4.2: A pseudocode of the RC4 pseudorandom number generator

```
i = 0
j = 0
while Needed
    i = (i + 1) mod 256
    j = (j + S[i]) mod 256
    swap the values of S[i] and S[j]
    rNum = S[(S[i] + S[j]) mod 256]
    output rNum
end
```

From a cryptanalysis view, this algorithm means that there are $2^{8 \cdot 256} = 2^{2048}$ possible keys for the RC4 cipher, resulting in $256! \cdot 256^2 = 2^{1700}$ possible states. This vast amount of states makes it quite difficult to reconstruct keys from internal states (Schneier, 1996).

4.4.3 Block Ciphers

The distinction between block and stream ciphers is often misleading since block ciphers are used as well to encrypt streams of data. But instead of encrypting every single character after another, block ciphers (as their name already implies) encrypt a finite set of data from plaintext.

Block ciphers are defined as follows:

> *"For any given key k, a block cipher specifies an encryption algorithm for computing the n-bit ciphertext for a given n-bit plaintext, together with a decryption to a given n-bit ciphertext."* (Knudsen, 2011, p. 152)

In other words, block ciphers encrypt blocks of plaintext into blocks of ciphertext under the action of a secret symmetric key (same key for en- and decryption). The cipher itself has two important parameters, which are the *block size* and the *key size*. The block sizes vary between implementations and fields of application, but typically are 64, 128 or 256 bits in length. The key size may vary as well but is more vital to the cryptosystem as it determines the number of permutations that are actually generated. E.g., with a block size b and a key size k there are 2^k possible keys and 2^b permutations per key. This means there are $(2^b)!$ permutations for each b-bit input block. Even more obfuscation is achieved by interchanging several blocks (Knudsen and Robshaw, 2011). Block ciphers are fundamental to modern cryptography, as they are ranked among the most widely used cryptographic mechanisms. The two most common block ciphers in modern cryptography are the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). The latter provides more security as it has the advantage that it is based on DES and covers the possible attacks on DES (Paar and Pelzl, 2010).

4.4.4 AES

In October 2000, the National Institute of Standards and Technology (NIST) announced that Rijndael (named after its two inventors Joan Daemen and Vincent Rijmen) would become the Advanced Encryption Standard (AES) as the US federal standard for the encryption of electronic data. Technically speaking, the standard can be described as a key-iterated block cipher that operates on blocks of 128 bits in length. Because of the possible key sizes, which can be 128, 196 or 256 bits, the different variants of AES are referred as AES-128, AES-196 and AES-256. The encryption of plaintext is performed throughout several numbers of rounds n_r which depend on key and block size. In the terminology of Rijndael, n_k is defined by the key size, divided by 32 (a sequence with a size of 4 bytes) and n_b is defined by the block length, divided by 32 (NIST, 2001) and (Daemen and Rijmen, 2011).
The number of rounds n_r (see Table 4.1) can be defined by (Daemen and Rijmen, 2011):

$$n_r = \max(n_k, n_b) + 6 \qquad \text{with} \quad \frac{n_k}{32} \quad \text{and} \quad \frac{n_b}{32} \tag{4.9}$$

The algorithm of AES internally operates on a two-dimensional array of bytes, the so-called State. The State is built up of a rectangular array of four rows of bytes, each containing n_b bytes. With the block size of 128 bits, this means the array consists of 16 elements each containing one byte. This structure represents the State array $s[r, c]$, where r is the row number and c is the column number with r,$c \in \{0, 1, 2, 3\}$. The cipher uses the State array to copy the bytes of the input array $in_0, in_1, in_2, ..., in_{15}$ to the State array $s[r, c]$ and subsequently to the output array $out_0, out_1, out_2, ..., out_{15}$ (illustrated in Figure 4.3) (NIST, 2001):

$$\begin{aligned} s[r, c] = in[r + 4c] \qquad & out[r + 4c] = s[r, c] \\ \text{for} \quad 0 \leq r < 4 \quad \text{and} \quad & 0 \leq c < n_b \end{aligned} \tag{4.10}$$

AES type	AES-128	AES-192	AES-256
$\mathbf{n_r}$	10	12	14

Table 4.1: Number of rounds for the AES variants

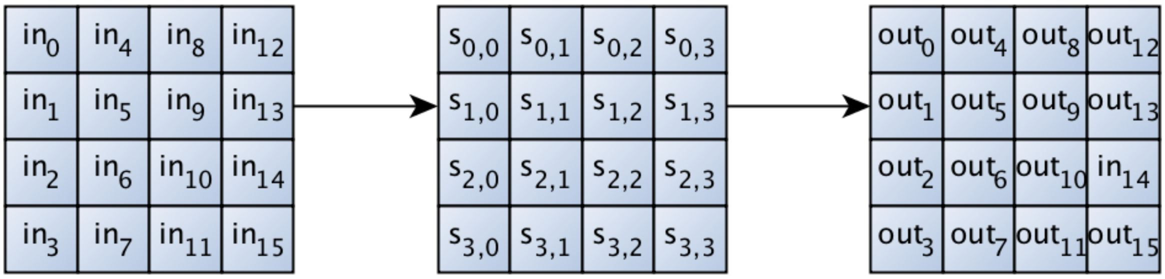


Figure 4.3: AES State array s, input in and output out: the AES State array s copies the input $in_0 - in_{15}$ to the output $out_0 - out_{15}$ (NIST, 2001, p. 9).

According to Paar and Pelzl (2010), the three so-called layers within AES are:

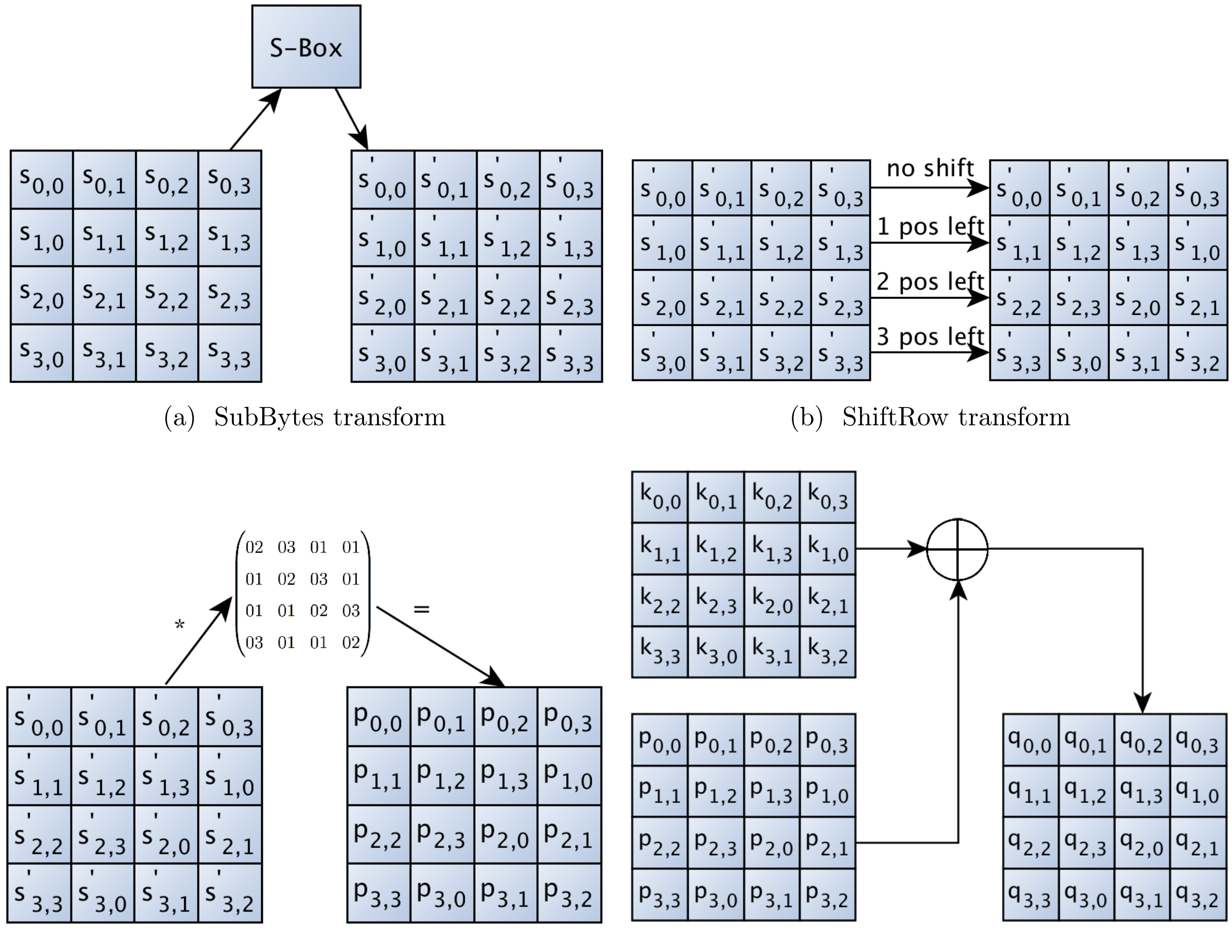
- The **Key Addition layer** that implies a 128-bit *round key* (the size comes from the block size) that has been derived from the main key. The *round key* is used in a logical XOR-operation against the State.
- The **Byte Substitution Layer** that introduces *confusion* to the data by means of nonlinearly transforms using the lookup/substitution table S-Box.
- And the **Diffusion Layer** to permute data on a byte level (ShiftRows) and to mix blocks of four bytes (MixColumn).

The following four transformations (see also Figure 4.4) are applied by the AES cipher each round, except the final one (NIST, 2001) and (Vaudenay, 2005):

- **SubBytes()**,
- **ShiftsRows()**,
- **MixColumns()**,
- **AddRoundKey()**.

Within **SubBytes** each byte of the State is transformed by a nonlinear byte substitution using a substitution table, the S-Box. In **Shiftrows**, a circular shift of all rows is performed by shifting each row number r of the 4x4 State matrix $s[r, c]$ by r positions to the left. **MixColumns** is used to mix each column of the State matrix to enable *diffusion*. Therefore, each column is multiplied with a 4x4 $GF(2^8)$ matrix M (see equation 4.11). During the preliminary round and at the end of each round, **AddRoundKey** is performed to add a *round key* $k_{r,c}$ to the State. To do this, XOR-operations are applied for each byte of the State against the derived user-supplied key $k_{r,c}$ (NIST, 2001) and (Vaudenay, 2005) and (Paar and Pelzl, 2010).

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{pmatrix} = \begin{pmatrix} p_{0,c} \\ p_{1,c} \\ p_{2,c} \\ p_{3,c} \end{pmatrix}, \quad M = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}, c = 0, 1, 2, 3 \tag{4.11}$$


S-Box
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$
no shift
1 pos left
2 pos left
3 pos left
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$ $s'_{1,0}$
$s'_{2,2}$ $s'_{2,3}$ $s'_{2,0}$ $s'_{2,1}$
$s'_{3,3}$ $s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$
*
02 03 01 01
01 02 03 01
01 01 02 03
03 01 01 02
=
$p_{0,0}$ $p_{0,1}$ $p_{0,2}$ $p_{0,3}$
$p_{1,1}$ $p_{1,2}$ $p_{1,3}$ $p_{1,0}$
$p_{2,2}$ $p_{2,3}$ $p_{2,0}$ $p_{2,1}$
$p_{3,3}$ $p_{3,0}$ $p_{3,1}$ $p_{3,2}$
$k_{0,0}$ $k_{0,1}$ $k_{0,2}$ $k_{0,3}$
$k_{1,1}$ $k_{1,2}$ $k_{1,3}$ $k_{1,0}$
$k_{2,2}$ $k_{2,3}$ $k_{2,0}$ $k_{2,1}$
$k_{3,3}$ $k_{3,0}$ $k_{3,1}$ $k_{3,2}$
$q_{0,0}$ $q_{0,1}$ $q_{0,2}$ $q_{0,3}$
$q_{1,1}$ $q_{1,2}$ $q_{1,3}$ $q_{1,0}$
$q_{2,2}$ $q_{2,3}$ $q_{2,0}$ $q_{2,1}$
$q_{3,3}$ $q_{3,0}$ $q_{3,1}$ $q_{3,2}$

(a) SubBytes transform

(b) ShiftRow transform

(c) MixColumn transform

(d) AddRoundKey transform

Figure 4.4: The four transformations **SubBytes**, **ShiftsRows**, **MixColumns** and **AddRoundKey** used by AES (NIST, 2001) and (Vaudenay, 2005) and (Paar and Pelzl, 2010).

Decryption with AES is carried out with the corresponding inverse operation applied in the right order shown in Table 4.2:

Encryption	**Decryption**
SubBytes: use S-Box	InvSubBytes: use inverse S-Box^{-1}
ShiftRows: rotate to the left	InvShiftRows: rotate to the right
MixColumns: multiply by 4x4-matrix	InvMixColumns: multiply by 4x4-matrix^{-1}
AddRoundKey (self-inverse)	AddRoundKey (self-inverse)

Table 4.2: Decryption with AES (Knudsen and Robshaw, 2011)

4.5 Asymmetric/Public-Key Algorithm

The invention of Public-Key algorithms (also called asymmetric key algorithms) goes back to the year of 1979, where Whitfield Diffie and Martin Hellman proposed "new directions in cryptography" (Diffie and Hellman, 1976). This milestone paper introduced how two parties could communicate privately with two different keys using a public channel (see Figure 4.5). The first key, the encryption key, is different and cannot be calculated (within a meaningful time) from the second key, the decryption key. In other words, it is not necessary to keep the Public-Key for encrypting a plaintext secret as long as the Private-Key for decrypting the ciphertext is confidential. With this approach, even a stranger could encrypt messages but still cannot decrypt a ciphertext without the corresponding Private-Key. By means of this approach, the key is only a secret of one person/entity and not a secret of a pair or group of entities like with symmetric key algorithms.
An analogy true to life is a conventional mailbox on the corner of a street: a letter can be put into the mailbox by everyone (i.e encrypt), but only an entity with the Private-Key (i.e the postman) can retrieve (i.e decrypt) letters (Vaudenay, 2005).

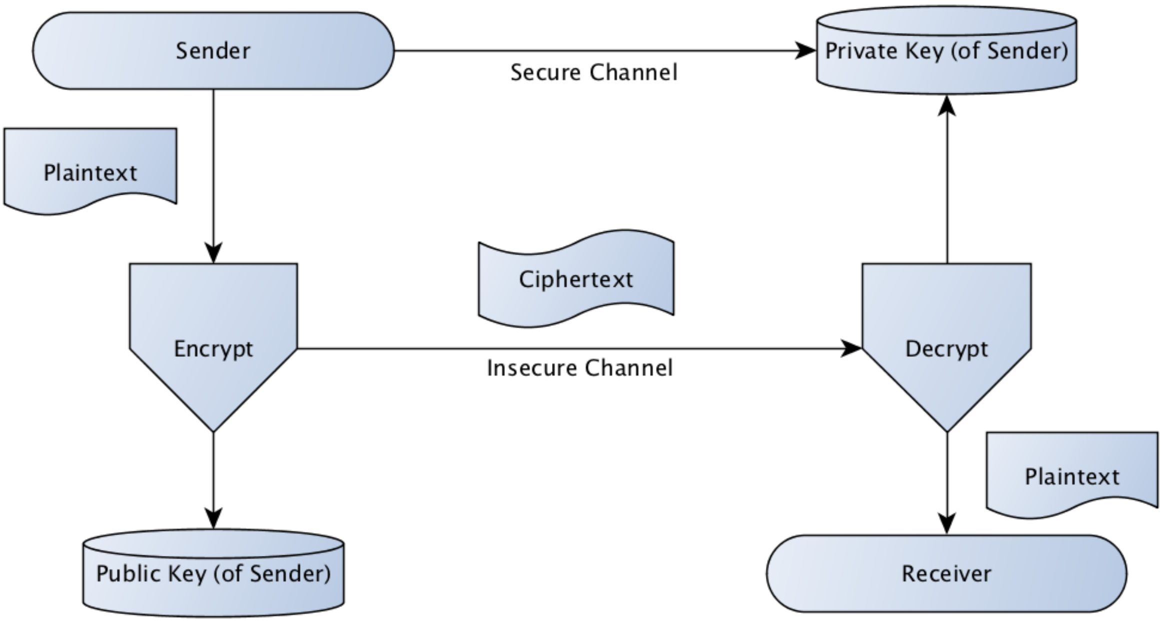


Figure 4.5: Basic proceeding of asymmetric key algorithms

In Vaudenay (2005), a Public-Key cryptosystem is defined by:

- A probabilistic algorithm (also called pseudorandom key generator) which outputs the public k_{pub} and private k_{priv} key pair,
- An encryption algorithm **Enc** (can be probabilistic) which outputs the ciphertext c from a plaintext using the Public-Key k_{pub},

- And a decryption algorithm **Dec** which outputs the plaintext p from the ciphertext c by using the Private-Key k_{priv}.

