
Ashif Iqbal

# Cybersecurity in Smart Metering: Mitigating risks and ensuring reliability

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

23 February 2024

# Abstract


| | |
|---|---|
| Author: | Ashif Iqbal |
| Title: | Cybersecurity in Smart Metering: Mitigating risks and ensuring reliability |
| Number of Pages: | 35 + 3 pages |
| Date: | 23 February 2024 |
| Degree: | Bachelor of Engineering |
| Degree Programme: | Information Technology |
| Professional Major: | Mobile Solutions |
| Supervisors: | Harri Mäkelä, R&D Manager, Landis+Gyr<br>Toni Spännäri, Senior Lecturer, Metropolia University of Applied Sciences |


---


The use of smart metering systems is increasing very fast. The more devices are connected to the network, the more there are security threats. The thesis was commissioned by Landis+Gyr, which is a leader in the energy sector. This study dives into the cybersecurity part of the smart metering ecosystem.

The goal of this study is to present a detailed theoretical background and an example of the implementation of Transport Layer Security (TLS) in JMS, HTTP and Jetty routes for smart metering. The examples provide basic understanding of the role of backend development to improve data security for smart metering systems. Furthermore, the study explores potential threats to smart meters, strategies and best practices in cybersecurity, potential technological innovations, and future directions.

Research was conducted on the security threats to smart metering systems and their mitigation strategies. The thesis also presents a practical example of a security solution using TLS, a keypair and the Java programming language to protect the communication and secure data in smart metering ecosystems.

The study provides a comprehensive analysis of cybersecurity challenges and ways to overcome them. The practical example describes how security can be configured at a general level. Basic introduction of the smart meters, their communication protocols, and their role in the energy sector is discussed in the study. The study also discusses Landis+Gyr's contribution to smart energy and smart metering field.

This study aims to bridge the research gap between the cybersecurity aspects of smart metering, encouraging a secure and reliable environment in the field of Smart Metering as a Service.


---

## Contents

## List of Abbreviations

PLC – Power Line communication

P2P – Point to Point

Zigbee – wireless communication protocol for a small amount of data in short distances using low power digital radios

Wi-SUN – Wi-SUN (Wireless Smart Ubiquitous Networks) is a wireless communication network for use in smart meters, smart cities, and other Internet of Things (IoT) applications.

TLS – Transport Layer Security, cryptographic protocol to secure communication on a network

JETTY – HTTP web server for a Java Servlet container

# 1. Introduction

## 1.1 Background

Digital transformation is a current phenomenon, and smart metering is considered a core component of energy management systems. Smart meters offer real-time data on energy consumption, supporting more optimized energy usage and integrating different renewable energy sources. However, as smart energy continues to evolve, the connectivity and complexity of smart metering systems are increasing, exposing them to security vulnerabilities and posturing challenges to the reliability and integrity of smart metering systems. (Kohout, Lieskovan and Mlynek, 2023.)

Landis+Gyr, a global leader in utility and energy management solutions, plays a key role in deploying the smart metering infrastructure. This final year project was commissioned by the company. The collaboration and working experience with the company has given this study a unique opportunity to analyze the practical aspects of cybersecurity in smart metering and provide insights into real-life solutions and impediments.

## 1.2 Problem Statement

Cybersecurity in smart metering is a complex topic and the integration of cybersecurity measures in smart metering is necessary to protect against different events like data breaches, unauthorized access, and potential disruption to energy availability. This study discusses cybersecurity aspects of smart metering, including best practices, challenges, and risk mitigation strategies in the smart metering industry, particularly from the perspective of market leaders like Landis+Gyr. It is especially important to take cybersecurity measures in smart metering seriously because everything runs on energy these days and any breach will cause significant problems and panic. (Tweneboah-Koduah et al., 2018.)

### 1.3 Objectives

The objective of this study is to give an overview of current security models for smart metering, focusing on technologies, rules and regulations, industry standards and best security practices.

First, the objective was to study the current security practices, challenges, and security solutions of Landis+Gyr for smart metering technology. Another goal is to provide guidance for enhancing cybersecurity in smart metering, contributing to the bigger goal of managing energy better and securely with Smart Metering as a Service.

### 1.4 Scope and Limitations

The thesis will focus on the cybersecurity part of Smart Metering as a Service, with an emphasis on Landis+Gyr, which is an important company in the smart metering and utility industry. The study will explore the theoretical and practical implementation of security measures. It will also describe how to consider technology, regulations, and organizational aspects.

The study is limited by access to proprietary information, the rapidly evolving nature of technology, and the specific focus on Landis+Gyr, which means the entire industry is not represented.

## 2. Theoretical Background: Cybersecurity in Smart Metering with a Focus on Landis+Gyr

### 2.1 History and Evolution

Landis+Gyr, which was founded in 1896, has historical significance in the energy sector, and it is one of the main players in the energy sector. The company's rich history offers innovation, resilience, and adaptability. Stutz (2017) explores the

company's evolution, especially during the period of the Hungarian uprising and east-west trade from 1953-1967. His research emphasizes Landis+Gyr's adaptability and strategic planning in navigating complicated political and economic landscapes. This historical context provides a base for understanding the company's strength and its capability to adapt to the dynamic nature of the energy sector. Beyond these events, the company's continued growth in the face of global economic recessions, technological disruptions, and changing regulatory landscapes further strengthens its reputation as a determined player in the energy domain.

## 2.2 Business Models and Value Networks

In the area of smart metering, innovative and sustainable business models are the heart of success. Landis+Gyr's standing as an industry forefront is reinforced by its advanced business strategies and practices. Bonakdar, Bjelajac, and Strunz (2014) researched Landis+Gyr's approach to calculating and analyzing business models within value networks. Their research highlighted Landis+Gyr's innovative strategies, which have made it one of the leaders in utility and smart metering industries. The knowledge provided by the research emphasizes how Landis+Gyr effectively navigates the complexities of the utility and energy domains, creating value for the stakeholders with its unique business models.

## 2.3 Workflow Management and Operational Efficiency

Operational excellence is non-negotiable in industries that prioritize precision, reliability, and customer satisfaction. Landis+Gyr's unwavering commitment to these principles is manifest in its precise workflow management practices. Iten (1996) provides a detailed investigation of Landis+Gyr's workflow management practices, underlining the company's commitment to streamlined processes and effective management. The research highlights the necessity of these practices in ensuring the successful deployment and operation of smart metering solutions. For a company like Landis+Gyr, that kind of workflow management is influential in delivering secure and reliable energy management solutions.

### 2.4 Cybersecurity in Smart Metering

The confluence of the energy sector with digital technologies presents opportunities and challenges. Cybersecurity emerges as a vital concern in this relationship. Smart metering systems, with their elegant web of interconnected devices and real-time data exchanges, are interesting targets for cyber attackers. Protecting the integrity, confidentiality, and availability of these systems is crucial. Modern research points out the multifaceted nature of cyber threats targeting smart metering infrastructures. (Markopoulou, 2022.)

Landis+Gyr has a proactive approach towards cybersecurity of smart metering. They often collaborate with cybersecurity experts and strictly follow the international security standards in their meters. The company is committed to providing high level security and protection for their smart metering solutions. Those practices help the company and the smart metering industry to survive different cyber security threats.

