PEOPLEs' DEMOCRATIC REPUBLIC of ALGERIA

Ministry of Higher Education and Scientific Research

Ferhat Abbas University Setif 1

Faculté des Sciences

Department of Computer Science

Specialization in Networks and Distributed Systems

Security for IoT based-Energy Internet

A thesis presented by:

Chahrazed BEN REBBOUH

In fulfillment of the requirements for the degree of

Doctor of Sciences

Approved by Examination Committee:

Prof. Fouzi HARRAG	Ferhat Abbas University Setif 1	President
Prof. Houssem MANSOURI	Ferhat Abbas University Setif 1	Supervisor
Dr. Sarra CHERBAL	Ferhat Abbas University Setif 1	Co-supervisor
Dr. Chirihane GHERBI	Ferhat Abbas University Setif 1	Examiner
Dr. Hamza DRID	University of Batna 2	Examiner
Dr. Abdelmalek BOUDRIES	Abderrahmane Mira University Bejaia	Examiner

2024/2025

Abstract

With the rapid integration of the Internet of Things (IoT) into the energy sector, the concept of the Energy Internet (EI) has emerged, enabling real-time data monitoring and seamless interconnection between traditional smart grids and renewable energy resources. This paradigm promises enhanced efficiency and sustainability in energy management. However, the explosive growth of interconnected devices and the heterogeneity of the EI ecosystem introduce critical security challenges that remain significant barriers to progress. The main objective of this thesis is to address these security challenges and mitigate vulnerabilities within the EI. To this end, we propose four security solutions leveraging emerging technologies such as blockchain, post-quantum, quantum cryptography, and machine learning techniques, ensuring the continued and secure operation of the EI.

Keywords: Energy Internet, Internet of Things, Smart Grid, Security, Authentication, Blockchain, Cryptography, Post-Quantum, Quantum, Machine Learning.

ملخص

مع الاندماج السريع لإنترنت الأشياء في قطاع الطاقة، ظهر مفهوم إنترنت الطاقة، مفهوم يتيح مراقبة البيانات في الوقت الفعلي والربط بين الشبكات الذكية التقليدية وموارد الطاقة المتجددة. يساعد هذا النموذج في تعزيز الكفاءة والاستدامة في إدارة الطاقة. ومع ذلك، فإن النمو المذهل للأجهزة المترابطة وعدم تجانس البنى التحتية للطاقة المتجددة يطرحان قضايا أمنية حرجة ويظلان عائقاً كبيراً أمام التقدم. الهدف الرئيسي من هذه الأطروحة هو التغلب على المشكلات الأمنية والتخفيف من نقاط الضعف في إنترنت الطاقة، وفي هذا السياق اقترحنا أربعة حلول أمنية باستخدام التقنيات الناشئة مثل البلوكتشين، التشفير الكمومي وما بعد الكمومي، وتقنيات تعلم الآلة، مما يضمن استمرار التشغيل الآمن.

الكلمات المفتاحية: إنترنت الطاقة، إنترنت الأشياء، الأمن، المصادقة، البلوكتشين، التشفير الكمومي، تعلم الآلة.

Acknowledgment

In the name of Allah, the Most Gracious, the Most Merciful. I thank Allah for the strength and blessings He has given me to complete this work.

I sincerely thank **Prof. Houssem Mansouri**, my thesis supervisor, for his constant support, advice, and kindness. His guidance and encouragement helped me finish this work successfully.

A special thanks to **Dr. Sarra Cherbal**, my thesis co-supervisor, for her incredible help and patience. Her thoughtful feedback, daily support, and belief in me kept me motivated, both in my work and personally. Without her, this thesis would not have been possible.

I am deeply grateful to **Prof. Al-Sakib Khan Pathan** for his amazing dedication. He spent countless hours, even late at night, reviewing my papers and guiding me. His commitment greatly influenced my research journey.

I also thank **Prof. Soufiene Djahel**, **Dr. Hui Zhou**, and **Dr. Messai Mohamed-Lamine** for their mentorship during my internship. Their advice, meetings, and teamwork gave me the opportunity to learn and grow as a researcher.

On a personal note, I would like to express my heartfelt gratitude to my sisters, **Sara** and **Fatima**. Thank you for always being there for me; your love and support have been my anchor during difficult times.

To my dear friends **Asma**, **Rihab**, **Amina**, **Kaouther**, **Rania**, and **Sarah**—thank you for making every day brighter. Your friendship helped me through this journey.

*To the memory of my beloved **mother**,*

whose dream was to see me earn my doctorate

May Allah grant her Jannah

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

AS Authentication Server

AVISPA Automated Validation of Internet Security Protocols and Applications

BAN Burrows–Abadi–Needham

BANs Building Area Networks

BB84 Bennett-Brassard-84

BP Buyer Prosumer

CA Certification Authority

CBE Certificate-Based Encryption

CEI Community Energy Internet

CEIC Community Energy Internet Cluster

CGS charging stations

CH Chameleon Hash

CL-AtSe Constraint-Logic-based Attack Searcher

CNN convolutional neural network

CPS Cyber-Physical System

CVP Closest Vector Problem

DHT Distributed Hash Table

DL Deep Learning

DNN deep neural network

DNP3 Distributed Network Protocol 3

DoS Denial-of-Service

DPoS Delegated Proof of Stake

DPoW Delayed Proof of Work

DT decision tree

EBC Energy Blockchain

ECC Elliptic Curve Cryptography

ECDLP Elliptic Curve Discrete Logarithm Problem

ECDSA Elliptic Curve Digital Signature Algorithm

EI Energy Internet

ER Energy Router

EVs Electric Vehicles

FHE Fully Homomorphic Encryption

FPGA Field Programmable Gate Array

G2V Grid-to-Vehicle

GANs Generative Adversarial Networks

GC Grid Companies

GGH Goldreich-Goldwasser-Halev

HANs Home Area Networks

HE Homomorphic Encryption

HLPSL High-Level Protocol Specification Language

IANs Industrial Area Networks

ICTs Information and Communication Technologies

IDS intrusion detection system

IEDs Intelligent Electronic Devices

IIoT Industrial Internet of Things

IoE Internet of Energy

IoT Internet of Things

IoV Internet of Vehicles

KNN k-nearest neighbors algorithm

LWE Learning With Error

MDMS Meter Data Management System

ML Machine Learning

MTU Master Terminal Unit

N11U Network Testing Tool

NIST National Institute of Standards and Technology

NSA National Security Agency

OFMC On-the-fly Model-Checker

P2P Peer-To-Peer

PBFT Practical Byzantine Fault Tolerance

PDCPs Power Distribution and Controller and Prosumers

PHE Partially Homomorphic Encryption

PKI Public Key Infrastructure

PoB Proof of Burn

PoS Proof of Stake

PoW Proof of Work

PQC post-quantum cryptography

PUF Physical Unclonable Function

PV photovoltaic

QBER Quantum Bit Error Rate

QKD quantum key distribution

RAs Registration Authorities

RF random forest

ROM Random Oracle Model

RSA Rivest-Shamir-Adleman

RTC Real-Time Clock

RTUs Remote Terminal Units

SATMC SAT-based Model-Checker

SCADA Supervisory Control and Data Acquisition

SDN Software-Defined Networking

SG Smart Grid

SHAP SHapley Additive exPlanations

SHE Somewhat Homomorphic Encryption

SIS Short Integer Solution

SM Smart Meter

SoC System-on-a-Chip

SP Seller Prosumer

SVC support vector classifier

SVP Shortest non-zero Vector Problem

TLS Transport Layer Security

V2G Vehicle-to-Grid

VPN Virtual Private Network

XAI eXplainable Artificial Intelligence

General introduction

The global shift towards clean and sustainable energy systems has emerged as a key strategic priority for sustainable development [1]. In this context, the use of renewable energy resources occupies first place as an alternative to fossil fuels, giving rise to a new energy network called Energy Internet (EI), combining the existing power grid, renewable energy resources, Smart Grid (SG), and Internet of Things (IoT) [2]. This can involve the integration of sensors, smart devices and communication technologies to improve the performance, monitoring and management of energy-related systems.

Security holds paramount importance in the context of the EI. Given that these systems involve the collection, processing, and sharing of sensitive data regarding energy consumption and distribution, any security breach could lead to severe consequences, ranging from loss of user data confidentiality to major disruptions in energy supply. Smart devices and communication networks used in the EI are often attractive targets for cyber-attacks, as compromising them could directly impact the security of critical infrastructure and the daily lives of users. Therefore, robust security measures need to be put in place at all levels of the EI, from the design of devices and networks to the management of data and software platforms [3]. This includes strong user and device authentication, encryption of sensitive data in transit and at rest, continuous monitoring of networks to detect suspicious activities, and the establishment of rapid and effective incident response protocols in case of an attack. By ensuring the security of the EI, we not only guarantee the protection of critical infrastructure and user data, but also promote trust in the widespread adoption of these technologies for more efficient and sustainable energy management [4].

Problem statement

Given that the SG is a fundamental component of the EI, it can be assumed that the EI inherits all the security vulnerabilities of the SG. In the first place, we have the centralized nature, the main architecture of the SG reveals that each region has a centralized utility center responsible for management and control. This center acts as a third party, which can sometimes be compromised [5]. Secondly, as energy demand increases, it becomes more difficult for the system to adapt and respond effectively to growing needs, posing scalability issues [6]. Thirdly, with regard to recent advances in security, we are facing the emergence of quantum computers, which are exceptionally fast compared to classical computers, leading to the risk of quantum attacks that threaten existing cryptographic methods, requiring more robust defense strategies [7]. Fourthly, the EI is a business network, means its design allows individual customers to participate simultaneously in the production and distribution of energy. It is therefore crucial to identify and mitigate unauthorized access and malicious activity within the system, requiring advanced techniques for effective intrusion detection [8].

Goals and contributions

Numerous security strategies have been proposed to guarantee the security of EI. However, despite this increase, a major gap remains in the existing research literature, namely the absence of security solutions that address multiple security threats particularly quantum attacks. Thus this thesis aims to fill this gap by proposing new solutions to strengthen its resilience against emerging threats. However, tackling all these challenges together is a difficult multi-tasking objective, in this thesis we adopt a step-by-step approach where we first solve each challenge separately and then combine the proposed solutions in a way that maximizes benefit while preserving performance. To this end, we have proposed four solutions: Our first solution in Chapter 4 is based on blockchain. By using blockchain, we addressed the challenges of scalability, centralization, and third-party involvement. In the second solution (Chapter 5), we focused on resistance to quantum attacks using post-quantum cryptography (PQC). PQC relies on mathematical problems thought to be difficult to solve efficiently, even for quantum computers.

In the third solution presented in Chapter 6, we have combined the first solution (based on blockchain) with the second solution (based on PQC) to create a hybrid solution, this solution inherits the properties of both approaches. The last solution in Chapter 7, relies on Deep Learning (DL) and eXplainable Artificial Intelligence (XAI) to detect intrusions and suspicious activity.

Dissertation outline

All chapters in this thesis are manuscripts of our articles that have been published or submitted in scientific journals. The thesis include seven chapters divided into two main parts: background and contributions. The background part overview the EI concept, key technologies, security threats and requirements, existing solutions, and open challenges. While the contributions part includes our proposed solutions to address the security challenges in EI. This dissertation is structured as follows:

- Chapter 1 introduces the term EI, while providing an in-depth understanding of the key technologies involved in the basic EI architecture. It also discusses the main security issues and the security measures required.
- Chapter 2 provides an overview of existing solutions in the literature review that mitigate threats within EI security, and also discusses the limitations and challenges associated with these solutions.
- Chapter 3 summarizes the techniques and methods used to assess the security of the proposed contributions in part II, including the evaluation metrics and tools used for simulation.
- Chapter 4 presents an enhanced mutual authentication protocol for IoT-based EI using blockchain technology. The proposed protocol integrates blockchain-based security mechanisms to guarantee secure communication between various IoT devices. The implementation was carried out on Hyperledger Fabric, a popular blockchain platform. While performance is evaluated using Caliper benchmarking tool, security properties are also assessed using Burrows–Abadi–Needham (BAN) logic and ProVerif tool.
- Chapter 5 proposes a defense mechanism against quantum computer attacks in IoT-based EI using the Goldreich-Goldwasser-Halev (GGH) cryptosystem and quantum key distribution (QKD).

These techniques offer the best security features and contribute to secure authentication, eavesdropping detection and resistance to the best-known attacks. The proposed protocol is evaluated using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool to demonstrate the security properties it guarantees.

- Chapter 6 in this chapter, a combined approach from Chapters 4 and 5 is presented, integrating PQC and blockchain technology to enhance security and authentication in EI. The proposed solution benefits from the advantages of both solutions to ensure the confidentiality, integrity, and privacy of critical information, reinforcing the resilience of the EI. The security of the proposed protocol is validated using BAN logic and the Proverif tool.

- Chapter 7 provides an introduction to a robust intrusion detection system (IDS) specifically designed for the environment and constraints of SGs. This IDS uses the convolutional neural network (CNN) model to effectively identify and neutralize potential security threats. The proposed CNN model is complemented by the integration of the SHapley Additive exPlanations (SHAP) algorithm to enhance the transparency of the decision-making process.

Part I

Background

Chapter 1

Security issues and requirements in Energy Internet

1.1 Introduction

The adoption of renewable energy resources as a primary energy source has attracted a great deal of industry attention in recent years, due to the need to find environmentally-friendly energy solutions. In response to this need, EI has emerged as a transformative network designed to integrate diverse renewable energy sources into a coherent, efficient and intelligent energy system.

This chapter offers a comprehensive exploration of EI, starting with a detailed discussion of its fundamental components, such as IoT, energy storage systems, SGs. We examine the architectural framework that enables these components to interconnect and operate seamlessly, ensuring optimal energy distribution and management. In addition, this chapter examines the potential threats and vulnerabilities inherent in EI. by doing this, it highlights the critical security requirements needed to protect and maintain network resilience.

1.2 Energy Internet (EI)

The terminology associated with the concept of the EI is particularly inconsistent, creating confusion in both academic and industry communities. Various studies and papers have referred to the concept using terms such as "Internet of Energy (IoE) [9, 10, 11]" or "Smart Grid 2.0 [12, 13, 14]." This inconsistency goes beyond linguistic differences and includes variations in the architectural frameworks proposed to define the components and functionalities of this network. Such ambiguity not only prevents a clear discourse on EI, but also complicates efforts to standardize systems and interoperate technologies in this area.

Different definitions have been proposed by authors in the literature, Parvin et al. [15] describe EI as an evolution of the SG, integrating its features with those of the IoT. Within this framework, EI facilitates the exchange of data and information, encouraging seamless communication and coordination between energy networks. Joseph et al. [16] explored the concept of EI based on two key studies: Rifkin [17, 18] and Tsoukalas [19, 20]. Rifkin defines EI as the third industrial revolution, while Tsoukalas considers it as the successor of the SG. These differences show that there is no single agreed model for EI.

In our perspective, the EI is a new concept that incorporates an idea or vision of future power systems, which integrate different types of distributed and renewable energy resources, a SG and Information and Communication Technologies (ICTs). The definition provided by Laroussi et al. [21] offers a comprehensive overview of existing perspectives, drawing on Jeremy Rifkin's book "*The Third Industrial Revolution* [18]." This definition emphasizes five key concepts:

- An urgent transition to renewable energies as the primary energy source.
- Converting buildings into micro-energy systems that autonomously produce electricity and distribute surplus power to the grid or other consumers.
- Strategically deploying storage technologies alongside renewable energy systems to enhance grid flexibility and seamlessly integrate dispersed renewable sources.
- Leveraging existing infrastructure to facilitate the shift from a centralized to a decentralized energy sector.
- Electrifying the transport sector to reduce oil dependence while developing an interactive electricity network.

In summary, the term "EI" is inspired by the traditional internet, but rather than connecting computers and devices to exchange information, EI links various energy resources. It interconnects distributed renewable energy systems, such as solar panels, wind turbines, electric vehicles, and storage systems to exchange energy. Figure 1.1 provides a global overview of the EI.

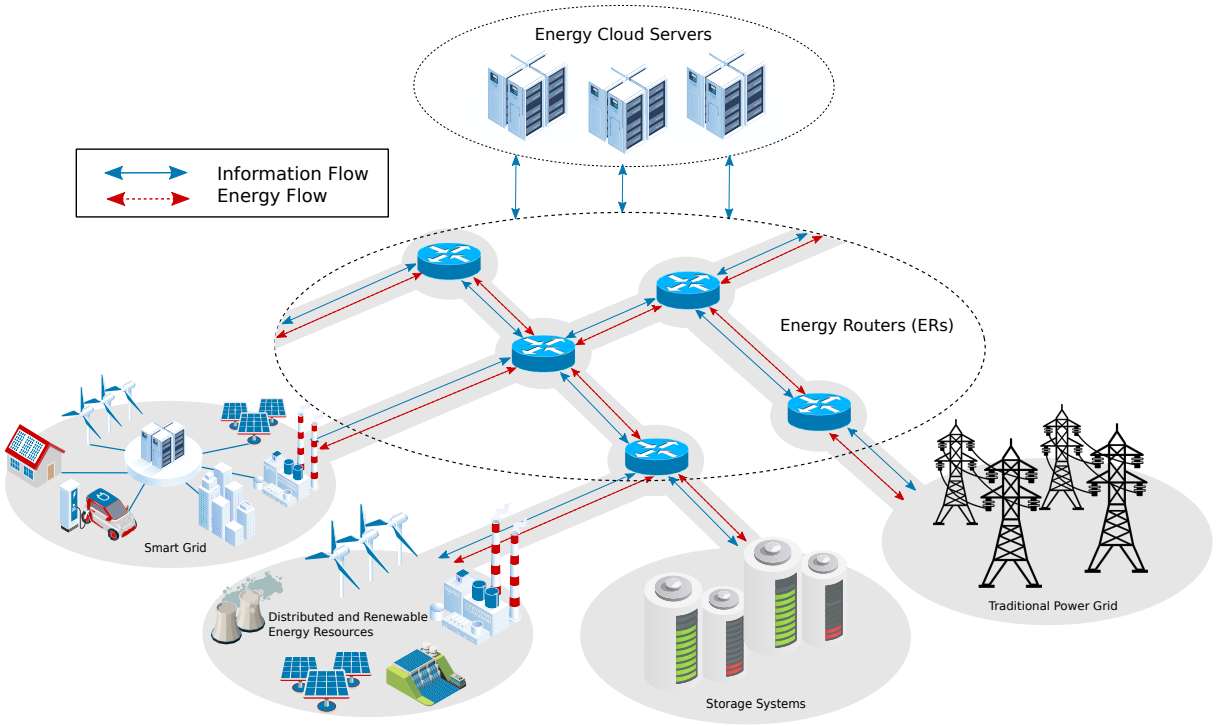


Figure 1.1: Global view of the Energy Internet.

1.3 Energy Internet key technologies

Implementing the EI is still a challenging task, here we discuss some key technologies that are essential for such implementation.

1. **Smart Grid (SG):** SG serves as the foundational building blocks of the EI, representing an advancement in electrical infrastructure through the integration of communication technologies with traditional power grid components, enhancing energy generation, transmission, and distribution [3]. Unlike the traditional power grid, which supports unidirectional transmission, the SG enables bidirectional communication and energy flow among its various components.

The SG can be modernized as a Cyber-Physical System (CPS) illustrated in Figure 1.2, where it incorporate two layers, the physical layer, also known as the power system layer, encompasses key components such as energy generation, transmission, and distribution, as well as energy end-users, including homes, buildings, and industries. The communication within this layer is facilitated by Home Area Networks (HANs), Building Area Networks (BANs), and Industrial Area Networks (IANs), which enable seamless data exchange between these entities [22]. Central to the operation of a SG is the Supervisory Control and Data Acquisition (SCADA) system, which provides real-time monitoring, data acquisition, and control of the grid's components, such as substations and transformers. SCADA systems are essential for maintaining the stability of the grid and its operation efficiency, enabling operators to remotely control and automate various grid functions [23]. SCADA, along with distribution and microgrid control and consumer-side management, represent the cyber layer of the SG, while wired and wireless communication serves as the cyber-physical link.

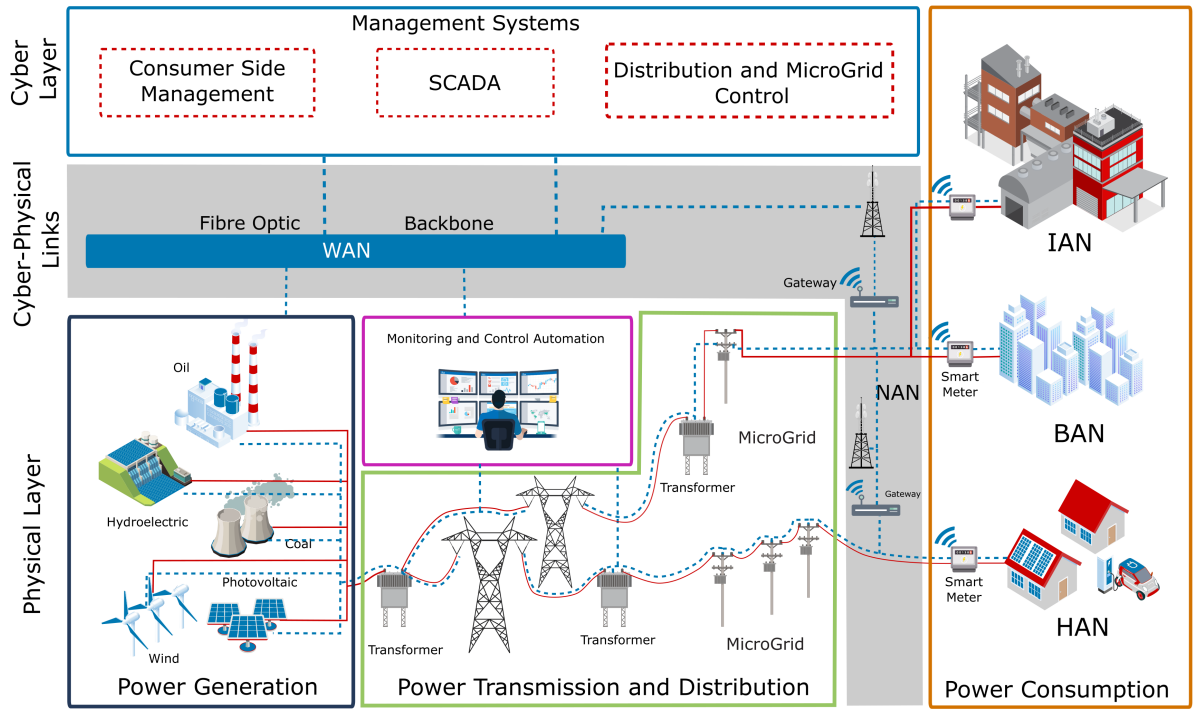


Figure 1.2: SG architecture as a CPS.

2. **Internet of Things (IoT):** IoT is a global network that connects physical objects via the internet, enabling smart devices to collect, process and share data about themselves and their environment [24]. In the energy sector, IoT plays a crucial role in providing sensing and actuation capabilities, which are essential for improving operational efficiency and facilitating communication between different components. One of the key features of the EI is its seamless integration with the IoT, enabling extensive control and monitoring of all activities and functionalities within the SG [2].

3. **Electric Vehicles (EVs) and V2G technology:** The integration of EVs into the power grid via Vehicle-to-Grid (V2G) technology offers a valuable opportunity to improve energy efficiency and enhance grid stability, thus promoting the development of sustainable cities. EVs can operate as both consumers and prosumers. In prosumer mode, made possible by V2G technology, EVs supply electricity to the grid, while in Grid-to-Vehicle (G2V) mode, they act as consumers, drawing electricity as a load [25]. EVs also act as mobile energy storage devices [26], offering a flexible and innovative approach to managing energy resources. Integrating EV with EI benefit both fields.