Generally the encryption of a plaintext P is done by using the Public-Key k_{pub} (see equation 4.12), whereas decryption is realized with the associated Private-Key k_{priv} (see equation 4.13). The so-called one-way functions and trapdoor one-way permutations are applied for the proceeding of en/decryption. This approach additionally implies the advantage of applying digital signature schemes of message authentication and non-repudiation (Schneier, 1996).

$$Enc_{k_{pub}}(P) = C \qquad \text{C} \ldots \text{ciphertext} \tag{4.12}$$

$$Dec_{k_{priv}}(C) = P \qquad \text{P} \ldots \text{plaintext} \tag{4.13}$$

One-way functions constitute the fundamental principle of Public-Key algorithms. On the one side, they are quite simple to compute on every input, but they are highly complex to invert on the other side. Mathematically speaking, the computation of $y = f(x)$ should be sufficiently fast, whereas the inverse computation $x = f^{-1}(y)$ should be unacceptably slow in computation unless someone is given some trapdoor information (Paar and Pelzl, 2010).
According to Paar and Pelzl (2010), there are three major types of Public-Key algorithms (in respect to their one-way function) that are of practical relevance:

Integer-Factorization Schemes are based on the factoring problem: it is presumed that factoring large integers is quite difficult. E.g., the famous RSA (named after its inventors Ron Rivest, Adi Shamir and Leonard Adleman) algorithm avails itself of the factoring problem.

Discrete Logarithm Schemes rely on the so-called discrete logarithm problem[5]: using modulo arithmetics to make the mathematical problem more intractable than with the factoring problem. A famous example for the usage of the discrete logarithm problem is the Digital Signature Algorithm (DSA).

Elliptic Curve Cryptography (ECC) Schemes operate over points on an elliptic curve. By means of scalar point multiplication of points from a securely held elliptic curve, the algorithm makes it almost infeasible to solve the mathematical problem. The most established examples are the Elliptic Curve Diffie-Hellman[6] key exchange (ECDH) and the Elliptic Curve Digital Signature Algorithm[7] (ECDSA).

[5] `http://modular.math.washington.edu/edu/124/lectures/lecture8/html/node5.html`, Accessed: 3.26.2013

[6] `http://csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_Revision1_Mar08-2007.pdf`, Accessed: 4.2.2013

[7] `http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf`, Accessed: 4.2.2013

In some cases, a Private-Key k_{priv} is used for encryption and a Public-Key k_{pub} is used for decryption. E.g., this is performed to authenticate an entity, as only the entity holds its Private-Key; nobody else is able to encrypt a plaintext without the specific Private-Key.

4.5.1 Rivest-Shamir-Adleman (RSA) Algorithm

The Rivest-Shamir-Adleman algorithm, often referred as RSA (named after its three inventors Ron Rivest, Adi Shamir, and Leonard Adleman), is one of the widely-used asymmetric key encryption algorithms. It is based on the presumed difficulty of factoring very large prime numbers, the integer factorization problem: the multiplication of two large prime numbers is computationally easy, but factoring its product is related with high computational effort. The involved Public- and Private-Keys are functions of a pair based on these large prime numbers. The keys are usually larger than 1024 bits, and therefore it is quite difficult to recover the plaintext from the Public-Key or the ciphertext once their product is built (Schneier, 1996) and (Delfs and Knebl, 2007) and (Paar and Pelzl, 2010).
Encryption and decryption with RSA is proceeded straightforward with the Public-Key $k_{pub}(n, e)$ and Private-Key $k_{priv} = d$ using modulo arithmetic where $d, e, n, x, y \in \mathbb{Z}$ (Paar and Pelzl, 2010):

$$y = e_{k_{pub}}(P) \equiv P^e \mod n \qquad \text{P...plaintext} \tag{4.14}$$

$$x = d_{k_{priv}}(C) \equiv y^d \mod n \qquad \text{C...ciphertext} \tag{4.15}$$

Every entity within an RSA cryptosystem has its own pair of Public-Key k_{pub} and Private-Key k_{priv}. Therefore, RSA key generation is performed by each entity using the following key generation algorithm (Delfs and Knebl, 2007) and (Paar and Pelzl, 2010):

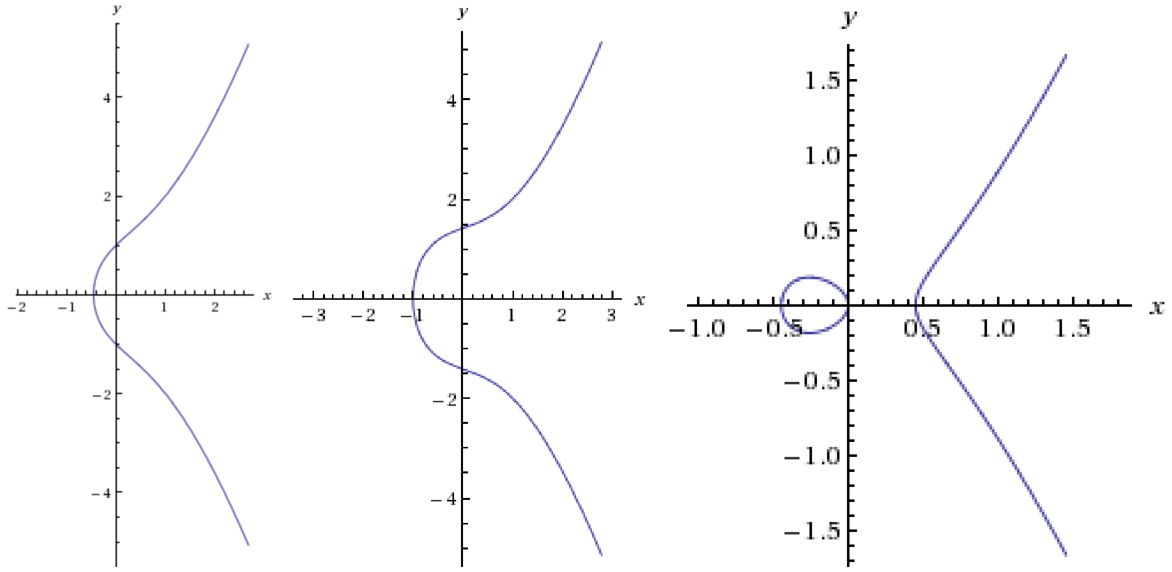
1. Choose large distinct primes p and q to calculate $n = p \cdot q$.
2. Choose e that is prime to $\Phi(n) = (p - 1) \cdot (q - 1)$, so that $gcd(e, \Phi(n)) = 1$ (gcd...greatest common divisor).
3. d can be generated from $e \cdot d \equiv 1 \mod \Phi(n) \cdot (n, d)$ by:
   $d = e^{-1} \mod ((p - 1) \cdot (q - 1))$.
4. Use the pair (n, e) as Public-Key k_{pub} and (n, d) as Private-Key.

4.5.2 Elliptic Curve Cryptography (ECC)

Elliptic curve cryptography (ECC) takes advantage of the characteristics of a elliptic curve: an elliptic curve can be described as a finite set of points $s(p)$ with $p(x, y)$ over the binary Galois fields $GF(2^n)$ or over the prime Galois fields $GF(prime)$ by (Jao, 2010):

$$y^2 = x^3 + a \cdot x + b \qquad a, b, x, y \in \mathbb{R} \tag{4.16}$$

The secret lies within the parameters a and b. It is almost impossible to reconstruct the elliptic curve, and therefore the implied set of points $s(p)$, without the corresponding a and b parameters. Any small aberration from the original a and b results in a significantly different elliptic curve (see Figure 4.6).



(a) a=2 , b=1

(b) a=1 , b=2

(c) a=-0.2 , b=0

Figure 4.6: Elliptic curves with different a and b parameters

Using this group of points as an underlying number system for ECC makes it quite hard to solve the elliptic curve discrete logarithm problem (ECDLP) (Jao, 2010).

According to Jao (2010) and Hankerson and Menezes (2011), significantly smaller operands (approximately 150-256 vs 1024-3072 bits) can be used for ECC, providing the same security level as RSA, but making them much faster.

This is to say because solving the ECDLP is much harder than solving the discrete logarithm or the factorization of integers. This may be one of the reasons why ECC is gaining popularity for many new applications, especially on embedded platforms (Paar and Pelzl, 2010).

For describing the encryption and decryption algorithm, like specified in Amounas and Kinani (2012), it is assumed that the Public- and Private-Key pair derives from the curve points from an elliptic curve E over a finite field $GF(prime)$. Furthermore, E is defined by a base point $O \in E$ of large prime order n which is publicly known. When a sender s communicates with a receiver r, the proceeding can be defined as follows (Jao, 2010) and (Amounas and Kinani, 2012):

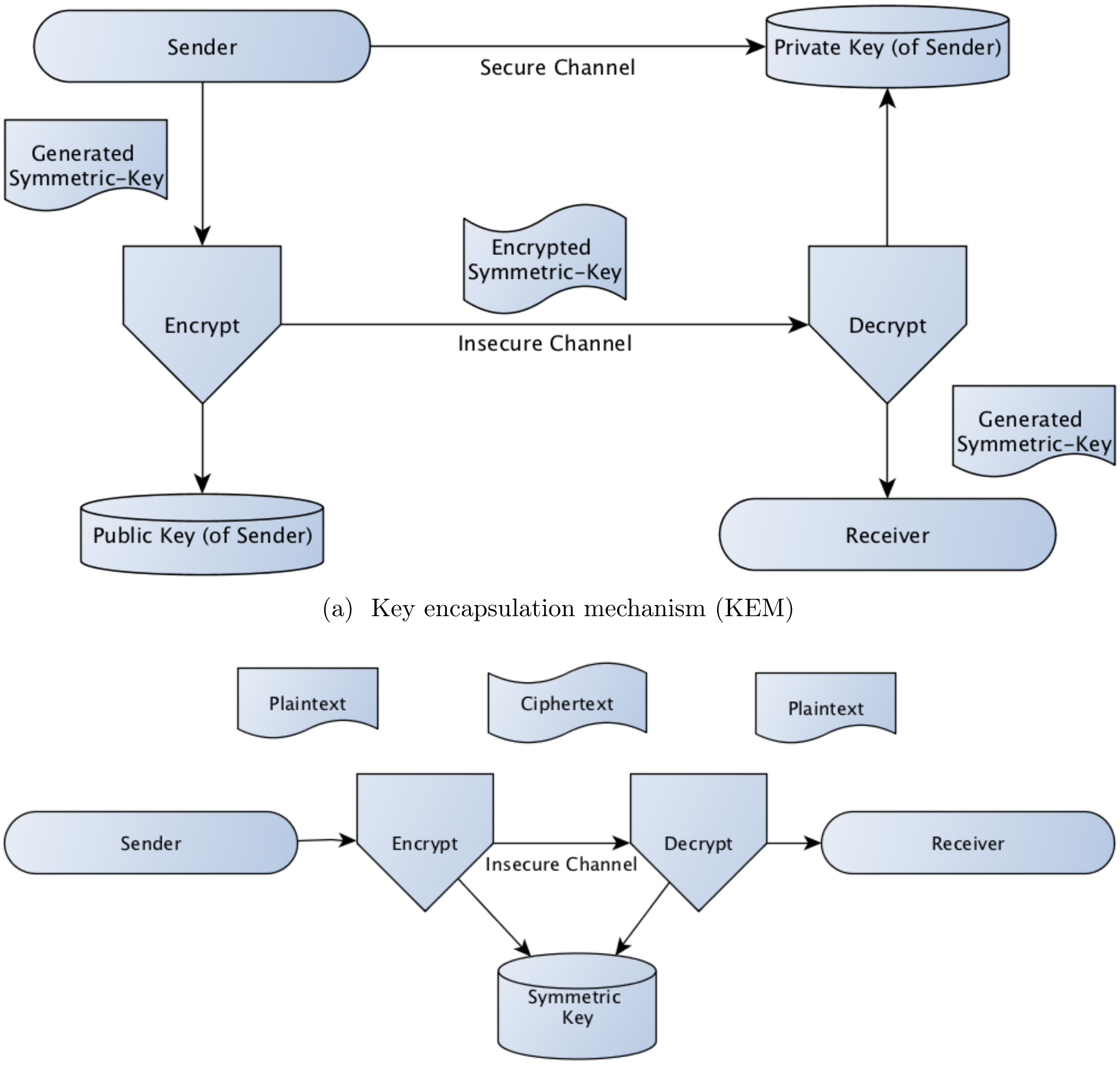
- First the Public- and Private-Key pair is generated by choosing a random integer $\alpha, 1 \leq \alpha < n$ and calculating $\beta = \alpha \cdot O$ (where α is the Private-Key and β the Public-Key).
- Then, s chooses a random integer $i_{rand}, 0 \leq i_{rand} < n$ and calculates $C_1 = i_{rand} \cdot O$ and $C_2 = i_{rand} \cdot \beta + P$ (where P is the plaintext).
- The ciphertext C is described by (C_1, C_2).
- To encrypt the ciphertext, r uses the known base point O, the parameters a, b, and α to perform $P = C_2 - \alpha \cdot C_1$.

4.6 Hybrid Encryption Algorithm

According to Schneier (1996), there are two major reasons why Public-Key algorithms are not a meaningful substitute for symmetric key algorithms:

- Firstly, Public-Key cryptosystems are at least 1000 times slower than symmetric key algorithms. They consume more resources, because of their use of two different keys and the one-way functions. Even though this statement was made in 1996, this assumption is still valid.
- Secondly, Public-Key algorithms are vulnerable to chosen plaintext attacks (CCA). E.g., as the encryption key is public and the the ciphertext is generated through $C = Enc(P)$, an attacker could (of course, for relatively short plaintexts) encrypt all possible plaintexts and compare them with the ciphertext. Even if the attacker is not able to reconstruct the Private-Key k_{priv}, this approach leads to the plaintext.

In hybrid encryption algorithms, a combination of symmetric key and Public-Key encryption is conducted. Therefore hybrid encryption algorithms consist of two parts (see Figure 4.7): the key encapsulation mechanism (KEM) and the data encapsulation mechanism (DEM). Within the KEM, a Public-Key encryption is performed to secure the so-called (symmetric) *session key* k_s. This session key k_s is used to encrypt the actual plaintext using symmetric key encryption, also called DEM (Cramer and Shoup, 2004) and (Kurosawa, 2011).
The motivation for hybrid encryption is obvious: taking advantage of the strengths from both encryption algorithms. Speed through symmetric key and security through asymmetric key encryption.