## 3. Infrastructure and Threat Landscape

### 3.1 Infrastructure Components

#### 3.1.1 Smart Meters

Smart meters are advanced devices made to measure electric, water, heat and gas energy consumption and provide the options to read data and manage the meter remotely. They offer many functionalities that facilitate remote monitoring and management of utility services. For example, functionalities like disconnect and firmware updates can be done remotely. (Mathas et al., 2020.)
These functionalities not only streamline operations for utility companies but also give consumers detailed insights of their energy usage pattern, enabling more informed energy use decisions.

Smart meters often use protocols like Zigbee, Wi-SUN, and cellular networks to connect with other devices or the head-end system, which is important for

seamless transmission of data and executing commands. Normally point to point (P2P) devices are connected directly to the head-end system, ensuring fast data transmission without the need for an intermediary device. On the other hand, Power Line Communication (PLC) devices are connected via a Data Concentrator, a setup that aggregates data of multiple meters before sending them to the head-end-system. (Ravichandra,2023)

These communication models optimize data flow and enhance the scalability and reliability of the smart metering infrastructure. Below there is a picture of E360 LTE smart meter from Landis+Gyr.


Figure1: E360 smart meter from Landis+Gyr. (Landis+Gyr)

E360 is a point to point (P2P) device which communicates directly to head-end system.

### 3.1.2 Data Concentrators

Data concentrators play a significant role in smart metering infrastructure. They are positioned strategically on substations to serve smart meters and head-end

systems. They process and aggregate data, making sure only relevant data is sent to the head-end system. (Ravichandra, 2023)
Data concentrators function as a nerve center in smart metering ecosystems. They efficiently filter and refine data from many meters before it reaches the main system. They have adaptive processing capabilities to ensure streamlined communication.

Power Line Communication or PLC devices use data concentrators to communicate. Their main function is to combine and process data, ensuring that only appropriate information is transmitted to the head-end system, optimizing bandwidth, and reducing data redundancy (Ravichandra, 2023).

Below is a picture of DC450 Data Concentrator from Landis+Gyr.


Figure 2: DC450 data concentrator from Landis+Gyr. (Landis+Gyr)

Data Concentrators like DC450 are connected to a set of Power Line Communication (PLC) meters and work as a communication channel between PLC smart meters and the head-end system.

### 3.1.3 Communication Networks

The choice of communication networks centers on regional infrastructure and specific utility requirements. There are different options from Power Line Communication (PLC) and Radio Frequency (RF) mesh networks to cellular networks for point-to-point (P2P) devices. These communication channels are secured with multiple security layers, including VPN tunnels, and the data transmitted is encrypted to safeguard against potential breaches. (Mathas et al., 2020; Ravichandra, 2023.)

As smart metering is evolving very fast, communication networks are crucial for transmitting data efficiently and reliably. To enable seamless communication between meters and utility systems, they serve as a backbone. With the latest technologies, communication networks improve service delivery and offer customer satisfaction by timely data collection, dynamic pricing, outage management, and enhanced distribution.

### 3.1.4 Head-end Systems.

Head-end systems are used for advanced analytics to process massive amounts of data received from the smart meters. This includes usage time, outage detection, and predict maintenance and detect different events like fraud detection, data tempering and sending an alarm. They are often integrated with IT systems like billing systems and some portals. (Mathas et al., 2020.)

The head-end system acts as the center in smart metering technology. It provides real-time data about energy consumption and network performance, which facilitates optimized grid management and responsive customer service. On top of data processing, it supports setting dynamic rates, enhances energy efficiency programs, and helps the resolution of service disruptions. With proper data analysis, the head-end system can contribute to the evolution of smart metering technology. (Mathas et al., 2020.)

## 3.2 Threat Landscape

### 3.2.1 Eavesdropping

Eavesdropping, or the unauthorized interception of communications, can have grave consequences, like data breaches. It can also reveal patterns in consumer's energy usage, leading to privacy concerns. (Mathas et al., 2020.) Creating strong encryption mechanisms for data in transit can deter eavesdroppers (Ravichandra, 2023.).

Eavesdropping in the context of smart metering might happen when an attacker intercepts the communication between the meter and the head-end system. For example, if the network or data in transit is unsecure, an attacker can get information about consumption patterns. It could reveal personal information and even the attacker can know when the home is unoccupied.

### 3.2.1 Tampering

There can be diverse types of tampering. Physical tampering including meter manipulation, like using magnets to interfere with meter operation, can cause wrong reading of metering data. Malicious software can be used to change readings or disrupt operations. (Ravichandra, 2023.)

To protect against tampering, utilities should have a multifaceted approach to address physical and cyber threats. By using advanced encryption, utilizing tamper-detection technologies, and enforcing strong access controls, risks can be mitigated.

Tamper-evident seals and regular on-site inspection can prevent physical tempering of the meters. Keeping a secure boot process and firmware signing can prevent unauthorized software modifications. (Mathas et al., 2020.)

### 3.2.2 Replay Attacks

In a replay attack, attackers capture legitimate commands and replay them later to create unauthorized actions, such as server disruptions. For example, an attacker can capture a certain command and use it later to cause service disruptions. (Mathas et al., 2020.)

Replay attacks are a significant threat to the security of HTTP or JMS routes. For example, on HTTP routes, attackers can capture an authentic API call for a sensitive action, such as modifying user privilege, and replay it to achieve malicious outcomes. In JMS based messaging, replaying compromised messages could lead to unauthorized processing or server disruption.

It can be prevented by using time-stamped command and sequence numbers and making sure commands are only acted upon once. (Ravichandra, 2023.)

### 3.2.3 Spoofing

Spoofing is when attackers pretend to use a legitimate device, inject corrupted or wrong data, creating wrong billing and grid management decisions. Spoofing undermines the reliability of smart metering systems by impersonating real devices to inject wrong data. This can change energy readings and disrupt operations. (Mathas et al., 2020.)

Spoofing can be prevented by mutual authentication process, which will ensure both parties in a communication exchange are legitimate (Ravichandra, 2023). Mutual authentication is a two-way authentication where client and server both need to authenticate before the communication starts. Several types of authentication methods like keypair and biometric authentication can be used for mutual authentication process.

### 3.2.4 Denial of Service (DoS) Attacks

DoS attacks aim to overpower systems, making them unavailable. In the context of smart metering, a successful DoS attack could disrupt the flow of critical metering data, affecting billing, grid management, and more. Implementing rate-limiting, traffic filtering, and redundancy can help mitigate the impact of DoS attacks.

### 3.2.5 Vulnerabilities in Communications Protocols

There are different communication protocols which smart meters use, and they have their vulnerabilities. For example:

- Zigbee: Zigbee is vulnerable to ‘man-in-the-middle’ attacks, where communication can be intercepted and potentially altered. (Mathas et al., 2020)
- Power Lan Communication (PLC): PLC communication can be disrupted by noise interference. (Ravichandra, 2023)
- Point To Point (P2P): P2P devices are connected by cellular networks like GPRS, Edge, 3G, 4G and 5G networks. They are robust networks and susceptible to jamming attacks. (Mathas et al., 2020)

Those communication protocols are vital for smart meters and any vulnerabilities on those can make the whole system vulnerable. Using strong encryption methods like pre-configured keys, mutual authentication, device authentication and secure tunneling protocols can be used to secure communication.