4. **Energy storage system:** Energy demand and supply are both subject to variations, which can lead to grid disruptions. To mitigate this, the surplus energy generated from renewable resources is stored in energy storage systems, ensuring its availability in the event of a power shutdown or grid breakdown. There are various types of energy storage systems, including batteries, supercapacitors and flywheels [27].

5. **Energy Router (ER):** When energy supply exceeds local demand, Energy Routers (ERs) play a crucial role. These routers dynamically adapt to variations in demand and supply by scheduling, converting and controlling energy, ERs differ from one another depending on the task assigned [27]. ERs ensure efficient management and distribution of surplus energy [4].

6. **Smart Meter (SM):** A Smart Meter (SM) is an advanced utility meter with both measurement and communication capabilities. It measures energy consumption data, enabling remote readings and secure display on home appliances. SMs can receive remote instructions, such as switching between credit and prepayment modes, or updating tariff information. Their main functions are to provide consumers with data on their energy consumption so that they can better control their consumption and costs, and to send information to utilities for peak load management and pricing strategies.

Within the energy infrastructure, SMs enable consumers to know how and when they consume energy, and how much they pay per kilowatt-hour of energy. This improves price transparency, enables more accurate billing, and speeds up outage detection and restoration by the utility [28].

1.4 Security threats and attacks

In the field of EI security, threats are a common concern and difficult to manage because EI involves multiple connected infrastructures. Ignoring these threats without taking proper security measures can lead to successful attacks.

1.4.1 Traditional attacks

Traditional attacks include a range of well-known threats, such as Man-in-the-Middle (MITM) attacks, impersonation attacks, replay attacks, and Denial-of-Service (DoS) attacks. These attacks exploit vulnerabilities in communication, authentication, and system availability, posing significant risks to security and functionality.

- **Man-in-the-Middle (MITM) attack:** The attacker basically sits in the middle of a communication event and tries to hijack it by eavesdropping or cutting and intercepting the data exchanged between the communicating legitimate parties [29]. While in many cases, the intercepted data may not be altered right away, often the attacker may use the recorded data at a later time to cause harm or cause confusion in the system. When the data are actively altered, it is an attack on the data integrity. Hence, sensitive installations like SG or power lines can be severely affected by this. The main objective of MITM is to manipulate messages; in particular, it often targets communication between hubs and networks or vehicles, in order to influence system accountability [30].

- **Impersonation attack:** An impersonation attack occurs when an adversary poses as a legitimate user or device in order to gain unauthorized access to a system or network. This is often done by imitating the target's credentials, identifiers or communication patterns [31].

- **Replay attack:** A replay attack involves an adversary capturing a valid data transmission (e.g. authentication messages) and replaying it later to trick the system into granting unauthorized access or performing unintended actions [32].

- **Denial-of-Service (DoS) attack:** A Denial of Service (DoS) attack aims to disrupt the availability of a system, service or network by overwhelming it with excessive requests or malicious traffic, rendering it inaccessible to legitimate users [33].

- **Brute force attack:** A brute force attack is a systematic method by which an attacker attempts to guess passwords or credentials by trying every possible combination until the correct one is found. This attack is often automated using tools that quickly generate and test potential passwords [34].

- **Eavesdropping attack:** An eavesdropping attack occurs when an unauthorized person intercepts, modifies or deletes data sent between two devices. This type of attack takes advantage of unsecured or weakly encrypted networks, enabling the attacker to access sensitive information during transmission. Often referred to as "sniffing" or "snooping", eavesdropping can be carried out in a number of different ways. For example, it can be passive, when the attacker simply listens to the communication, or active, when the attacker goes further by manipulating the data or posing as a legitimate user in order to steal more information [35].

- **Session hijacking:** Session hijacking occurs when an attacker impersonates a legitimate user by stealing or manipulating session credentials (e.g. cookies, session IDs) to gain unauthorized access to an active communication session. This attack exploits vulnerabilities in protocols such as TCP or web applications where session management lacks strong authentication or encryption [36].

- **Information disclosure** Refers to the unintentional exposure of sensitive data due to inadequate security controls [37]. The concept is implicit in discussions about maintaining confidentiality in communication protocols and the handling of structured/unstructured data.

- **Perfect forward secrecy:** Ensures that the session keys used in encrypted communications cannot be compromised, even if long-term secret keys are leaked [38]. To achieve this, temporary session keys are generated by ephemeral key exchanges (e.g. Diffie-Hellman or Kyber algorithms) [39].

1.4.2 Quantum attack

Quantum computers are another cutting-edge development that store data and perform calculations using the principles of quantum physics. While they can outperform classical supercomputers in specific tasks, this unprecedented computational power also introduces a new category of threats known as quantum attacks. These attacks jeopardize the security and confidentiality of widely used cryptographic algorithms such as Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC), which rely on mathematical problems that quantum computers can solve efficiently [40]. Acknowledging this emerging risk, major institutions such as the U.S. National Security Agency (NSA) and the National Institute of Standards and Technology (NIST) have initiated efforts to transition toward quantum-resistant cryptographic standards. Notably, NIST has launched a competition to identify and standardize public-key algorithms resilient to quantum threats, with the final selection expected by 2026 [41].

1.5 Security requirements

Addressing security issues in the EI is a complicated task that needs to be handled systematically. By defining and verifying each security requirement individually, we can effectively address security issues in the EI. Figure 1.3 illustrates the main security requirements of the EI, which are discussed below:

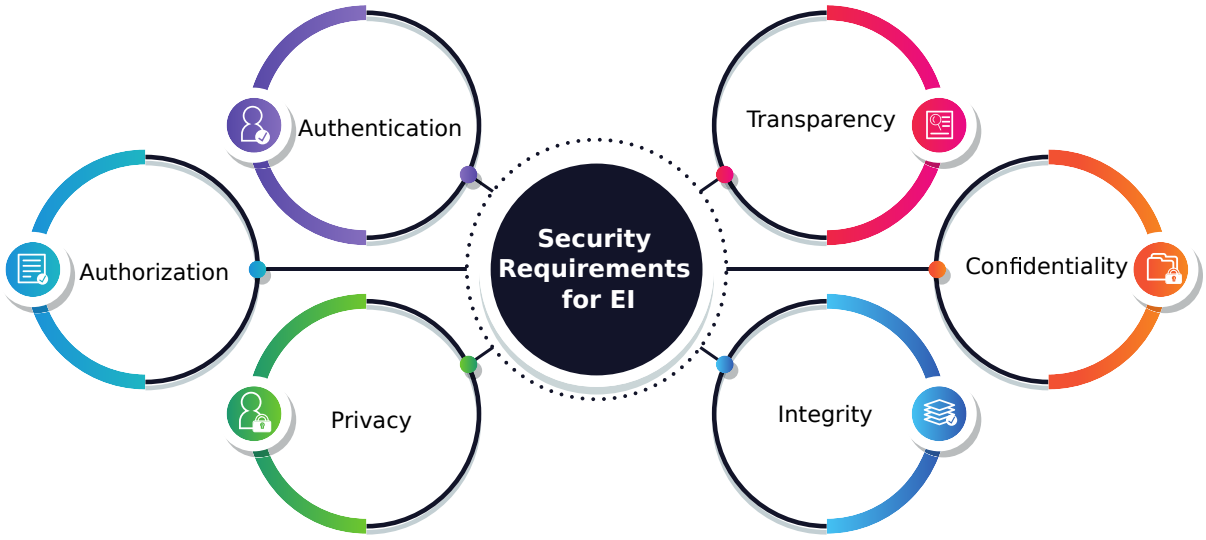


Figure 1.3: Security requirements for EI.

- **Authentication:** Authentication can be defined as the process of verifying that a person is genuinely the one who claims to be, i.e. guaranteeing the authenticity of the communication between the parties. It makes communication trustworthy [42, 43].

- **Privacy:** In simple terms, it is about controlling with who and how a user's personal information is shared [42]. In the context of the EI, energy consumption information can provide a model of personal life, which can simply be used by a thief, in the real world.

- **Authorization:** Authorization focuses on controlling user access to network resources [44], In other words, it determines the type of actions the user is allowed to perform and the type of resources they are allowed to access, which is highly recommended in the EI network, as it includes energy resources and any unauthorized access to these resources would be costly.

- **Integrity:** Guarantees that the data is accurate, i.e., that it has not been modified by a third party other than the authentic sender and receiver [45], which is a vital characteristic in energy systems.

- **Transparency:** Transparency refers to the clear and open visibility of energy-related data, processes and transactions within the energy network. This transparency ensures that all stakeholders, including consumers, suppliers, and regulators, have access to accurate and timely information on energy production, distribution and consumption.

- **Confidentiality:** While authorization focuses on access control, confidentiality involves hiding private information from unauthorized users or devices [46]. Again, the difference between privacy and confidentiality is that privacy is deeply related to personal or private data while the confidentiality in this context is a general requirement of keeping data secret or hidden from an entity who is not entitled to possess it.

1.6 Conclusion

The EI represents the future of power systems, making its security a key priority. This chapter has provided an overview of the EI concept, discussing the essential key technologies that support its implementation, as well as its architecture, security threats and requirements.

In the following chapter, we will explore the security challenges of the EI in greater depth, with a specific focus on existing security solutions. The discussion will analyze the strengths and limitations of current approaches across different categories.

Chapter 2

Existing security solutions and their limitations

2.1 Introduction

Security is a key factor in the successful deployment of the EI. As the EI evolves to incorporate a wide range of renewable energy resources and distributed technologies, it simultaneously expands the attack surface of the network. This increased complexity introduces new vulnerabilities that, if left unaddressed, could undermine the reliability and resilience of the entire system. To ensure secure and efficient operation, it is essential to implement robust security mechanisms capable of detecting, preventing, and mitigating cyber-attacks.

In this chapter, we present a detailed overview of the security landscape. Section 2.2 provides a comprehensive examination of existing solutions that have been proposed to improve EI security. This analysis covers various methodologies and techniques, assessing their effectiveness and limitations in mitigating threats within the EI environment. Section 2.3 analyzes these approaches to identify best practices and potential improvements. Subsequently, Section 2.4 explores the challenges and outstanding issues that need to be addressed to establish robust and secure EI protocols.

2.2 Security solutions for Energy Internet

Numerous studies have been conducted to enhance the security of the EI, leading to the development of various proposed solutions. We classify these solutions into five main categories: traditional cryptography, PQC and quantum cryptography, blockchain-based, Machine Learning (ML)-based, and hybrid solutions.

2.2.1 Traditional cryptography based solutions

Cryptography is considered the oldest method in security solutions, with various types and methods depending on the field of application.

2.2.1.1 Cryptographic techniques

ECC is a well-known and widely used cryptographic method, as it offers better security with the same key length as RSA, and is well suited to devices with limited resources such as IoT devices, hence it is considered a lightweight asymmetric encryption method [47]. An elliptic curve is a set of points described by the equation [48]: $y^2 = x^3 + ax + b$. Where a and b are chosen so that the curve has no singularities (i.e. $4a^3 + 27b^2 \neq 0$). One of the basic ECC operations is scalar (or point) multiplication, which involves multiplying a point P on the elliptic curve by an integer k to obtain a resultant point Q, denoted as $Q = k.P$. The multiplication is performed using a repeated addition operation. As shown in Figure 2.1, which is the elliptic curve used in Bitcoin named secp256k1 with the equation: $y^2 = x^3 + 7$. To add the two points P and Q, we draw a straight line that passes through the two points, the intersection of this line is $-R$, the resulting point is the reflection of $-R$ on the x-axis which is R. Given the point R, it is difficult to find P, which is the Elliptic Curve Discrete Logarithm Problem (ECDLP) [45].

ECDLP is a crucial challenge within ECC, involving the discovery of a scalar multiple, $k \in \mathbb{Z}_q^*$, such that $Q = k \cdot G$, where Q is a point on an elliptic curve $E(\mathbb{F}_p)$. Solving the ECDLP is recognized to be computationally infeasible within polynomial time. Consequently, even with the most advanced computing technology available today, determining the secret integer k from the public parameters of the elliptic curve $E(\mathbb{F}_p)$, and the points G and Q, requires an impractical amount of time. The ECC generated keys will then be used in the Elliptic Curve Digital Signature Algorithm (ECDSA), which is a digital signature

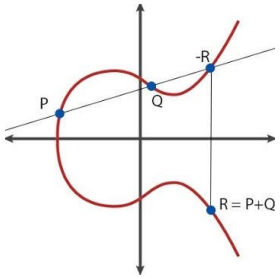


Figure 2.1: secp256k1 point addition.

algorithm widely used in security systems such as Bitcoin and Transport Layer Security (TLS) [49].

The Schnorr signature algorithm relies on ECDLP and presents numerous benefits compared to alternative signature schemes. It guarantees strong security characteristics including unforgeability, non-repudiation, and authenticity. Additionally, it incorporates batch verification, enabling the concurrent validation of multiple signatures, thus lowering computational expenses. The Schnorr signature algorithm comprises three main steps [50]: key generation, signature generation, and signature verification.

- **Key generation:** The procedure initiates by selecting a random number, $r \in \mathbb{Z}_q^*$. Subsequently, compute $P = r \cdot G$. The pair (r, P) forms the user's public-private key pair, with r serving as the private key and P as the public key. It is imperative to safeguard the private key and refrain from disclosing it, whereas the public key can be distributed openly.

- **Signature generation:** The process start with the selection of a random scalar, $k \in \mathbb{Z}_q^*$. Subsequently, R is computed as $R = k \cdot G$. Following this, a cryptographic hash function H is employed to derive $e = H(R||P||m)$, where m represents the message intended for signing. Ultimately, the signature on m is denoted as (R, s), with $s = r + k \cdot e$.

- **Signature verification:** To authenticate the signature, the recipient of the message and signature initially obtains the point s on the elliptic curve from the signature pair (R, s). Then, the recipient calculates $P + R \cdot e$. Finally, the recipient verifies whether $s \cdot G \stackrel{?}{=} P + R \cdot e$.
  To prove the correctness of this verification, we expand $s.G$ as follows: $s \cdot G = (r + k \cdot e) \cdot G = r \cdot G + k \cdot e \cdot G$. Since $r \cdot G = P$ and $k \cdot e \cdot G = R \cdot e$, we obtain: $s \cdot G = P + R \cdot e$.
  This shows that if $s \cdot G = P + R \cdot e$, the signature is valid, confirming the authenticity and integrity of the message.

Homomorphic Encryption (HE) is a type of encryption that allows one to perform a series of operations on encrypted data and obtain the same result as performing these operations on unencrypted data [51]. HE can be classified into three types according to the operations they support: Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE), and Fully Homomorphic Encryption (FHE). In PHE, an operation can be performed an infinite number of times on the ciphertext, while in SHE, a multiplication or addition can be performed with a limited number of operations. FHE combines the two, as a multiplication or addition can be performed an unlimited number of times.

The hash function is a one-way function, which means that it is easy to calculate the hash of a message, but the reverse option is not possible. Calculating the hash of a message of any length will give a result of a fixed length [52], such as SH-256 gives a 256-bit result, while MD5 gives a 128-bit result.

Certificate-Based Encryption (CBE) is a digital certificate-based cryptographic method, in which the digital certificate is issued by a Certification Authority (CA) and contains public keys and identity information. CBE relies on Public Key Infrastructure (PKI) to authenticate users and their public keys, enabling secure and verified encryption and decryption processes [53].

2.2.1.2 Related works on traditional cryptography based solutions

Table 2.1 summarizes the most important works on cryptography-based solutions. It evaluates the performance of each proposed cryptographic protocol in terms of communication, computation, and storage costs, as these metrics are the primary criteria for comparing such security solutions. Detailed calculations for these metrics are provided in Chapter 3.

Verma et al. [54] introduce a solution for secure data aggregation in EI-based SM to grid communication, denoted as PF-DA, which serves as a pairing-free encryption technique. The proposed solution eliminates the requirement for pairing operations, thereby achieving a reduction in computational cost through the utilization of CBE. Through a comprehensive security analysis, PF-DA demonstrates resilience against various attacks. The authors evaluate the performance of the proposed scheme in terms of computational, communication, and storage costs.

Khan et al. [55] introduce an authentication and key agreement protocol designed to facilitate V2G communication. The proposed protocol integrates ECC and hash functions, encompassing three principal phases: initialization, registration of both the vehicle and the grid, and the key agreement phase.

Table 2.1: Traditional cryptography based solutions.

Ref	Year	Secured Feature	Verification Methods	Cryptosystem	Performance
[54]	2021	Privacy Integrity	N/A	CBE, Hash	- Communication Cost= 2048 bits - Computational Cost= 70 ms - Storage Cost= 1024 bits
[55]	2021	Authentication Confidentiality	AVISPA, BAN Logic, ROM	ECC, Hash	- Communication Cost=896 bits - Computational Cost=0.46 ms - Storage Cost = N/A
[56]	2022	Authentication Confidentiality	ROM	ECC, Hash	- Communication Cost=1144 bits - Computation Cost=26.906 ms - Storage Cost = N/A
[57]	2022	Authentication Confidentiality	ROM	ECC	- Communication Cost=1504 bits - Computational Cost=200.6 ms - Storage Cost = N/A - Energy Consumption= 24.07 mJ
[58]	2022	Privacy Integrity	N/A	PHE (Paillier)	- Communication cost = N/A - Computational Cost=54.0389 ms - Storage cost = N/A
[59]	2022	Authentication Integrity	N/A	ECC, Hash	- Communication Cost = N/A - Computational cost=312.4 ms - Storage cost= N/A
[60]	2023	Authentication	Scyther, ROM	ECC, Hash	- Communication Cost=1308 bits - Computational Cost=5.332 ms - Storage Cost=640 bits

N/A: Not Applicable.

To assess its security, the protocol undergoes evaluation within the Random Oracle Model (ROM) as an adversary model and employs BAN logic as its security model. Furthermore, the authors utilize AVISPA as a verification tool. Performance evaluation of the protocol focuses primarily on communication and computational costs.

The authors of [56] introduce a secure framework tailored for facilitating key agreement and authentication between EVs and SG infrastructure, employing ECC and hash functions. Within this framework, EVs are required to initiate a secure session with charging stations. The proposed scheme comprises three distinct phases: initialization, encompassing the generation of public and private key pairs, registration within the grid server database; and key agreement to establish a session key. Furthermore, the authors assess the security of their scheme within the ROM. In terms of performance evaluation, they analyze communication and computational costs, while storage costs remain unaddressed.

Safkhani et al. [57] introduce an enhanced protocol building upon the framework proposed by Khan et al. [61], which had been found vulnerable to multiple attacks. The refined protocol integrates Physical Unclonable Function (PUF) technology. In the original scheme, the compromise of a user entity enables an attacker to successfully extract secret keys. Consequently, Safkhani et al. [57] incorporate PUFs, equipping each user with this technology. This addition ensures that even if an attacker infiltrates a user entity, retrieval of secret keys stored within the PUF becomes infeasible. The proposed scheme encompasses four distinct steps: initialization, registration, login and key agreement, and password and identifier modification. The security of the proposed scheme is examined using the ROM, alongside an assessment of its performance in terms of computational, communication, and energy costs.

Rao et al. [58] introduce a novel multi-dimensional system designed to ensure the integrity and privacy of user data. Central to their proposal is the integration of Paillier homomorphic encryption, wherein every SM within the network encrypts electricity consumption data before transmitting it to the edge server. Subsequently, the edge server undertakes data aggregation operations utilizing hash functions and super incremental sequences. Authentication of the aggregated data is facilitated through batch signature verification. The performance evaluation of this proposed system primarily focuses on computational costs, while communication and storage expenses remaining unassessed.

In the realm of EI security, the strategy of assigning unique identities to edge devices and regularly updating their private keys based on embedded timestamps is pivotal. This approach, exemplified by [59], ensures robust security measures. For instance, a SM with the ID "SM001" may see its private key refreshed every 24 hours, reflecting its validity period (e.g., SM001 $||$ 2025.05.01-2026.05.02). During inter-device communication, these updated private keys authenticate devices, maintain data integrity, and ensure confidentiality. Incorporating Real-Time Clock (RTC) information and timestamp-based mechanisms reduces the risk of key leakage. The authors conducted an evaluation of the computational cost associated with their protocol. Itoo et al. [60] introduced an authentication framework for V2G communication based on ECC. This framework aims to facilitate the establishment of communication sessions between EVs and charging stations (CGS). The proposed protocol consists of four distinct phases: initialization, registration, authentication, and password update. The security of the protocol is examined through assessment using the ROM and the Scyther tool. Additionally, performance evaluation encompasses computational, communication, and storage costs as key metrics.

2.2.2 Post-quantum and quantum cryptography based solutions

Significant advancements have been made in the field of quantum computing in recent years, even though it is still in its early stages. Currently, the most powerful quantum computer available is IBM's Osprey, which has 1121 qubits [62]. However, the RSA-2048 encryption standard, which is considered relatively basic, requires 4000 qubits for potential decryption by quantum computers. To address this gap, IBM is planning to release a new quantum computer with over one million qubits [62], highlighting the importance of developing cryptographic systems that can withstand quantum attacks. In this context, there has been a significant focus on research aimed at addressing these vulnerabilities and developing strong cryptographic methods.

2.2.2.1 PQC and quantum cryptography approaches

Quantum cryptography stands out as a promising approach such as QKD, utilizing the principles of quantum mechanics. However, the complexity and high cost associated with quantum cryptography make it difficult to integrate into current devices [63].

To address these challenges, a novel cryptographic discipline known as PQC or quantum-resistant cryptography has emerged. This cryptographic approach aims to resist both classical and quantum attacks by adopting NP-hard problems, which are computationally challenging even for quantum computers. By embracing such methodologies, the cryptography community aims to enhance data security in the face of advancing quantum threats. PQC can be classified into four main classes: lattice-based, code-based, hash-based, and multivariate cryptography [64]. However, lattice-based cryptography schemes, considered as the most promising candidates in the NIST competition were based on hard lattice problems, such as the Closest Vector Problem (CVP) and the Shortest non-zero Vector Problem (SVP). These two problems are fundamental in the field of lattice cryptography, but they are not practical for lattice construction. Novel and suitable problems have been introduced, such as NTRU, Learning With Error (LWE), and Short Integer Solution (SIS), which form the basis of current PQC [65]. Therefore, CRYSTALS-DILITHIUM and FLCON were selected in the digital signature category in 2022 [66, 67].

2.2.2.2 Related works on PQC and quantum cryptography based solutions

In this section, we outline the most important works that rely on PQC and quantum cryptography to secure EI, which are summarized in Table 2.2.

The work in [68] analyzes the security and effectiveness of a quantum Virtual Private Network (VPN) for securely transmitting power grid data. Using QKD, a quantum key is generated at User A's terminal and securely shared with User B through a quantum channel. This key is then used for encryption and decryption within a VPN channel, enabling secure communication between the two users. Rigorous testing with the Network Testing Tool (N11U) demonstrates the quantum VPN's capability to enhance security, while evaluations of throughput, delay, and encryption efficiency confirm its performance. The quantum-based approach provides an additional layer of security by detecting any interception attempts during key exchange, ensuring a highly secure communication channel.

Jia et al. [69] outlines a conceptual framework for ensuring secure communication within the EI using quantum technology. This work confirms the robust security measures implemented in corporate system communication by utilizing QKD methods that rely on optical fiber modes. Furthermore, the paper discusses network configurations, service accessibility, and system evaluation for scenarios in power system protection.

The authors of [70] implemented a PQC scheme called FrodoKEM, on hardware-constrained SM devices using System-on-a-Chip (SoC) architecture. This scheme is designed to secure data communication between SMs and the Meter Data Management System (MDMS), specifically targeting protection against eavesdropping attacks. Given that the use of PQC can lead to higher computational costs, they employed a Field Programmable Gate Array (FPGA) to accelerate cryptographic operations, thereby enhancing overall efficiency.