(a) Key encapsulation mechanism (KEM)

(b) Data encapsulation mechanism (DES)

Figure 4.7: Basic proceeding of hybrid encryption algorithms: first the symmetric key is encrypted with the Public-Key algorithm. After the symmetric key is securely exchanged, messages are encrypted with the symmetric key and transmitted via public channels.

4.7 Selective Encryption Algorithm

The idea of selective encryption methods is to encrypt only a relatively small (but crucial) portion of the plaintext, while leaving the remaining part unencrypted. The motivation usually emerges from trading confidentiality against performance. Figure 4.8 describes the basic approach for selective encryption.

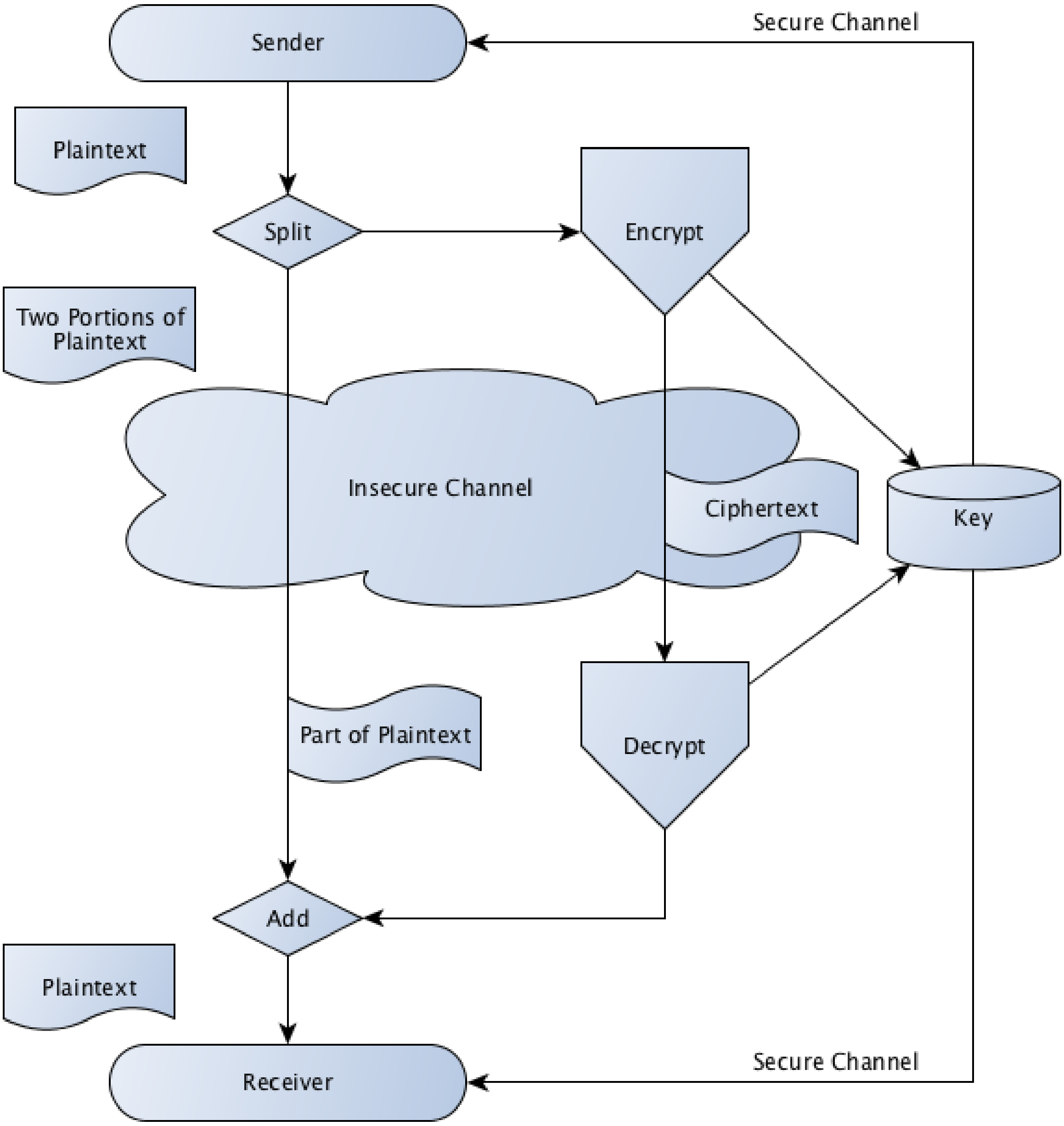


Figure 4.8: Basic approach for selective encryption: the plaintext is split into a ciphered and a plaintext portion. The receiver decrypts the ciphertext and combines it with the unencrypted portion to retrieve the plaintext.

This idea is widely used in multimedia security, where the encryption of the entire bitstream from multimedia data could be too expensive or simply require too much processing power. E.g., in video streaming the encryption of the entire multimedia data, which is usually large, implies a vast overhead from encryption. This implies a significant influence to the required real-time operation. There are numerous examples in the field of multimedia security, where selective encryption techniques are implemented to dramatically reduce the computation time. These examples reach from selective encryption of still images (Pommer and Uhl, 2003), to the protection of specific layers in videos (Vijayalakshmi et al., 2010), often to serve the purpose of Digital Rights Management (DRM)[8].

However, this thesis pays more attention to the following two approaches for selective encryption of multi-resolution smart meter data:

- An adaption of the generic selective encryption method (Lundin and Lindskog, 2010),
- The encryption of data from wavelet analysis (see chapter 5) (i.e. decomposition) proposed by Engel (2011) and Engel (2013), as well as Pommer and Uhl (2003).

The latter is discussed in more detail in chapter 5.

4.8 Generic Selective Encryption Method (GSEM)

In Lundin and Lindskog (2010), the generic selective encryption for zero-, first- and second-order languages is investigated regarding its security implications. Even if data that originates from a text message is quite different than smart meter data, they have the common ground of recurring patterns. In terms of languages the repeating patterns are the words that consist of letters from a limited alphabet. Referred to smart meter data, the recurring patterns are the dates: E.g., a daily consumption is displayed by 96 meter readings which are merely distinct by their timestamp and the accumulated consumption value. Therefore the generic selective encryption method could pose a meaningful approach to secure smart meter data.

4.8.1 Generic Selective Encryption Approach

The generic selective encryption approach (Lundin and Lindskog, 2010) divides a plaintext P into n equally sized fractions P_i, where $0 \geq i < n$. Thus the plaintext P can be established by binary concatenation $\oplus$ of its fractions:

$$P = P_1 \oplus P_2 \oplus P_3 \oplus \ldots \oplus P_n \tag{4.17}$$

In Figure 4.9, a selective encrypted plaintext with four encrypted fractions (gray colored) and six unencrypted fractions (white colored) is shown.

[8] Access control technology to limit the use of digital content after sale.

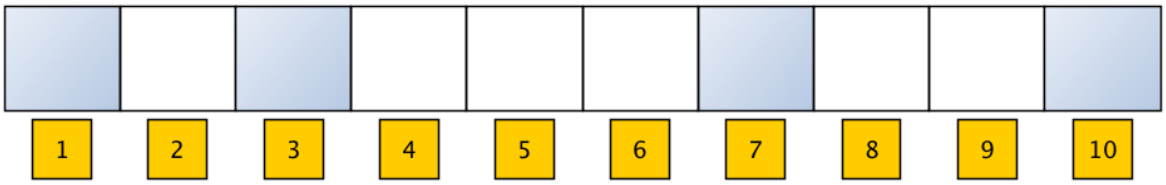


Figure 4.9: Example of a selectively encrypted plaintext with four encrypted parts (gray colored) and six unencrypted parts (white colored) (Lundin and Lindskog, 2010)

.

The three entities that are involved in the encryption process are:

- P, the plaintext to selectively encrypt,
- The vector $\vec{v}$ to control the apportionment of the encrypted and unencrypted fractions,
- $E(P)$, the encrypted plaintext.

4.9 Comparison of Encryption Algorithms

Irrespective of symmetric or asymmetric key algorithms, the secrecy of a cryptosystem is not solely depending on the key length. The redundancy within the plaintext often could lead to deep insights or even bypassing a cryptosystem. Shannon (1949) describes in his milestone paper from 1949 a Secret-Key cryptosystem from the information theory perspective. Regarding his proposal, perfect secrecy for a cryptosystem is provided, if a ciphertext allows no conclusions about the plaintext. Furthermore a statistical analysis on the ciphertext should be frustrating and due to its high effort not worthwhile.
Shannon (1949) defines two techniques to disguise redundancy within a plaintext:

- **Diffusion** - Every character from the key and plaintext should influence as many parts of the ciphertext as possible (use of permutations).
- **Confusion** - With the use of simple substitution, the relationship between the plaintext and the ciphertext is covered.

4.9.1 The Rationale for Choosing AES-256 as Encryption Algorithm

According to Schneier (1996), asymmetric key cryptography and symmetric key cryptography solve different problems: symmetric key algorithms fit best for encrypting data whereas asymmetric key algorithms are more suitable for key management, authentication and verification processes. As authentication and verification processes are not mandatory for this research, symmetric key cryptography plays a more significant role for securing smart meter data. The decisive advantage of symmetric key cryptography over asymmetric key cryptography is performance. This comes from the assumption that Public-Key algorithms produce more computational overhead due to their one-way functions and their key-pairs (two different keys are required).

On closer examination of symmetric key algorithms, there are two available approaches: the use of stream ciphers and the use of block ciphers. As already mentioned, the smart meter data is stored on the flash memory of the smart meter using the CSV file format. For data encryption this means, the amount of input data is known beforehand and it is therefore not necessary to process the plaintext on a character basis.

Furthermore, in terms of confusion and diffusion, block ciphers seem to fit well as they are designed to provide sufficient confusion and high diffusion through their algorithms. This comes from the fact that block ciphers diffuse the information from one plaintext character into several ciphertext characters. The last question that needs to be answered, is which of the proposed block ciphers fits best for securing the smart meter data.

As AES is still used as the US federal standard and international standard for the encryption of classified electronic data, the the block cipher AES is considered as safe enough to meet the demands of this research. Therefore AES is chosen as encryption algorithm for the multi-resolution smart meter data.

4.10 Defining the Trade-off in Security and Performance

As smart meters are, due to their hardware constrains, not capable of carrying out demanding cryptography, the computational effort must be reduced. Reducing the computational effort by leaving major parts of the metering data in plaintext means to simultaneously reduce security as well. To prevent the creation of detailed load profiles on the one hand, and to avoid to overburden the smart meter on the other hand, an appropriate trade-off between security and performance needs to be identified. However, defining a trade-off in security and performance may constitute a major challenge. A well fitting method, such as proposed by this research, is the evaluation of the required amount of encrypted data for selective encryption methods and the re-traceability of the load profiles from the remaining unencrypted segments.

This evaluation is performed by comparing the results from two approaches: the use of the default CSV data representation and the wavelet data representation, considering the amounts of ciphered data on a “best-effort” basis. A comparison of the several time requirements for securing the metering data, provides sufficient characteristics for specifying a cost matrix. The desired trade-off between security and performance can be defined by two metrics. Firstly, the PSNR between the selectively encrypted data and the plaintext data gives information about the secrecy and the required amount of encrypted data. Secondly, the similarity between the selectively ciphered data and the unciphered data provides additional validation of the trade-off. The aim for the proposed selective encryption methods is to make it (at least almost) impossible for an attacker to reconstruct the load profile from unencrypted data.

5 Wavelets as Data Representation

As the prior discussed approach with the CSV file format for data representation may provide good encryption results, its trade-off between security and performance may be too one-sided on the security part: trading too much performance for security. The following chapter describes the usage of an alternative approach that relies on wavelets as data representation format for the smart meter data. More specifically, the multi-resolution analysis (MRA) from discrete wavelet transforms (DWT) is analyzed to justify how MRA could be used to secure multi-resolution smart meter data.

5.1 Introduction to Wavelets

> *"The concept of **wavelet** originates from the study of signal analysis, i.e., from the need in certain cases to analyze a signal in the time and frequency domains simultaneously. The crucial advantage of wavelet analyses is that they allow us to decompose complicated information contained in a signal into elementary functions associated with different time scales and different frequencies and to reconstruct it with high precision and efficiency."* (Shima and Nakayama, 2010, p. 449)

According to Christensen (2010) and Shima and Nakayama (2010), a wavelet can be described as a mathematical function of finite length and fast decaying oscillating waveforms to split a given function into different scale components. This wavelet theory provides an analysis of a signal within a vector space $L^2(\mathbb{R})$. In $L^2(\mathbb{R})$ orthonormal bases are constructed where an orthonormal set of vectors, with the length of one, build the vector space. Many applications in signal processing take place within this normed $L^2(\mathbb{R})$ vector space, as a detailed time and frequency analysis of signals can be conducted. For example, if b_k with the coefficients c_k serves as an orthonormal basis for $L^2(\mathbb{R})$, all functions $f \in L^2(\mathbb{R})$ can be described by (Christensen, 2010):

$$f = \sum_{k=0}^{\infty} c_k \cdot b_k \qquad \text{where } k \in \mathbb{Z} \tag{5.1}$$

5.1.1 Definition of Wavelet

In consideration of real-value functions, a wavelet $\psi(t)$ displayed over time t, needs to have a localized waveform that meets the following criteria (Shima and Nakayama, 2010):

1. The integral of the wavelet $\psi(t)$ equals zero: $\int_{-\infty}^{\infty} \psi(t)dt = 0$.
2. The integral of the square wavelet $\psi(t)^2$ leads to normalization: $\int_{-\infty}^{\infty} \psi(t)^2 dt = 1$.

3. The Fourier transform[9] $\Psi(\omega)$ of $\psi(t)$ needs to satisfy the so-called **admissibility constant** C_Ψ, whose value depends on the chosen wavelet: $C_\Psi \equiv \int_0^\infty \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty$.

5.1.2 Haar Wavelet

As already mentioned, a real-valued wavelet $\psi(t)$ requires an orthonormal basis in $L^2(\mathbb{R})$. The Haar wavelet (see Figure 5.1) meets these requirements as it is orthogonal and symmetric in nature. This means that Haar wavelets have linear phase characteristics and therefore no corruption of the input signal during filtering is performed (Gao and Yan, 2011).