## 3.3 Best Practices and Mitigation Strategies

Here is list of best practices to secure smart metering communication and to mitigate cybersecurity risks.

- Continued monitoring can detect abnormal behaviors, indicating potential security breaches. (Ravichandra, 2023)
- Segmentation of the network ensures that breaches do not spread out. (Mathas et al., 2020)
- Human errors can also cause significant security risks. Regular security training and awareness can mitigate this risk. (Ravichandra, 2023)

- Regular updates of the system software, firmware and cryptographic keys ensure that known problems are fixed, reducing potential attack vectors.
- Multi-factor authentication can add an additional layer of security, especially for remote access to the system.

The complex nature of smart metering infrastructure, combined with its significant role in modern energy management, makes it a remarkably interesting target for attackers. A robust understanding of the associated threat landscape, combined with proactive mitigation strategies, is important to ensure the security and reliability of the smart metering infrastructure. (Ravichandra, 2023.)

# 4. Strategies and Best Practices

The ever-increasing connectivity and complexity of smart metering systems have made them vulnerable to a vast amount of cyber security threats. To ensure security and reliability of the smart metering infrastructure, it is imperative to adopt robust strategies and best practices. This chapter explores the various strategies and best practices that can be deployed to safeguard smart metering systems.

## 4.1 Regular Software and Firmware Upgrades

Updating the software and firmware versions in smart metering infrastructure is a primary line of defense against cyber threats. Regular updates are important in addressing known vulnerabilities and enhancing the overall security of the system. (Markopoulou, 2022.)

By carefully addressing vulnerabilities, utility companies can prevent potential exploits and attacks. The effects of continuous monitoring of future and current threats on the smart metering system should be tested. Some processes can even be automated where possible.

## 4.2 Robust Authentication and Encryption Protocols

It is particularly important to ensure the integrity, confidentiality, and security of data in transit. Implementing robust authentication and encryption protocols can prevent eavesdroppers and prevent unauthorized access to smart meters. (Sun et al., 2021.)

By ensuring that only legitimate devices and systems can communicate with each other, utility companies can prevent many attacks like spoofing and man-in-the-middle attacks. Authentication protocols are important, such as mutual authentication on the client and server side, where client and server must authenticate themselves before the communication starts. Digital Certificates and public keys also can be used to improve security.

Encryption protocol protects the integrity of the data when it is on the network. Data should be end-to-end encrypted from origin to destination. It is very important for utilities to continuously update themselves by utilizing affective encryption and authentication techniques.

## 4.3 Periodic Security Audits and Vulnerability Assessment

To stay ahead of security threats, it is necessary to assess the security posture of the smart metering system regularly. By running security audits regularly, potential weaknesses and vulnerabilities can be identified, allowing utilities to act proactively. By realizing the potential attack vectors, utilities can implement measures to counteract them. (Kohut, Lieskovan & Mlynek, 2023)

Security audits can help identify system misconfigurations and backdated security practices. Systematical evaluation of a smart metering system can help find problems on the system before they are exploited.

## 4.4 Employee and User Education

Most of the time, security breaches occur due to human mistakes and oversight. Proper education for employees and customers about potential threats and best practices can significantly reduce risks of inappropriate security lapses (Tweneboah-Koduah & S. et al, 2018). By practicing a culture of cyber security awareness, companies can ensure that their employees and customers become the first line of defense against security attacks.

Employees must be educated about using strong passwords, how to detect phishing attempts and how to handle sensitive data. Customers should also have at least a basic level understanding of the security of the devices they use to minimize the risks of security breaches from their side.

## 4.5 Intrusion Detection and Prevention Systems

As cyber security threats are becoming increasingly sophisticated, it is necessary to have systems in place to detect and prevent malicious activities in real time. Intrusion detection and prevention systems can monitor network traffic, identify anomalies, and take necessary steps to prevent potential security breaches and protect the information (Olivares-Rojas et al., 2022). Utility companies like Landis+Gyr can respond to threats and reduce potential damage by monitoring the network actively.

## 4.6 Network Segmentation

Segmentation of the network is highly effective to protect against potential breaches. Companies can ensure that if one segment of the network is compromised, it does not spread to other parts of the network by segmenting the network (Markopoulou, 2022). Network segmentation is amazingly effective in large infrastructures like Landis+Gyr, where a breach in one segment can have a significant impact on the entire system.

Network segmentation separates different network zones, such as public and private networks. They can be separated by firewalls and other security measures to limit access to potential attackers. This approach is very important in smart metering systems because sensitive data should be protected while allowing necessary communications.

## 4.7 Backup and Disaster Recovery plan

In the event of a security breach or a system malfunction, it is necessary to have a backup and disaster recovery plan to ensure recovery after the event. Regular backup is particularly important to utilities because data loss can severely impact utility companies. On top of regular backup, a nicely planned and defined disaster recovery plan can ensure the rapid restoration of services in the aftermath of breaches. By preparing for a possible worst-case scenario with a proper backup and disaster recovery plan, companies can ensure the continuity of service and minimize disruptions. (Tweneboah-Koduah et al., 2018.)

## 4.8 Zero Trust Architecture

Implementing zero-trust architecture ensures that every device, user, and combination is authenticated and authorized. This approach runs on a principle of “never trust, always verify,” ensuring that even internal network traffic is also treated as a potential problem. Implementing zero trust can significantly improve the system by reducing the attack surface and preventing lateral movement within the network. (Markopoulou, 2022; Sun et al., 2021.)

## 4.9 Threat Intelligence Sharing

Collaboration between different smart metering companies and participating in threat intelligence sharing can provide insights into coming threats and attack patterns. By staying updated about the possible and newer security threats

utilities can enhance their security by implementing countermeasures. (Olivares-Rojas et al., 2022; Tweneboah-Koduah & S. et al, 2018.)
Threat intelligence sharing enables utility companies to collectively enhance their protection against current and future threats by sharing information about upcoming security threats, attack patterns, and security solutions. Having a collaborative approach to cybersecurity of smart metering infrastructure can benefit the whole energy industry.

## 4.10 Secure Development Lifecycle (SDLC)

Security should be ensured from the start of the development cycle. Adapting a secure development lifecycle ensures that security measures are integrated at every stage of development. Regular code reviews, security testing, and security assessments can identify and rectify potential security flaws. (Kohut, Lieskovan & Mlynek, 2023; Markopoulou, 2022.)

Belowe is a diagram of a secure software development lifecycle.