The work in [71] presents a quantum-secure privacy-preserving protocol for SM authentication in SGs. It enhances security through mutual authentication, key agreement and attack resistance, while guaranteeing identity confidentiality and message integrity. The protocol uses semi-QKD and hash functions for secure key distribution and data integrity, demonstrating efficient performance with lower overhead and optimized energy consumption.

Table 2.2: PQC and quantum cryptography based solutions.

Ref	Year	Secured Feature	Algorithm Family	Performance Metrics	Challenges and Limitations
[68]	2017	Confidentiality Integrity	QKD	- Throughput - Delay	- Cost considerations arise due to the potential expense of quantum VPN equipment and infrastructure. - The absence of established technical specifications and standards for quantum VPN in the power industry may pose regulatory challenges.
[69]	2019	Confidentiality Integrity	QKD	- Quantum Key Consumption - Data Encryption Success	- QKD schemes, particularly those relying on optical fiber modes, may encounter limitations concerning transmission range and signal degradation over extended distances. - The implementation of quantum secure communication technology within practical power system applications poses challenges due to its inherent complexity.
[70]	2022	Confidentiality Integrity	Lattice (FrodoKEM)	- Execution time - Relative time improvement - Processing time - Resource usage	- Like most PQC systems, FrodoKEM is still in its early stages of implementation. Although theoretical calculations demonstrate its performance, it's the real-world implementation that will make the difference.
[71]	2024	Authentication Privacy Integrity	Semi-QKD	Lower computation and communication overhead, energy efficiency	Requires implementation of semi-QKD, potential complexity in real-world deployment without quantum capabilities
[72]	2021	Confidentiality Authorization	Lattice	- Theoretical evaluation: computation and space complexity. - Experimental evaluation: time, space, and communication	- Security proofs for proposed ACE schemes outlined complexity and reliance on specific assumptions, like LWE which may hinder practical adoption.

2.2.3 Blockchain based solutions

Blockchain is an innovative technology that emerged with the introduction of Satoshi Nakamoto, who implemented it in Bitcoin in 2009 [73]. Although blockchain technology existed before the creation of Bitcoin, it was Bitcoin that marked its first practical application [74, 75].
Blockchain consists of decentralized and distributed ledger relying on cryptographic methods [76, 77]. Figure 2.2 (adopted from [75]) illustrates this ledger containing a series of linked or concatenated blocks, each containing a group of transactions. These transactions collectively calculate the Merkle root (the hash of all individual transactions within a block). This calculation uses a hierarchical data structure known as a Merkle tree, also known as a binary hash tree [78, 75].

Users initiate transactions by exchanging cryptocurrencies with each other. These transactions are then broadcast across the network, and each node in the network maintains a memory pool where it stores unconfirmed transactions. Miners (or validators) select a group of transactions from the memory pool for inclusion in a new block, with higher-reward transactions being prioritized.

Miners create the block by forming the header section, which includes information such as the previous block hash, the Merkle root, and the nonce number (the header specifications depend on the blockchain platform). In the case of Proof of Work (PoW), the first node to solve the mathematical puzzle earns the privilege of adding a new block. At this point, other nodes must verify the newly added block.

Blockchain can be classified into three categories: public, private, and consortium [79]. In a **public blockchain**, individuals can join the network, observe transactions, and actively participate in the consensus process without the need for authentication [75, 80]. Unlike the **private blockchain**, where pre-selected nodes can join the network, allowing for the identification of nodes in case of issues during block addition, this type of blockchain exhibits a form of centralization with a single controlling organization manipulating the blockchain. The **consortium blockchain** combines the benefits of both public and private blockchains. In contrast to a single organization being solely responsible for creating blocks, as seen in private blockchains, a consortium blockchain involves multiple organizations. Furthermore, unlike public blockchains that permit anyone to join the network, only nodes affiliated with a specific organization are allowed to join [81, 82].

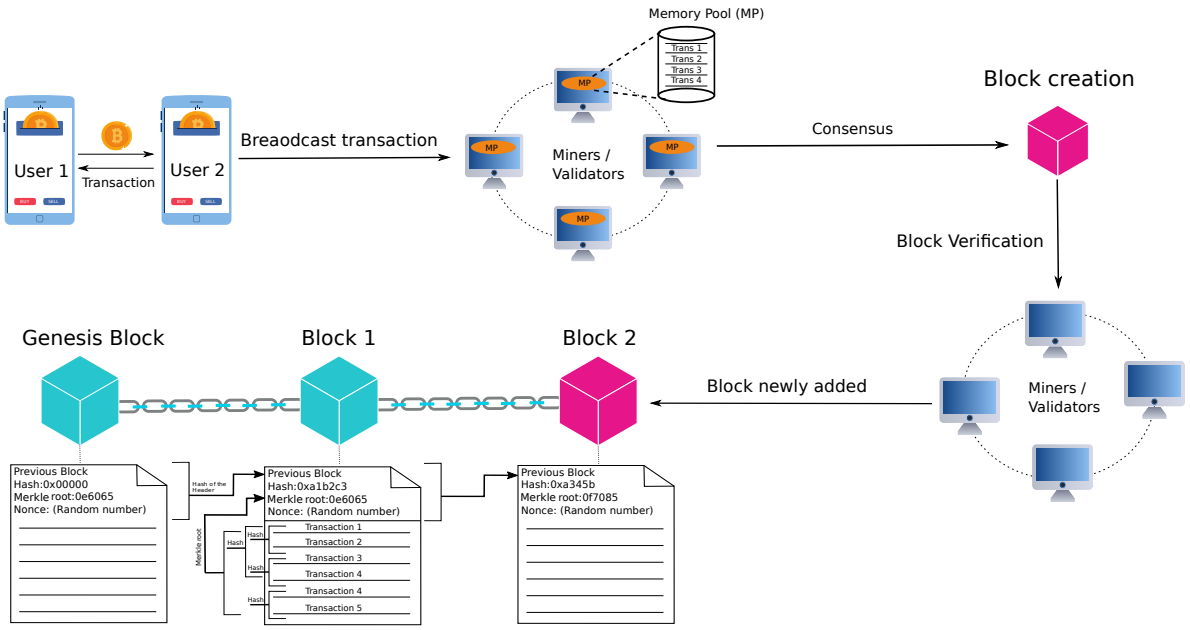


Figure 2.2: Transaction flow in blockchain.

2.2.3.1 Consensus algorithms

Blockchain security relies primarily on the consensus algorithm, which can be defined as the method of agreements between all participants, guaranteeing the validity of each block added to the blockchain [24]. There are many different types of consensus algorithms, of which, here we will highlight the most well-known ones.

1. **Proof of Work (PoW):** The main idea of the PoW consensus algorithm is to solve a cryptographic puzzle. This puzzle is difficult to solve but easy to verify. Miners continue generating a nonce value until the result of the hash function is below a predefined target. An example of the target value starts with a certain number of zeros. The target value is adjusted periodically to regulate the rate at which new blocks are added to the blockchain. In Bitcoin, for example, the target is set to maintain an average block time of 10 minutes. After every 2016 blocks, an adjustment is applied to ensure the consistency of the block creation rate [83].

2. **Proof of Stake (PoS):** In PoS, the need for solving computationally costly cryptographic puzzles is eliminated which is the problem in PoW [84, 85, 86]. The selection of validators is based on their stake, representing the coins they currently hold or have earned. This implies that validators with a greater number of coins have a higher probability of adding blocks. The rationale behind this

selection is that a validator who has risked more coins is considered more legitimate. However, this approach raises concerns about centralization, as it could lead to a centralized entity having the role of adding blocks to the network. To address this issue, a solution has been proposed: the randomization of the validator selection process [87, 88].

3. **Delegated Proof of Stake (DPoS):** DPoS represents an evolved iteration of PoS, effectively addressing the inefficiencies in computational resource utilization seen in PoW and mitigating centralization concerns present in PoS [89]. DPoS introduces a mechanism where predetermined nodes, known as “stakers," engage in a voting process to select “delegate nodes" responsible for block addition. Although DPoS successfully tackles centralization issues linked to the wealthiest nodes in PoS, it introduces a distinct form of centralization associated with a limited number of pre-selected nodes [83].

4. **Proof of Burn (PoB):** PoB enables the selection of the next block’s validator through a process of "burning" coins [83, 90]. The more coins you burn, the greater the likelihood of being chosen as the validator for the next block. Burning is accomplished by rendering the coins unspendable, achieved by transferring them to another account where they cannot be utilized. This method serves as a tangible demonstration of one’s commitment to the system [83].

5. **Delayed Proof of Work (DPoW):** Both PoS and PoW are susceptible to a 51% attack [91], a 51% attack occurs when a single miner or a group of miners gains control of more than half of a blockchain network’s total computing power. To address this vulnerability, DPoW introduces a solution by incorporating an additional blockchain known as the "notary chain." The original chain, which contains transactions, is then designated as the "target chain." In DPoW, miners compete to create new blocks on the notary chain. However, instead of computing the hash of transactions, the notary chain involves the hash of blocks [83]. This approach adds an extra layer of security because in the event of an attack on the target chain, the attacker must also manipulate the notary chain to alter the record of blocks [92].

2.2.3.2 Related works on blockchain based solutions

Several initiatives have been undertaken to enhance the security of EI and its components through the use of blockchain technology. Table 2.3 summarizes the works related to blockchain-based solutions, outlining their advantages and drawbacks.

Chen et al. [93] propose an authentication protocol based on the Practical Byzantine Fault Tolerance (PBFT) consensus mechanism. Authentication requests are issued by the terminals to the master node. The latter orchestrates the PBFT consensus with the slave nodes. Once the authentication has been successfully completed, the master node generates a session key and adds the terminal information to the blockchain.

Lu et al. [94] propose integration of Software-Defined Networking (SDN) and blockchain for secure distributed energy trading, using the Distributed Hash Table (DHT) network for transactions, that speeds up the process of finding information about data stored in the cloud. Sellers produce energy and store it in devices, and then buyers purchase through the manager and use Bloom's filter to obtain the best offers. Transactions between seller and buyer are completed and stored on the blockchain. However, the authors did not provide specific details on the SDN implementation.

In their subsequent work, Lu et al. [95] introduce a blockchain-based approach to the trading matching system. They implemented a black-box matching process, in which a trusted agent node performs the matching operations. This approach maximizes benefits for both parties. It ensures the protection of users' trade information and preserves the security and credibility of the matching scheme.

Ding et al. [96] focus on developing Energy Blockchain (EBC), a system that operates without the need for trusted intermediaries. EBC is designed as a secure blockchain framework for peer-to-peer energy trading within Industrial Internet of Things (IIoT) environments. IIoT entities engage directly in peer-to-peer energy trading using EBC. The system utilizes aggregators and energy purchasers/sellers to facilitate transactions, which are recorded in blocks, validated by authorized entities, and securely added to the blockchain, ensuring transparency. Additionally, the system ensures privacy and security through robust wallet security measures, transaction authentication, and data unforgeability.

The authors of [97] propose an electricity trading platform within a Community Energy Internet Cluster (CEIC), leveraging blockchain and Ethereum smart contracts to facilitate collaboration among multiple Community Energy Internet (CEI) systems within a specific region. The primary objective is to prevent

double spending. The proposed platform enables households to securely and transparently buy and sell excess electricity within the CEIC. Participants, acting as consumers or producers of energy, possess individual digital wallets linked to the blockchain network for secure transactions. Smart contracts execute transactions when both parties agree on the energy amount and price, ensuring automated and secure payment and delivery.

Si et al. [98] introduce an innovative privacy-preserving blockchain scheme tailored for smart parks integrating diverse energy sources. SMs diligently collect data on individual electricity consumption, employing encryption and perturbation techniques to safeguard privacy. Subsequently, the encrypted data is securely stored on the blockchain, facilitating analysis through gradient descent analysis. This approach enables accurate load change prediction, identification of energy demand patterns, and optimization of energy scheduling, all while prioritizing user privacy.

The paper [99] presents a framework that leverages Hyperledger Fabric blockchain technology to streamline clean energy transactions at photovoltaic (PV) electric vehicle charging stations. The framework establishes collaboration between a PV organization responsible for energy production and an EV organization managing charging stations. EVs are recharged at these stations equipped with SMs, which record energy consumption via off-chain blockchain technology, simplifying energy management for both EV and PV administrators. This integration guarantees the transparency, security, and efficiency of energy transactions.

The work in [100] presents a multi-blockchain scheme for authentication and authorization across domains, addressing the complex network integration within the EI, which encompasses interconnected systems such as the SG, IoT devices, and distributed renewable energy resources. Each domain initializes its blockchain network for authentication, using Cuckoo filters for efficient storage of user authentication data. When a user in domain A seeks access to resources in domain B, the Authentication Server (AS) in domain B checks the user's status, requests authentication information from the supervisory blockchain, issues a random challenge to the user and completes the authentication process by verifying the user's signature. In addition, the AS in domain A can revoke a user if necessary, while cross-domain access control is managed by role-based access control, facilitating secure data exchange in the event of successful authentication.

Table 2.3: Blockchain based solutions.

Ref	Year	Secured Features	Blockchain Platform	Advantages	Drawbacks
[93]	2018	Authentication	HyperLedger Fabric	Distributed and decentralized authentication.	PBFT consensus poses problems of latency, overhead and scalability as the number of nodes in the network increases.
[94]	2019	Privacy	N/A	Implementation of anonymous transactions and cloud data storage to protect user privacy.	The use of blockchain, SDN, and DHT technologies can introduce complexity in implementation and maintenance.
[95]	2020	Privacy	N/A	Improves efficiency, fairness and privacy in energy trading.	Dependence on agent nodes raises centralization issues, and scalability issues are not addressed in depth.
[96]	2021	Privacy	N/A	Applying the Stackelberg game to credit-based lending optimizes loan pricing and maximizes the utility of lending banks and borrowers in energy trading scenarios.	Potential lack of specific details on technical implementation, experimental validation or real-world deployment.
[97]	2021	Integrity	N/A	The use of the double auction mechanism with smart contracts facilitates the optimal matching of energy supply and demand.	Integrating decentralized energy transactions into existing energy markets and regulatory frameworks may require coordination and alignment with traditional utilities and network operators, raising integration issues.
[98]	2021	Privacy	N/A	By relying on distributed gradient descent and blockchain, the system enables precise analysis of energy consumption while preserving user privacy.	The computing resources required to run the distributed gradient descent algorithm, the blockchain network and the encryption processes may impose additional resource requirements on the system.
[99]	2022	Transparency	Hyperledger Fabric	Improving the efficiency of energy management through the integration of blockchain technology.	The framework does not include sufficient privacy protection measures to safeguard sensitive data.
[100]	2023	Authentication Authorization	HyperLedger Fabric	The use of a separate blockchain for each domain offers scalability in the efficient management of authentication data.	The use of multi-blockchain networks and cuckoo filters can make system architecture more complex.
[101]	2023	Transparency Integrity	Ethereum Virtual Machine	The use of Ethereum smart contracts guarantees the transparency and security of energy transactions, while off-chain processing minimizes the costs associated with blockchain transactions.	Integrating smart energy meters with Ethereum smart contracts requires technical expertise and resources, which poses deployment challenges.

N/A: Not Applicable.

The authors of [101] introduced an energy trading system leveraging Ethereum smart contracts and smart energy meters. In the proposed system, sellers and prosumers engage on the energy trading platform. Sellers are required to generate energy tokens adhering to the ERC20 token standard to denote their energy units. Upon agreement between the seller and prosumer regarding the energy price, the smart energy meters update the price information on the smart contract deployed on the blockchain, utilizing an off-chain concept. Payment is then seamlessly processed through the smart contract using cryptocurrency, thereby enabling the smooth transfer of energy tokens.

2.2.4 ML and DL based solutions

ML techniques are widely used for tasks such as classification, regression, and density estimation. They form the core of various applications, including computer vision, fraud detection, bioinformatics, malware detection, authentication, and speech recognition [102]. In this context, IDSs that rely on ML are designed to accurately identify and counter cyberthreats while maintaining efficient performance on varied datasets. By learning what is considered normal behavior, these models are able to flag up any deviation as a potential anomaly [103].

2.2.4.1 ML and DL models

There are many ML models used in the design of IDSs. This section briefly introduces the commonly used models, which are also discussed in this chapter.

Deep neural networks (DNNs) form the basis of many modern ML models. DNNs are capable of finding and learning representations from raw data, and performing feature learning and classification [104]. The convolutional neural network (CNN) is a specialized type of DNN, which uses convolutional layers to extract spatial features, making it highly effective for image and video processing [105].

For sequential data tasks, Long Short-Term Memory (LSTM) networks are commonly used. LSTMs are designed to capture long-term dependencies in data [106], though they can be computationally intensive and may face challenges with training speed. To address these issues, Gated Recurrent Units (GRUs) were developed. GRUs simplify the LSTM architecture by reducing the number of gates, which lessens computational load while maintaining similar performance in handling sequential information.

In scenarios where data privacy and decentralization are critical, Federated Learning (FL) offers a valuable approach. FL enables models to be trained across multiple devices or servers without sharing raw data, thereby preserving user privacy and leveraging diverse data sources.

On a different note, traditional ML techniques remain valuable for many tasks. Decision Trees (DT) provide a straightforward method by splitting data based on feature values, resulting in interpretable models. Building on this concept, Random Forests (RF) employ an ensemble of decision trees to improve prediction accuracy and reduce the risk of overfitting. Additionally, Support Vector Machines (SVM) are used for both classification and regression tasks; they work by finding the optimal hyperplane that separates data into distinct classes, making them robust for a range of problem types.

Explainable AI (XAI) techniques play a crucial role in making ML models more transparent and understandable, particularly in security applications where trust and clarity are essential. For instance, SHapley Additive exPlanations (SHAP) helps quantify the contribution of each input feature to the model's prediction, giving insight into which factors most influenced the outcome [107]. Similarly, Local Interpretable Model-agnostic Explanations (LIME) generates local approximations of a model's behavior to explain why a specific decision was made. These XAI methods not only enhance the interpretability of complex models like DNNs and CNNs but also support cybersecurity experts in validating and refining their IDSs.

2.2.4.2 Related works on ML and DL based solutions

Recent notable studies have focused on improving IDSs to effectively counter the growing cyber threats. The works discussed in this section are summarized in Table 2.4.

Tariq et al. [108] proposed a fog-edge-enabled SVM based FL IDS for SGs. This system addresses the privacy risks and latency issues associated with cloud-based IDS models by leveraging fog computing and FL to train IDS models on edge devices. The proposed model enhances intrusion detection performance and ensures data privacy by sharing only learning parameters rather than raw data.

Similarly, Osa et al. [109] designed and implemented a DNN-based IDS aimed at enhancing the detection of cyber-attacks on computer networks. The system utilizes a neural network with six hidden dense layers, each employing the ReLU (Rectified Linear Unit) activation function, and a softmax activated output layer. The study also applied the synthetic minority oversampling technique (SMOTE) and random sampling techniques to address data imbalance within the CICIDS2017 dataset [110].

Expanding on this, Khacha et al. [111] developed an IDS tailored for IoT networks by integrating CNN, LSTM, and GRU models, which were tested across both realistic and simulated traffic datasets. In a comparable context, Saadouni et al. [112] developed a hybrid deep learning model utilizing CNN and GRU architectures to enhance intrusion detection for IIoT networks. The model was evaluated using the Edge-IIoTset dataset [113] for both binary and multi-class classifications.

Further contributions to this field include those by Altunay et al. [114] who introduced an hybrid IDS using a combination of CNN and LSTM networks tailored for IIoT networks. Their model was evaluated using two datasets, UNSW-NB15 [115] and X-IIoTID [116], achieving high accuracy rates in both binary and multi-class classifications.

Moreover, Le et al. [117] proposed an IDS based on ensemble trees, specifically using DF and RF classifiers. Their method aims to enhance attack detection performance while providing explanations for the ML model predictions through the SHAP method. This approach addresses the complexity and resource demands of DNN, offering a more interpretable solution that supports cybersecurity experts in optimizing their decisions. The system was evaluated on the NF-BoT-IoT-v2 [118], NF-ToN-IoT-v2 [119], and IoTDS20 [120] datasets.

Sharma et al. [121] developed a deep learning-based IDS for IoT networks using a DNN model, which was evaluated using the UNSW-NB15 dataset [115]. Initially, the system achieved an accuracy of 84%, which increased to 91% after addressing class imbalance with synthetic data generation using Generative Adversarial Networks (GANs). In another study, Sharma et al. [122] proposed an IDS based on both DNN and CNN models, employing XAI techniques such as LIME and SHAP to improve model interpretability. This approach was tested on the NSL-KDD [123] and UNSW-NB15 [115] datasets.

Younisse et al. [124] proposed an IDS leveraging CNN and SHAP for explainability. Their study analyzed feature importance using KDE plots to explain SHAP results, enhancing the understanding of feature contributions in CNN models. This method helps in selecting relevant features and improving model performance by focusing on critical data aspects. The system was tested on the KDD 99 dataset [125].

Table 2.4: ML and DL based solutions.

Ref	XAI tool	Dataset	ML/DL model
[108]	N/A	CICIDS2017, NSL-KDD	FL, SVM
[109]	N/A	CICIDS 2017	DNN
[111]	N/A	Edge-IIoTset, NSL-KDD	CNN, LSTM, GRU
[112]	N/A	Edge-IIoTset	CNN, GRU
[114]	N/A	UNSW-NB15, X-IIoTID	CNN, LSTM
[117]	SHAP	IoTDS20, NF-BoT-IoT-v2, NF-ToN-IoT-v2	DT, RF
[121]	N/A	UNSW-NB 15	DNN
[122]	SHAP, LIME	UNSW-NB 15, NSL-KDD	DNN, CNN
[124]	SHAP, LIME	KDD 99	CNN

N/A: Not Applicable.

2.2.5 Hybrid based solutions

The use of traditional cryptography, PQC or even blockchain and quantum cryptography can enhance EI security. Another strategy is to combine these solutions to create a hybrid solution that benefits from the strengths of each, making security more robust. In this section, we explore the notable hybrid solutions in this field (see Table 2.5).

2.2.5.1 Related works on hybrid based solutions

Guan et al. [126] proposed a hybrid system, which consists of two levels: the first involves SMs in the form of private blockchain nodes, in which they store the amount of energy held by each user; the second level involves managers in the form of a consortium blockchain nodes that is responsible for balancing energy demand and supply; the user's energy information is encrypted using ECDSA and the hash function.

The authors of [127] propose a blockchain-based on lattice cryptosystem for secure EV charging in the EI, employing PBFT consensus. It addresses the risk of injecting malicious code in EV-charging station communication. Blockchain integration provides transparency within the EI-EV interface. The power grid initiates hierarchical key generation for cryptographic key generation. Power Distribution and Controller and Prosumers (PDCPs) validate charging information using public-private key pairs, while aggregators ensure secure communication and integrity of energy transactions, enhancing reliability in the EI and EV interface.

Table 2.5: Hybrid based solutions.

Ref Year	Secured Features	Security Solutions	Cryptosystem	Advantages	Challenges and Limitation
[126] 2020	Privacy Integrity	Blockchain Traditional cryptography	ECDSA and hash function	The decentralization nature of the proposed scheme.	Using two types of blockchain in the same system is twice as complex and costly.
[127] 2021	Transparency Confidentiality	Blockchain PQC	Lattice	Decentralization and quantum resistance of the proposed solution.	The implementation involved 3 organizations with 4 peers, which is not enough to prove the scalability of the solution, which is a major problem in EI.
[128] 2022	Authentication Privacy	Blockchain Traditional cryptography	ECC and CH	Users authenticate anonymously in the EI, bolstering privacy and confidentiality in transactions.	Secure storage and handling of private keys are imperative, posing challenges in key management and security maintenance.
[129] 2022	Privacy	Blockchain Traditional cryptography	ECC	Anonymity, transparency and distribution of the voting process.	Using the PoW consensus is too complicated in terms of computing costs.
[130] 2023	Confidentiality Integrity	Blockchain Traditional cryptography	ECC and hash function	The immutability and tamper-resistance properties are guaranteed through the use of blockchain.	The assumption of a trusted GC is not always practical or realistic in all EI environments.
[131] 2023	Authentication Integrity	Blockchain PQC	Lattice	Decentralization and transparency through the integration of blockchain technology.	Integration of complex cryptographic mechanisms with blockchain may challenge scalability.