The Haar wavelet function ψ_{Haar} is mathematically defined by (Christensen, 2010):

$$\psi_{Haar}(t) = \begin{cases} 1 & \text{for} \quad 0 \leq t < \frac{1}{2}, \\ -1 & \text{for} \quad \frac{1}{2} \leq t < 1, \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

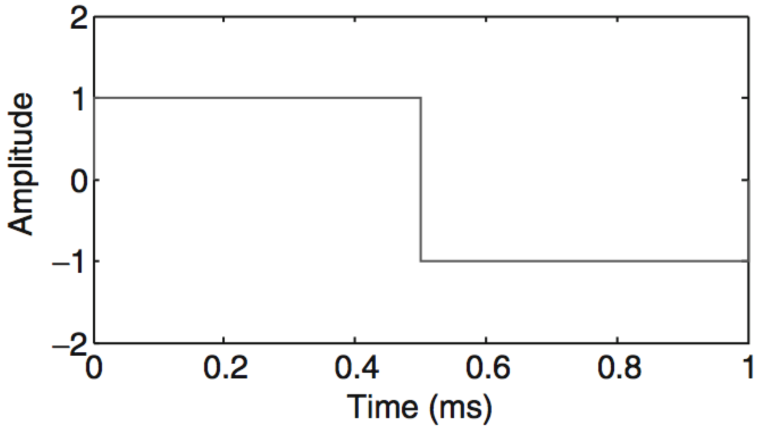


Figure 5.1: The Haar wavelet (Gao and Yan, 2011, p. 61)

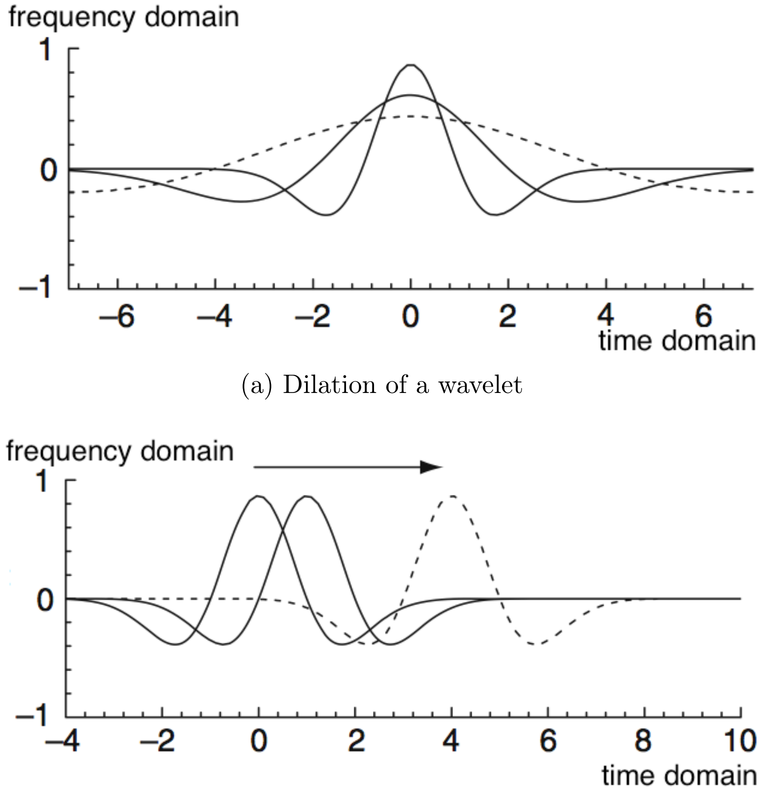
5.2 Wavelet Transform

According to Freeman and Quiroga (2013) and Mallat (2009), the wavelet transformation describes a process, where a signal of one independent variable t is mapped onto a function of two independent variables a and b, to create dictionary time-frequency *atoms*.

The variables a and b are used as parameters to manipulate the function form of the wavelet. Parameter a, also known as dilation parameter, defines the dilation and contraction of the wavelet within the time domain (see Figure 5.2a). Whereas parameter

[9] `http://www.thefouriertransform.com/`, Accessed: 4.17.2013

b, also known as translation parameter, guides the movement of the wavelet along the time axis (see Figure 5.2b) (Shima and Nakayama, 2010).



(a) Dilation of a wavelet

(b) Translation of a wavelet

Figure 5.2: Translation and dilation of a wavelet (Shima and Nakayama, 2010, p. 451)

In other words, a wavelet transform is another form of representing a signal without any alternation of the input information. A wavelet transform $T(a, b)$ of a signal $s(t)$, using a wavelet $\psi(t)$ is mathematically defined by (Shima and Nakayama, 2010):

$$T(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(t) \cdot \psi_{cc}(\frac{t-b}{a}) dt \qquad \psi_{cc} \equiv \text{ complex conjugated base wavelet} \tag{5.3}$$

5.2.1 Discrete Wavelet Transform (DWT)

In cases where the input data is represented by a finite number of values, the use of the discrete wavelet transform (DWT) is reasonable. The greatest advantage of the DWT, in contrast to other wavelet transforms, is that the DWT reduces the redundancy in the wavelet coefficients. The redundancy reduction within the coefficients is performed by discretizing the dilation parameter a and the translation parameter b. A commonly applied approach for the discretization is the use of the dyadic grid wavelet $\psi_{j,k}(t)$, where $j, k \in \mathbb{Z}$ (Gao and Yan, 2011).

The dyadic grid wavelet is defined by (Gao and Yan, 2011):

$$\psi_{j,k}(t) = \sqrt{2^j} \cdot \psi(2^j t - k) \tag{5.4}$$

According to Gao and Yan (2011) and Christensen (2010), the DWT for the signal $s(t)$ is performed using the dyadic grid wavelet:

$$T_{j,k} = \int_{-\infty}^{\infty} s(t) \cdot \psi_{j,k}(t) dt = \int_{-\infty}^{\infty} s(t) \cdot \sqrt{2^j} \cdot \psi(2^j t - k) dt \tag{5.5}$$

Applying the Haar wavelet to perform the discrete wavelet transform, the following Haar discrete wavelet is defined (Gao and Yan, 2011):

$$\psi_{Haarj,k}(t) = \sqrt{2^j} \cdot \psi(2^j t - k) \quad \text{where} \quad \psi(t) = \begin{cases} 1 & \text{for} \quad 0 \leq t < \frac{1}{2}, \\ -1 & \text{for} \quad \frac{1}{2} \leq t < 1, \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

5.3 Inverse Wavelet Transform

To reproduce the original signal $s(t)$ from the wavelet transform $T(a, b)$, an inverse wavelet transform is performed. Considering the signal $s(t) \in L^2(\mathbb{R})$ as the function $f(t)$, the signal can be reconstructed by the following definition (Shima and Nakayama, 2010):

$$f(t) = \frac{1}{C_\Psi} \int_{-\infty}^{\infty} db \int_{0}^{\infty} \frac{da}{a^2} \cdot T(a, b) \cdot \psi_{a,b}(t) \tag{5.7}$$

5.3.1 Inverse Discrete Wavelet Transform (IDWT)

Applying the Haar wavelet to the inverse wavelet transform from equation 5.7, the inverse discrete wavelet transform (IDWT) for the signal $s(t)$ can be described by:

$$IDWT_{Haar} = s(t) = \frac{1}{C_\Psi} \int_{-\infty}^{\infty} db \int_{0}^{\infty} \frac{da}{a^2} \cdot T(a, b) \cdot \sqrt{2^a} \cdot \psi(2^a t - b) \tag{5.8}$$

5.4 Multi-Resolution Analysis (MRA)

The multi-resolution analysis (MRA) or also called multi scale approximation, describes a recursive method for performing discrete wavelet analysis and synthesis of signals or data within a vector space. Generally MRA is performed by recursively splitting a signal $s(t)$ into a *blurred* and a *detailed* portion. This proceeding is repeated until a set of wavelets with an orthonormal basis and satisfying details are constructed (Kaiser, 2011).

According to Shima and Nakayama (2010) and Gao and Yan (2011), a MRA of the vector space $L^2(\mathbb{R})$ consists of a set of closed subspaces V_j (see Figure 5.3), where $j \in \mathbb{Z}$, under the following conditions:

1. **Monotonicity**: The subspace V_j holds the following constitution: $\{0\} \subset \cdots \subset V_{-3} \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset V_3 \cdots L^2(\mathbb{R})$.
2. **Completeness**: All subspaces form a complete $L^2(\mathbb{R})$ and therefore the intersection of the different subspaces $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$ and the union operation $\bigcup_{j\in\mathbb{Z}} V_j = L^2(\mathbb{R})$.
3. **Dilation and translation regularity**: The signal $s(t)$ is only an element of V_j: $s(t) \in V_j$ if and only if $s(2t) \in V_{j+1}$.
4. **Existence of an orthonormal basis**: There exists a function $\phi(t) \in V_0$ where the set $\{\phi(t-j)\}$ describes an orthonormal basis for V_0.

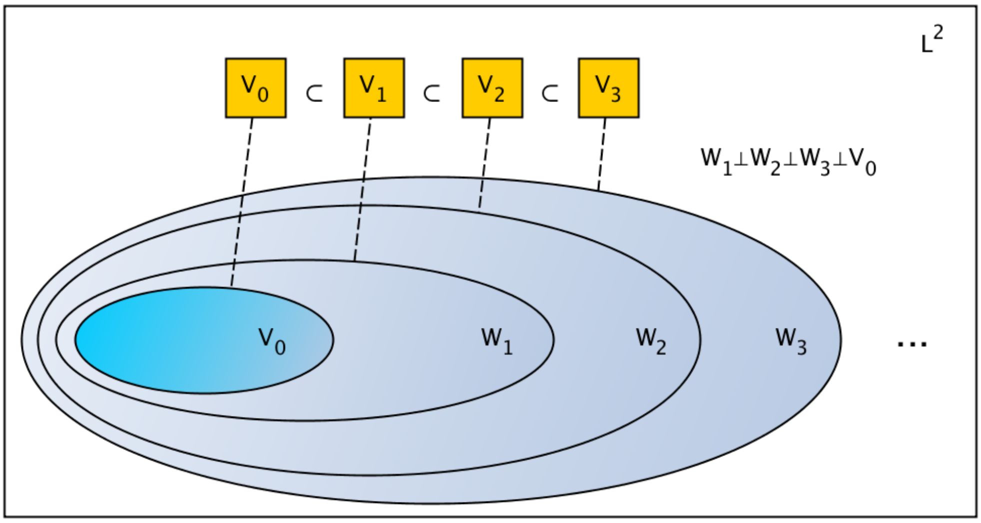


Figure 5.3: Relationship between wavelet subspaces where $V_1 = V_0 + W_0, V_2 = V_1 + W_1$ and so forth (Gao and Yan, 2011, p. 54).

5.4.1 Multi-Resolution Decomposition and Reconstruction

With the multi-resolution decomposition, Mallat (2009) proposes a recursive algorithm to separate a signal $s(t)$ into high-frequency and low-frequency components (subbands) at different scales. The high-frequency components of $s(t)$ are represented by the details and the low-frequency components are described by the approximations. Every detail d_j and approximation a_j within a specific scale j, derives from the previous approximation a_{j-1}. The first low-frequency subband holds the most information and is therefore again separated into details and approximations. The decomposition of the several low-frequency subbands can be carried out to a level j, to create a dyadic tree structure which contributes to the performance of the DWT.

Figure 5.4 illustrates the proceeding of multi-resolution decomposition and reconstruction of the signal $s(t)$. The signal is separated into high-frequency and low-frequency subbands by applying the high-pass g and the low-pass h filter. The coefficients from filtering are then decimated by two, which means every second data point is discarded. The reconstruction is performed by inserting zeros between the samples a_{j-1} and d_{j-1}, applying the inverse filters $g^{'}$ and $h^{'}$ to the coefficients and adding the outputs (Mallat, 2009).

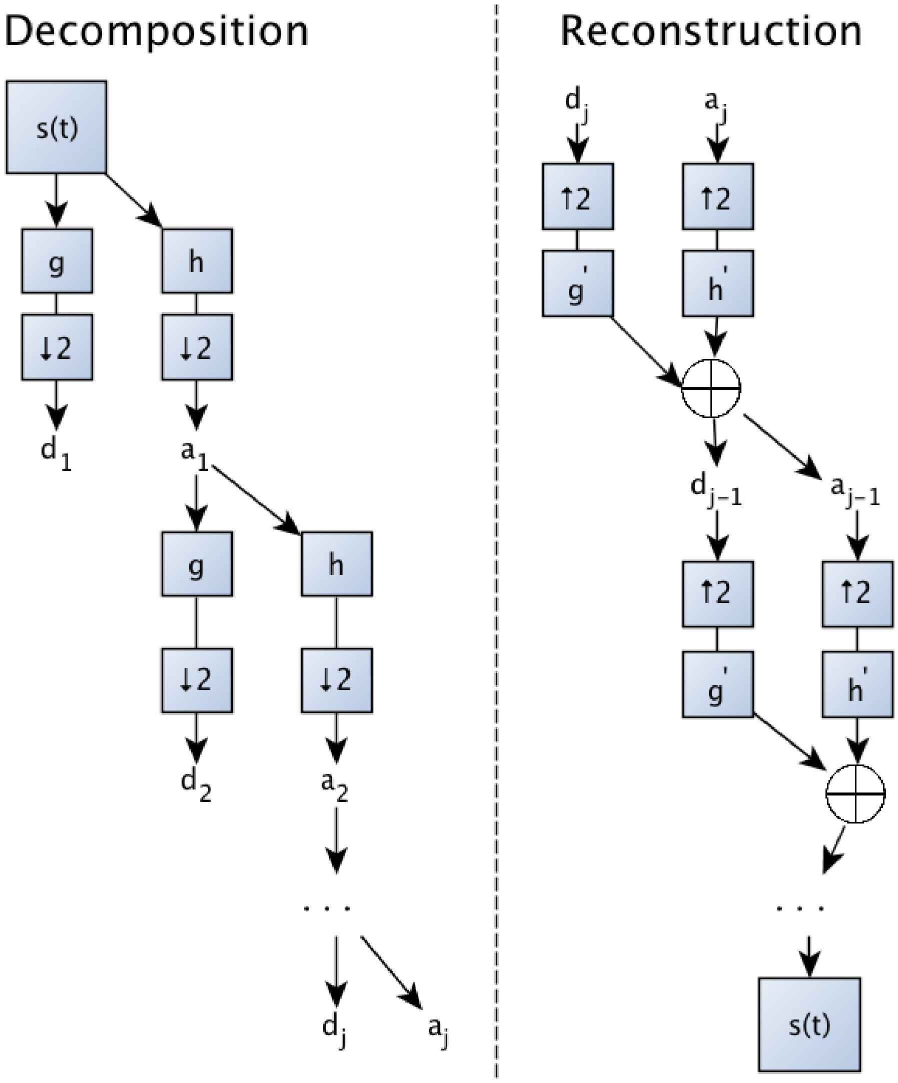


Figure 5.4: Proceeding of multi-resolution decomposition (left) and reconstruction (right) of the signal $s(t)$ (Mallat, 2009, p. 300).

In Figure 5.5, an example of a five-level multi-resolution decomposition of the signal $s(t)$ is described.

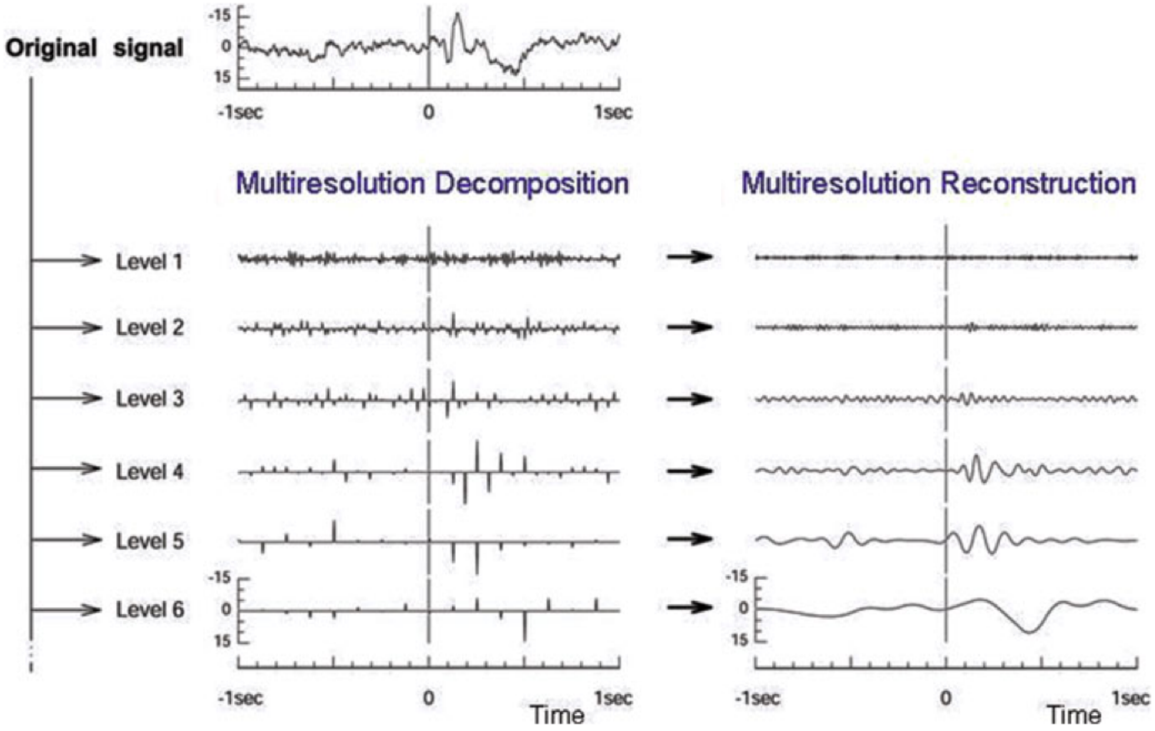


Figure 5.5: A five-level multi-resolution decomposition and reconstruction of the signal $s(t)$. The left part shows the wavelet coefficients of the signal $s(t)$ and the right part shows the several reconstructed waveforms for each scale (Freeman and Quiroga, 2013, p. 55).