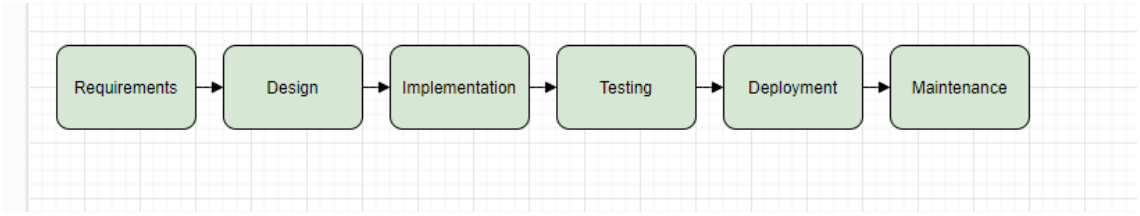


Figure 3: Diagram of software development lifecycle

The diagram describes the software development lifecycle. Implementing security from the initial phase of development, in this case from requirement phase, can reduce the costs caused and efforts required by future fixes. Being proactive in every part of the development cycle is necessary for ensuring the product meets the security requirements.

Smart metering has an extremely critical role in modern energy management and the security of the smart metering ecosystem is important. By adopting the

strategies and best practices outlined in this chapter, companies can ensure the security, reliability, and resilience of their smart metering system.

# 5. Implementation Example: Securing Communication with TLS and Keypair

## 5.1 Background and Objectives

This study is about the implementation of TLS security to secure JMS messaging, the Jetty web server, and HTTP routes in the smart metering system at Landis+Gyr. For security reasons, no code snippet from the real project or versions of the components will be disclosed. Instead, some dummy code will be used to give the reader some idea about how security can be configured. The main objective of this thesis is identifying the common vulnerabilities of a smart metering system and implementation might vary according to requirement.

In smart metering systems, securing communication is important for data integrity and system reliability. This example implementation will explore the best practices for implementing TLS to secure various components like JMS, Jetty, and HTTP routes for data transmission for smart metering at Landis+Gyr. However, cybersecurity in smart metering is a very big topic and there are so many types of security implementations for smart meters, but the focus of this study is to just secure communication with TLS and a keypair.

There can be several problems if JMS, JETTY, and HTTP routes are not secured with TLS security. For example, JMS messages can be intercepted during transmission, which can cause unauthorized access to sensitive data. An attacker can also intercept and modify the data in transit without detection. Without a secure authentication process, malicious entities can impersonate legitimate services.

Without TLS security, incoming and outgoing data would be in plaintext, which can cause eavesdropping. Attackers can also steal session cookies and act like

legitimate users. Sensitive and valuable information like API keys could be exposed and fall into the wrong hands during transactions.

Without TLS security on routes data packets can be intercepted and altered, credentials might be stolen by sniffing the network traffic, and lack of security can cause unauthorized users to access restricted routes.

Below is a diagram of the solution the study will provide to secure the communication routes in smart metering.

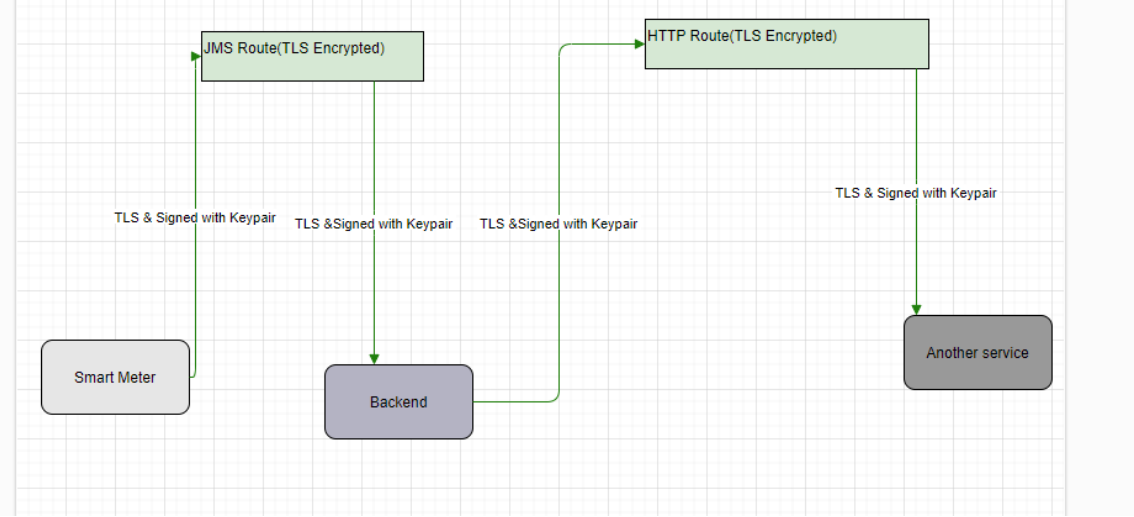


Figure 4: Demo of a smart metering system with a backend which is connected to another service via JMS and HTTP.

In this demo example, the metering data from smart meters to the backend coming via the JMS protocol is encrypted and then the message is processed and sent to another service using the HTTP route.

### 5.1.1 Challenges

Other main challenges faced by Landis+Gyr and other utility companies include:

- Unauthorized access to JMS broker.
- Potential data breaches over HTTP routes.
- Vulnerabilities in web servers leading to potential server exploits.
- Ensuring data integrity and preventing data tempering.

Cyber security in smart metering is a huge topic and this part will only focus on the specified challenges mentioned above. Proprietary code will not be shared because of confidentiality obligations and privacy considerations. Instead, some dummy code will be introduced to give readers some idea about how security can be configured at a fundamental level.

## 5.2 Implementation of TLS for JMS Broker

This part will describe how to encrypt JMS communication using TLS for a smart metering system.

### 5.2.1 Enabling TLS for JMS Broker

Enabling TLS/SSL to a JMS broker requires several steps. Keystore and truststore can be created for the brokers and clients with the Java keytool. They can be created using the Java keytool command-line utility. However, proper configuration of the brokers to use these stores is necessary and important.
First, a new keystore and a keypair needs to be generated with the “keytool” command. The keypair will be used by the JMS broker or server. Listing 1 shows the command for generating a keypair.

```
keytool -genkeypair -alias jmsbroker -keyalg RSA -keysize 2048 -keystore
broker-keystore.jks -validity 365
```

Listing 1. Command for generating a keypair.

“alias jmsbroker” is the keypair within the keystore, and “keystore broker-keystore.jks” is the keystore file to be created. Details about the organization and the keypair need to be entered in the following prompt, with a keystore password. After that, a Certificate Signing Request Should be generated, if the certificate is planned to be signed by a Certificate Authority (CA). Listing 2 shows the command for generating a certificate signing request.

```
keytool -certreq -alias jmsbroker -file jmsbroker.csr -keystore bro-ker-
keystore.jks
```

Listing 2. Command for a certificate signing request.

Then the certificate can be imported. CA's root and any intermediate certificates may need to be imported into the keystore. Listing 3 shows commands for importing root and intermediate certificates.

```
keytool -import -alias root -keystore broker-keystore.jks -trustcacerts -
file path/to/root/ca/certificate
keytool -import -alias intermediate -keystore broker-keystore.jks -trust-
cacerts -file path/to/intermediate/ca/certificate
```

Listing 3. Commands for importing root and intermediate certificates.