The authors of [128] propose an anonymous authentication scheme using blockchain, ECC, and Chameleon Hash (CH), called EVAA (Editable and Verifiable Anonymous Authentication). In this scheme, users register on the energy management platform by providing their personal information. Subsequently, the platform generates a unique identity certificate for each user, which is utilized to guarantee anonymous authentication.

Decision making in EI network is a challenging task which can directly affect the security of the network. The voting protocol proposed by Hu et al. [129] is dedicated to this issue. This proposed protocol ensures a fair and transparent voting process where all nodes in the network can participate in the decision making process, which helps in the management of EI network, including energy production, distribution and consumption.

Chen et al. [130] propose an access control protocol using blockchain, ECC and hash function. This proposed protocol assumes Grid Companies (GC) to be considered as a trusted entity and involves cryographic operations to generate secure access control on the communication between GC and power supply side nodes. In addition, it involves blockchain for immutable identity.

The authors of [131] propose a post-quantum blockchain scheme tailored for the EI within the Internet of Vehicles (IoV) framework. This innovative approach combines blockchain technology with lattice-based signatures, enhancing authentication and transaction verification processes. The scheme's performance is evaluated using key metrics, including public key size, private key size, and signature size, to ensure its effectiveness and efficiency.

2.3 Analysis and discussion

Based on the previously discussed security solutions, we can draw the following conclusions.

- **Blockchain-based solutions:** Integrating blockchain technology with the EI can significantly enhance security by providing robust protection mechanisms. However, as shown in Table 2.3, each scheme has its own advantages and drawbacks. Key observations include:
  - The consensus algorithm must be chosen carefully, considering both resource limitations and the size of the network.

  - While the deployment of multiple blockchains within the EI framework can improve security, it also increases resource requirements and adds complexity to the system architecture.
  - Blockchain is inherently designed to prevent centralization and eliminate third-party control. Therefore, any blockchain-based protocol must avoid central control to fully leverage these benefits.

- **Traditional cryptography:** This remains the most established and widely adopted approach for securing communications and data. Each cryptographic method comes with specific performance costs, and the overall effectiveness largely depends on its integration within the protocol. It is important to avoid unnecessary computational overhead to maintain system efficiency.

- **Post-quantum and quantum cryptography:** The adoption of PQC and quantum cryptography in securing the EI presents a promising avenue. Although quantum cryptography offers enhanced security, as shown in Table 2.2, its implementation requires substantial modifications to the existing power system infrastructure, making it less cost-effective. In contrast, PQC similar in appearance to traditional cryptographic methods but based on more complex mathematical problems, emerges as the most practical solution in the context of quantum computing.

- **ML-based solutions:** ML-based solutions offer high accuracy in detecting cyber threats and are capable of learning complex patterns from network data. However, while models such as DNNs, CNNs and sequential models such as LSTMs and GRUs can effectively identify anomalies, their success often depends on the balance between performance and interpretability. Techniques such as XAI help demystify these models by explaining how individual characteristics influence their predictions. Nevertheless, the computational complexity and data dependency of these ML models must be carefully managed to ensure that they fit well into the overall EI security architecture.

- **Hybrid solutions:** Combining multiple technologies can address a range of security challenges. However, such solutions may also inherit the limitations of the individual technologies employed. Therefore, any security solution developed for the EI must align with its unique network characteristics to be truly effective.

2.4 Challenges in the existing literature

Although extensive research efforts have been devoted to enhancing the security of the EI, many challenges persist in the existing literature. These issues span various technological layers and security paradigms, reflecting the complexity of securing a dynamic and heterogeneous energy ecosystem. The following points highlight key open issues that still require attention for the development of a robust and secure EI:

- **Scalability**: While blockchain offers promising security enhancements, its scalability remains a challenge. The increasing number of IoT devices in the EI ecosystem demands security solutions that can efficiently handle large-scale deployments without compromising performance.
- **Quantum-resistance**: The rise of quantum computing poses a significant threat to current cryptographic methods. Although PQC is being developed to counteract this threat, the transition to quantum-resistant algorithms is still in its early stages and requires further research and standardization.
- **Interoperability**: The EI involves various devices, systems, and technologies, making interoperability a critical concern. There is a need for standardized protocols and frameworks to ensure seamless communication and security across different components of the EI.
- **Real-time for security operations**: The real-time nature of data monitoring and interconnection in the EI introduce challenges in ensuring timely and secure data processing. We need further development for the techniques to securely handle and analyze real-time data streams without introducing latency or vulnerabilities.
- **Energy efficiency of security operations**: Security solutions must be designed to be energy-efficient, given the resource-constrained nature of many IoT devices in the EI. Balancing robust security measures with minimal energy consumption is a critical area requiring innovative approaches.
- **Hybrid security solutions**: Combining multiple security techniques, such as blockchain, traditional cryptography, and quantum technology, into hybrid solutions is a promising approach. However,

integrating these diverse methods into a cohesive and efficient security framework poses significant technical and operational challenges.

- **Decentralization**: The existing power grid relies mainly on a trusted central authority to manage and control the network. Transforming the EI into a fully decentralized network is a challenge, and requires a major effort in terms of security.

2.5 Conclusion

In this chapter, we reviewed existing solutions proposed to secure the EI, highlighting several emerging technologies, including blockchain, PQC, quantum and traditional cryptography, as well as hybrid and ML-based approaches. We have provided an overview of the fundamental ideas, advantages and limitations of each solution, enabling us to identify the main challenges and open issues arising from these limitations. The aim of this chapter is to set out our motivation for proposing targeted solutions to address these limitations. Before going into our contributions, the next chapter presents the techniques, tools and metrics used for evaluation and simulation, which will be used later in our contributions.

Chapter 3

Simulation and evaluation techniques, tools, and metrics

3.1 Introduction

To demonstrate the effectiveness of any security solution, it is crucial to thoroughly evaluate its performance. Although security is the primary objective, the solution's lightweight nature is equally important, as it ensures efficient implementation. Additionally, practical deployment offers valuable insights into its applicability in real-world scenarios. Consequently, this chapter details the techniques, tools, and metrics employed to simulate and evaluate our contributions presented in Chapters 4, 5, and 6.

3.2 Security verification techniques

The security verification process aims to validate the system's ability to mitigate various potential attacks. To achieve this, we employed a combination of techniques: formal methods such as BAN logic, ProVerif, and AVISPA, along with informal approaches based on analyzing common attack scenarios to assess the system's resilience against diverse threats.

3.2.1 BAN logic

BAN logic is a formal model widely employed in the literature to assess the security of key distribution and authentication protocols. The model entails transforming protocol messages into logical formulas, applying logical rules of inference to these formulas, and making reasonable assumptions to determine whether the protocol can achieve its intended security goals [132]. We describe some preliminary concepts and notations below to understand BAN logic better.

Notations:

- $P \mid\equiv X$: P believes X, or P would be entitled to believe X is fundamental to the logic, as it allows the principal P to behave as though X is true.
- $P \triangleleft X$: P observes X. A message containing X has been sent to P, which has the ability to read and echo X (possibly after performing some decryption procedures).
- $P \mid\sim X$: P once said X. At some point, the principal P, transmitted a message that contained the statement X. It is uncertain whether the message was dispatched in the distant past or during the present execution of the protocol, but it is certain that P held the belief in X at that time.
- $P \Rightarrow X$: P controls X, which mean P possesses jurisdiction over X. The principal P, has the power and expertise to make authoritative decisions concerning X.
- $\#(X)$: X is fresh, indicating that X has not been transmitted in any message during any prior executions of the protocol.
- $\langle X \rangle_Y$: X combined with Y.
- $\{X\}_Y$: X encrypted using secret key Y.
- $P \overset{k}{\leftrightarrow} Q$: P and Q use the shared key k to communicate.
- $\overset{k}{\mapsto} P$: P has k as a public key.

Rules:

- R_1 : The session keys rule $\frac{P|\equiv\#(X),P|\equiv Q|\equiv X}{P|\equiv P\overset{K}{\longleftrightarrow}Q}$ assuming that principal P considers the session key K to be fresh and that principal P and Q both believe in X, which is a necessary parameter of the session key, principal P concludes that she and Q have the same session key K.
- R_2 : The jurisdiction rule $\frac{P|\equiv Q\Rightarrow X,P|\equiv Q|\equiv X}{P|\equiv X}$ if P acknowledges Q's authority over X, and Q believes X, then P will believe that X is true.
- R_3 : The nonce-verification rule $\frac{P|\equiv\#(X),P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$ if P believes X is fresh, and P believes Q once said X, then P believes Q believes X.
- R_4 : The message-meaning rule $\frac{P|\equiv P\overset{K}{\rightleftharpoons}Q,P\triangleleft\langle X\rangle_K}{P|\equiv Q|\sim X}$ if P believes that the secrete key K is shared with Q and P sees $\langle X\rangle_K$ then, P believes Q said X.
- R_5 : The freshness-conjuncatenation rule $\frac{P|\equiv\#(X)}{P|\equiv\#(X,Y)}$ if P believes X is fresh, then, P believes that (X, Y) is fresh.
- R_6 : The public key encryption rule $\frac{P|\equiv\overset{K}{\longmapsto},P\triangleleft\{X\}_{K^{-1}}}{P|\equiv Q|\sim X}$ if P believes that Q has a public key K, and receives the message X encrypted using a secret key K^{-1}, then, P believes Q said X.
- R_7 : The belief rule $\frac{P|\equiv X,P|\equiv Y}{P|\equiv(X,Y)}$ if P believes X, and P believes Y, then, P believes (X, Y).

To apply BAN logic effectively, it is essential to start by idealizing the protocol, which involves abstracting the messages exchanged into a simplified form. In addition, the initial beliefs of each participant must be clearly defined, including assumptions about shared keys, the freshness of nonces and trust in specific entities. By systematically applying the rules of BAN logic to these idealized messages, new beliefs can be derived from the established assumptions. The aim is to demonstrate that each participant ultimately trusts the authenticity of the messages and that the intended security properties, such as mutual authentication or key agreement, are successfully achieved.

3.2.2 ProVerif

ProVerif is an automated tool that allows users to verify the security properties, such as confidentiality, authentication, and integrity, of security protocols. It is built using the Prolog language, which enables

the tool to explore the different states and behaviors of the protocol. The security protocol is analyzed in three segments: the declaration segment, the request segment, and the execution segment [133].

- **Declaration segment:** In this initial phase, the protocol is modeled by declaring its fundamental components. This includes specifying cryptographic primitives (e.g. encryption, decryption, digital signatures), data types, communication channels, keys and other relevant parameters. Declarations are expressed in a formal language that captures the structure and intended behavior of the protocol within the framework of the Dolev-Yao adversary model [134], in which the attacker is assumed to have full control over the network but cannot break cryptographic assumptions.

- **Request segment:** During this phase, the security properties to be verified are formally stated in the form of queries. These queries generally concern the confidentiality of certain secrets, mutual authentication or the maintenance of message integrity. The user defines these properties by constructing assertions or goals that the protocol must satisfy. This segment guides ProVerif analysis, focusing on the protocol specific security objectives.

- **Execution segment:** In the final phase, ProVerif processes security declarations and queries using an automatic reasoning engine. Using its Prolog-based resolution mechanisms, the tool explores all possible executions of the protocol to determine whether the specified properties are respected. ProVerif uses a form of over-approximation to ensure that the analysis completes, even in the presence of potentially infinite state spaces. If a property is not respected, ProVerif usually provides a counter-example or attack trace that illustrates how the security violation may have occurred, enabling potential vulnerabilities to be identified and addressed.

3.2.3 AVISPA

AVISPA is considered one of the most recognized and reliable tools for checking and verifying the effectiveness of security protocols against attacks and threats [135]. AVISPA provides a formal security language known as High-Level Protocol Specification Language (HLPSL), which is used to specify the security properties and problems. It is based on modular expressions and roles. For the analysis technique, there exists four back-ends: the On-the-fly Model-Checker (OFMC), the SAT-based Model-Checker

(SATMC), the Constraint-Logic-based Attack Searcher (CL-AtSe) and the TA4SP protocol analyzer [136]. The AVISPA work process can be broken down into several steps:

- **Specification:** The first step is to write an HLPSL specification that defines the protocol roles (e.g. initiator, responder), the sequence of message exchanges and the cryptographic operations involved. At the same time, the security properties to be verified, such as authentication, confidentiality and integrity, are explicitly stated.
- **Translation and pre-processing:** AVISPA then translates the HLPSL specification into an intermediate formal model that captures the essential behaviors of the protocol. This abstraction process eliminates non-critical details while preserving the logical structure and security requirements of the protocol.
- **Analysis via multiple analysis engines:** AVISPA then uses the four analysis back-ends for verification, each using a different methodological approach:
  - **OFMC** dynamically generates the protocol state space during analysis, exploring all possible execution paths to detect vulnerabilities in real time.
  - **SATMC** converts the verification problem into a Boolean satisfiability problem (SAT) and uses efficient SAT solvers to verify violations of the specified security properties.
  - **CL-AtSe** relies on constraint logic programming to systematically search for potential attacks by applying logical inference rules and constraints that capture the capabilities of the adversary.
  - **TA4SP** uses tree automata techniques to model protocol executions as trees, enabling the tool to verify security properties using automata theoretic methods.
- **Interpretation of results:** After the analysis, AVISPA provides information on the security of the protocol. If no attacks are found, the protocol is considered safe with regard to the properties specified under the assumed conditions. Conversely, if vulnerabilities are detected, AVISPA generates an attack trace or counterexample that outlines the sequence of actions an adversary could perform to compromise the protocol.

3.3 Simulation tools

In this section, we outline the simulation tools used in our study to evaluate the performance and security aspects of the proposed solutions. Specifically, we use Hyperledger Fabric, a permissioned blockchain platform designed for enterprise applications, and Caliper, a benchmarking tool used to measure key performance metrics of blockchain networks. These tools provide a comprehensive framework for assessing the operational efficiency and scalability of distributed ledger systems in a controlled environment.

3.3.1 Hyperledger Fabric (HLF)

Hyperledger Fabric (HLF) is a permissioned blockchain platform developed under the auspices of the Linux Foundation [137]. As a distributed ledger technology (DLT) framework designed to meet enterprise requirements, HLF distinguishes itself by enabling the execution of smart contracts in several general-purpose programming languages, including Node.js, Java and Go. The platform is characterized by its unique execute-order-validate architecture, which organizes the transaction process into three distinct phases: execution, order and validation. Unlike public blockchain systems, HLF assigns specific identities to all network nodes, which are classified into three roles: clients (initiators of transactions), peers (maintainers of ledger copies and executors of smart contracts), and orderers (managers of transaction sequencing and consensus) [138, 139, 140].

In addition, the HLF incorporates a Member Service Provider (MSP), an essential design element responsible for managing user identities and controlling access to the network. The MSP uses a CA to verify and authenticate users, ensuring that only authorized entities participate in the blockchain network.

3.3.2 Caliper benchmark

Caliper is a benchmarking tool designed to evaluate the efficiency of blockchain networks. The tool focuses on measuring three main metrics: transaction latency, throughput, and resource usage. Transaction latency quantifies the time it takes for a transaction to be recorded in the ledger, while throughput measures the number of transactions processed per second. Resource usage, meanwhile, monitors CPU and network memory consumption during the benchmarking process. By providing quantitative information

on these key performance indicators, Caliper is a valuable resource to assess the scalability and operational efficiency of blockchain implementations [141].

3.4 Evaluation metrics

The main metrics for evaluating the performance of the proposed security solutions are computational cost, communication cost and storage cost. These metrics provide a comprehensive assessment of cryptographic efficiency and load on the system.

- **Computational cost:** Refers to the amount of computing resources, such as processor cycles and memory usage, required to perform various cryptographic operations. This cost is usually measured in milliseconds (ms) to quantify the execution time. In this thesis, we adopt the computational costs required to perform each cryptographic operation based on the studies of Wang et al. [142] and Sadhukhan et al. [143], as shown in Table 3.1.

Table 3.1: Computational cost for various cryptographic operations.

Notations	Description	Time (ms)
$T_{se/d}$	Symmetric encryption/ decryption	0.0046
$T_{ae/d}$	Asymmetric encryption/ decryption	3.850
T_h	String to number hash	0.089
T_H	String to point hash	12.418
T_b	Bilinear pairing	48.660
T_a	ECC point addition	0.118
T_m	ECC point multiplication	2.226
T_{QKD}	Quantum Key distribution	0.0159

- **Storage cost:** Defined as the amount of data that must be stored locally by the system, usually quantified in terms of data size (bits).

- **Communication cost:** Refers to the volume of data exchanged between the different entities of the system. It is generally calculated as the product of the number of messages transmitted and the size of each message. Table 3.2 details the size of the various data elements used. Table 3.2 details the size of the various data elements used.

Table 3.2: Size of data elements used.

Notations	Description	Size (bits)
$\|ECC\|$	ECC point	320
$\|C_s\|$	One symmetric ciphertext	128
$\|H\|$	Hash outputs	256
$\|ID\|$	Identifications	64
$\|T\|$	Timestamps	32
$\|SK\|$	QKD key	128

3.5 Conclusion

This chapter detailed the techniques, tools, and metrics that will be used to verify and evaluate the contributions presented in the subsequent chapters. We explored several security verification methods, namely BAN logic, ProVerif, and AVISPA, to assess the resilience of the system to potential attacks. Additionally, we introduced simulation tools such as Hyperledger Fabric and Caliper, which are used to measure key performance parameters of the blockchain network.

Furthermore, we described the main evaluation metrics used to assess cryptographic operations, including computational cost, storage cost, and communication cost.

In the next chapter, we will present our proposed methods and protocols for improving the security of the EI system, taking into account the advantages and limitations of existing contributions.

Part II

Contributions

Chapter 4

SemAuth: Secure and Efficient Mutual Authentication Protocol in IoT-based EI using Blockchain

4.1 Introduction

The main characteristic of the EI is the integration of various technologies via the internet. However, this fusion poses new security challenges, particularly in terms of authentication between various IoT devices in the EI [3].

A significant amount of research has been dedicated to the security of EI, SG, and IoT. Despite the numerous authentication systems available, many of them do not meet essential security criteria. Primarily, these approaches depend on a central authority for both registration and authentication, thus reducing their decentralization. Additionally, relying on a central authority presents scalability issues, as the integrity of the system is compromised in case of authority failure. Moreover, several proposed solutions lack implementation specifics or fail to execute properly, making it challenging to ensure their effectiveness in real-world scenarios.

In this chapter, we propose **SemAuth** (Secure and Efficient Mutual Authentication Protocol), a blockchain-based solution for device authentication in IoT-based EI. Building upon the foundational work

of Wang et al. [142], our protocol addresses the limitations outlined in Section 4.2. Specifically, we extend their framework by integrating smart contracts to streamline user registration and signature verification processes. The enhanced protocol, along with its formal security analysis, is elaborated in Sections 4.3 and 4.4, respectively.

A comparative analysis against recent studies in Section 4.5 demonstrates the superior efficiency and robustness of SemAuth. To validate its practicality, we implement the protocol on the HLF platform, with performance evaluations conducted using the Caliper benchmarking tool (Section 4.6). For clarity, we adopt the abbreviation **SEMAGrid** (Secure and Efficient Mutual Authentication Protocol for Smart Grid under Blockchain) to reference Wang et al.'s original protocol throughout this chapter.

4.2 Overview of SEMAGrid protocol

Key notations employed in this work are summarized in Table 4.1.

Table 4.1: Key notations of SemAuth protocol.

Notation	Description
RA	Registration Authority
SM_i	Smart Meter
UC_j	Utility Center
BC	Blockchain
SC	Smart Contract
$\mathbb{G}$	A general cyclic group
$E(\mathbb{F}_p)$	An elliptic curve over a finite field $\mathbb{F}_p$
q	A large prime number
G	A generator of $\mathbb{F}_p$ with the order q
Z_q^*	A set of integers no larger than q
P_{pub}	The system master public key
k	The system master private key
h_1	A hash function from $\{0,1\}^*$ to $\mathbb{G}$
h_2	A hash function from $\{0,1\}^*$ to Z_q^*
Sig_{SM_i}, Sig_{UC_j}	Signature of SM_i and UC_j
HID_{SM_i}, HID_{UC_j}	Hashed ID of SM_i and UC_j
Δt	Maximum transmission delay

The authentication protocol outlined in [142] for SG systems incorporates a smart contract deployed on the blockchain. The system model is depicted in Figure 4.1.

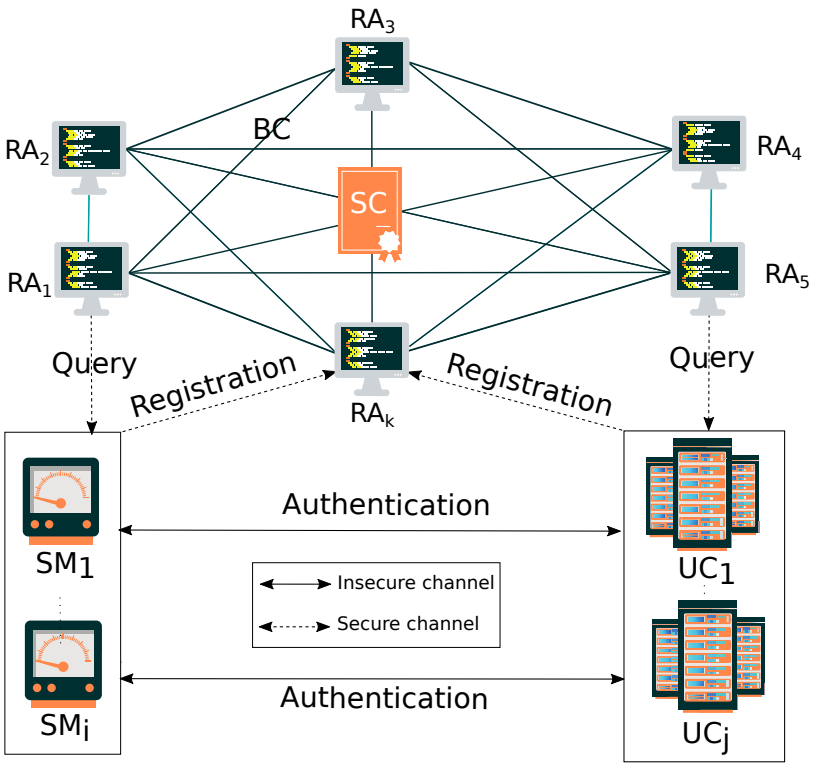


Figure 4.1: System model of SEMAGrid protocol.

The objective of this protocol is to decentralize the authentication process, requiring a sequence of steps to guarantee secure and authorized access to resources:

- The system initialization begins with the initiator generating a public and private master key using ECC. In this process, $k \in Z_q^*$ denotes the private key, and $P_{pub} = k \cdot G$ represents the public key.
- During the registration process, SM_i and UC_j sends the hash of the identifier to RA_k. Subsequently, RA_k verifies the ledger records to ascertain if the user has already been registered by invoking the *Registration* function in the smart contract. If the user's information is not found in the records, the *Registration* function proceeds to generate a public key and signature in the following manner:
  1. Randomly select: $r_{SM_i} \in Z_q^*$.
  2. Compute the public key: $R_{SM_i} = r_{SM_i} \cdot G$.
  3. Calculate the signature: $Sig_{SM_i} = k + r_{SM_i} \cdot e_{SM_i}$, where $e_{SM_i} = h_1(P_{pub}||R_{SM_i}||HID_{SM_i})$.
- RA_k transmits the public key R_{SM_i} and signature Sig_{SM_i} to SM_i. SM_i can validate the signature by

verifying if $Sig_{SM_i} \cdot G \stackrel{?}{=} P_{pub} + e_{SM_i} \cdot R_{SM_i}$. Upon registration, each party holds a signature and public key, which will be utilized to establish a session key during the authentication process.