6 Methodology

Reported in this chapter are the procedures used to solve the problem described in the introducing chapters. The methodological realization and the procedures used to provide an evaluation and scientific justification of the results and findings are discussed within this chapter.

6.1 Phase I: Planning

During the planning phase, operational definitions were established for this research. The foundation for these definitions are based on a review of literature, including works representing industrial remedies and standards described throughout previous chapters. Definitions were established for: the data representation format, selective encryption approaches, the encryption algorithm, the evaluation metrics and the conducted validations. Figure 6.1 servers as a structural diagram for the methodology of this research.

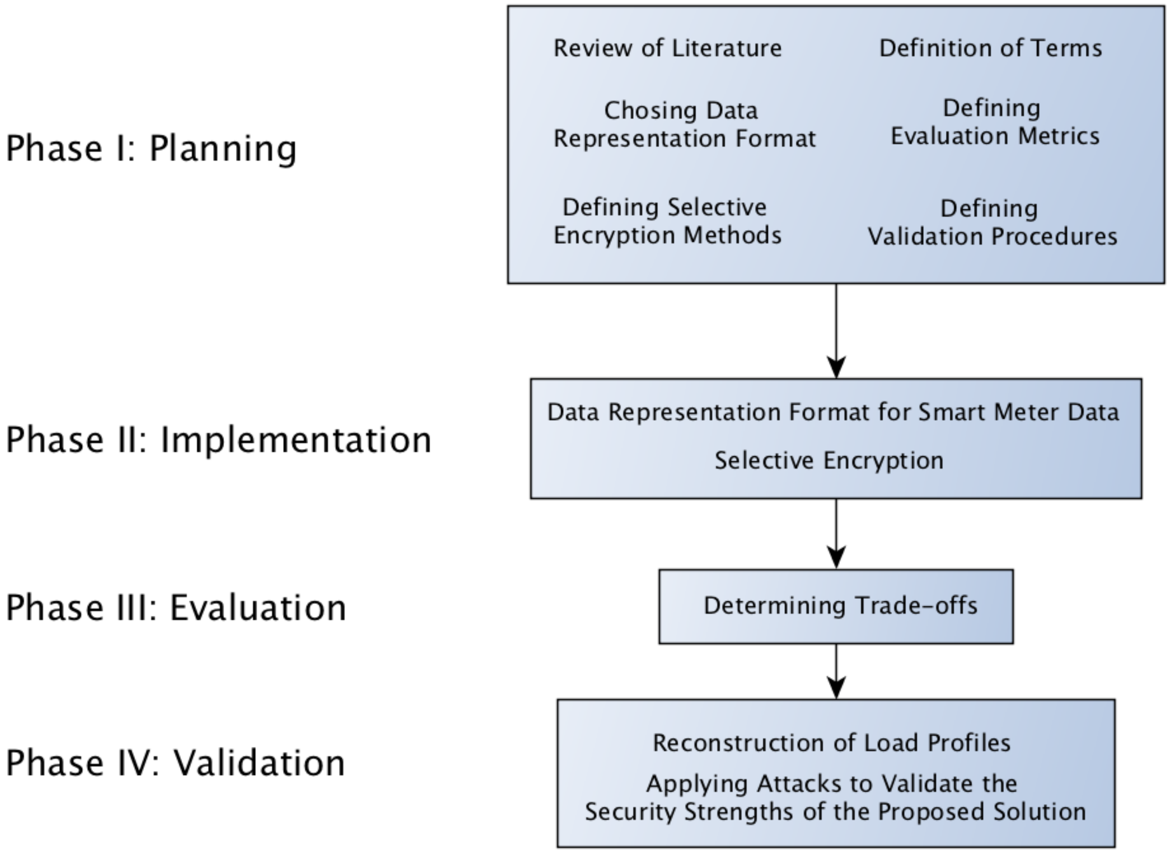

Phase I: Planning
Review of Literature
Definition of Terms
Chosing Data Representation Format
Defining Evaluation Metrics
Defining Selective Encryption Methods
Defining Validation Procedures
Phase II: Implementation
Data Representation Format for Smart Meter Data
Selective Encryption
Phase III: Evaluation
Determining Trade-offs
Phase IV: Validation
Reconstruction of Load Profiles
Applying Attacks to Validate the Security Strengths of the Proposed Solution

Figure 6.1: A flowchart for describing the methodology used in this research

6.2 Phase II: Implementation

During the implementation phase the proposed technologies and approaches from the previous sections are realized. The smart meter data from the CSV file format is analyzed within two different data representation domains: the CSV file format and the wavelet multi-resolution analysis. The selective encryption methods are performed independently for both of the data representation domains. The specific design decisions for the implementation are discussed in chapter 7.

6.3 Phase III: Evaluation

To provide an quantitative evaluation of the results, two evaluation metrics are defined.

6.3.1 Peak Signal-to-Noise Ratio (PSNR)

In signal processing the term peak signal-to-noise ration (PSNR) is generally used to determine the ratio between the maximum possible power of a signal (referred as valuation $valuation_s$ of a signal s) and the power of corrupting noise that affects the fidelity of the signal representation. With the PSNR as quality measure, the similarity between two signals, for example the unencrypted and the selectively encrypted smart meter data, can be determined. Usually the PSNR is expressed on a logarithmic decibel scale (Kotevski and Mitrevski, 2010).

For example, the PSNR between the unencrypted signal *orig* and the selective encrypted signal *enc* is calculated by (Mallat, 2009) and (Wang and Bovik, 2009):

$$PSNR(orig, enc) = 10 \cdot \log_{10} \frac{valuation_s}{MSE(orig, enc)} [dB] \tag{6.1}$$

The mean-squared error (MSE), as part of equation 6.1, provides a quantitive score to describe the degree of similarity between two signals (Wang and Bovik, 2009).

The MSE is defined by (Wang and Bovik, 2009):

$$MSE(orig, enc) = \frac{1}{N} \sum_{i=1}^{N} (orig_i - enc_i)^2 \tag{6.2}$$

As discussed by Wang and Bovik (2009), the use of the MSE as signal fidelity measure has the following advantages:

- **Simplicity**: The computation of MSE is performed parameter free and only by one multiply and two additions per sample.
- **Constant and direct interpretations of similarity**: All norms of two signals are described by non-negative values. Furthermore, the difference between two

signals equals zero if and only if all corresponding samples of the signals are equal.

- **Clear physical meaning**: Even after performing a transform to a signal, the energy of a signal distortion in the transform domain is still the same as in the signal domain.
- **Excellent metric for optimization**: With its properties of convexity, symmetry and differentiability, MSE represents an excellent metric in the context of optimization.
- **Trust in convention**: Historically the MSE has been used for a wide variety of signal processing applications. E.g., in signal compression, restoration, denoting, reconstruction, etc.

According to Kotevski and Mitrevski (2010), theoretically the highest PSNR would be 100 dB, but in reality the estimated value for image processing lies between 30 dB and 40 dB.

6.3.2 Similarity

The second evaluation metric used within this research is referred as similarity. The similarity between the two signals s_o and s_e, that have the same amount of elements N, is calculated by comparing each element of the original, unencrypted signal s_o with the corresponding element of the selectively encrypted signal s_e.
The similarity calculation is defined by:

$$Similarity(s_o, s_e) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \text{where} \quad x = \begin{cases} 1 & \text{for} \quad |s_{oi} - s_{ei}| < 0.01, \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

The similarity returns the amount of similar elements and therefore defines how different two signals are on a percentage scale. This evaluation metric is used in addition to the PSNR as quantitative measure to determine how many meter readings can still be retrieved after performing the selective encryption method.

6.3.3 Computational Effort

The computational effort is simply defined by measuring the computation time (in milliseconds ms) using different amounts of encryption with the several selective encryption approaches. As these results highly depend on the hardware configuration used for the evaluation, the evaluation is carried out on two diverse hardware environments.

6.4 Phase IV: Validation

The validation of the results and findings within this thesis has been defined as a systematic examination of the results from the evaluation process. This examination is carried out on a test data set of smart meter data described in section 3.3. To validate the proposed solutions for securing the smart meter data and defining reliable trade-offs between security and performance, the perspective of an attacker is covered. Therefore possible reconstructions of load profiles due to the secured smart meter data are analyzed. As an attacker probably could figure out easily which portions of the data are ciphered, these data could be interpolated. Possible attacks and how to avoid them is analyzed to underline the security strengths of the proposed solutions.

7 Implementation

The following section provides details about the implementation of the proposed selective encryption methods for multi-resolution smart meter data. The implementation is based on the introduced AES encryption method. For multi-resolution representation of the smart meter data, two appendages are pursued: firstly, the idea of the generic selective encryption method from Lundin and Lindskog (2010) and secondly, the use of wavelet analysis proposed by Engel (2011) and Engel (2013) is adapted to serve as a basis for data representation. For both strategies, the selective encryption methods are applied, which means portions of the data are encrypted.
This chapter focuses exclusively on design decisions proposed within this research and the implementation part. Results from the identified security trade-off and detailed evaluations are covered in chapter 8.

7.1 Development Environment

The software is developed using the integrated development environment (IDE) Eclipse Indigo Java Enterprise Edition for developers (version: Indigo Service Release 2) in combination with the Java Standard Edition Development Kit (JDK) version 1.6.0_45. As development and testing system, a MacBook Air Mid 2011 with the following hard- and software specifications is used:

- **Processor:** 1.7GHz dual-core Intel Core i5 with 3MB L3 cache
- **Memory:** 4GB of 1333 MHz DDR3
- **Graphics:** Intel HD Graphics 300 with 384MB memory
- **Storage:** Apple SSD SM256C with 256GB flash storage
- **Operating System:** Mac OS X 10.7.5 Lion
- **Java:** Java Standard Edition Development Kit (JDK) 1.6.0_45

7.2 Smart Meter Data

The smart meter data used for the implementation originate from the EasyMeter Q3D[10] model, a smart meter from the German manufacturer EasyMeter GmbH. The EasyMeter Q3D stores the meter readings in a CSV file on its flash memory. Each line of the CSV file represents the current electricity consumption with its date and time stamp for the corresponding reading. The meter readings cover six months and reach from September 3^{rd} 2012 until March 3^{rd} 2013. It is assumed that there are no power

[10] `http://www.easymeter.com/fileadmin/bilder/downloads/100125_Q3D_Produktblatt.pdf`, Accessed: 3.19.2013

outages or other faults that influence the creation of the metering data. Thus, the used smart meter data contain 96 readings per day and 17,472 readings for six months.

7.3 Selective Encryption Based on Different Data Representations

The default granularity of the used smart meter data is described by a 15 minute interval. This means there are 96 values per day and 17,472 values for six months. In terms of conditional access, like described in Efthymiou and Kajogridis (2010), the granularity of a load profile may be set up by encrypting certain readings to enable a multi-resolution representation. However, the approach of conditional access provides an efficient approach to secure customers' privacy, but does not deal with the issue of the high computational effort due to encryption. The idea of selective encryption addresses this issue of high computational effort by reducing the amount of data to encrypt. The proposed selective encryption methods are therefore more adequate than conditional access, talking about performance.

The consumption information from the smart meter data is separated by day and stored within a data structure which contains 96 values per day. This data structure is chosen to make both data representation formats comparable.

7.3.1 Selective Encryption Based on CSV

One way to provide selective encryption like proposed by Lundin and Lindskog (2010), is the use of a vector that identifies which part of the data is encrypted. The idea of this vector is adapted to be used for the selective encryption of smart meter data. For each encryption level ENC-Level a unique vector $\vec{v_{ei}}$, where $0 \leq i \leq 96$, is used. i defines the corresponding ENC-Level that stands for the recent amount of plain- and ciphertext. With 96 readings per day the ENC-Level reaches from 0 to 96, where 96 means all readings are encrypted. The vector $\vec{v_{ei}}$ keeps control over the encrypted and unencrypted smart meter data portions and splits them into plain- and ciphertext. $\vec{v_{ei}}$ is used for the selective encryption, as well as the decryption procedure like described in Figure 7.1. With this vector, the conventional underlying CSV file structure can be used.

The idea behind the implemented selective encryption method is to split the elements of the data structure into equivalent portions depending on the ENC-Level. This means the encryption is not only performed on the first elements of the data structure, but evenly spread. E.g., with ENC-Level 4: the first, twenty-fifth, fifth and seventy-fifth element is encrypted. With ENC-Level 24: every fourth element is encrypted, and so forth.

This approach may decrease the chances for successfully reconstructing the load profile for an attacker. The opportunities for reconstruction decreases linearly with the increase of encrypted readings. This approaches indicates, that almost every reading needs to be encrypted to secure a user's privacy. This assumption is evaluated later in chapter 8.

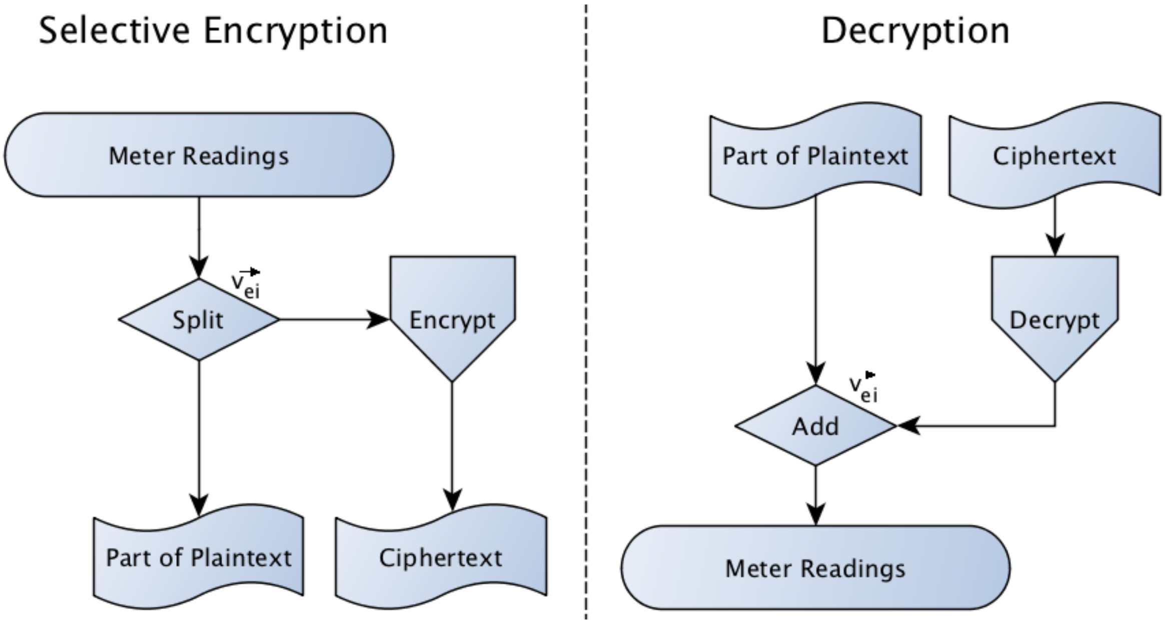


Figure 7.1: Selective encryption based on CSV data representation and ENC-Level vector $\vec{v_{ei}}$

7.3.2 Selective Encryption Based on Wavelets

Recalling the idea of the wavelet multi-resolution analysis from section 5.4, the multi-resolution smart meter data is represented by a number of vector spaces $\{V_0, V_1, V_2, \ldots, V_j\}$. Each vector space contains another vector space with a higher resolution, until the whole smart meter data is retrieved by the ambient space V_j.
In Engel (2011) and citeEngel2013, the first elements of the data structure contain the most significant information from the wavelet multi-resolution analysis. With the increase of the element number from the data structure, the relevance of the element decreases. This circumstance avails the ability to reduce the computational effort by only securing the most significant elements. It is unnecessary to encrypt the least significant elements, as they may not provide sufficient information for an attacker to retrieve the load profile. With this approach, computation effort is reduced.
The decomposition of the data during the MRA, where the information is split into high- and low-subbands, is carried out to a maximum level of five. This limitation comes from the 96 values per day. As the data (96 values per day) need to be a multiple of two, the maximum decomposition that can be carried out during the MRA is a decomposition level $decomp_{lvl}$ of five (see Figure 7.2).