After acquiring the signed certificate, it should be imported into the keystore. Listing 4 shows command for importing a certificate to the keystore.

```
keytool -import -alias jmsbroker -keystore broker-keystore.jks -file
jmsbroker.crt
```

Listing 4: Command for importing a certificate to keystore.

After the keystore, a truststore needs to be created. The truststore contains certificates from other parties that need to communicate with or from CA's that are trusted. If a self-signed certificate or certificate from a private CA is used, the certificates need to be imported to the truststore of the client that will connect to the JMS broker. Listing 5 shows command for importing certificates to the truststore.

```
keytool -import -alias jmsbroker -file jmsbroker.crt -keystore client-
truststore.jks
```

Listing 5. Command for importing certificates to truststore.

A mock JMS broker is configured to allow TLS for all JMS connections. For a Java-based simulation, a "BrokerService" should be set up with SSL/TLS. Listing 6 shows the configuration of a JMS broker for supporting TLS:

```
import org.apache.activemq.broker.BrokerService;
import org.apache.activemq.broker.SslContext;
```

```
public class SecureBroker {
    public static void main(String[] args) throws Exception {
        BrokerService broker = new BrokerService();

        // Set the SSL context for the broker
        SslContext sslContext = new SslContext();
        sslContext.setKeyStore("path/to/broker-keystore.jks");
        sslContext.setKeyStorePassword("keystorePassword");
        sslContext.setTrustStore("path/to/broker-truststore.jks");
        sslContext.setTrustStorePassword("truststorePassword");
        broker.setSslContext(sslContext);

        // Add a connector that uses SSL
        broker.addConnector("ssl://localhost:61617");

        broker.start();
    }
}
```

Listing 6. Configuring JMS broker to support TLS.

### 5.2.2 JMS Client Configuration

On the client side, the connection factory should be configured to use TLS and refer to the correct truststore to validate the broker's certificate. Listing 7 shows the configuration of the client side to support TLS.

```
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.JMSException;
import javax.jms.Session;
import org.apache.activemq.ActiveMQSslConnectionFactory;

public class SecureJmsClient {
    public static void main(String[] args) {
        try {
            // Specify the SSL properties
            System.setProperty("javax.net.ssl.trustStore", "path/to/cli-
ent-truststore.jks");
            System.setProperty("javax.net.ssl.trustStorePassword",
"truststorePassword");

            // Create a connection factory that uses SSL
            ActiveMQSslConnectionFactory connectionFactory = new Ac-
tiveMQSslConnectionFactory("ssl://localhost:61617");
```

```
            // Create and start a connection
            Connection connection = connectionFactory.createConnection();
            connection.start();

            // Create a session
            Session session = connection.createSession(false, Ses-
sion.AUTO_ACKNOWLEDGE);

            // Additional setup like creating consumers or producers...

        } catch (JMSException e) {
            e.printStackTrace();
        }
    }
}
```
Listing 7. Client-side configuration for TLS.

Cipher suite can be added with TLS to enhance the security on transport. A cipher suite is like a combination of encryption, authentication, and key exchange algorithms used during SSL/TLS handshake to establish a secure connection. Both clients and the server should support the chosen cipher suite for a successful handshake. Listing 8 shows how cipher suite can be added to JMS routes.

```
System.setProperty("https.cipherSuites", "TLS_AES_256_GCM_SHA384,TLS_EC-
DHE_RSA_WITH_AES_256_GCM_SHA384");
```
Listing 8. Java code to add the cipher suite.

On top of TLS security, signing the data with a keypair gives an extra layer of protection to the data in transit. Keypair security is based on public key cryptography, which involves a private and a public key. The private key is secret, and the public key is shared. Data encrypted with the public key can only be decrypted with the private key, which ensures authenticity.

There are various ways to generate a keypair like using a Hardware Security Module (HSM) or by using various cryptographic libraries or tools, such as OpenSSL, a widely used cryptographic toolkit, and Python, a strong programming language with extensive libraries for cryptography.

In this example, the Java keytool will be used to create a new keypair. The same keypair used in TLS can be used, but it is a good practice to have different keypairs for different purposes. Listing 9 shows the command for generating a keypair.

```
keytool -genkeypair -alias signingKey -keyalg RSA -keysize 2048 -sto-
retype JKS -keystore signingKeystore.jks -validity 365
```

Listing 9. Command for generating a keypair.

For JMS messaging, the message will be signed before sending. This is illustrated below on listing 10.

```
from("direct:jmsStart")
    .process(exchange -> {
        String messageToSign = "JMS message to sign";
        byte[] signedMessage = signData(messageToSign.getBytes(),
"path/to/signingKeystore.jks", "keystorePassword", "signingKey");
        exchange.getIn().setBody(signedMessage);
    })
    .to("activemq:queue:signedMessages");
```

Listing 10. Example code for signing JMS messages.

When receiving the data, the recipient can verify the signature using the public key. This will ensure that the data has not been tempered. Example code for the verification process is shown below on listing 11.

```
public boolean verifySignature(byte[] data, byte[] signature, String key-
storePath, String keyAlias) throws Exception {
    KeyStore keystore = KeyStore.getInstance(KeyStore.getDefaultType());
    keystore.load(new FileInputStream(keystorePath), null);
    Certificate cert = keystore.getCertificate(keyAlias);
    PublicKey publicKey = cert.getPublicKey();

    Signature sig = Signature.getInstance("SHA256withRSA");
    sig.initVerify(publicKey);
    sig.update(data);
    return sig.verify(signature);
}
```

Listing 11. Example code for verifying the signature with the public key.

### 5.2.3 Testing and Validation

The following example illustrates what happens if billing data needs to be sent from the meter to the back end of a smart metering system. The following example illustrates how billing data can be sent. Listing 12 shows some dummy billing data on XML format.

```
<EnergyUsageReport>
    <MeterID>123456789</MeterID>
    <Timestamp>2024-02-06T12:00:00Z</Timestamp>
    <EnergyConsumed>350.5</EnergyConsumed> <!-- Kilowatt-hours (kWh) -->
</EnergyUsageReport>
```

Listing 12. Dummy meter data for testing.

Without TLS security, the data is transmitted over the network as plain text. This means anyone with access to the network potentially intercepts the data and reads the content. Listing 13 is an example of what the data will look like without TLS.

```
[PLAINTEXT TRANSMISSION]
<EnergyUsageReport>
    <MeterID>123456789</MeterID>
    <Timestamp>2024-02-06T12:00:00Z</Timestamp>
    <EnergyConsumed>350.5</EnergyConsumed>
</EnergyUsageReport>
```

Listing 13. Unsecured plaintext message without TLS.

Without TLS, risks of eavesdropping, tempering and impersonation attacks are high. However, if the communication is encrypted by TLS, the process makes it unreadable to anyone except the actual recipient, who has the decryption key. Listing 14 is a conceptual representation of how the encrypted data would not be readable.

```
[TLS ENCRYPTED TRANSMISSION]
EncryptedData(GibberishTextHereThatRepresentsTheEncryptedFormOfTheXML)
```

Listing 14. Encrypted message after implementing TLS.

Thus, securing JMS routes with TLS can improve confidentiality, integrity, and authentication.