- To generate a session key, SM_i chooses a random value, $a \in Z_q^*$, and computes two values: $A = a \cdot G$ and $V_1 = a \cdot Sig_{SM_i}$. Subsequently, SM_i transmits the tuple $\{R_{SM_i}, A, t_1, HID_{SM_i}, V_1\}$ to UC_j, where t_1 denotes a timestamp.

- Upon receiving the tuple $\{R_{SM_i}, A, t_1, HID_{SM_i}, V_1\}$ from SM_i, UC_j first verifies the freshness of the timestamp t_1 ($t_1 - t_1' \leq \Delta t$) and then validates the integrity of V_1 using the equation: $V_1 \cdot G^2 \stackrel{?}{=} A \cdot (h_1(P_{pub}||R_{SM_i}||HID_{SM_i}) \cdot R_{SM_i} + P_{pub})$. If the equation is correct, UC_j selects a random number $b \in Z_q^*$ and computes $B = b \cdot G$. Subsequently, UC_j generates the session key as: $SK_{UC_j} = h_2(A||B) \oplus Sig_{SM_i}$, along with $V_2 = Sig_{UC_j} \cdot A \cdot B \oplus Sig_{SM_i}$. Finally, UC_j sends the tuple $\{B, SK_{UC_j}, V_2, t_2\}$ as a response to SM_i.

- Upon receiving the tuple $\{B, SK_{UC_j}, V_2, t_2\}$ from UC_j, SM_i verifies the freshness of the timestamp: $t_2 - t_2' \leq \Delta t$. If the timestamp is fresh, SM_i proceeds to check the validity of V_2 and SK_{UC_j} as follows: $V_2 \stackrel{?}{=} a.B.(P_{pub} + h_1(P_{pub}||R_{UC_j}||HID_{UC_j}).R_{UC_j}) \oplus Sig_{SM_i}$ and $SK_{UC_j} \oplus Sig_{SM_i} \stackrel{?}{=} h_2(A||B)$. If both verifications succeed, SM_i computes the session key, $SK_{SM_i} = h_2(A||B) \oplus Sig_{SM_i}$, and stores it along with SK_{UC_j} as the session keys for future communication with UC_j.

While the SEMAGrid protocol offers certain advantages, it is susceptible to a variety of security vulnerabilities. These vulnerabilities encompass a range of potential attacks, including, but not limited to, MITM attacks, session hijacking, impersonation attacks, and information disclosure. A detailed discussion of each of these security vulnerabilities is presented below:

- **MITM attack:** In the authentication phase, an adversary $\mathcal{A}$ can execute an MITM attack by intercepting communication between SM_i and UC_j. The attack proceeds as follows (a visual illustration of this attack is presented in Figure 4.2):

  1. Over a public channel, SM_i transmits the tuple $\{R_{SM_i}, A, t_1, HID_{SM_i}, V_1\}$ to UC_j.
  2. The adversary $\mathcal{A}$ intercepts the message, locally saves the value of A, and generates a new value $A' = a'.G$, to calculate a new $V1' = Sig_{SM_i}.a'$. Subsequently, it transmits the regenerated tuple: $\{R_{SM_i}, A', t_1^*, HID_{SM_i}, V_1'\}$ to UC_j.

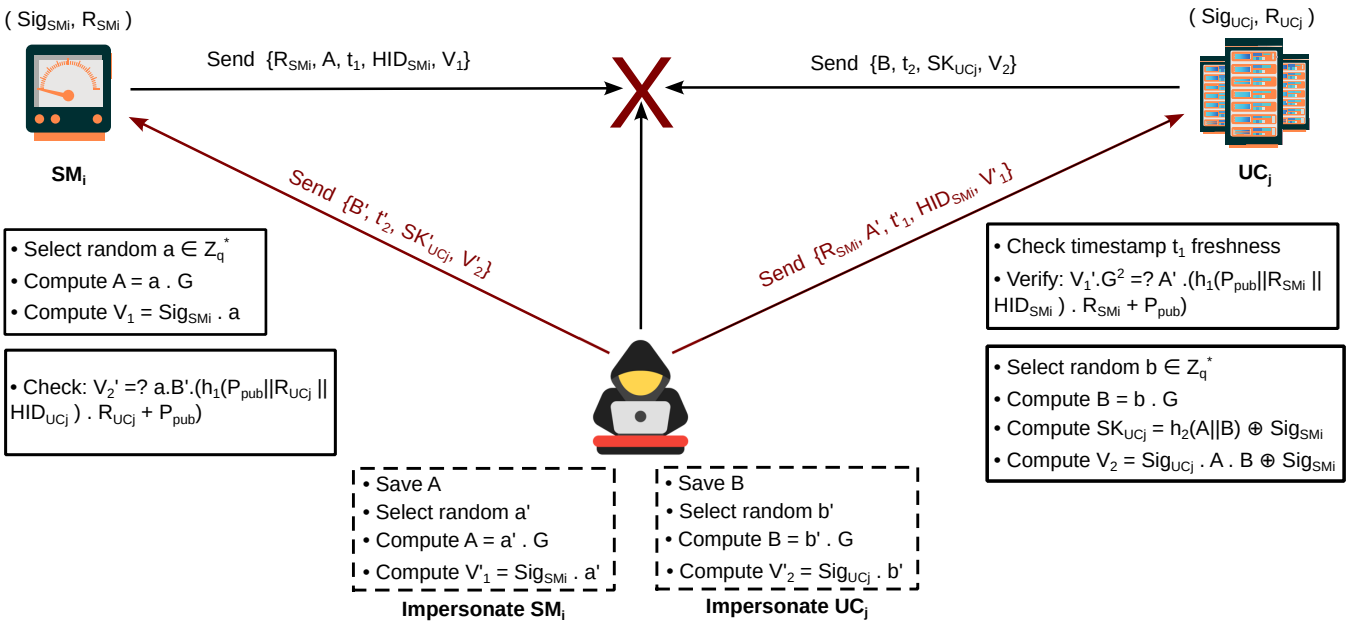


Figure 4.2: MITM attack on SEMAGrid protocol.

3. UC_j verifies the equation:

$$\begin{aligned} V_1'.G^2 &\stackrel{?}{=} A'.(h_1(P_{pub}||R_{SM_i}||HID_{SM_i}).R_{SM_i} + P_{pub}) \\ &\stackrel{?}{=} a'.G(h_1(P_{pub}|||R_{SM_i}||HID_{SM_i}).R_{SM_i} + P_{pub}) \\ &\stackrel{?}{=} a'.G(e_{SM_i}.R_{SM_i} + k.G) \\ &\stackrel{?}{=} a'.G(e_{SM_i}.r_{SM_i}.G + k.G) \\ &\stackrel{?}{=} a'.G^2(e_{SM_i}.r_{SM_i} + k) \\ V_1'.G^2 &= Sig_{SM_i}.a'.G^2 \end{aligned} \tag{4.1}$$

Which will be true, UC_j then generates a random number b, and compute $B = b.G$, $SK_{UC_j} = h_2(A'||B) \oplus Sig_{SM_i}$, and $V_2 = Sig_{UC_j}.A'.B \oplus Sig_{SM_i}$.

4. The adversary $\mathcal{A}$ intercepts the message, in the same way as the first one, he will save B locally, select a random number b', compute the new value of B' using b': $B' = b'.G$, $V_2' = Sig_{UC_j}.A.B' \oplus Sig_{SM_i}$, and $SK'_{UC_j} = h_2(A||B')$. Then transmitted the newly generated tuple: $\{B', SK'_{UC_j}, V_2', t_2^*\}$ to SM_i.

5. SM_i verifies the equation:

$$\begin{aligned} V_2' &\stackrel{?}{=} a.B'.(P_{pub} + h_1(P_{pub}||R_{UC_j}||HID_{UC_j}).R_{UC_i}) \oplus Sig_{SM_i} \\ &\stackrel{?}{=} a.B'.(k.G + h_1(P_{pub}||R_{UC_j}||HID_{UC_j}).R_{UC_i}) \oplus Sig_{SM_i} \\ &\stackrel{?}{=} a.b'.G.(k.G + h_1(P_{pub}||R_{UC_j}||HID_{UC_j}).R_{UC_i}) \oplus Sig_{SM_i} \\ &\stackrel{?}{=} A.b'.G.(k + h_1(P_{pub}||R_{UC_j}||HID_{UC_j}).R_{UC_i}) \oplus Sig_{SM_i} \\ &\stackrel{?}{=} A.B'.(k + e_{UC_j}.R_{UC_i}) \oplus Sig_{SM_i} \\ &= A.B'.Sig_{SM_i} \oplus Sig_{SM_i} \end{aligned} \tag{4.2}$$

   As the verification is true, SM_i will compute the SK'_{SM_i} as follows: $SK'_{SM_i} = h_2(A||B') \oplus Sig_{SM_i}$.

6. In the other side UC_j will also compute SK'_{UC_j} is the same way: $SK'_{UC_j} = h_2(A'||B) \oplus Sig_{UC_j}$.
7. A final verification should be performed in each party: $SK'_{UC_j} \oplus Sig_{SM_i} \stackrel{?}{=} h_2(A'||B)$ in UC_j side, and $SK'_{SM_i} \oplus Sig_{UC_j} \stackrel{?}{=} h_2(A||B')$ in SM_i side.
8. In a normal situation, SK'_{SM_i} should be equal to SK'_{UC_j}, but in this case, $SK'_{SM_i} \neq SK'_{UC_j}$, while this cannot be detected by SM_i and UC_j, and the attacker $\mathcal{A}$ has here generated two keys: SK'_{SM_i} to communicate with SM_i and SK'_{UC_j} to communicate with UC_j.

- **Session hijacking:** In the same way, as demonstrated in MITM attack, an attacker can intercept and take control of a session that has been established between two communicating parties using SK'_{UC_j} and SK'_{SM_i}. This allows the attacker to hijack the session and gain unauthorized access to any sensitive information that is exchanged between the parties.

- **Impersonation attack:** An unauthorized individual gaining access to restricted information or systems may be a significant security concern, and an adversary $\mathcal{A}$ could potentially achieve this by impersonating a legitimate participant in the communication using valid session keys, SK'_{UC_j} and SK'_{SM_i}.

- **Information disclosure:** Upon carrying out the MITM attack, an intruder can intercept and read the messages being transmitted between the two parties, leading to the compromise of sensitive information and systems. The attacker can gain control of the established session utilizing the session keys, SK'_{UC_j} and SK'_{SM_i}.

Emphasizing the gravity of the security vulnerability present in SEMAGrid protocol, it is crucial to address and resolve these security concerns to prevent any potential harm to users.

4.3 Proposed scheme

Our proposed protocol aims to enhance the security of SEMAGrid protocol, by addressing the identified security limitations. To achieve this, we have designed an improved protocol that specifically targets the authentication and key establishment phases while keeping the initialization and registration phases identical to those in SEMAGrid protocol. During the modified phase, registered entities, SM_i and UC_j can establish a session key by following a set of defined steps (see Figure 4.3).

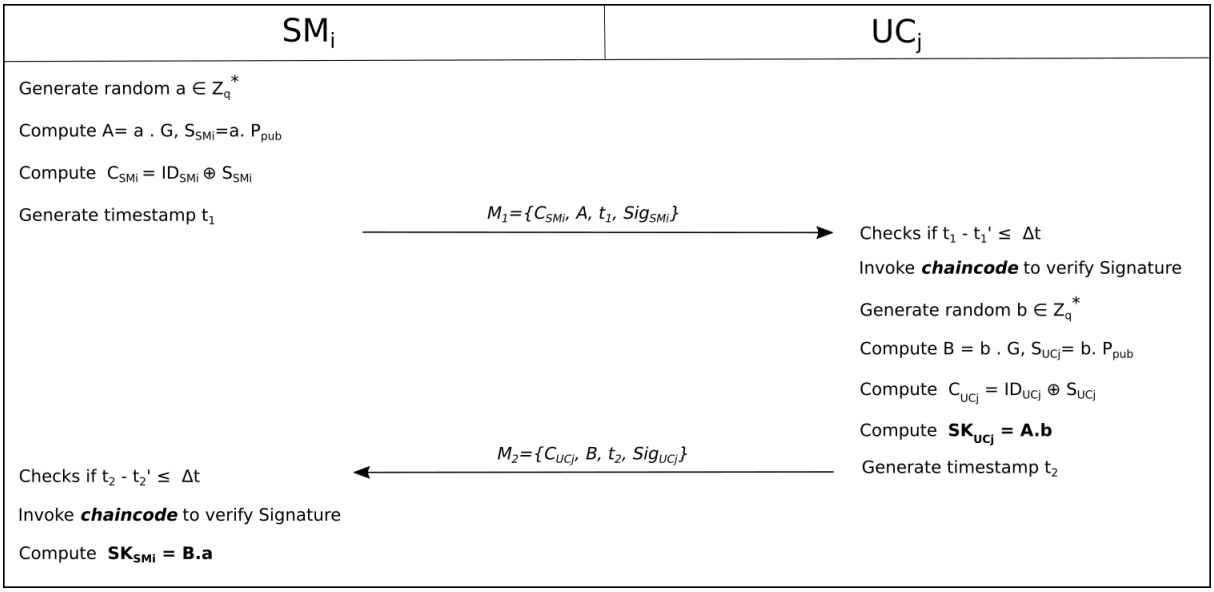


Figure 4.3: Modified phase in SemAuth protocol.

1. To begin, SM_i selects a random number, $a \in Z_q^*$ and computes: $A = a.G$ to be used later in the computation of the session key, and Sig_{SM_i}. SM_i computes $S_{SM_i} = a.P_{pub}$ and $C_{SM_i} = ID_{SM_i} \oplus S_{SM_i}$, to be used later to verify the signature ownership, where ID_{SM_i} is the identifier for SM_i. SM_i then sends the tuple $\{C_{SM_i}, A, t_1, Sig_{SM_i}\}$ to UC_j.

2. Upon receiving the tuple, UC_j verifies the signature using the *SignatureVerify* function from smart contract to check if SM_i is the owner of the signature by verifying the following:

$$Sig_{SM_i} \stackrel{?}{=} P_{pub} + h_1(P_{pub}||R_{SM_i}||h_1(C_{SM_i} \oplus (A.k))).R_{SM_i} \tag{4.3}$$

If the verification does not match, UC_j rejects the authentication; Otherwise, UC_j selects a random number, $b \in Z_p^*$ and computes: $B = b.G$, $S_{UC_j} = b.P_{pub}$, and $C_{UC_j} = ID_{UC_j} \oplus S_{UC_j}$. UC_j then computes the session key, $SK_{UC_j} = A.b$, and sends the tuple $\{C_{UC_j}, B, t_2, Sig_{UC_j}\}$ to SM_i.

3. SM_i also verifies the signature using the *SignatureVerify* function to confirm that UC_j is the owner of the signature. If the verification is successful, SM_i computes: $SK_{SM_i} = B.a$, and stores it for future communication.

Prove: The correctness of Equation 4.3 is explained as follows:

$$\begin{aligned}
Sig_{SM_i}.G &\stackrel{?}{=} P_{pub} + h_1(P_{pub}||R_{SM_i}||h_1(C_{SM_i} \oplus (A.k))).R_{SM_i} \\
&\stackrel{?}{=} P_{pub} + h_1(P_{pub}||R_{SM_i}||h_1(C_{SM_i} \oplus (a.G.k))).R_{SM_i} \\
&\stackrel{?}{=} P_{pub} + h_1(P_{pub}||R_{SM_i}||h_1(C_{SM_i} \oplus (a.P_{pub}))).R_{SM_i} \\
&\stackrel{?}{=} P_{pub} + h_1(P_{pub}||R_{SM_i}||h_1(C_{SM_i} \oplus S_{SM_i}))).R_{SM_i} \\
&\stackrel{?}{=} P_{pub} + h_1(P_{pub}||R_{SM_i}||h_1(ID_{SM_i})).R_{SM_i} \\
&\stackrel{?}{=} k.G + e_{SM_i}.r_{SM_i}.G \\
&= (k + e_{SM_i}.r_{SM_i}).G
\end{aligned} \tag{4.4}$$

Prove: The correctness of a session key is described as follows: $SK_{UC_j} \stackrel{?}{=} A.b \stackrel{?}{=} a.G.b \stackrel{?}{=} B.a = SK_{SM_i}$.

4.4 Security evaluation

In this section, we provide a detailed analysis of how our proposed enhancement offers a robust defense against commonly known attacks through informal security analysis. Additionally, we verify that our solution satisfies the necessary security properties with a formal security analysis using BAN logic and the ProVerif tool.

4.4.1 Informal security analysis

- **MITM attack:** In order to run an MITM attack, an intruder is required to execute a series of steps. Initially, the attacker must generate a random number $a' \in Z_p^*$ and calculate $A' = a'.G$. The attacker then computes $S'_{SM_i} = a'.P_{pub}$. Next, the attacker generates a valid identification ID and computes $C'_{SM_i} = ID \oplus S'_{SM_i}$. The attacker then sends these parameters along with the signature of the legitimate user to the recipient.

  Upon receiving these parameters, the *SignatureVerify* function is invoked by UC_j to verify the authenticity of the signature. Since the ID that was sent by the attacker does not match the ID_{SM_i} used to generate the signature, the authentication request is rejected and the invocation of the chaincode ends. Consequently, the MITM attack is effectively mitigated.

- **Impersonation attack:** An attacker could potentially send a message to UC_j containing SM_i's valid ID and other parameters that have been generated or intercepted by the attacker. However, due to the computational complexity of determining SM_i's ID, it would be challenging for the attacker to pass the smart contract check. Even if the attacker were to intercept C_{SM_i}, it would be impossible to impersonate SM_i. As a result, it can be concluded that the enhanced scheme is resistant to impersonation attacks.

- **DoS attack:** The distributed architecture of blockchain presents a notable advantage in mitigating DoS attacks. This is achieved by storing authentication records across multiple nodes in the network, making it challenging for attackers to focus on a single point of vulnerability and interfere with the authentication process.

- **Replay attack:** To prevent replay attacks, timestamps can be used to verify the validity of a message's timestamp compared to the most recently received one. This approach ensures the integrity and freshness of transmitted data. In our protocol, each party checks the freshness of the timestamp upon receiving a message by verifying whether: $t_1 - t'_1 \leq \Delta t$. This enhancement significantly strengthens the protocol's ability to resist replay attacks.

- **Decentralization:** Our enhanced protocol heavily relies on the use of blockchain technology, which provides an inherent level of security due to its decentralization. The decentralized nature of

blockchain mitigates the risk of centralized attacks, making it much more difficult for attackers to manipulate the system. Furthermore, the difficulty of carrying out a 51% attack on a blockchain network (as it is expensive, especially for large scale network) adds an extra layer of security to our protocol, ensuring that the system remains secure and resilient against attacks.

- **Perfect forward secrecy:** Our protocol provides the crucial property of perfect forward secrecy, which is a fundamental feature in cryptography that ensures the confidentiality of past and future sessions, even if an attacker gains access to the private key used for session encryption. This is achieved by generating a unique session key for each session, independent of the long-term private key. The use of A and B to establish a one-time session key in our protocol is the key element that guarantees perfect forward secrecy.

- **Mutual authentication:** Mutual authentication is a critical security feature that ensures the legitimacy of both parties involved in a communication session. In our protocol, mutual authentication is achieved by utilizing a shared secret key. To authenticate each other, both parties must independently compute $SK_{SM_i} = a.B$ or $SK_{UC_j} = b.A$, where the values of a and b are only known to the legitimate parties. This process confirms that each party is in possession of the correct shared secret key, which is essential for secure communication.

4.4.2 Formal security analysis

- **BAN logic:** We examine the security of proposed protocol through the application of BAN logic where our objective is to accomplish the following four goals:

  $G_1 : SM_i \mid\equiv (SM_i \overset{SK}{\longleftrightarrow} UC_j)$

  $G_2 : UC_j \mid\equiv (SM_i \overset{SK}{\longleftrightarrow} UC_j)$

  $G_3 : SM_i \mid\equiv UC_j(SM_i \overset{SK}{\longleftrightarrow} UC_j)$

  $G_4 : UC_j \mid\equiv SM_i(SM_i \overset{SK}{\longleftrightarrow} UC_j)$

  The idealized transmitted messages are presented as follows:

  $M_1 : SM_i \longrightarrow UC_j : \{C_{SM_i}, A, t_1, Sig_{SM_i}\}$

  $M_2 : UC_j \longrightarrow SM_i : \{C_{UC_j}, B, t_2, Sig_{UC_j}\}$

The foundational presumptions of the enhanced protocol are delineated as follows:

$A_1 : SM_i \mid\equiv \#(a)$

$A_2 : UC_j \mid\equiv \#(b)$

$A_3 : SM_i \mid\equiv UC_j \Rightarrow B$

$A_4 : UC_j \mid\equiv SM_i \Rightarrow A$

$A_5 : SM_i \mid\equiv SM_i \overset{A}{\leftrightarrow} UC_j$

$A_6 : UC_j \mid\equiv UC_j \overset{B}{\leftrightarrow} SM_i$

A_1 and A_2 denote that a and b are randomly generated numbers by SM_i and UC_j, respectively. Given that A is equal to a multiplied by G and B is equal to b multiplied by G, we can derive A_3 and A_4.

The main proofs are stated as follows:

- We demonstrate that our improved method accomplishes G_1 and G_3. Now, starting from the mapping $M_2 : UC_j \longrightarrow SM_i : \{C_{UC_j}, B, t_2, Sig_{UC_j}\}$, we have, $SM_i \triangleleft (C_{UC_j}, B, t_2, Sig_{UC_j})$. Using S_1 and A_5 along with the rule R_4 (see Section 3.2.1 for a detailed list of rules), we derive $S_2 : SM_i \mid\equiv UC_j \mid\sim B$. Using S_2, A_2, and applying the rule R_5 and R_3, we obtain $S_3 : SM_i \mid\equiv UC_j \mid\equiv B$. Based on S_3, A_3, and the rule R_2, we infer that $SM_i \mid\equiv B$. Since $SK_{SM_i} = B.a$, we use A_1, S_4, and S_3 to deduce $G_1 : SM_i \mid\equiv (SM_i \overset{SK_{SM_i}}{\longleftrightarrow} UC_j)$. Employing G_1, A_3, and the rule R_1, we derive $G_3 : SM_i \mid\equiv UC_j \mid\equiv (SM_i \overset{SK_{SM_i}}{\longleftrightarrow} UC_j)$.
- We show that our improved method achieves G_2 and G_4. Again, starting from the mapping $M_1 : SM_i \longrightarrow UC_j : \{C_{SM_i}, A, t_1, Sig_{SM_i}\}$, we have, $UC_j \triangleleft (C_{SM_i}, A, t_1, Sig_{SM_i})$. Using S_1 and A_6 along with the rule R_4, we derive $S_2 : UC_j \mid\equiv SM_i \mid\sim A$. Using S_2, A_1, and applying the rules R_5 and R_3, we obtain $S_3 : UC_j \mid\equiv SM_i \mid\equiv A$. Based on S_3, A_4, and the rule R_2, we infer $UC_j \mid\equiv A$. Since $SK_{UC_j} = A.b$, we use A_2, S_4, and S_3 to deduce $G_2 : UC_j \mid\equiv (UC_j \overset{SK_{UC_j}}{\longleftrightarrow} SM_i)$. Employing G_2, A_4, and the rule R_1, we derive, $G_4 : UC_j \mid\equiv SM_i \mid\equiv (UC_j \overset{SK_{UC_j}}{\longleftrightarrow} SM_i)$.