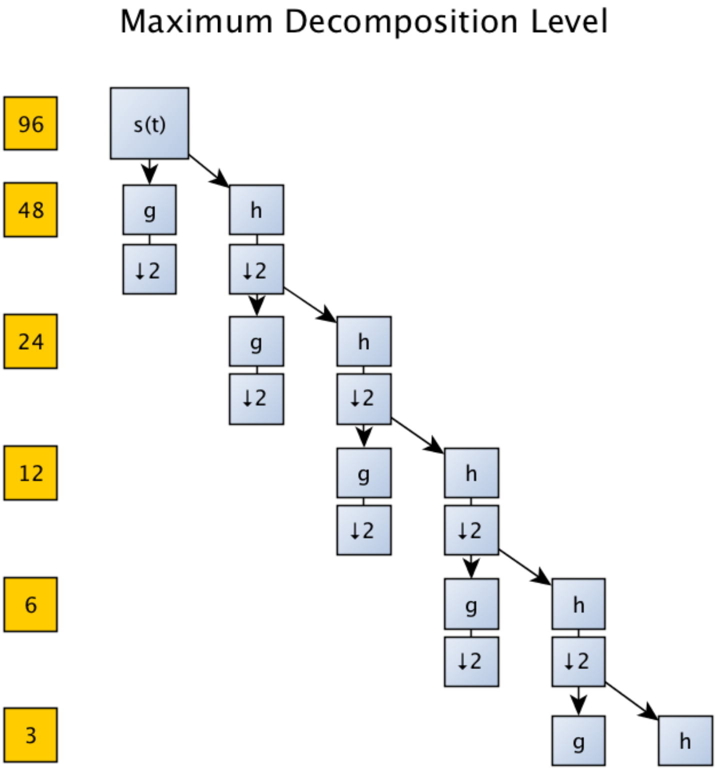


Figure 7.2: Maximum decomposition level $decomp_{lvl}$ with 96 readings per day

8 Results and Findings

The findings of the developed and evaluated selective encryption methods are outlined within this chapter. The results are solely based on the proposed implementation from section 7.

8.1 Evaluation Environment

For the evaluation process, two evaluation units with different hardware and software configurations are utilized. The first evaluation unit is a MacBook Air Mid 2011 (see section 7.1 for more details). The second unit is the single-board computer Raspberry Pi in revision B with the following hard- and software configurations:

- **Processor:** Broadcom BCM2835 ARMv6 ARM1176JZF-S with 700MHz
- **Memory:** 512MB SDRAM
- **Graphics:** Broadcom VideoCore 4 shared with the BCM2835 processor
- **Storage:** Transcend SDHC Class 4 memory card with 4GB storage capacity
- **Operating System:** Soft-float and Hard-float Debian Linux called *Raspbian wheezy* and *Soft-float Debian wheezy*[11]
- **Java:** Java Standard Edition Development Kit (JDK) 1.7_10 for Linux ARM v6/7 Soft-float Application Binary Interface (ABI) and the Java Development Kit (JDK) 1.8 Early Access with JavaFX for ARM[12] Hard-float ABI.

As suggested by the Oracle JDK for the ARMv6/7 processor type, the slower performing software floating point (Soft-float) Debian distribution is evaluated. The actual decision is made to avoid being restricted to only run java applications, which could lead to inoperability on the Raspberry Pi. This comes from the way the Oracle Java virtual machine (JVM) for the ARM architecture operates on floating point values: they are passed into integer registers. This means that with the JDK, the java source code can be compiled directly on the Raspberry Pi to evade miscalculations, which would have a devastating effect on cryptography. The second evaluation is carried out using JDK 8 with Hard-float ABI.

[11] `http://www.raspberrypi.org/downloads`, Accessed: 4.23.2013

[12] `http://jdk8.java.net/fxarmpreview/javafx-arm-developer-preview.html`, Accessed: 5.5.2013

8.2 Critical Point (CP) for the Trade-off

The aim of this research is to define a reliable trade-off between performance and security with the use of selective encryption methods. This trade-off is implied as a decision to be made with full comprehension of both: the upside and downside from the selected amount of encrypted data. This amount is referred as "critical point" and can be defined as a borderline for the required amount for encryption. If a smaller amount of data is encrypted than defined by the "critical point", the security strength is highly questionable. If more data is encrypted, resources are wasted. The "critical point" therefore specifies the least amount for encryption, where the original smart meter data and encrypted smart meter data are completely independent.

8.3 Results for the CSV as Data Representation

The following section provides the results and findings for the use of the CSV file format as data representation. The results are described by the categories: PSNR (see section 8.3.1), similarity (see section 8.3.2), computational effort (see section 8.3.3) and attacks against the selective encryption method (see section 8.3.4). The results originate from the comparison between the original, unencrypted data (original smart meter data) and the selective encrypted smart meter data.
For evaluation, the use of the CSV file format means, that all meter readings are copied to the programs internal data structure without any alteration. This is to mention, as each element of the programs data structure represents the corresponding plain meter reading. With the granularity of 15 minutes, there are 96 readings and therefore 96 elements in the data structure per day. For example, the element with the number 37 represents the electricity consumption at 8:00 a.m. The element with the number 41 represents the consumption at 10:00 a.m. and so forth.

8.3.1 PSNR

The assumption from section 7.3.1, where it was presumed that almost every reading needs to be encrypted to secure a user's privacy, can be confirmed. In Figure 8.1, the decrease of the PSNR is identified with the increase of encrypted readings from the smart meter data, is shown. The PSNR reaches the "critical point" (CP) if, and only if the whole data is encrypted. Using the PSNR, the CP can be defined as 6.614 dB. With the increase of the amount of encrypted data, the reconstruction decreases and security increases. To provide most possible security, this means there are no shortcuts for the required amount of encryption: all readings need to be encrypted.

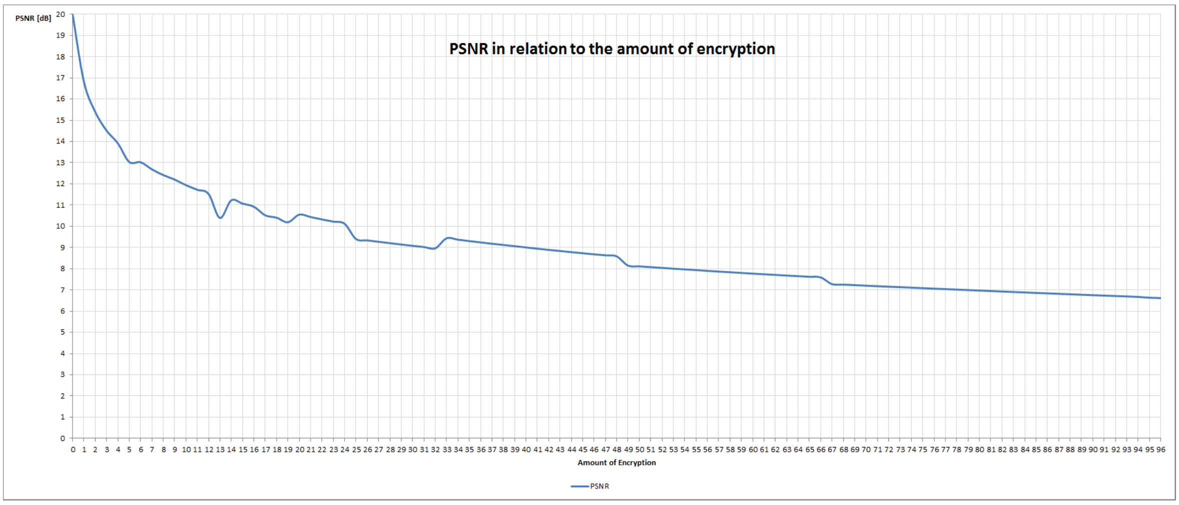


Figure 8.1: PSNR decreases almost linearly using CSV

8.3.2 Similarity

The second evaluation metric, the similarity between original and encrypted smart meter data (see Figure 8.2), underlines the results from PSNR: security increases linearly with the increase of encrypted smart meter data. Therefore, reliable security for the smart meter data is only provided, if all data is encrypted.

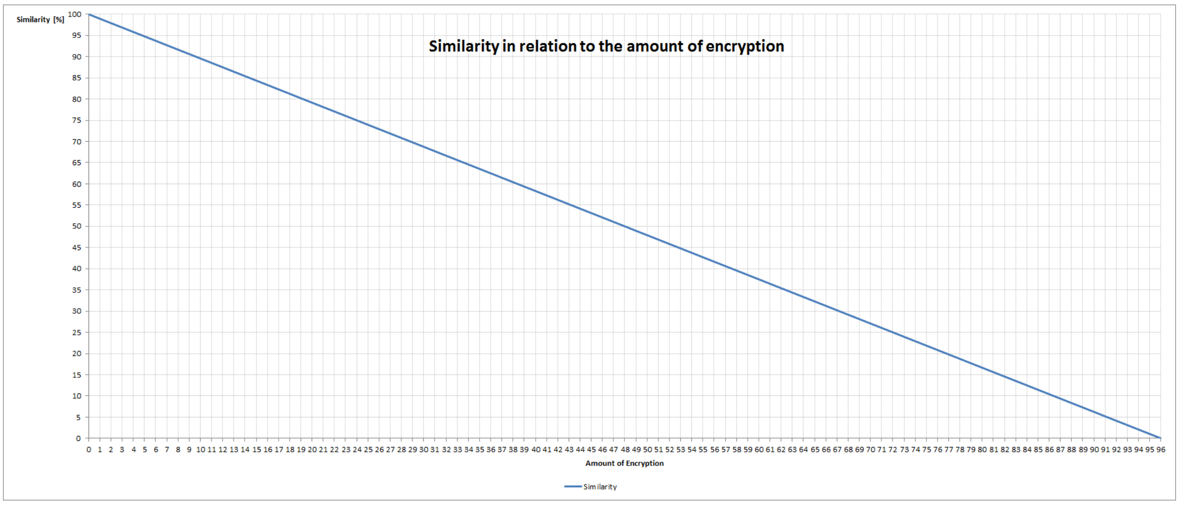


Figure 8.2: Security increases linearly with the increase of encrypted elements

8.3.3 Computational Effort

The computational effort is established by determining the required time for encryption. Figure 8.3 illustrates the computational effort for the several encryption levels on the three evaluation environments. The PC environment performes the fastest processing. With the embedded system, the Raspberry Pi, computation takes approximately eight times longer than with the PC. Contrary to the expectations that with Hard-float ABI processing is carried out more efficient than with Soft-float ABI, the results show

that there is no performance increase with Hard-float ABI. Quite the contrary can be discovered: the use of Hard-float ABI causes a slight decrease in performance due to higher requirements of the operation system with Hard-float ABI.
Related to the results from section 8.3.1 and 8.3.2, the CP is defined by the encryption level of 96, which means the entire smart meter data needs to be encrypted. In Table 8.1, a comparison between the three evaluation environments with the appropriate CP are shown.

Evaluation environment	CP	Computation time
PC	96	136ms
Raspberry Pi Soft-float	96	1024ms
Raspberry Pi Hard-float	96	1099ms

Table 8.1: Computational effort using CSV data representation

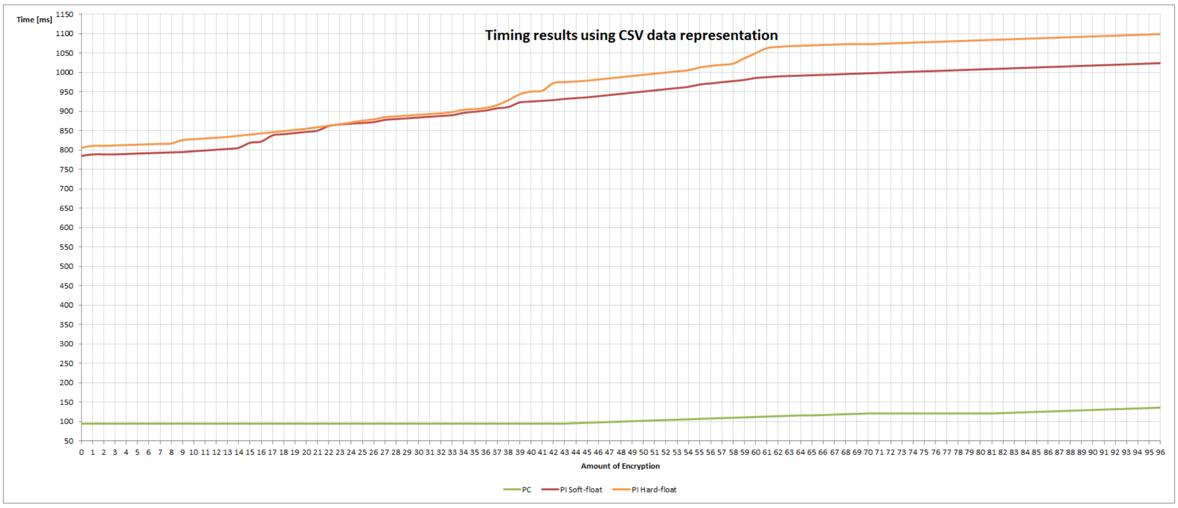


Figure 8.3: Timing results from the three evaluation environments using CSV data representation

8.3.4 Attacks Against the Selective Encryption Method

One attack against the proposed selective encryption approach is the reconstruction of load profiles. This kind of attack is made possible, as firstly, an attacker is able to interpret the plain data portions of the smart meter data and secondly, therefore is able to interpolate the encrypted data portions by determining the averages between the neighbor readings. Like described in Figure 8.4 and 8.5, the attacker is able to create reconstructed load profiles from the selectively encrypted smart meter data. The behavioral insights from the reconstructed load profiles are almost equal to the load profiles from the original smart meter data. There are only little deviations (see Table 8.2) due to building the average of the neighbor readings.