### 5.2.4 Outcomes

By implementing SSL/TLS with a cipher suite and keypair signing for JMS routes can improve security in smart metering, enhance confidentiality and integrity between clients and brokers. It stops unauthorised access and data breaches, which are crucial because smart metering systems handle sensitive information. The implementation requires special attention to certificate management but pays off well by protecting against eavesdropping and tempering. This approach not only secures data in transit but also strengthens trust in smart metering communication, providing indispensable assistance for maintaining privacy and system reliability in a smart metering environment.

## 5.3 Implementing TLS for securing HTTP and JETTY routes

### 5.3.1 Transferring from HTTP to HTTPS

Acquiring SSL/TLS certificates for a Jetty server involves several steps. The steps involve generating a keypair and a Certificate Signing Request (CSR) to install the certificates on the server.
First, a key pair needs to be generated. A keypair is a combination of a public and private key. The public key will be included in the CSR, which needs to be submitted to Certificate Authority (CA) to obtain the SSL/TLS certificate. Listing 15 shows the command for generating a keypair.

```
keytool -keystore keystore.jks -alias yourdomain -genkey -keyalg RSA -
keysize 2048
```

Listing 15. Command for generating a keypair.

keystore.jks is the keystore file where the private key and certificate will be stored. yourdomain is a name for the keypair and certificate (alias).

During the generation process, details such as organizational information and fully qualified domain name (FQDN) for the server should be entered. The details will be embedded in the CSR.
For generating CSR, following command needs to be executed from listing 16.

```
keytool -keystore keystore.jks -alias yourdomain -certreq -file your-
domain.csr
```
Listing 16. Command for generating CSR.

"your-domain.csr" is the file containing the CSR. This file should be submitted to a Certification Authority (CA). CA will validate the domain and issue a TLS/SSL certificate for the domain. There are several CAs, such as Let's Encrypt, DigiCert, and Comodo. After certificates are received, they need to be imported.

First, import the root certificate and intermediate certificates into the keystore with the below commands on listing 17.

```
keytool -keystore keystore.jks -import -alias root -file rootCA.crt
keytool -keystore keystore.jks -import -alias intermediate -file interme-
diateCA.crt
```
Listing 17. Commands for importing root and intermediate certificates.

After that, the domain's certificate should be imported with the command below on listing 18.

```
keytool -keystore keystore.jks -import -alias yourdomain -file your-
domain.crt
```
Listing 18.  Command for importing a domain certificate.

"rootCA.cert" is the CA's root certificate. "intermediateCA.crt" is the intermediate certificate and "yourdomain.crt" is the SSL/TLS certificate.

Listing 19 shows an example of configuring Jetty/HTTP routes to use SSL/TLS with the Java programming language.

```
import org.eclipse.jetty.server.Server;
```

```
import org.eclipse.jetty.util.ssl.SslContextFactory;
import org.eclipse.jetty.server.ServerConnector;

public class SecureJettyServer {
    public static void main(String[] args) {
        // Create a new instance of the Jetty server
        Server server = new Server();

        // Create a new instance of the SslContextFactory
        SslContextFactory.Server sslContextFactory = new SslContextFac-
tory.Server();

        // Set the path to the keystore
        sslContextFactory.setKeyStorePath("/path/to/keystore.jks");

        // Set the keystore password
        sslContextFactory.setKeyStorePassword("password");

        // Set the key manager password (if different from the keystore
password)
        sslContextFactory.setKeyManagerPassword("password");

        // Optionally, set the path to the truststore
        sslContextFactory.setTrustStorePath("/path/to/keystore.jks");

        // Optionally, set the truststore password
        sslContextFactory.setTrustStorePassword("password");

        // Create a ServerConnector for the server, using the sslCon-
textFactory
        ServerConnector sslConnector = new ServerConnector(server,
sslContextFactory);

        // Set the port that the server will listen on for secure connec-
tions
        sslConnector.setPort(8443);

        // Add the connector to the server
        server.addConnector(sslConnector);

        try {
            // Start the server
            server.start();
            // Join the server thread to prevent exit
            server.join();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

```
}
```
Listing 19. Code for configuring HTTP routes with Java.

A cipher suite can be added with the code illustrated below on listing 20.

```
 String[] cipherSuites = {
    "TLS_AES_256_GCM_SHA384",
    "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384"
};
sslContextFactory.setIncludeCipherSuites(cipherSuites);
```
Listing 20. Code for adding a cipher suite.

For an extra layer of security, HTTP and Jetty routes can also use a signed message. There is example code below on listing 21 for signing the data before sending it to the network.

```
import org.apache.camel.builder.RouteBuilder;
import java.security.Signature;
import java.security.KeyStore;
import java.security.PrivateKey;
import java.security.cert.Certificate;

public class HttpsSignedRouteBuilder extends RouteBuilder {
    @Override
    public void configure() throws Exception {
        from("direct:start")
            .process(exchange -> {
                String dataToSign = "This is a message to sign";
                byte[] signedData = signData(dataToSign.getBytes(),
"path/to/signingKeystore.jks", "keystorePassword", "signingKey");
                exchange.getIn().setBody(signedData);
            })
            .to("jetty:https://localhost:8443/myService");
    }

    private byte[] signData(byte[] data, String keystorePath, String key-
storePassword, String keyAlias) throws Exception {
        KeyStore keystore = KeyStore.getInstance(KeyStore.getDe-
faultType());
        keystore.load(new FileInputStream(keystorePath), keystorePass-
word.toCharArray());
        PrivateKey privateKey = (PrivateKey) keystore.getKey(keyAlias,
keystorePassword.toCharArray());
        Signature signature = Signature.getInstance("SHA256withRSA");
```

```
        signature.initSign(privateKey);
        signature.update(data);
        return signature.sign();
    }
}
```
Listing 21. Code for signing the data before sending it over HTTP.

Upon receiving, the receiver can verify the signature using the public key. Listing 22 shows example code for verifying the signature with a public key.

```
public boolean verifySignature(byte[] data, byte[] signature, String key-
storePath, String keyAlias) throws Exception {
    KeyStore keystore = KeyStore.getInstance(KeyStore.getDefaultType());
    keystore.load(new FileInputStream(keystorePath), null);
    Certificate cert = keystore.getCertificate(keyAlias);
    PublicKey publicKey = cert.getPublicKey();

    Signature sig = Signature.getInstance("SHA256withRSA");
    sig.initVerify(publicKey);
    sig.update(data);
    return sig.verify(signature);
}
```
Listing 22. Code for verifying the signature with a public key.

### 5.3.2 Testing and Validation

It can be assumed that data has come in the XML format from the meter to the backend. Now that data needs to be transported to another service. As a result, the data is first converted to JSON as in listing 23 and sent to an HTTP endpoint.

```
{
 "MeterID": "123456789",
    "Timestamp": "2024-02-06T12:00:00Z",
    "EnergyConsumed": 350.5
  }
```
Listing 23. Mock meter data in JSON.

Without SSL the data is sent as plaintext, so a potential attacker can intercept the communication and read the data exactly as it is in listing 24.