- **ProVerif:** The specifics of the declaration and query segments are provided in Figures 4.4 and 4.5, respectively. The running segment encompasses the SM_i, UC_j, and RA processes, as well as the

main process, which are presented in Figures 4.6, 4.7, and 4.8. In the declaration segment, variables are defined with their respective types, communication channels are established, and functions are defined in the query segment to deduce the security of the enhanced scheme. The running segment is composed of four sub-processes: ProcessSMi, ProcessUCj, ProcessRA, and the main process, which represent the operations of SM_i, UC_j, RA, and the main process, respectively. The outcomes of the queries are depicted in Figure 4.9, which clearly demonstrates the security of the enhanced scheme, and the protection of the identifiers, ID_{SM_i}, ID_{UC_j}, and the session keys, SK_{SM_i}, SK_{UC_j}.

```
(*--------------------------------channel-------------------------*)
free ch: channel.
free schi:channel[private].
free schj:channel[private].
(*--------------------------------constant------------------------*)
free HIDi: bitstring.
free HIDj: bitstring.
free Sigi: bitstring.
free Sigj: bitstring.
free IDi: bitstring [private].
free IDj: bitstring [private].
free Ski: bitstring [private].
free Skj: bitstring [private].
free Ri: bitstring.
free Rj: bitstring.
const G: bitstring.
const Ppub: bitstring.
const k: bitstring[private].
(*----------------------------functions--------------------------*)
fun conc(bitstring,bitstring):bitstring. (*String concatination*)
fun H(bitstring):bitstring.(*Hash function*)
fun add(bitstring,bitstring):bitstring. (*addition function*)
fun mult(bitstring, bitstring):bitstring.(*multiplication function*)
fun XOR(bitstring,bitstring):bitstring. (*exclusive-OR*)
(*-------------------reductions &equations----------------------*)
reduc forall A:bitstring,B:bitstring;
XORagain(XOR(A,B),B)=A.
equation forall A:bitstring,B: bitstring;
add (A,B)=add(B,A).
(*--------------------------------------------------------------*)
```

Figure 4.4: The specifics of the declaration segment.

```
(*-----------------------------events-----------------------------*)
event beginEntitySMi(bitstring).
event endEntitySMi(bitstring).
event beginEntityUcj(bitstring).
event endEntityUCj(bitstring).
(*-----------------------------querys-----------------------------*)
query attacker(IDi).
query attacker(IDj).
query attacker(Ski).
query attacker(Skj).
query id:bitstring; inj-event(endEntitySMi(id))==>
inj-event(beginEntitySMi(id)).
query id:bitstring; inj-event(endEntityUCj(id))==>
inj-event(beginEntityUcj(id)).
(*------------------------------------------------------------------*)
```

Figure 4.5: The specifics of the query segment.

```
(*-----------------------------Process SMi------------------------*)
let ProcessSMi =
       let HIDi =H(IDi)in
       out(schi,HIDi);
       in(schi,(Sigi:bitstring,Ri:bitstring,ei:bitstring));
(*-------------------------Registration SMi----------------------*)
event beginEntitySMi(IDi);
new a: bitstring ;
new t1: bitstring ;
if mult(Sigi,G)= add(Ppub,mult(ei,Ri)) then
let A = mult(a,G) in
let V1 = mult(Sigi, a) in
out(ch, (Ri,A,t1, HIDi,V1));
in(ch,(rB:bitstring,Skj:bitstring,rV2:bitstring,rt2:bitstring));
if rV2=mult(mult(a,rB),XOR(mult(add(Ppub,H(conc(conc(Ppub,Rj),
HIDj))),Ri),Sigi) ) then
if XOR(Skj,Skj)=H(conc(A,rB)) then
let Ski=XOR(H(conc(A,rB)),Sigi) in
event endEntitySMi(IDi)
else 0.
(*------------------------------------------------------------------*)
```

Figure 4.6: The details of SM_i segment.

```
(*-----------------------------Process UCj------------------------*)
let ProcessUCj=
       let HIDj =H(IDj)in
       out(schj,HIDj);
       in(schj,(Sigj:bitstring,Rj:bitstring,ej:bitstring));
(*-------------------------Registration UCj----------------------*)
event beginEntityUcj(IDj);
new b: bitstring ;
new t2: bitstring ;
if mult(Sigj,G)= add(Ppub,mult(ej,Rj)) then
in(ch, (rRi:bitstring,rA:bitstring,rt1:bitstring,rHIDi:bitstring,
rV1:bitstring));
if mult(rV1,mult(G,G))=mult(add(mult(H(conc(rHIDi,conc(Ppub,rRi))),
rRi),Ppub),rA) then
let B=mult(b,G) in
let Skj=XOR(H(conc(rA,B)),Sigi) in
let V2=XOR(mult(mult(Sigj,rA),B),Sigi) in
out(ch,(B,Skj,V2,t2));
event endEntityUCj(IDj)
else 0.
(*------------------------------------------------------------------*)
```

Figure 4.7: The details of UC_j segment.

```
(*-----------------------------Process RA-------------------------*)
let SMiReg =
       in(schi,rHIDi:bitstring);
       new ri: bitstring;
       let Ri = mult(ri,G) in
       let ei = H(conc(conc(Ppub, Ri), rHIDi)) in
       let Sigi = add(k, mult(ri, ei)) in
       out(schi,(Sigi,Ri,ei));
       0.

let UCjReg =
       in(schj,rHIDj:bitstring);
       new rj: bitstring;
       let Rj = mult(rj,G) in
       let ej = H(conc(conc(Ppub, Rj), rHIDj)) in
       let Sigj = add(k, mult(rj, ej)) in
       out(schj,(Sigj,Rj));
       0.
let ProcessRA = UCjReg | SMiReg.
process ProcessRA | ProcessSMi | ProcessUCj
(*------------------------------------------------------------------*)
```

Figure 4.8: The details of RA and main process segment.

```
------------------------------------------------------------------------
Verification summary:

Query not attacker(IDi[]) is true.

Query not attacker(IDj[]) is true.

Query not attacker(Ski[]) is true.

Query not attacker(Skj[]) is true.

Query inj-event(endEntitySMi(id)) ==> inj-event(beginEntitySMi(id)) is true.

Query inj-event(endEntityUCj(id)) ==> inj-event(beginEntityUcj(id)) is true.

------------------------------------------------------------------------
```

Figure 4.9: The results of queries.

4.5 Comparative analysis

This section evaluates the enhanced protocol's performance, considering various factors such as security, computational cost, and communication cost. The assessment provides insights into the effectiveness of the upgraded protocol and its overall suitability for real-world applications.

4.5.1 Security features

Table 4.2 presents a comparative analysis of the security features of our enhanced protocol with other existing protocols. It should be noted that the symbol "✓" denotes the protocol's effectiveness in mitigating the particular attack. In contrast, the symbol "×" indicates the protocol's vulnerability to the attack in question, and "N/A" indicates "not applicable".

Table 4.2: Comparing security features of SemAuth protocol with recent works.

Ref	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
[142]	✓	✓	✓	×	✓	×	×	✓	✓	✓
[143]	✓	×	✓	✓	✓	✓	✓	×	×	×
[144]	✓	✓	✓	✓	✓	✓	✓	×	×	×
[145]	✓	✓	✓	✓	✓	✓	✓	×	×	✓
[146]	✓	N/A	✓	✓	✓	✓	✓	×	×	✓
[147]	✓	N/A	N/A	✓	✓	✓	✓	×	×	✓
[148]	✓	N/A	✓	✓	✓	✓	✓	×	×	✓
[149]	✓	N/A	✓	✓	✓	✓	✓	×	×	✓
[150]	✓	N/A	✓	✓	✓	✓	✓	×	×	✓
SemAuth	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

F1: MITM attack, **F2:** DoS attack, **F3:** Replay attack, **F4:** Impersonation attack, **F5:** Mutual authentication, **F6:** Session hijacking, **F7:** Information disclosure, **F8:** Batch verification, **F9:** RA decentralization, **F10:** Support blockchain-based solution.

Table 4.2 reveals that our proposed SemAuth protocol provides comprehensive security features compared to the referenced works. Notably, previous protocols like [142] and [143] are vulnerable to common attacks such as MITM, session hijacking, and information disclosure, which SemAuth effectively mitigates. While some protocols like [144] and [145] address a wider range of attacks, they often lack full decentralization or blockchain support, limiting their applicability in dynamic EI environments. Our protocol distinguishes itself by offering resistance to all listed traditional attacks (MITM, DoS, Replay, Im-

personation, Session hijacking, Information disclosure) while also supporting mutual authentication and blockchain-based decentralization with multiple RAs. This holistic approach ensures a more robust and resilient security framework for IoT-based EI, addressing critical gaps in existing solutions.

4.5.2 Computational cost

The computational cost of the improved cryptographic scheme is compared with seven other protocols [142, 144, 145, 146, 147, 149, 150]. The assessment focuses solely on the authentication phase and session key agreement, which constitute the enhanced part of the proposed protocol. A detailed description of the notations used and the computational cost calculation process is provided in Section 3.4.

Table 4.3: Comparing computational costs of SemAuth protocol with recent works.

Ref	SM_i **/ IoT node**	UC_j **/ Service provider**	**Total time (ms)**
[142]	$2T_h + 4T_m + T_a$	$2T_h + 6T_m + T_a$	199.782
[144]	$4T_h + 6T_m + 1T_b$	$5T_h + 4T_m + 1T_b$	297.311
[145]	$3T_h + 3T_m + 1T_a + 2T_b + 1T_{e/d}$	$3T_h + 3T_m + 1T_a + 2T_b + 1T_{e/d}$	255.1762
[146]	$11T_h + 4T_m + 1T_a$	$11T_h + 4T_m + 1T_a$	161.5552
[147]	$7T_h + 4T_m + 1T_a + 2T_{e/d}$	$7T_h + 4T_m + 1T_a + 2T_{e/d}$	160.8524
[149]	$7T_h + 4T_m$	$5T_h + 1T_a$	80.862
[150]	$7T_h + 4T_m + 1T_a$	$7T_h + 4T_m + 1T_a + 2T_{e/d}$	160.8432
SemAuth	$3T_m$	$3T_m$	119.514

The results, shown in Table 4.3, indicate that the computational overheads of Wu et al.'s scheme [144] and Fan et al.'s scheme [145] are notably high, as they rely on bilinear pairing operations, contributing to a cost of 297.311 and 255.1762, respectively. In contrast, other protocols reduce the computational cost by utilizing only ECC instead of bilinear pairing operations. Despite the reduced cost of the SEMAGrid protocol [142], it still suffers from several shortcomings.

The improved protocol presented in this work addresses the issues identified in earlier protocols, resulting in a lightweight scheme. As illustrated in Table 4.3, the experimental findings demonstrate that the proposed scheme outperforms all of the referenced protocols in terms of computational efficiency.

4.5.3 Communication cost

The communication cost associated with the authentication procedure is determined by the total number of bits transmitted between the participating entities. To evaluate this cost, various parameters and their respective bit sizes are taken into account (see Section 3.4).

In the proposed scheme, both entities SM_i and UC_j exchange two messages, namely M_1 and M_2, each containing three ECC points of size 160 bits and a timestamp of size 32 bits. The transmission of messages M_1 and M_2 requires 512 bits each. Therefore, the overall communication cost of the proposed scheme is 1024 bits. The communication costs associated with related schemes [142, 144, 145, 146, 147, 149, 150] have also been evaluated using the same approach and are summarized in Table 4.4.

Table 4.4: Communication cost comparison of SemAuth protocol with recent works.

Ref	SM_i **/ IoT node (bits)**	UC_j **/ Service provider (bits)**	**Total cost (bits)**
[142]	1018	928	1964
[144]	1376	1376	2752
[145]	672	800	1472
[146]	1760	1856	3616
[147]	928	800	1728
[149]	832	992	1824
[150]	992	832	1824
SemAuth	512	512	1024

As demonstrated in Table 4.4, SemAuth exhibits significantly lower communication costs compared to all other schemes. With a total communication cost of 1024 bits, it drastically outperforms protocols like [146] (3616 bits) and [144] (2752 bits). This efficiency is primarily due to the optimized message structure of SemAuth, which focuses on transmitting only essential ECC points and timestamps. The reduced data volume per transaction minimizes network bandwidth consumption and lowers latency, making SemAuth highly suitable for resource-constrained IoT devices within the EI. This efficiency, combined with its robust security features, positions SemAuth as a highly practical solution for real-world deployment where communication overhead is a critical consideration.

4.6 Implementation

The practical deployment of our blockchain-based security framework is organized into two primary phases: (i) network infrastructure setup and configuration, and (ii) performance evaluation and benchmarking. These phases are elaborated in the following subsections.

4.6.1 Network deployment

To implement the blockchain-based solution, a network topology was designed to include multiple smart grids, represented as organizations within the HLF network. Specifically, the network includes two organizations, denoted as SG_A and SG_B, both utilizing the same communication channel C for transaction exchange (see Figure 4.10).

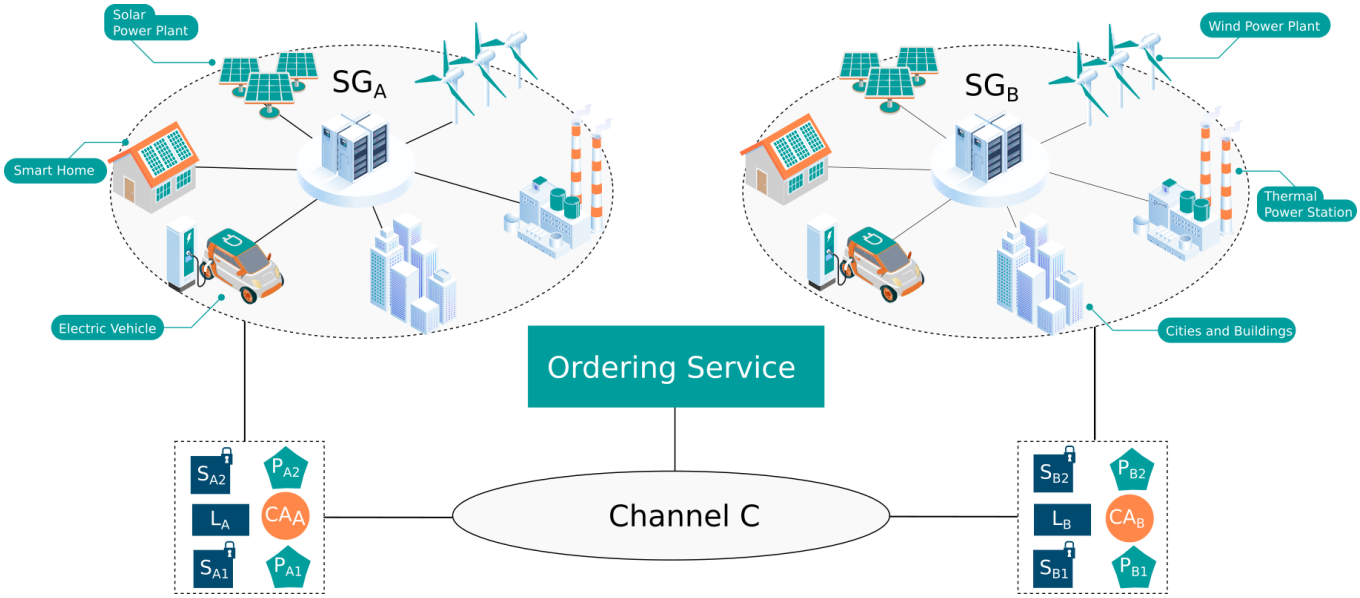
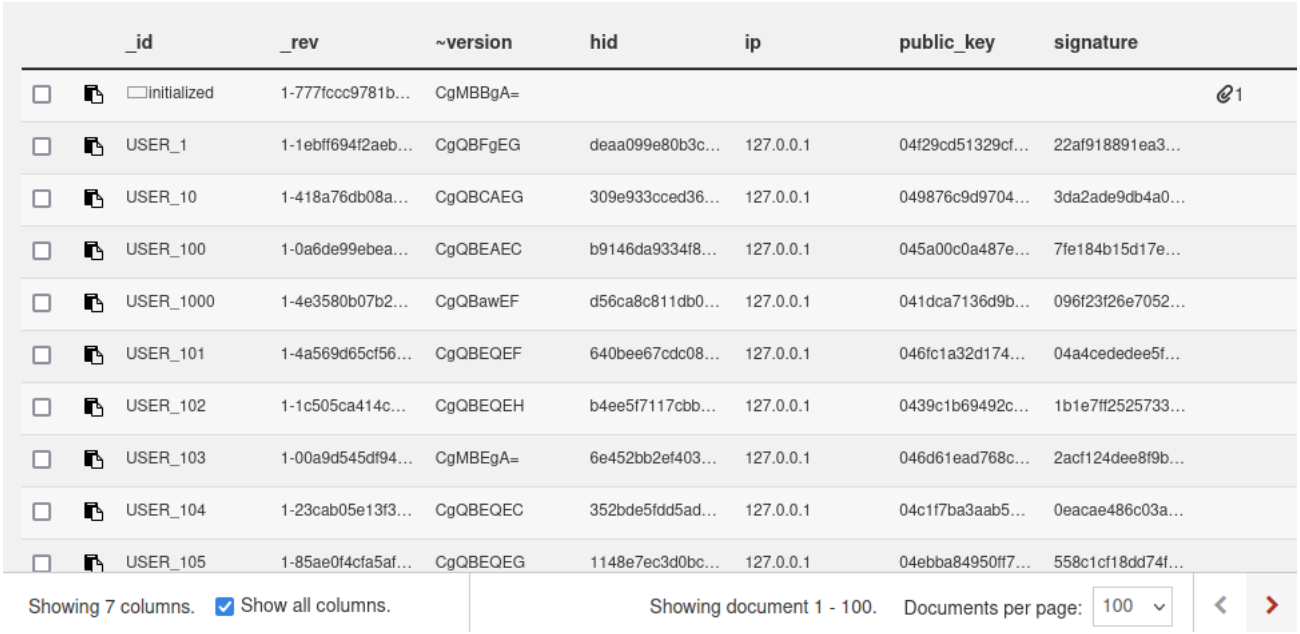


Figure 4.10: EI configuration based on HLF.

Each organization includes two peers, namely P_{A1}, P_{A2}, P_{B1}, and P_{B2}, respectively, which acted as the RA in our network model (Figure 4.1). These peer nodes maintained identical copies of the blockchain ledger L, which was associated with the network channel C. To ensure consensus among nodes in the network, we employed the Raft consensus algorithm for ordering service, with three ordering nodes deployed to process the consensus. Additionally, each organization has its own local certificate authority, CA_A and CA_B. The smart contract installation package is S_{A1}, S_{A2}, S_{B1}, and S_{B2} for P_{A1}, P_{A2}, P_{B1}, and P_{B2},

respectively.



	_id	_rev	~version	hid	ip	public_key	signature	
☐	initialized	1-777fccc9781b...	CgMBBgA=					1
☐	USER_1	1-1ebff694f2aeb...	CgQBFgEG	deaa099e80b3c...	127.0.0.1	04f29cd51329cf...	22af918891ea3...	
☐	USER_10	1-418a76db08a...	CgQBCAEG	309e933cced36...	127.0.0.1	049876c9d9704...	3da2ade9db4a0...	
☐	USER_100	1-0a6de99ebea...	CgQBEAEC	b9146da9334f8...	127.0.0.1	045a00c0a487e...	7fe184b15d17e...	
☐	USER_1000	1-4e3580b07b2...	CgQBawEF	d56ca8c811db0...	127.0.0.1	041dca7136d9b...	096f23f26e7052...	
☐	USER_101	1-4a569d65cf56...	CgQBEQEF	640bee67cdc08...	127.0.0.1	046fc1a32d174...	04a4cededee5f...	
☐	USER_102	1-1c505ca414c...	CgQBEQEH	b4ee5f7117cbb...	127.0.0.1	0439c1b69492c...	1b1e7ff2525733...	
☐	USER_103	1-00a9d545df94...	CgMBEgA=	6e452bb2ef403...	127.0.0.1	046d61ead768c...	2acf124dee8f9b...	
☐	USER_104	1-23cab05e13f3...	CgQBEQEC	352bde5fdd5ad...	127.0.0.1	04c1f7ba3aab5...	0eacae486c03a...	
☐	USER_105	1-85ae0f4cfa5af...	CgQBEQEG	1148e7ec3d0bc...	127.0.0.1	04ebba84950ff7...	558c1cf18dd74f...	

Showing 7 columns. ☑ Show all columns. Showing document 1 - 100. Documents per page: 100

Figure 4.11: CouchDB showing the database for our HLF network.

To effectively store and manage the network's state, we utilized CouchDB, a NoSQL database renowned for its efficiency in querying and indexing data. A screenshot of the CouchDB interface for P_{A1} (shown in Figure 4.11) displays the database containing attributes such as hash ID, public key, and signature. It is worth noting that each peer has a copy of this CouchDB.

Overall, our network topology and deployment strategy were carefully designed to ensure the security, reliability, and scalability of our blockchain-based solution. By leveraging the decentralized and transparent nature of the blockchain, we were able to create a network that facilitated secure communication and transaction exchange among multiple smart grids while maintaining privacy and confidentiality for each organization. We employed the GO programming language to define our chaincode, which includes the implementation of the *Registration* function illustrated in Figures 4.12 and 4.13. This function serves as a crucial component of the system by enabling user registration through the generation of a public-private key pair and the creation of a signature. Additionally, it generates a JSON-encoded byte array of the user object, which can be used for further processing. Finally, the function securely stores the user data in the ledger using the *PutState* method and returns the user object. Overall, the *Registration* function plays a critical role in ensuring the integrity and security of the system's user registration process.

```go
func (s *SmartContract) Registration(stub shim.ChaincodeStubInterface, args []string)
sc.Response {

    if len(args) != 5 {
        return shim.Error("incorrect number of arguments, expecting 4")
    }

    userHashID := args[1]
    privateKeyStr := args[2]
    ipAddress := args[3]
    k := args[4]

    // Decode the private key of user and the private key of the system
    privateKeyBytes, _ := hex.DecodeString(privateKeyStr)
    kBytes, _ := hex.DecodeString(k)

    // Get system public key
    systemAsBytes, _ := stub.GetState("System")
    system := new(System)
    _ = json.Unmarshal(systemAsBytes, system)

    Ppub := system.PublicKey

    // Decode the public key of the system
    PpubBytes, err := hex.DecodeString(Ppub)

    curve := elliptic.P256()
    privateKey := new(ecdsa.PrivateKey)
    privateKey.D = new(big.Int).SetBytes(privateKeyBytes)
    privateKey.PublicKey.Curve = curve
    privateKey.PublicKey.X, privateKey.PublicKey.Y = curve.ScalarBaseMult(privateKeyBytes)

    publicKeyBytes := elliptic.Marshal(curve, privateKey.PublicKey.X, privateKey.PublicKey.Y)
    publicKeyStr := hex.EncodeToString(publicKeyBytes)
```

Figure 4.12: Registration function part 1.

```go
// Calculate e = Hash(Ppub || publicKey || Hash(id))
h := sha256.New()
h.Write(PpubBytes)
h.Write(publicKeyBytes)
h.Write([]byte(userHashID))

e := new(big.Int).SetBytes(h.Sum(nil))
if err != nil {
    return shim.Error(fmt.Sprintf("failed to decode k: %s", err))
}
// Calculate sig = k + r * e
K := new(big.Int).SetBytes(kBytes)

r := new(big.Int).SetBytes(privateKeyBytes)

Sig := new(big.Int).Mul(r, e)
Sig.Add(Sig, K)
SigBytes := Sig.Bytes()
signatureStr := hex.EncodeToString(SigBytes)

user := &User{
    HID:       userHashID,
    PublicKey: publicKeyStr,
    Signature: signatureStr,
    IP:        ipAddress,
}
// Convert the User object to a JSON-encoded byte array
userBytes, err := json.Marshal(user)
if err != nil {
    return shim.Error(fmt.Sprintf("failed to marshal user data to JSON: %s", err))
}
// Store the JSON-encoded byte array in the ledger using PutState
err = stub.PutState(args[0], userBytes)
```

Figure 4.13: Registration function part 2.