Encryption level	Maximum deviation [kW]	Average deviation [kW]
48	0.118	0.003
64	0.319	0.013

Table 8.2: Deviation between original and reconstructed meter readings

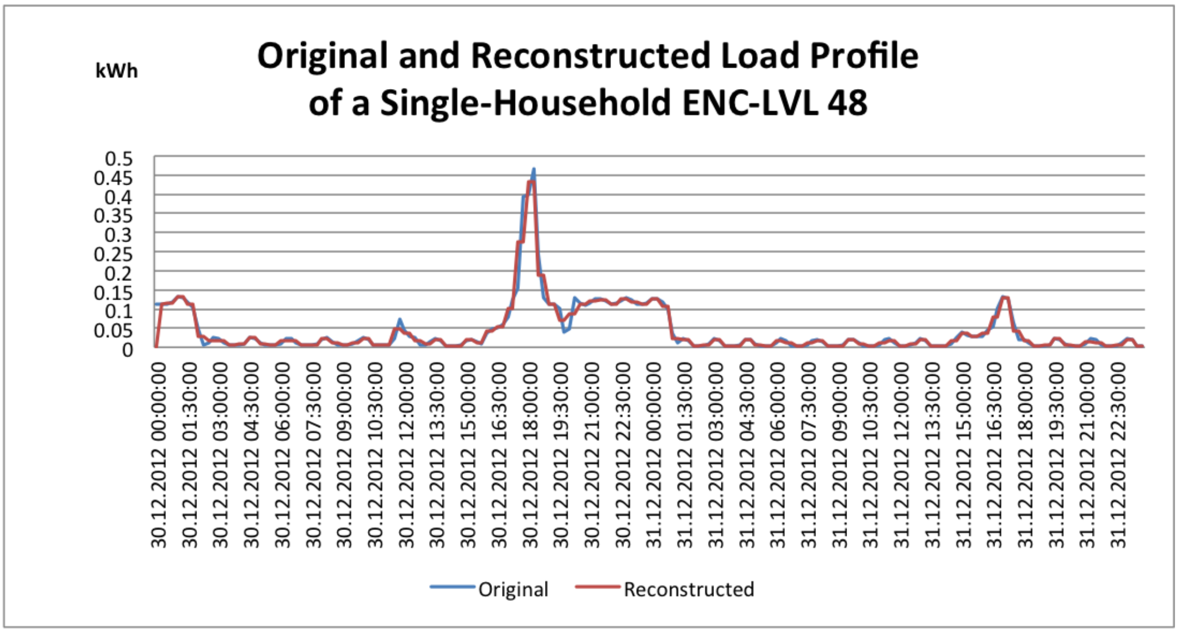


Figure 8.4: Reconstructed load profile using CSV and encryption level 48

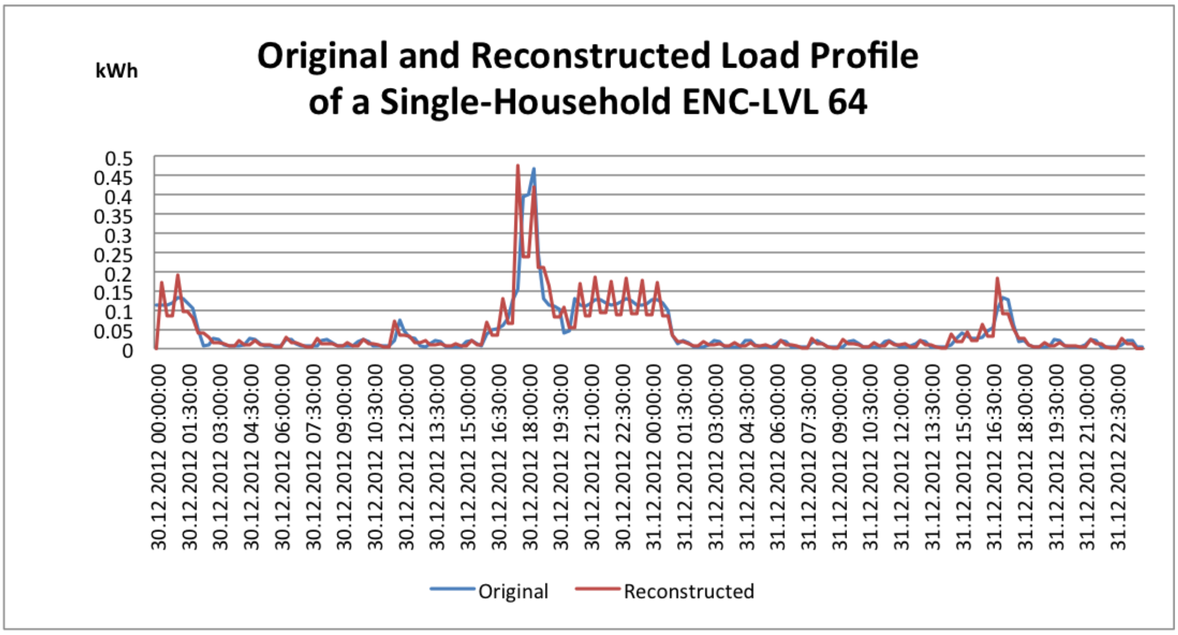


Figure 8.5: Reconstructed load profile using CSV and encryption level 64

8.4 Results for the MRA as Data Representation

Within this section the results and findings from applying selective encryption with the use of wavelet multi-resolution analysis (MRA) are reported. As seen in section 8.3, the results from the evaluations are described by the categories: PSNR (see section 8.4.1), similarity (see section 8.4.2), computational effort (see section 8.4.3) and attacks against the selective encryption method (see section 8.4.4). During the implementation phase the MRA was conducted with the decomposition levels ($decomp_{lvl}$) of 1, 2, 3, 4 and 5. Therefore, the results for the several decomposition levels are analyzed in contrast to each other.

8.4.1 PSNR

With the use of the MRA as data representation, an abrupt decrease of the PSNR is identified as data is encrypted. Depending on the used decomposition level, the CPs are defined between 5.918 and 7.031 dB. In Table 8.3, a decrease of the CP is followed by an increase. The floating points during MRA are rounded differently for each decomposition level such that the PSNR results diversify. Considering the PSNR (see Figure 8.6) this means, with the MRA selective encryption can be performed efficiently: security is still provided while major parts of the smart meter data remain in plaintext.

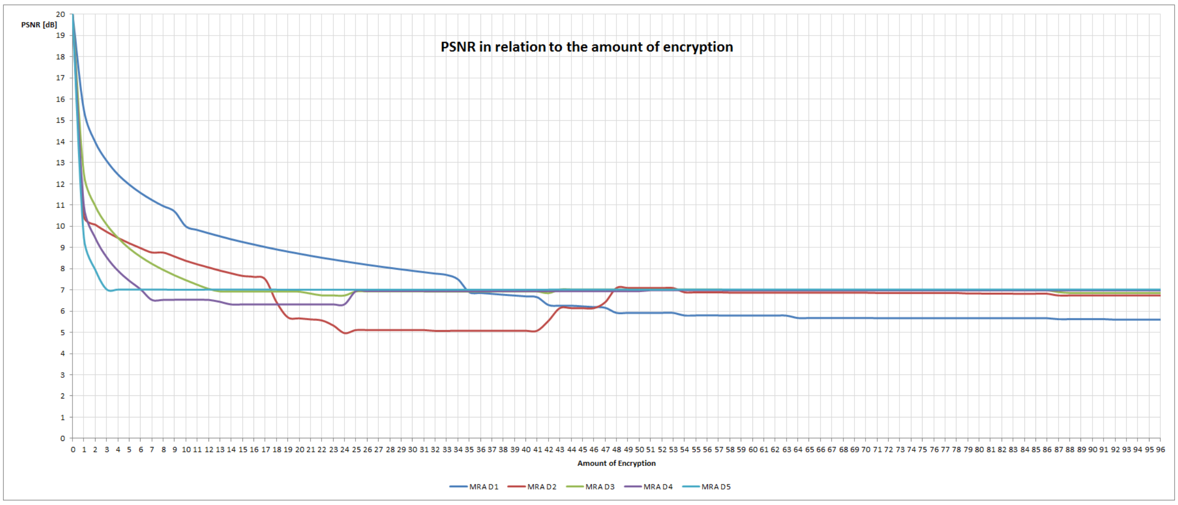


Figure 8.6: PSNR decreases abrupt applying selective encryption with use of MRA

decomp$_{\text{lvl}}$	**CP**	**PSNR** [dB]
1	48	5.918
2	24	4.965
3	12	7.052
4	6	7.031
5	3	7.031

Table 8.3: CPs for different $decomp_{lvl}$ using MRA

8.4.2 Similarity

By means of the similarity between original and encrypted smart meter data (see Figure 8.7), the shifting of the CP between the several decomposition levels is shown. With decomposition level one, half of the smart meter data needs to be encrypted to provide sufficient security. With decomposition level two, only a fourth of the smart meter data requires encryption. For every increase of the decomposition level during the MRA, the amount for encryption is cut in half.

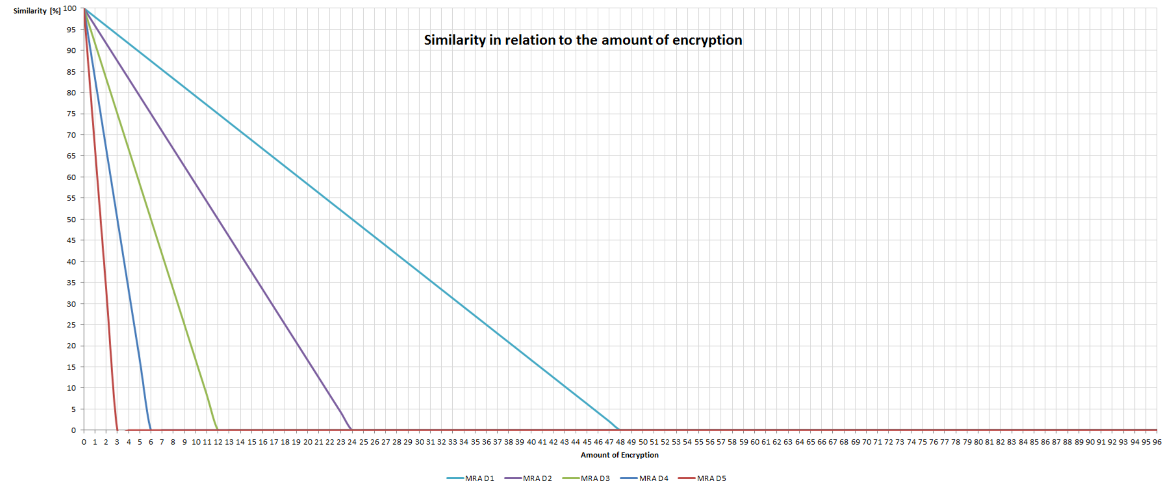


Figure 8.7: The shifting of the CP with respect to the decomposition level

8.4.3 Computational Effort

In respect to section 8.4.1 and 8.4.2, the established CPs highly depend on the decomposition level $decomp_{lvl}$. Table 8.4 provides a comparison between the three evaluation environments, the $decomp_{lvl}$, CPs and the timing results to identify the most efficient $decomp_{lvl}$ for the MRA. For all three evaluation environments the optimum $decomp_{lvl}$ is defined by five. Therefore, Figure 8.3 uses $decomp_{lvl}$ of five to compare the computation efforts for the evaluation environments. The fastest processing, as already proofed by section 8.3.3, is achieved using the PC environment. Like described in Figure 8.8, the computation takes approximately eight times longer on the Raspberry Pi. Using the Hard-float ABI, the initialization process is established with 100ms more in processing time than with Soft-float ABI. Therefore, the use of Hard-float ABI is slightly less efficient in processing than Soft-float ABI.

Evaluation environment	**$\mathbf{decomp_{lvl}}$**	**CP**	**Computation time**
PC	1	48	113ms
	2	24	99ms
	3	12	97ms
	4	6	97ms
	5	3	95ms
Raspberry Pi Soft-float	1	48	996ms
	2	24	825ms
	3	12	795ms
	4	6	739ms
	5	3	724ms
Raspberry Pi Hard-float	1	48	1127ms
	2	24	970ms
	3	12	916ms
	4	6	905ms
	5	3	855ms

Table 8.4: The most efficient $decomp_{lvl}$ is defined by five

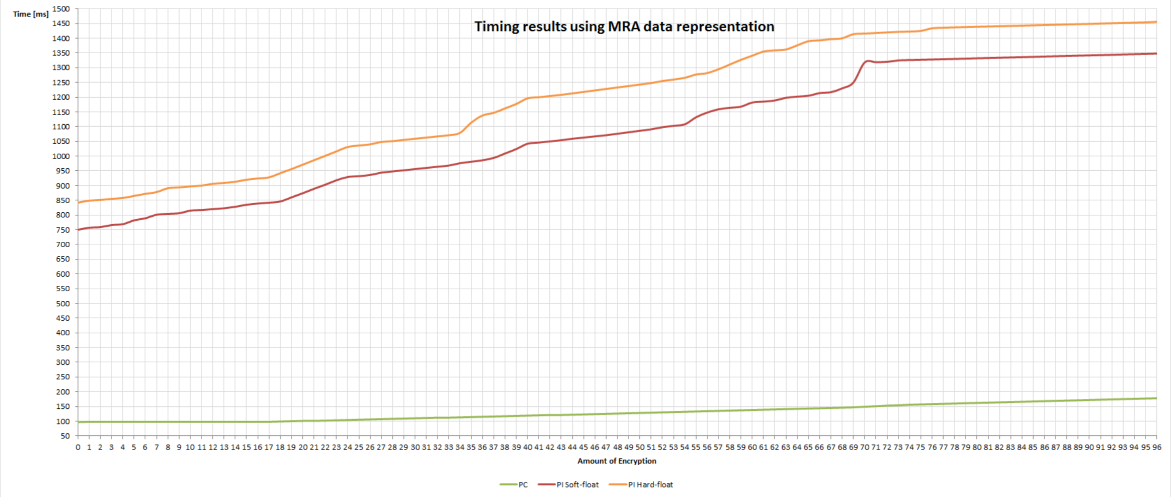


Figure 8.8: Timing results from the three evaluation environments using wavelet MRA

8.4.4 Attacks Against the Selective Encryption Method

Unlike with CSV (see section 8.3.4), the amount of reconstructable smart meter data with MRA is insufficient to create the original load profile. However, to increase the chances of reconstructing the original smart meter data, an attacker is provided with the information which parts of the data are encrypted. With this information the encrypted portions could be exchanged with zero, before the signal is reconstructed during the MRA. In Table 8.5 the average increases for the several decomposition levels are shown. As a result of the described attack, an average increase of the PSNR can be achieved if less data is encrypted than defined by the CP. However, by means of the similarity no qualitative improvements can be achieved and therefore security for the smart meter data still remains.

decomp$_{\text{lvl}}$	**Average increase of PSNR [%]**	**Similarity increase [%]**
1	10.411	0.0
2	18.332	0.0
3	1.217	0.0
4	2.947	0.0
5	0.840	0.0

Table 8.5: Increase of the PSNR for several *decomp$_{lvl}$* applying exchange attack

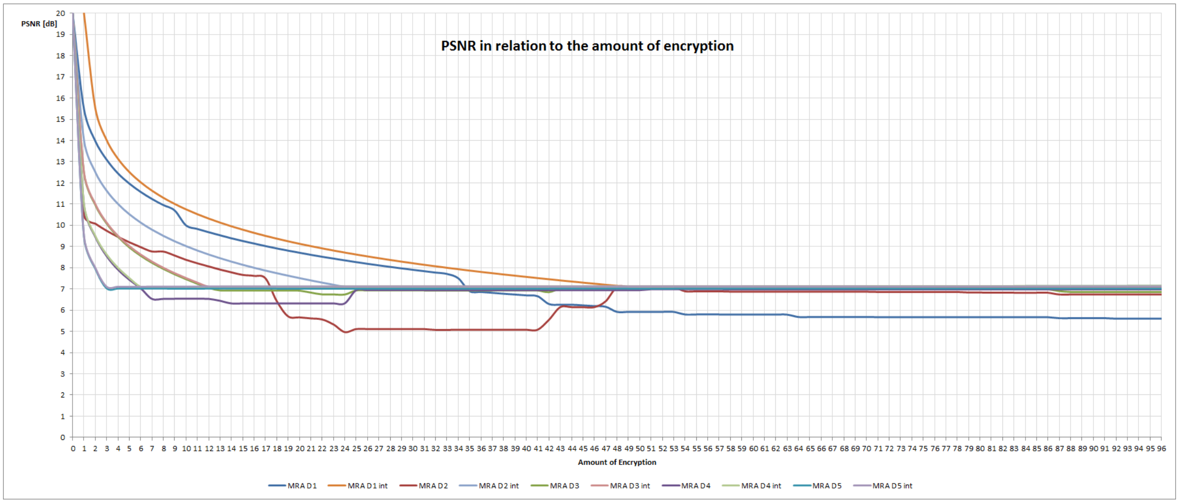


Figure 8.9: Average increase of PSNR due to exchanging ciphered portions with zero

9 Discussion

This final section of the research project provides a summary of new insights gathered from the implementation phase, as well as the evaluation phase. The gained results and findings are opposed with the expected results from the planning phase. The pending research questions are answered and a perspective for future work is given.