```
[PLAINTEXT TRANSMISSION OVER HTTP]
```

```
{
  "MeterID": "123456789",
  "Timestamp": "2024-02-06T12:00:00Z",
  "EnergyConsumed": 350.5
}
```
Listing 24. Plaintext message without TLS.

However, when the connection is encrypted with TLS, the data is encrypted before leaving the client and stay encrypted till it reaches the destination. Listing 25 shows encrypted messages when TLS was used.

```
[ENCRYPTED TRANSMISSION OVER HTTPS]
GibberishTextHereThatRepresentsTheEncryptedFormOfTheJSON
```
Listing 25. Encrypted message with TLS enabled.

### 5.3.3 Outcomes

By implementing SSL/TLS for HTTP and Jetty routes, data can be protected in transit, which provides protection against data interception and manipulation. It is necessary for security compliances with data protection standards and user trust. The learning curve involves certificate management of SSL and configuring the server. After overcoming that, it can offer robust protection for data in transit. The TLS security measure is vital for smart metering as well as any web service, ensuring sensitive data will be handled properly and in a trustworthy way.

With keypair signing, data is signed so the receiver can verify the data was indeed sent by the sender and was not tempered. TLS alone can ensure secure transmission, but there is no functionality to verify the integrity of the data or the sender, which can be accomplished by using signed messages with a keypair.

### 5.3.4 Certificate Management

Certificates can be rotated and revoked for enhanced security. Certificates can be replaced with the old ones before the existing certificate is expired or compromised. This maintains confidentiality, integrity, and availability of secure communication between meter and other services. Certificate rotation can be

automated as manual rotation can be time-consuming and error prone. There are tools available on the market to automate the process such as Let's Encrypt, AWS Certificate Manager, or HashiCorp vault. The tools can automatically issue, renew, and replace certificates without risk of downtime due to expired certificates.

One important thing is to revoke compromised and old certificates to prevent their further use and remove them from the trust chain. It is highly recommended to keep detailed report of all certificate issuance, rotation and revocation activities for auditing, troubleshooting, and compliance purpose.

# 6. Risk Mitigation and Flexibility Management

## 6.1 Risk Mitigation

The integration of meters and information technology has created new types of risks to the grid, even a simple attack can cause massive disruption and affect energy availability.

### 6.1.1 Identifying and Prioritizing Risks

Identifying risks in smart metering is important because the grid is getting more and more dependent on information technology, and the security is becoming a key issue. The entire grid could potentially be attacked, but end-devices and the distribution side are most vulnerable.

There can be different types of attacks like denial of service, brute force attack, botnet attack and replay attack. There can be other types of attacks to the end device like malware, viruses, and worms to steal sensitive data. Therefore, proper defence mechanisms like admission, authentication and access control of devices should be verified to minimize the risks. (Dehalwar; Kalam; Kolhe; Zayegh, 2017)

### 6.1.2 Implementing Controls

Security controls need to be implemented to protect against possible attacks. Implementing robust security controls, including encryption, authentication, access control and security audits can be helpful mitigating the risks. For example, using TLS to protect smart metering routes can potentially stop an attacker from intercepting data on transit. Educating employees about possible security threats is necessary for utility companies, as threat levels and ways are changing all the time.

### 6.1.3 Monitoring

Continuous monitoring and regular review about possible security threats helps to adapt to the ever-changing nature of cyber threats. Power Quality monitoring is a popular way for reliable operation of smart meters. Smart metering with Power Quality monitoring can bring significant improvement in securing smart meters and grids. (Madhu; Vyjatanthi; Modi, 2019)

### 6.1.4 Data Integrity

Data integrity is important for the reliability of smart metering systems to ensure that the data that have been collected, processed, and stored is accurate, complete, and consistent. Data integrity is essential for smart metering services because it can affect billing, grid management and customers' trust. Inaccurate data processing can cause wrong billing and customers' data can also be compromised if not handled properly. (Almasabi et al, 2024)

Maintaining data integrity in smart metering systems is multifaceted and involves technical, operational and security dimensions. These challenges include securing data from unauthorised access, protecting data from manipulation and loss, and ensuring the accuracy of data collection and processing.

To ensure data integrity in smart metring systems, several strategies, technologies, and planning are needed. Data should be encrypted end-to-end to

protect against possible attacks. Using TLS on smart metering routes can be a great way protect data on transit. By implementing robust cybersecurity measures like encryption, authentication, access control, data validation and adopting a layered security approach are critical for protecting data integrity in smart grids. (Roy and Debbarma, 2024)

## 6.2 Flexibility Management

### 6.2.1 Modular Design

Modular design can enhance security in smart metering systems by separating individual components. In that way, a component can be updated or changed without affecting the entire system. By implementing modular design, if a certain part of the infrastructure is attacked, the compromised part of the system can be separated if needed. Modular design improves system resilience against cyber-attacks and makes it easier to integrate new technologies. Targeted security measures can be provided also for each component with a modular design approach. (Hseiki et al, 2024.)

### 6.2.2 Scalable Solutions

Scalable solutions to secure smart metering infrastructure ensure that when the system expands, security protocols can be adjusted to support new devices and increased data flow without compromising the system’s integrity. Scalability is important for maintaining security in dynamic environments, where the nature and volume of threats are evolving with technology. (Boakye-Boateng et al, 2023)

### 6.2.3 Adopting Agile Methodologies

Adopting agile methodologies to protect against cyber threats in smart metering ensures flexibility, fast response to attacks and continuous improvement. By adopting agile methods, the development of security measures becomes iterative, which enables utility companies to adapt to new security threats.

## 7. Technological Solution and Innovation

The energy sector is transforming rapidly with the integration of the latest technologies and innovations. The transformation is enhancing the efficiency of energy distribution, consumption and making way for more sustainable and secure energy systems. Below, some key technologies that might shape the future of smart metering are presented.

### Advance Metering Infrastructure (AMI)

Advance Metering Infrastructure (AMI) is a step forward from the traditional metering systems. AMI systems use two-way communication between utilities and consumers of smart meters. They allow instant data collection, remote management, and enhance customers' engagement. Some advantages of AMI technology are accurate billing, dynamic pricing, and energy monitoring. They substantially improve energy reliability and operational efficiencies. (Houda et al., 2020.)

AMI improves the finding and response to security concerns by facilitating two-way communication between users and companies. AMI systems have enhanced encryption protocols to ensure the safety of data in transit.