4.6.2 Performance measurement

The benchmark was conducted using the Hyperledger Caliper tool, with simulations performed on personal computers configured with a 4-core CPU (Intel Core i5, 2.40GHz), 8GB of RAM, and a 256GB SSD.

The performance of the proposed blockchain network is evaluated and compared with two existing systems: the SEMAGrid protocol and the protocol proposed by Tanwar et al. [151]. The comparison focuses on throughput and latency during the invocation of the Registration function, where both the proposed protocol and SEMAGrid follow identical instructions. Results from the implementation are compared with those reported for Tanwar et al.'s system. Additionally, throughput and latency are assessed for the *SignatureVerify* function, with a comparison against SEMAGrid. Since the original SEMAGrid paper did not include an implementation, the protocol was implemented independently to ensure a fair comparison. The performance evaluation is structured into two scenarios: the first presents results for the *Registration* function, and the second presents results for the *SignatureVerify* function.

- **Scenario 1: Registration function**

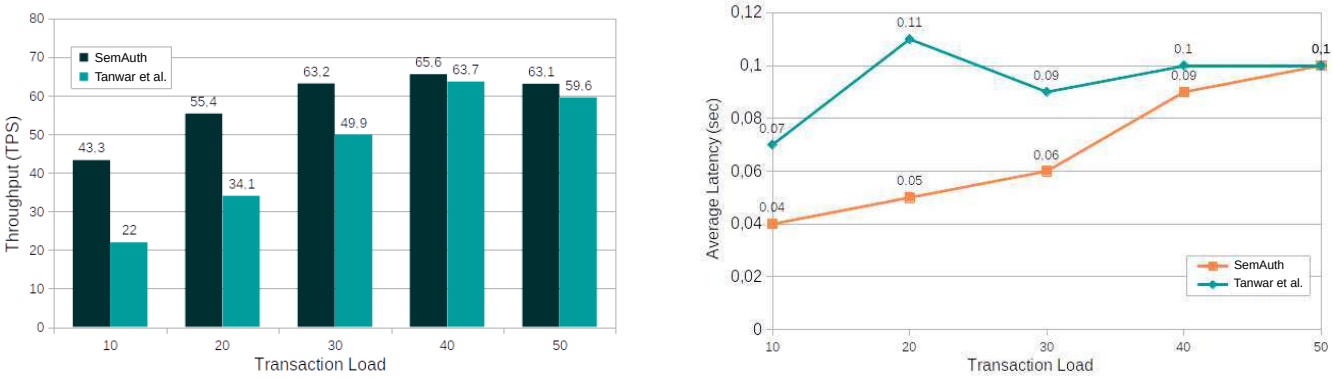
  In this scenario, we measured the performance of the network when executing the *Registration* function of the chaincode using the configuration illustrated in Table 4.5.

  The performance evaluation results illustrated in Figure 4.14 show that the evaluated blockchain network exhibited notably lower transaction latency compared to the system proposed by Tanwar et al. [151]. This improvement is attributed to the higher throughput achieved by the evaluated

Table 4.5: Banchmark configuaration (Registration function).

Parameters	Configuration
Rounds	5
Transaction per round	1000
Transaction mode	Write
Rate (TPS)	50 to 250

network. The findings indicate an inverse relationship between latency and throughput, where increased throughput corresponds to reduced latency. As shown in Figure 4.14, the network reached a peak throughput of 31 transactions per second (tps).



(a) Throughput.

(b) Average latencies.

Figure 4.14: Registration function.

However, it is important to note that the performance of a blockchain network is affected by several server-side factors, such as the complexity of the chaincode used, the choice of consensus mechanism, and the type of data-store utilized. A complex chaincode can increase the processing time of transactions, while the choice of consensus mechanism can affect resource utilization. Similarly, the type of data-store used can impact the speed and efficiency of data retrieval and storage. By considering these factors, we were able to gain a comprehensive understanding of the overall performance of our blockchain network.

- **Scenario 2: Verification function**

  In this scenario, we measured the performance of the network when executing the *SignatureVerify* function of the chaincode. Table 4.6 shows the configuration used in this scenario. The aim of

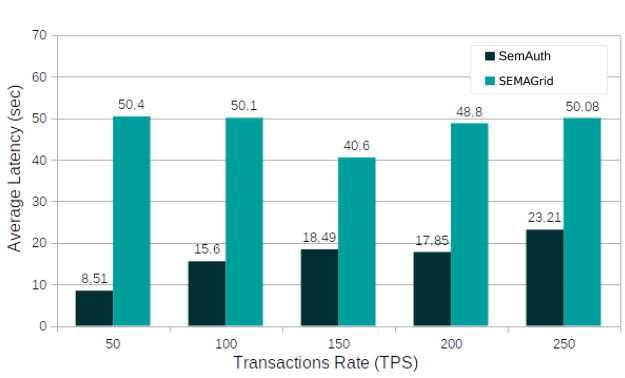
Table 4.6: Banchmark configuration (SignatureVerify function).

Parameters	**Configuration**
Rounds	5
Transaction Load	10 to 60
Rate control type	fixed-load
Transaction duration	1

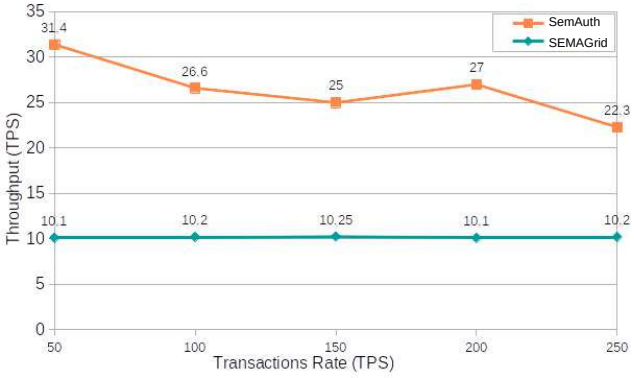
varying the transaction load was to assess the scalability of our blockchain network while the use of fixed-load with asset control rate control type ensured that the network was not overloaded with too many transactions at once. The duration of each transaction was set at 1 second to ensure a fair comparison of performance across all rounds. Latency observations were recorded for each round to evaluate the speed of transaction processing.

The latency observed in the invoke function of our blockchain network was notably higher compared to the latency observed in query mode (see Figure 4.15). This can be attributed to the fact that the invoke function involves creating and updating new records in the blockchain, which requires more processing time and resources compared to simply querying existing records. Additionally, in our blockchain network, the invoke function involved executing more complex chaincode logic, which also contributed to the higher latency observed. On the other hand, query mode involves retrieving and reading existing records from the blockchain, which is a less resource-intensive operation and typically results in lower latency.

Figure 4.15 illustrates the superior performance of our blockchain network in terms of transaction latency and throughput compared to SEMAGrid system. Our protocol's utilization of simple signature verification operations, as opposed to the more complex operations used in SEMAGrid protocol, is largely attributed to this superior performance. This demonstrates the effectiveness of our approach in achieving higher performance in blockchain network.



(a) Average latencies.



(b) Throughput.

Figure 4.15: SignatureVerify function.

4.7 Conclusion

This chapter presented a blockchain-based solution for device authentication in IoT-based EI networks. The solution employs a smart contract deployed on a blockchain platform to manage the registration and signature verification of users, utilizing multiple registration authorities distributed across the network to enhance resilience against potential attacks. The security and performance of the proposed authentication mechanism were validated through formal verification tools such as ProVerif and BAN logic, as well as performance benchmarking with Caliper.

The proposed solution offers several improvements compared to existing blockchain-based approaches in the EI domain. It is fully decentralized, which strengthens its robustness against attacks. It also provides a scalable and efficient authentication process, as demonstrated by the evaluation results. Moreover, it builds upon an established authentication protocol already used in smart grid systems, supporting compatibility with current infrastructure.

The next chapter addresses another critical area of concern in security, with a focus on the emerging threats introduced by quantum computing.

Chapter 5

LightQ: A Lightweight Security Scheme to defend against Quantum Attack in IoT-based EI

5.1 Introduction

Quantum computers are another cutting-edge development that store data and perform calculations using the principles of quantum physics. While quantum computers can be helpful for specific tasks and outperform our best supercomputers, they also pose a threat to the security and confidentiality of existing cryptographic methods, introducing a new security risk known as a quantum attack. To address this challenge, we propose a scheme to resist attacks from quantum computers by combining two promising methods: GGH lattice-based cryptography and QKD, also known as quantum cryptography. The former is based on complex mathematical problems, while the latter is based on quantum physics [152]. Our contribution is based on the main idea of using quantum physics [153], which is hard or even impossible to attack by a quantum computer [154]. We use a QKD protocol to exchange keys and take advantage of the security provided by quantum physics. Additionally, we use GGH public-key cryptography, which is robust and immune to quantum computing improvements [155]. By integrating QKD with the GGH cryptosystem, we can ensure that the exchanged keys are both secure and authenticated. This approach

can help to mitigate the risks associated with cyber-attacks on the EI, thus contributing to a more secure and resilient energy infrastructure.

5.2 Overview of GGH and QKD

In this section, an overview will be provided on the GGH lattice-based cryptography and QKD, which are the two building blocks that have been employed in the mechanism to resist quantum attacks. The fundamental concepts of GGH lattice-based cryptography, including its operational mechanism, will be described. Additionally, the basic principles of QKD, which employs quantum mechanics to enable secure communication between two parties, will be discussed along with an explanation of how it works. Key notations employed in this work are summarized in Table 5.1.

Table 5.1: Key notations of LightQ scheme.

Notations	**Description**
det	Determinant
$\mathcal{H}$	Hadamard ratio
P_{BP}, S_{BP}	Buyer Prosumer (BP) public and private key pair
P_{SP}, S_{SP}	Seller Prosumer (SP) public and private key pair
Sk	Shared secret key between BP and SP
t_i	Timestamp
Δt	Maximum transmission delay
q	Random bits $\in$ (0,1)
b_i	Random basis $\in$ (+,x)
S_i	Polarized photons

5.2.1 The GGH cryptosystem

The GGH cryptosystem is a lattice-based public key encryption scheme introduced by Goldreich, Goldwasser, and Halevi in 1997 [156]. To understand the cryptosystem, it is necessary to have a clear understanding of lattices and their fundamental properties. In this section, we provide an overview of these concepts.

- **Lattice:** A lattice is a set of linear combinations of vectors $v_1, v_2, ..., v_n$, where $v_1, v_2, ..., v_n$ are linearly independent vectors in $\mathbb{R}$, and the coefficients of the linear combinations are integers in $\mathbb{Z}$. Mathematically, a lattice L can be expressed as [157]:

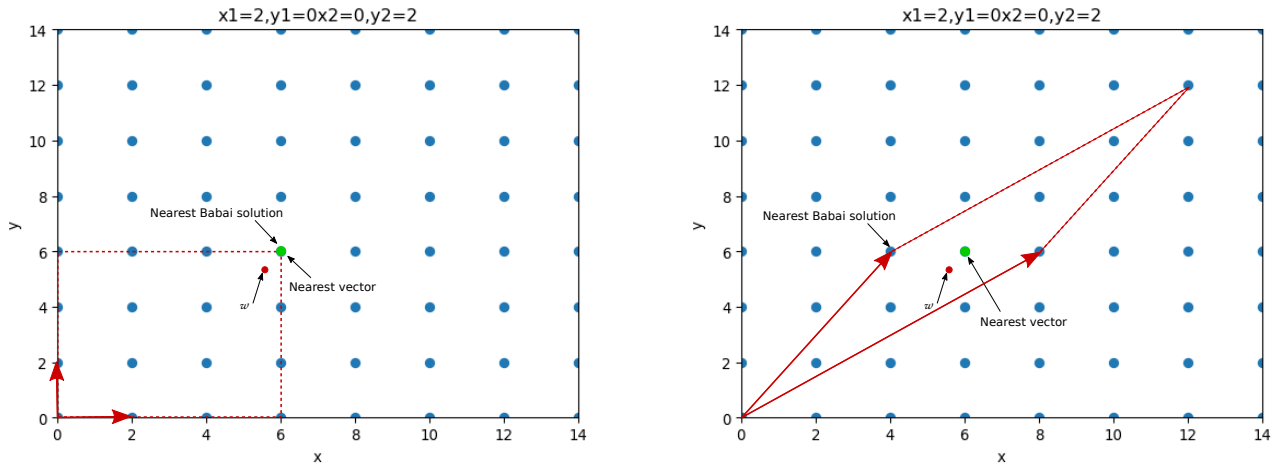
$$L = a_1v_1 + a_2v_2 + ... + a_nv_n, \text{where } a_1, a_2, ..., a_n \in \mathbb{Z} \tag{5.1}$$

The dimension of L is defined as the number of vectors in a basis for L, where the basis is any independent vectors that generate L.

- **The Hadamard ratio:** The Hadamard ratio of a basis $B = v_1, v_2, ..., v_n$ is defined as follows [157]:

$$\mathcal{H}(B) = \left(\frac{det(L)}{|v_1| \cdot |v_2|...|v_n|}\right)^{1/n}, \text{where } 0 < \mathcal{H}(B) \leq 1 \tag{5.2}$$

The Hadamard ratio is used to distinguish between good and bad basis, where a good basis (see Figure 5.1a) is one with a Hadamard ratio close to 1, while a bad basis (see Figure 5.1b) is one with a Hadamard ratio close to 0. A basis with a high Hadamard ratio means that the degree of orthogonality among the vectors in a lattice basis increases.



(a) "Good basis" (b) "Bad basis"

Figure 5.1: Babai's algorithm using Good and Bad basis.

- **'Good' and 'Bad' basis:** Figure 5.1 depicts two distinct bases employed for representing the same lattice. The first basis, illustrated in Figure 5.1a, is characterized as "Good" due to the substantial

orthogonality among the vectors (the red row). Within this context, we introduce an arbitrary point denoted as 'w' located within the lattice. Our objective is to identify the closest lattice point to 'w' based on the orthogonal basis. Employing Babai's algorithm, we obtain the resultant point (depicted as the green point), which accurately represents the nearest lattice point to 'w'. This highlights the effectiveness of Babai's algorithm when a "Good" basis is employed. Conversely, the second basis, illustrated in Figure 5.1b, is deemed "Bad" due to the significantly small angle between the basis vectors. In this scenario, when utilizing Babai's algorithm, the nearest point to 'w' obtained in the lattice is distant from the actual nearest lattice point.

- **The Shortest Vector Problem (SVP):** Involves finding a non-zero vector v in a lattice L that minimizes the Euclidean norm. Specifically, the goal is to determine the shortest non-zero vector in the given lattice [157].

- **The Closest Vector Problem (CVP)**: Requires finding a vector $v \in L$ that is closest to a given vector $w \in \mathbb{R}$, which involves minimizing the Euclidean norm $|w - v|$ when w is not in L [157].

- **Babai's algorithm:** Is a lattice-based method that solves the CVP in a lattice. It finds the closest lattice point to an arbitrary vector by representing it as a linear combination of lattice basis vectors and rounding the coefficients to the nearest integer. The algorithm is effective when the basis vectors are sufficiently orthogonal to each other but may yield an erroneous lattice point if they are highly non-orthogonal.

  A step-by-step breakdown of Babai's algorithm (refer to Algorithm 1):

  1. Write the arbitrary vector 'w' as a linear combination of lattice basis vectors: $w = c_1 \cdot v_1 + c_2 \cdot v_2 + ... + c_n \cdot v_n$, where '$v_1, v_2, ..., v_n$' are the basis vectors of the lattice, and '$c_1, c_2, ..., c_n$' are coefficients.

  2. Set a_i as the floor value (rounded down to the nearest integer) of each coefficient c_i for $i = 1, 2, ..., n$. In other words, $a_i = \lfloor c_i \rceil$.

  3. Return the vector 'v' obtained by taking the linear combination of the lattice basis vectors using the rounded coefficients: $v = a_1 \cdot v_1 + a_2 \cdot v_2 + ... + a_n \cdot v_n$.

Algorithm 1 Babai's Algorithm

Input
$w \in \mathbb{R}$ /* an arbitrary vector */
Output
$v \in L$ /* closest lattice vector to w */
1: Write $w = c_1 \cdot v_1 + c_2 \cdot v_2 + ... + c_n \cdot v_n$ with $c_1, ..., c_n \in \mathbb{R}$.
2: Set $a_i = \lfloor c_i \rceil$ for $i = 1, 2, ..., n$.
3: Return the vector $v = a_1 \cdot v_1 + a_2 \cdot v_2 + ... + a_n \cdot v_n$.

The GGH cryptosystem is a secure public-key encryption scheme that is based on the hardness of the SVP and CVP problems on lattices. The private key is generated based on a good basis lattice, while the public key is derived from a bad basis lattice with a low Hadamard ratio. Encryption is performed using the public key, and decryption involves finding the closest vector in the good basis lattice to the ciphertext vector using the Babai's algorithm. The security of the GGH cryptosystem depends on the assumption that it is difficult to find the good basis lattice given only the bad basis lattice.

5.2.2 The QKD protocol

The earliest and most widely used QKD protocol is Bennett-Brassard-84 (BB84) [158, 159]. This protocol is based on the polarization of photons. It is an attractive solution because an eavesdropper cannot copy a quantum particle state, making it inherently possible to validate that a transmitted key is secure. In this protocol, Alice and Bob, as the sender and receiver, respectively, must share both a classical and a quantum channel. The classical channel is a typical electrical wire, and the quantum channel is a fiber optic cable (see Figure 5.2). The protocol works as follows:

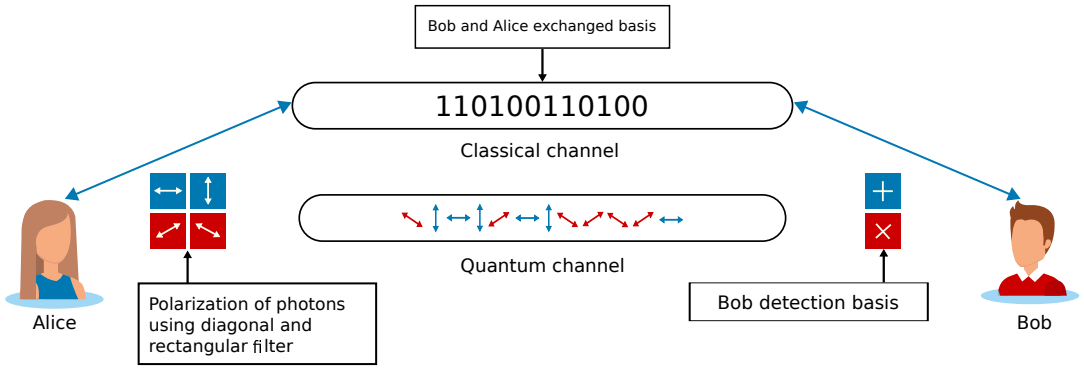
1. Alice first generates a sequence of bits to transmit to Bob. She then chooses a random polarization basis from the rectangle (+) to the diagonal (x) to convert the binary bits to qubits.

2. After receiving the photons, since Alice does not share her basis with Bob, Bob chooses a random basis to measure the received message.

3. After measuring the received message and obtaining the bits, Bob is not sure if they match Alice's. Therefore, Alice and Bob exchange their basis over the classical channel.

4. However, after comparing the basis, they eliminate the respective bits in their bit chains if they measure the photon with a different basis. The remaining bits should be identical for Alice and Bob. The bits corresponding to the correctly measured photons are used as a secret key.

The security of the BB84 protocol is further enhanced by the concept of Quantum Bit Error Rate (QBER). QBER is the percentage of bits that differ between Alice and Bob's measurements due to noise in the quantum channel, which can be caused by environmental factors or an eavesdropper intercepting and altering the photons. To ensure the security of the key, Alice and Bob need to estimate the QBER and eliminate the bits where they have a different measurement result. By repeating the protocol multiple times and using error-correcting codes, they can obtain a secret key with a low QBER and a high level of security [159].



Alice's bit sequence	1	1	0	1	0	0	1	1	0	1	0	0
Alice's photon polarization	↘	↕	↔	↕	↗	↔	↕	↘	↗	↘	↗	↔
Bob's detection basis	×	+	×	+	+	+	+	×	+	×	×	×
Bob's measurement	1	1	1	1	1	0	1	1	0	1	0	1
Basis comparison	✓	✓	✗	✓	✗	✓	✓	✓	✗	✓	✓	✗
Key	1	1	-	1	-	0	1	1	-	1	0	-

Figure 5.2: BB84 protocol.

5.3 System model

Figure 5.3 illustrates our system model, which consists of three entities: the Buyer Prosumer (BP), the Seller Prosumer (SP), and the ER. We assume that the energy trading occurs through Peer-To-Peer (P2P) communication between BP and SP, where the security of the transaction depends on the negotiation

phase data exchange.

- *SP*: A prosumer with surplus energy production from installed solar panels can act as an SP and sell energy to other prosumers on the network or inject it into the electrical system.
- *BP*: A prosumer who needs energy can act as a buyer prosumer. When the prosumer requires energy that the electrical system cannot provide, they act as a BP to purchase power from the SP.
- *ER*: The ER is responsible for scheduling, converting, and monitoring power to ensure the bidirectional flow of information and energy. Our system model assumes the ER is directly integrated into the smart home to simplify the task.

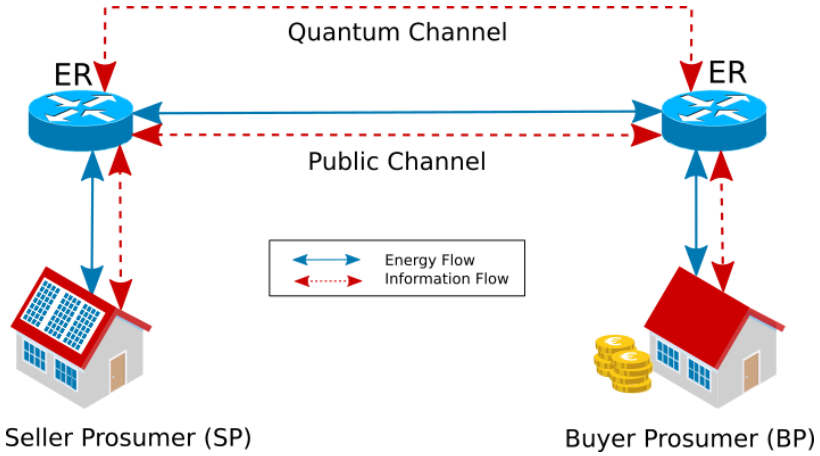


Figure 5.3: System model of LightQ scheme.

5.4 Proposed scheme

This section outlines the various phases of the proposed scheme, including initialization, shared key agreement, and authentication. A flow chart summarizing our scheme is presented in Figure 5.4.