9.1 Summary

Within this thesis selective encryption methods were applied and analyzed to determine a trade-off in security and performance for securing smart meter data. The main objective was to investigate whether selective encryption is applicable for securing multi-resolution smart meter data or not. The use of CSV as data representation format, as already assumed within the planning phase, proved to be very inefficient as data representation format for selective encryption methods: even with the advantage of negligible small overhead from pre-computing (preparing the meter readings for computation), the encryption amount and therefore the computational effort remains very high. This comes from the fact that almost every meter reading needs to be encrypted as an attacker could interpolate the encrypted data portions. Even the approach of the generic selective encryption method, where evenly split portions are encrypted, does not contribute to security. To prevent an attacker from reconstructing the load profile, the entire smart meter data needs to be encrypted using the CSV file format.

On the contrary, the observation has shown that by means of the wavelet multi-resolution analysis (MRA), selective encryption can be utilized to reduce operating expenses. Due to the transformation during MRA the data representation of the smart meter data is altered: the most important information is represented by the first sub-band. As a result of this design, it is satisfactory to secure the first data portions. An attacker is not able to perform the inverse discrete wavelet transform (IDWT), and therefore reconstruct the load profile, without the major coefficients from the first symbols.

Considering the computational effort, 27.941% less computation time is required using selective encryption methods with MRA for PC and 29.296% for the embedded system Raspberry Pi, providing the same security strengths as encrypting the entire smart meter data.

9.2 Conclusion

While encryption contributes to security strength, it was shown that encryption highly increases computational efforts. The observation emerges that it is not imperatively required to encrypt the entire data to secure the smart meter data, but the most significant portions. With CSV as data representation all data portions are equally important and therefore selective encryption cannot be applied. The use of MRA, on the other hand, enables a meaningful usage of selective encryption methods to reduce the computational efforts, because the data is transformed into a format with most and least important data portions. Thus, only the small most significant portion is encrypted. Especially for embedded systems, like as smart meters, the reduction of computation effort is vital: the hardware components have constrains in computation power. With this thesis light-weight selective encryption methods for securing multi-resolution smart meter data are provided.

9.3 Future Work

The results and findings of this thesis point to several directions for future work:

- **Wavelet variation** - The haar wavelet proofs to be quite efficient as data representation format for selective encryption compared to CSV as data representation. But there are several other wavelets that could be analyzed in respect to their behavior and performance for selective encryption. It may be relevant to clarify whether a different wavelet reduces the computational effort or not.
- **Wrapper for low-level programming languages** - The Java Virtual Machine (JVM) carries out the entire computation for this project. Speaking of performance, additional performance increases may be achieved by low-level languages such as assembly language. This is to say, as assembly languages allow a strong correspondence between the programming language and the architecture's machine code instructions, to increase the computational performance. This may be achieved due to a wrapper where encryption and decryption is performed in assembly language and invoked in Java.
- **Protocols for increase of interoperability** - What was neglected during this research, is the data aggregation of the smart meter data. The Advanced Metering Infrastructure (AMI) defines the communication channels, as well as the encryption and decryption of data for several smart meters within a specific domain. The way the data is collected and transmitted may not matter for this thesis, but the demands for interpretation of the smart meter data does matter. Using the MRA should not affect data processing by the AMI negatively. Therefore, the next step for this work is to ensure interoperability for specific smart meters.
- **Fault analysis** - The proposed security by this work highly relies on the Advanced Encryption Standard (AES). One more aspect of a future study is the survey of the strength of the implemented AES methods. Despite some of the

inherent flaws in AES, AES is safe enough to be trusted for confidential data. However, this statement needs to be reviewed for the developed software in Java as well, as Java is known to focus on interoperability and not on nondisclosure of software.

List of Figures

List of Tables

Listings

References

Amounas Fatima and Kinani El Hassan El. ECC Encryption and Decryption with a Data Sequence. In *Applied Mathematical Sciences*, volume 6, pages 5039–5047, 2012. URL `http://www.m-hikari.com/ams/ams-2012/ams-101-104-2012/index.html`. Accessed: 04.02.2013.

Biham Eli and Dunkelman Orr. Cryptanalysis of the A5/1 GSM Stream Cipher. In *Progress in Cryptology - INDOCRYPT 2000*, volume 1977 of *Lecture Notes in Computer Science*, pages 43–51. Springer Berlin, Heidelberg, 2000. ISBN 978-3-540-41452-0.

BMWFJ . *138. Verordnung: Intelligente Messgeräte-Einführungsverordnung - IME-VO.* Bundesministers für Wirtschaft, Familie und Jugend, 2012.

Bonneau Joseph. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *2012 IEEE Symposium on Security and Privacy (SP)*, pages 538–552, 2012.

Christensen Ole. An Introduction to Wavelet Analysis. In *Functions, Spaces, and Expansions*, Applied and Numerical Harmonic Analysis, pages 159–180. Birkhäuser Boston, 2010. ISBN 978-0-8176-4979-1.

Cramer Ronald and Shoup Victor. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. In *SIAM Journal on Computing*, volume 33, pages 167–226, Philadelphia, PA, USA, January 2004. Society for Industrial and Applied Mathematics.

Daemen Joan and Rijmen Vincent. Rijndael. In van Tilborg Henk C.A. and Jajodia Sushil, editors, *Encyclopedia of Cryptography and Security*, pages 1046–1049. Springer US, 2011. ISBN 978-1-4419-5905-8.

Delfs Hans and Knebl Helmut. *Introduction to Cryptography - Principles and Applications.* Springer, Berlin, Heidelberg, second edition, 2007. ISBN 978-3-540-49244-3.

Depuru Soma Shekara Sreenadh Reddy, Wang Lingfeng, Devabhaktuni Vijay, and Gudi Nikhil. Smart Meters for Power Grid - Challenges and Issues and Advantages and Status. In *Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES*, pages 1–7. IEEE, March 2011.

Diffie Whitfield and Hellman Martin. New directions in cryptography. In *IEEE Transactions on Information Theory*, volume 22, pages 644–654, Piscataway, NJ, USA, November 1976. IEEE Press.

DOC . *State & County QuickFacts.* U.S. Department of Commerce, United States Census Bureau, June 2013. URL `http://quickfacts.census.gov/qfd/states/00000.html`. Accessed: 07.05.2013.

DOE . *Energy Independence and Security Act of 2007 (EISA).* DEPARTMENT OF ENERGY, October 2007. URL `http://www.gpo.gov/fdsys/pkg/PLAW-110publ140/pdf/PLAW-110publ140.pdf`. Public Law No 110-140, 121 STAT 1693.

Doris Elizabeth and Peterson Kim. Government Program Briefing: Smart Metering. http://www.nrel.gov/docs/fy11osti/52788.pdf, September 2011. Accessed: 03.18.2013.

Edison Electric Institute . Smart Meters and Smart Meter Systems: A Metering Industry Perspective - An EEI-AEIC-UTC White Paper. March 2011. URL http://www.eei.org/issuesandpolicy/grid-enhancements/Documents/smartmeters.pdf. Accessed: 07.31.2013.

Efthymiou Costas and Kajogridis Georgios. Smart Grid Privacy via Anonymization of Smart Metering Data. In *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 238–243, October 2010.

Engel Dominik. Conditional Access Smart Meter Privacy Based on Multi-Resolution Wavelet Analysis. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, volume 45, pages 1–5, New York, USA, 2011. ACM. ISBN 978-1-4503-0913-4.

Engel Dominik. A Wavelet-based Approach to End-user Privacy in Smart Metering. In Kempter Guido and Lofer Hans-Jürgen, editors, *Proc. 7. Forschungsforum der Österreichischen Fachhochschulen*, pages 182–184. Wissenschaftlicher Verlag Berlin, 2013. Accessed: 07.20.2013.

European Comission . Directive 2009/72/EC, July 2009. URL http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:211:0055:0093:EN:PDF. Accessed: 02.27.2013.

Fontaine Caroline. RC4. In van Tilborg Henk C.A. and Jajodia Sushil, editors, *Encyclopedia of Cryptography and Security*, pages 1031–1032. Springer US, 2011. ISBN 978-1-4419-5905-8.

Freeman Walter J. and Quiroga Rodrigo Quian. Wavelets. In *Imaging Brain Function With EEG*, pages 49–64. Springer New York, 2013. ISBN 978-1-4614-4983-6.

Gao Robert X and Yan Ruqiang. *Wavelets - Theory and Applications for Manufacturing*. Springer, Berlin, Heidelberg, 2011. ISBN 978-1-441-91545-0.

Greveler Ulrich, Justus Benjamin, and Löhr Dennis. Hintergrund und experimentelle Ergebnisse zum Thema Smart Meters und Datenschutz, 2011. URL http://www.its.fh-muenster.de/greveler/pubs/smartmeter_sep11_v06.pdf. Accessed: 02.08.2013.

Hankerson Darrel and Menezes Alfred. Elliptic curve cryptography. In van Tilborg Henk C.A. and Jajodia Sushil, editors, *Encyclopedia of Cryptography and Security*, pages 397–397. Springer US, 2011. ISBN 978-1-4419-5905-8.

IDABC . Interoperable Delivery of European eGovernment Services to public Administrations, Business and Citizens, January 2005. URL http://ec.europa.eu/idabc/en/document/3782/5584.html. Accessed: 15.03.2013.

Jao David. Elliptic Curve Cryptography. In Stavroulakis Peter and Stamp Mark, editors, *Handbook of Information and Communication Security*, pages 35–57. Springer, Berlin, Heidelberg, 2010. ISBN 978-3-642-04116-7.

Kaiser Gerald. Multiresolution Analysis. In *A Friendly Guide to Wavelets*, Modern Birkhäuser Classics, pages 139–175. Birkhäuser Boston, 2011. ISBN 978-0-8176-8110-4.

Knudsen Lars R. Block Ciphers. In van Tilborg Henk C. A. and Jajodia Sushil, editors, *Encyclopedia of Cryptography and Security*, pages 152–157. Springer US, 2011. ISBN 978-1-4419-5906-5.

Knudsen Lars R. and Robshaw Matthew. *The Block Cipher Companion*. Springer, Berlin, Heidelberg, 2011. ISBN 978-3-642-17342-4.

Kotevski Zoran and Mitrevski Pece. Experimental Comparison of PSNR and SSIM Metrics for Video Quality Estimation. In Davcev Danco and Gómez Jorge Marx, editors, *ICT Innovations 2009*, pages 357–366. Springer, Berlin, Heidelberg, 2010. ISBN 978-3-642-10780-1.

Kurosawa Kaoru. Hybrid Encryption. In van Tilborg Henk C.A. and Jajodia Sushil, editors, *Encyclopedia of Cryptography and Security*, pages 570–572. Springer US, 2011. ISBN 978-1-4419-5905-8.

Lisovich Mikhail A., Mulligan Deirdre K., and Wicker Stephen B. Inferring Personal Information from Demand-Response Systems. In *Security Privacy, IEEE*, volume 8, pages 11–20, January-February 2010.

Lu Ning, Du Pengwei, Guo Xinxin, and Greitzer Frank L. Smart Meter Data Analysis. In *Transmission and Distribution Conference and Exposition (T D),IEEE PES*, pages 1–6, May 2012.

Lundin Reine and Lindskog Stefan. Security Implications of Selective Encryption. In *Proceedings of the 6th International Workshop on Security Measurements and Metrics*, volume 9 of *MetriSec '10*, pages 1–8, New York, USA, 2010. ACM. ISBN 978-1-4503-0340-8.

Mallat Stephane. *A Wavelet Tour of Signal Processing - The Sparse Way*. Academic Press, Amsterdam, Boston, 3rd edition, 2009. ISBN 978-0-12-374370-1.

Molina-Markham Andres, Danezis George, Fu Kevin, Shenoy Prashant, and Irwin David. Designing Privacy-Preserving Smart Meters with Low-Cost Microcontrollers. In Keromytis Angelos D., editor, *Financial Cryptography and Data Security*, volume 7397 of *Lecture Notes in Computer Science*, pages 239–253. Springer, Berlin, Heidelberg, 2012. ISBN 978-3-642-32945-6.

NIST . *Announcing the ADVANCED ENCRYPTION STANDARD (AES) (FIPS PUB 197)*. National Institute of Standards and Technology, November 2001.

Paar Christof and Pelzl Jan. *Understanding Cryptography - A Textbook for Students and Practitioners*. Springer, Berlin, Heidelberg, 2010. ISBN 978-3-642-04101-3.

Pommer Andreas and Uhl Andreas. Selective encryption of wavelet-packet encoded image data: efficiency and security. volume 9, pages 279–287, Berlin, Heidelberg, 2003. Springer.

Schneier Bruce. *Applied Cryptography - Protocols, Algorithms, and Source Code in C.* Wiley, New York, 2nd edition, 1996. ISBN 0-471-12845-8.

Sen Gupta Sourav, Chattopadhyay Anupam, Sinha Koushik, Maitra Subhamoy, and Sinha Bhabani P. High Performance Hardware Implementation for RC4 Stream Cipher. In *IEEE Transactions on Computers*, volume 62, pages 730–743, 2013.

Shannon Claude Elwood. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423, October 1948. URL `http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf`. Accessed: 05.25.2013.

Shannon Claude Elwood. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28:656–715, October 1949. URL `http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf`. Accessed: 03.25.2013.

Shima Hiroyuki and Nakayama Tsuneyoshi. Wavelet transformation. In *Higher Mathematics for Physics and Engineering*, pages 449–480. Springer, Berlin, Heidelberg, 2010. ISBN 978-3-540-87863-6.

Statistics Austria . Private Households 1985 to 2011, March 2012. URL `http://www.statistik.at/web_en/statistics/population/households_families_living_arrangements/households/032309.html`. Accessed: 01.09.2013.

Vaudenay Serge. *A Classical Introduction to Cryptography - Applications for Communications Security.* Springer, Berlin, Heidelberg, 2005. ISBN 978-0-387-25464-7.

Vijayalakshmi V., Varalakshmi L.M., and Sudha G. Florence. Efficient Encryption of Intra and Inter Frames in MPEG Video. In Meghanathan Natarajan, Boumerdassi Selma, Chaki Nabendu, and Nagamalai Dhinaharan, editors, *Recent Trends in Network Security and Applications*, volume 89 of *Communications in Computer and Information Science*, pages 93–104. Springer, Berlin, Heidelberg, 2010. ISBN 978-3-642-14477-6.

Wang Zhou and Bovik Alan C. Mean Squared Error: Love it or leave it? A new look at Signal Fidelity Measures. volume 26, pages 98–117, 2009.