### Artificial Intelligence and Machine Learning

Artificial Intelligence (AI) and Machine Learning are at the lead of driving predictive analysis and automation in the energy sector. They can analysemassive amounts of data generated by smart meters and predict energy consumption patterns. AI and Machine Learning optimize energy operations and anticipate to maintenance request other smart metering events. They can also help utility companies to make the right decisions, reduce operational costs, and enhance systems' reliability. AI and Machine Learning algorithms can identify suspicious patterns of cyber-attacks like spoofing and tempering. They are

continuously improving and adapting security measures and improving defences by learning from each incident. AI driven detection systems can automatically adjust to security protocols to protect against dynamic security threats.

### Blockchain Technology

Blockchain technology can offer a secure framework for energy transactions, by allowing peer-to-peer energy trading and decentralized energy resource management. Blockchain offers data integrity to facilitate trust among the members in smart energy ecosystems. The blockchain technology can also improve billing processes, certify renewable energy resources, and support the development of microgrids, leading to a more robust and sustainable energy system. (Houda et al., 2020.)

By using smart contracts of Blockchain Technology, billing can be automated, and other data transaction methods like metering data and metering events data can be secured to reduce fraud and errors.

### Edge Computing

Edge Computing processes data locally, which reduces the need of transmitting sensitive data to potentially weak networks. This decreases latency and limits the attack surface for security threats. Implementing Edge Computing in smart metering can improve data privacy and security by authorizing local data processing and storage, providing an additional layer of security by decentralizing data collection points. (Minh et al., 2022.)

## 8. Conclusion

The goal of the thesis was to examine the essential aspects of cybersecurity in smart metering systems. Firstly, it investigates the common vulnerabilities of smart metering systems. Secondly, it offers general solutions for mitigating risks combined with the digitalization of smart metering services. The theoretical

background provides a solid foundation for cybersecurity challenges in smart metering. The practical example provides a basic solution on how security can be configured to encrypt data on transit. The study also discusses the contributions of Landis+Gyr's to the field.

The study starts with a detailed theoretical background about smart meters, their vulnerabilities and how to secure them from possible threats. The practical study with example implementation provides security solutions on a basic level with the implementation of TLS and keypair encryption for securing JMS, JETTY/HTTP routes. No real code was used from Landis+Gyr because of privacy reasons. On top of this, the study pointed out the importance of having strong security for the smart metering infrastructure.

The thesis has covered the general topic of data integrity and risk mitigation strategies and how they can be implemented. It has also discussed future technologies such as Artificial Intelligence, Edge Computing and Blockchain and Advanced Metering Infrastructure (AMI) and their role in smart metering systems. It has also pointed out the importance of continuous learning, adaptation, and innovation to mitigate the cyber security risks.

The study investigated the possible threats to smart meters and suggested some strategies to protect the smart metering infrastructure. The security of smart metering is very important because data breaches can cause unauthorized access and even affect energy availability. The challenge is to ensure data integrity, and securing communication with TLS and a keypair is a useful way to protect the data in transit.

In conclusion, the thesis provides detailed analysis of cybersecurity risks and mitigation strategies in the smart metering ecosystem. The practical example highlights the role of backend development on the security of smart meters. It also provides valuable understanding and guidance for improving the security and reliability of smart metering systems against the growing number of cyber threats.

## 9. References

Almasabi S., Shaf A., Ali T. Ali, Zafar M., Irfan M. Irfan and Alsuwian T., "Securing Smart Grid Data With Blockchain and Wireless Sensor Networks: A Collaborative Approach," in IEEE Access, vol. 12, pp. 19181-19198, 2024.

Boakye-Boateng K., Ghorbani A. A. and Lashkari A. H., "Securing Substations with Trust, Risk Posture, and Multi-Agent Systems: A Comprehensive Approach," 2023 20th Annual International Conference on Privacy, Security and Trust (PST), Copenhagen, Denmark, 2023, pp. 1-12.

Bonakdar, A., Bjelajac, B. and Strunz, A. (2014) “Landis+Gyr: Designing and analyzing business models in value networks,” in Management of the Fuzzy Front End of Innovation. Cham: Springer International Publishing, pp. 281–287.

Dehalwar V., Kalam A., Kolhe M. L. and Zayegh A., "Review of detection, assessment and mitigation of security risk in smart grid," 2017 2nd International Conference on Power and Renewable Energy (ICPRE), Chengdu, China, 2017, pp. 1077-1081.

Houda Z. A. E., Hafid A. and Khoukhi L., "Blockchain Meets AMI: Towards Secure Advanced Metering Infrastructures," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 2020, pp. 1-6.

Hseiki H. A., El-Hajj A. M., Ajra Y. O., Hija F. A. and Haidar A. M., "A Secure and Resilient Smart Energy Meter," in IEEE Access, vol. 12, pp. 3114-3125, 2024.

Iten, A. (1996) “Workflow-Management bei Landis & Gyr,” in Praxis des Workflow-Managements. Wiesbaden: Vieweg+Teubner Verlag, pp. 253–269.

Kohout, D., Lieskovan, T. and Mlynek, P. (2023) “Smart metering cybersecurity-requirements, methodology, and testing,” Sensors, 23(8).

https://www.landisgyr.eu/

Madhu G. M., Vyjayanthi C. and Modi C. N., "Design and Development of a Novel IoT based Smart Meter for Power Quality Monitoring in Smart Grid Infrastructure," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 2204-2209.

Markopoulou, Dimitra. (2022). Tackling cybersecurity challenges in the energy and water sectors in the context of the cybersecurity and sectoral regulatory frameworks: the case of smart metering systems in the new digitalised environment. International Review of Law, Computers & Technology. 37(1), pp. 52–77.

Mathas, C.-M., Grammatikakis, K.-P., Vassilakis, C., Kolokotronis, N., Bilali, V.-G., & Kavallieros, D. (2020). Threat landscape for smart grid systems. Proceedings of the 15th International Conference on Availability, Reliability and Security.Review of detection, assessment and mitigation of security risk in smart grid.

Minh, Quy Nguyen, Van-Hau Nguyen, Vu Khanh Quy, Le Anh Ngoc, Abdellah Chehri, and Gwanggil Jeon. 2022. "Edge Computing for IoT-Enabled Smart Grid: The Future of Energy" Energies 15, no. 17: 6140.

Olivares-Rojas J. C., Reyes-Archundia E., Gutiérrez-Gnecchi J. A., Molina-Moreno I., Cerda-Jacobo J. and Méndez-Patiño A., "Towards Cybersecurity of the Smart Grid Using Digital Twins," in IEEE Internet Computing, vol. 26, no. 3, pp. 52-57, (2022)

Ravichandra A. (2023). Detecting and real time threat analysis in smart grid networks. International Journal of Scientific Research in Engineering and Management. 07. 10.55041/IJSREM25168.

Roy S. D. and Debbarma S., "Enhancing Cyber-Resilience of Power Systems' AGC Sensor Data by Time Series to Image Domain Encoding," in IEEE Transactions on Smart Grid. doi: 10.1109/TSG.2024.3361014.

Stutz, C. (2017) "The strategic cognition view of issue salience and the evolution of a political issue: Landis & Gyr, the Hungarian uprising and east-west trade, 1953–1967," in Issues in Business Ethics. Cham: Springer International Publishing, pp. 139–166.

Sun C. -C., Sebastian Cardenas D. J., Hahn A., and Liu C. -C., "Intrusion Detection for Cybersecurity of Smart Meters," in IEEE Transactions on Smart Grid, vol. 12, no. 1, pp. 612-622, Jan. 2021.

Tweneboah-Koduah, Samuel & Tsetse, Anthony & Azasoo, Julius & Endicott-Popovsky, Barbara. (2018). "Evaluation of cybersecurity threats on smart metering system," in Latifi, S. (ed.), Information Technology - New Generations. Advances in Intelligent Systems and Computing, vol. 558, pp. 199-207. Cham: Springer International Publishing, pp. 199–207.