5.4.1 Initialization

When BP and SP join the network, they generate their public and private keys using the GGH cryptosystem through the following steps:

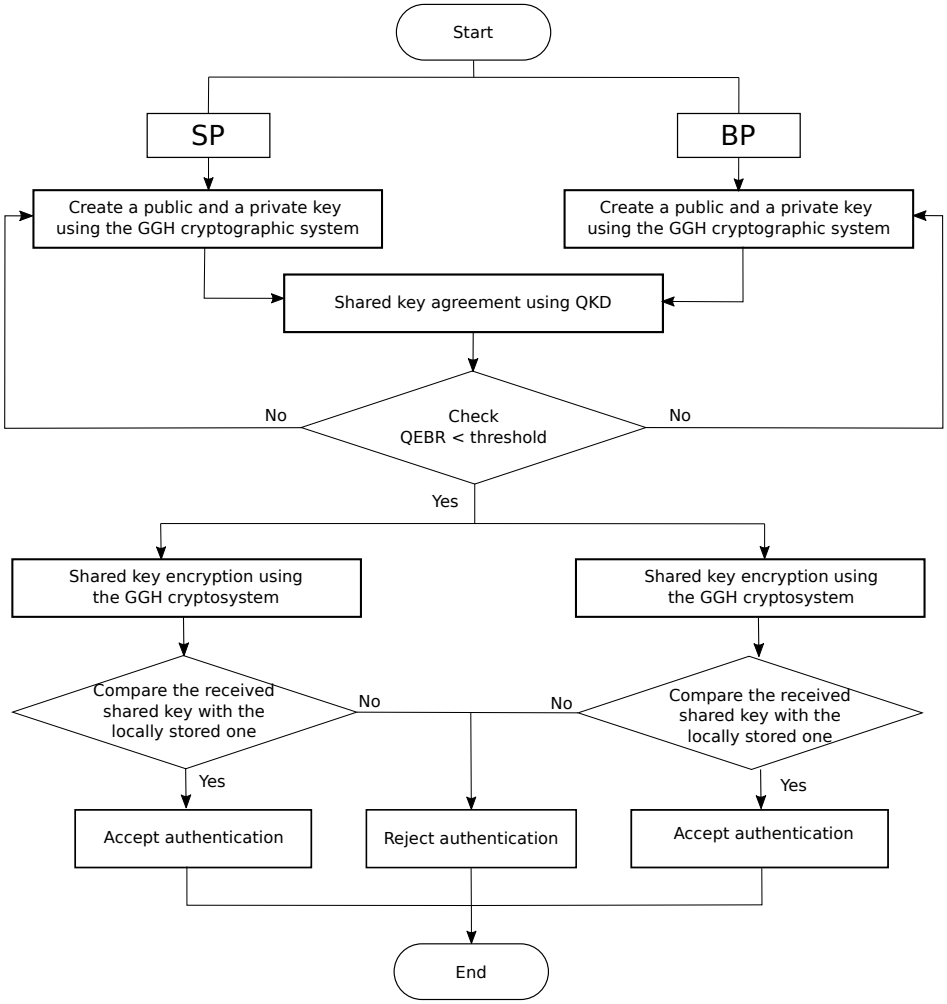


Figure 5.4: Flowchart depicting the workflow of the LightQ scheme.

- Select a good basis $V = v_1, ..., v_n$ as their private key and evaluate its quality using the Hadamard ratio.
- Choose an integer coefficient matrix U with $det(U) = \pm 1$.
- Compute a bad basis $w_1, ..., w_n$ by using the rows of $W = UV$.
- Disclose the public key $w_1, ..., w_n$.

5.4.2 Shared key agreement

The BB84 protocol is employed for the shared key agreement, as indicated in Algorithm 2.

Algorithm 2 Shared key agreement

Input
$q, b_0, b_1, QBER$
Output
Sk /* shared symmetric key */
Local:
S_0, S_1 /* polarized photons */
1: BP selects a random q and b_0
2: Based on q and b_0, we get S_0
3: BP sends S_0 to SP
4: SP selects a random b_1
5: Based on b_1, we get S_1
6: BP and SP exchange b_0 and b_1 on classical channel
7: BP and SP calculate the QBER from their received bits
8: **if** QBER $\geq$ 11% **then**
9: Abort the protocol and start over
10: **else**
11: Based on b_0 and b_1, BP and SP determine Sk
12: **end if**

The objective is for both parties to select the same shared key while preventing any potential eavesdropper from acquiring it. An overview of this process is illustrated in Figure 5.5.

- Initially, the BP encodes its bit sequence by choosing its basis randomly without revealing its choices to anyone. Then, the photons S_0 are sent to the SP via the quantum channel.
- The SP chooses a random basis b_1, and measures S_0 randomly according to quantum basis chosen.
- Based on the basis chosen, the SP completes the measurement and will get a set of results represented in S_1.
- SP compares its choices of basis b_1 with those of BP and identifies the subset of bits corresponding to the cases where they have both chosen the same basis. The similar bits are preserved and then, the other bits (the different bits) are deleted. BP also performs the same operation.
- BP and SP then share a subset of their results over public channels. BP and SP compare the bit sequences and then perform an error analysis. Communication over a quantum channel is secure if the QBER is less than 11%. This threshold is derived from information-theoretic security proofs in QKD, specifically in the BB84 protocol. BP and SP use the remaining bits for error analysis

to compose their encryption key. If the QBER exceeds 11%, the program is interrupted, and the protocol is restarted.

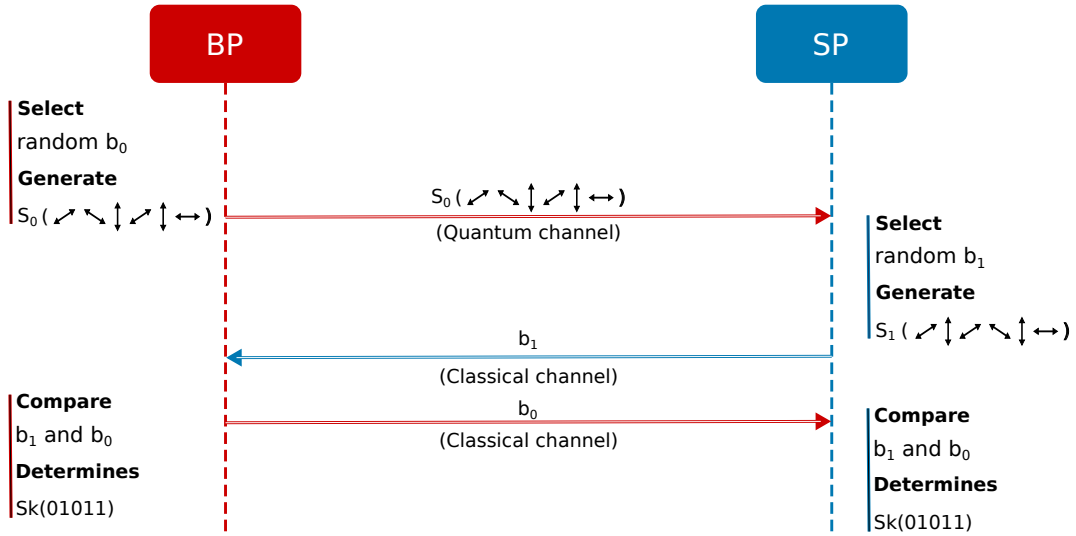


Figure 5.5: Shared key agreement phase of LightQ scheme.

5.4.3 Authentication

The authentication process between the BP and SP is a critical step in ensuring secure communication (see Figure 5.6). It involves the following steps:

1. Initially, both the BP and SP have the same shared key Sk, which was established in phase 5.4.2.

2. Each party concatenates Sk with a timestamp t_i to create a message. The timestamp is included to prevent modification of the message at a later stage. The message is then encrypted using the recipient's public key and sent over a public channel:

$$e = Encrypt(Sk||t_1, P_{BP}) \tag{5.3}$$

   (see Algorithm 3 for the encryption process).

3. The recipient receives the encrypted message and decrypts it using its private key:

$$(Sk'||t_1') = Decrypt(e, S_{BP}) \tag{5.4}$$

(see Algorithm 4 for the decryption process).

The recipient then checks that the current timestamp t_1' is fresh, i.e., $|t_1' - t_2| \leq \Delta t$.

4. If the timestamp is fresh and the decrypted message matches the locally stored Sk, then the sender is authenticated. Otherwise, the message is discarded.

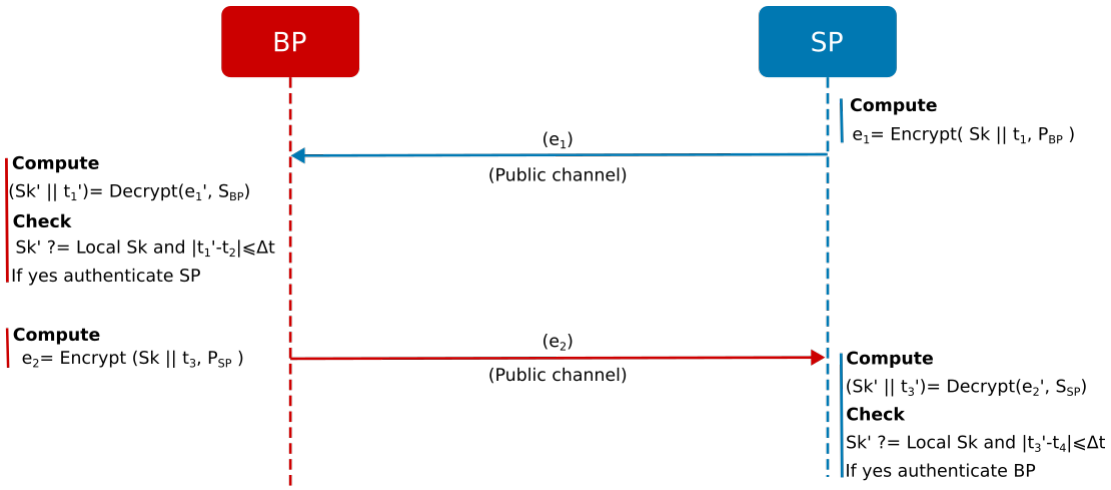


Figure 5.6: Authentication phase of LightQ shceme.

Algorithm 3 Encrypt

Input

m, P_{BP} /* plaintext, public key of the recipient*/

Output

e /* ciphtertext */

1: SP Choose random small vector r
2: Use BP public key P_{BP} to compute $e = mW + r$
3: Send the ciphtertext e to BP

Algorithm 4 Decrypt

Input

e, S_{PB} /* ciphtertext, private key of the recipient*/

Output

m /* plaintext */

1: Use Babai's algorithm to compute the vector $v \in L$ closest to e
2: Compute vW^{-1} to recover m

5.5 Security evaluation

In this section, we carry out an informal security analysis to examine the security features of our proposed scheme. In addition, we verify the security of the system using the AVISPA tool.

5.5.1 Informal security analysis

- **Quantum attack:** Our system resists quantum attacks by using QKD to generate the shared secret key *Sk*. QKD uses qubits, which are exchanged via a quantum channel. To obtain the secret key *Sk*, an attacker would have to intercept the exchanged photons and guess the transmission basis. However, the intercepted message can be detected because the photons completely change state once they are filtered with a basis sequence. In addition, we use GGH cryptography in the next phase, which is also based on lattice cryptography. To break GGH cryptography, we need to solve the GGH decision problem, which is a well-known hard problem. Consequently, our system offers strong protection against quantum attacks using both QKD and GGH cryptography mechanisms.

- **MITM attack:** MITM attacks are a major concern in secure communication. In such attacks, a malicious third party intercepts the communication between two legitimate parties and manipulates the messages in transit to deceive them. If the attacker intercepts the authentication message, they would need to have knowledge of the shared secret key and store it in memory to transform the message into another legitimate one. However, the use of QKD in the shared key agreement phase ensures that the secret key cannot be accessed by the attacker, making it impossible for them to successfully launch such an attack. Another scenario is when an attacker intercepts the public key and tries to perform an MITM attack by using the receiver's public key. To succeed in this attack, the attacker would need to compute a valid private key for the receiver's public key. However, this is not possible because the private key is computed by solving the SVP on the lattice generated by the public key. If the original public key is chosen from a good basis, then the lattice generated by the attacker's public key will be far from the original lattice, making it hard to solve the SVP problem on it, even for quantum computers. Therefore, the attacker cannot compute a valid private key for the receiver's public key. As a result, the MITM attack would be detected.

- **Replay attack:** In our scheme, we achieve this by including the message transmission time as a timestamp, which is encrypted and embedded within the message itself. This approach ensures that if an attacker attempts to replay the message by generating a valid time t_1' within a specific time frame Δt, the receiver can detect any tampering by verifying that the encrypted timestamp contained within the received message matches the original one sent. To successfully carry out a replay attack, the attacker would need to decrypt the message in order to modify the encrypted transmission time. However, this task is infeasible without knowledge of the receiver's secret key. Through this mechanism, our scheme effectively prevents replay attacks by verifying the freshness of the timestamp and detecting any attempts to manipulate the message's transmission time.

- **DoS attack:** A DoS attack is characterized by sending large amounts of data to the same site almost simultaneously to deny its service. In our scheme, we rely on valid timestamps to verify when a message was received. It will be rejected if many messages have been received with identical timestamps. In this way, DoS attacks will be avoided.

- **Impersonation attack:** An impersonation attack involves an attacker trying to pass themselves off as a legitimate user by using some of their private information. In the event that an attacker successfully launches a replay attack on an authentication request, they still won't be able to decrypt a message from a legitimate user without the private key. As only the rightful user has access to this key, our scheme is highly resilient to impersonation attacks.

- **Brute force attack:** The size of the shared secret key in our scheme is determined by the QBER. Without knowing the final key size, a brute force attack that tries all possible cases cannot determine the secret key. Furthermore, even if the attacker is able to measure or calculate the key, it does not compromise the security of the system as we use public key cryptography for message exchange.

- **Mutual authentication:** Mutual authentication means that each party verifies the legitimacy of the other. In our scheme, the authentication is based on the shared secret key. Each party has to compute $Encrypt(Sk||t_1, P_{BP}/P_{SP})$ where Sk is known only by the legitimate party.

- **Eavesdropping detection:** Our scheme uses QKD in the authentication phase to ensure the security and confidentiality of the system. QKD employs photons and qubits in a quantum channel to pre-

vent eavesdropping, allowing only the intended parties to obtain the shared secret key. This key is used solely for authentication, ensuring the confidentiality of information and preventing unauthorized access. By leveraging QKD, we provide a robust mechanism for preventing eavesdropping and guaranteeing the security of our system.

5.5.2 Formal security analysis

To verify the security properties of our scheme during the transmission of data between BP and SP, we implemented two main roles: BP and SP. Next, we defined the role of the session that contains the basic roles with their arguments. The role of the environment contains the constants and composition of sessions. Finally, we defined the proposed scheme's security objectives. The results obtained are presented in Figure 5.7. They illustrate the achievement of our proposed scheme on all the goals and security requirements. Our proposed scheme is **SAFE** under OFMC and CL-AtSe back-end and therefore, it is safe against all attacks.

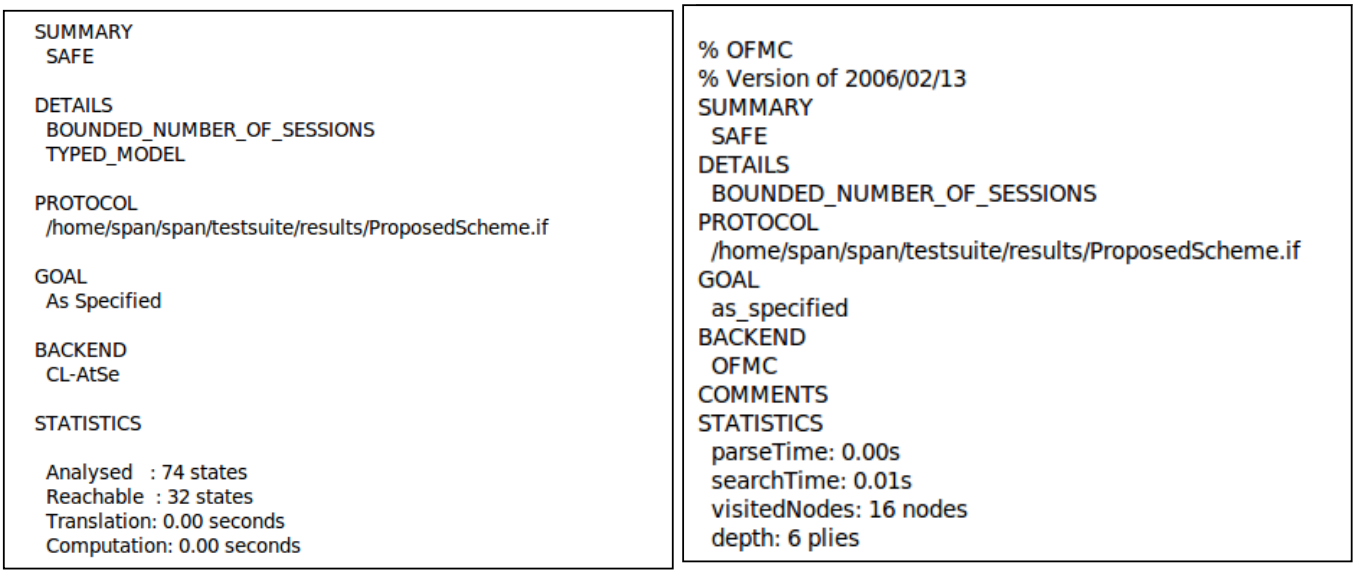


Figure 5.7: Results of LightQ scheme using OFMC and CL-AtSe back-ends of AVISPA.

5.6 Comparative analysis

To assess the effectiveness of the proposed system, we carried out a comparative analysis with existing solutions. Our evaluation focuses on security features, computational cost and communication overhead. The comparison highlights the advantages of our approach in resisting various cyber threats, including quantum attacks, while maintaining efficiency. The following subsections provide a detailed discussion of these aspects.

5.6.1 Security features

A comprehensive comparison of the proposed scheme with recent works was conducted to assess its effectiveness. Table 5.2 presents the comparison results, demonstrating that the scheme satisfies all the specified security properties and is the only one capable of resisting quantum attacks. In contrast, other schemes lack essential properties. For instance, the schemes in [160, 161, 162] are vulnerable to eavesdropping, DoS, and quantum attacks, while the work in [163] fails to provide security against eavesdropping and quantum attacks. The proposed scheme stands out as a reliable and robust solution for secure communication in the presence of quantum adversaries.

Table 5.2: Comparing security features of LightQ scheme with recent works.

Ref	F1	F2	F3	F4	F5	F6	F7	F8
[144]	×	✓	✓	✓	×	×	✓	✓
[160]	×	✓	✓	×	×	✓	✓	✓
[161]	×	✓	✓	×	×	✓	✓	✓
[162]	×	✓	✓	✓	×	✓	✓	✓
[163]	×	✓	✓	×	×	✓	✓	✓
[164]	×	✓	✓	×	×	×	✓	✓
[165]	×	✓	×	×	×	✓	✓	✓
LightQ	✓	✓	✓	✓	✓	✓	✓	✓

F1: Quantum attack, **F2:** Replay attack, **F3:** MITM attack, **F4:** DoS attack, **F5:** Eavesdropping detection, **F6:** Brute force attack, **F7:** Impersonation attack, **F8:** Mutual authentication, ✓denotes the resistance of the system to the attack in question, × denotes the vulnerability of the system to the attack in question.

Table 5.2 clearly demonstrates that the LightQ scheme provides superior security features compared to all other referenced works. While other schemes address some traditional attacks like replay, MITM,

DoS, impersonation, none of them offer explicit resistance to quantum attacks, which is a critical vulnerability in the evolving threat landscape. LightQ uniquely leverages QKD and GGH cryptography to provide robust protection against quantum adversaries, as well as enabling eavesdropping detection. This comprehensive security posture makes LightQ a future-proof solution, addressing a fundamental gap in existing cryptographic protocols for IoT-based EI systems. The ability to defend against quantum attacks, coupled with resilience to classical threats, highlights LightQ's significant advancement in the field.

5.6.2 Computational cost

The computational overhead of the proposed scheme was evaluated, focusing on the authentication and key agreement phases, which are commonly utilized in other schemes such as [160, 161, 162, 163].

To obtain the computational cost of QKD, which is not readily available, the QKD protocol was simulated on a local machine. The open-source QKD code available on GitHub (2017) was used for the simulation. The simulation was conducted on a desktop machine with an Intel Core i5-9700K CPU, 8 GB RAM, and Linux Mint 21 operating system. The encryption method was implemented in Python 3.10.6, and its execution time was measured using the timeit module in Python. A series of tests was performed using 228 qubits, with the encryption process repeated 5,000 times. As the resulting key size depended on the QBER, key sizes ranging from 0 to 228 bits were obtained. To account for variability, the average execution time for each key size was calculated, and the results were plotted (see Figure 5.8).

The results presented in Table 6.3 indicate that while the computational cost of the approach proposed in [160] is lower than our scheme, it suffers from several attacks as listed in Table 5.2. On the other hand, compared to [144, 161, 163, 164, 165, 166], our scheme provides better efficiency and security. A comparative graph is provided in Figure 5.9.

5.6.3 Communication cost

Assuming the parameters mentioned in Section 3.4, we analyze the communication cost for the authentication and shared key agreement phase. ECC points, hash function outputs, identity, and timestamps are represented by 320, 256, 64, and 32 bits, respectively.

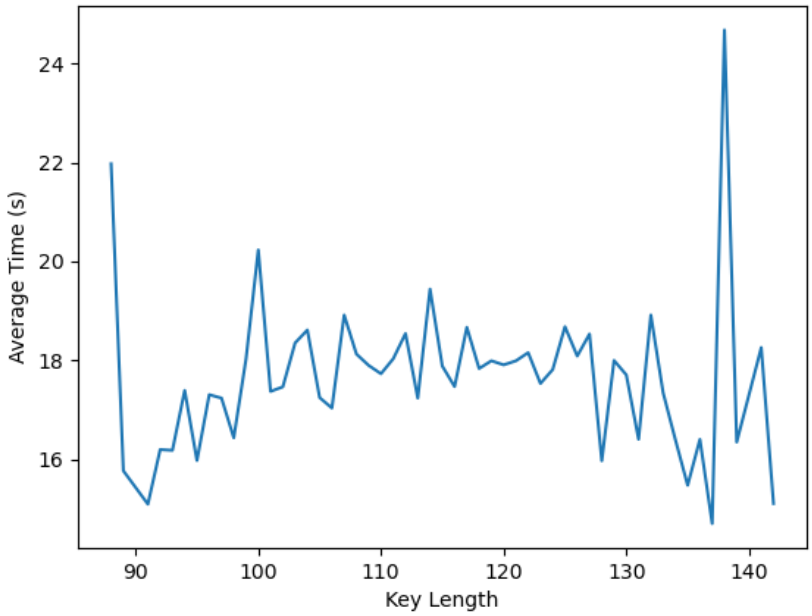


Figure 5.8: Execution time for QKD with different key size.

Table 5.3: Computational cost comparison of LightQ scheme with recent works.

Scheme	Operation	Total (ms)
[144]	$2T_b + 11T_m + 2T_a$	122.042
[160]	$12T_h + 4T_m$	9.972
[161]	$8T_m + 6T_h + 2T_a$	18.578
[163]	$2T_b + 4T_a + 8T_m + 2T_h + 2T_H + 2T_{se/d}$	140.623
[164]	$10T_m + 4T_a$	22.732
[165]	$8T_m + 4T_a$	18.28
[166]	$4T_b + 7T_m + 2T_a + 2T_H$	429.934
LightQ	$4T_{ae/d} + 1T_{QKD}$	15.415

In our scheme's authentication phase, two messages, $M_1\{e_1\}$ and $M_2\{e_2\}$, are exchanged. This involves concatenation of the result of encryption Sk with a timestamp, requiring a total of 320 bits. It is worth noting that the GGH cryptosystem produces a ciphertext of the same length as the plaintext.

In [164], the authentication phase employs two messages: $\{X_i, Y_i, K_{ip}, ID_i, t_i\}$ and $\{X_j, Y_j, K_{jp}, ID_j, t_j\}$. This includes six ECC points, two timestamps, and two identities, requiring a total of 2112 bits. In [163], four ECC points, two timestamps, two identities, and one hash function output are included in the two messages exchanged during the authentication phase: $M_1\{ID, t_1, \sigma_1, \sigma_2\}$ and $M_2\{ID, t_3, U_1, U_2, C_1\}$. This results in a total of 1600 bits. In [160], two messages, $M_1\{V_i, Z_j, SKV_{ij}, T_2\}$ and $M_2\{R_B, SKV_{ij}*, T_3\}$, are used in the authentication phase. This involves two ECC points, three timestamps, and four hash outputs,

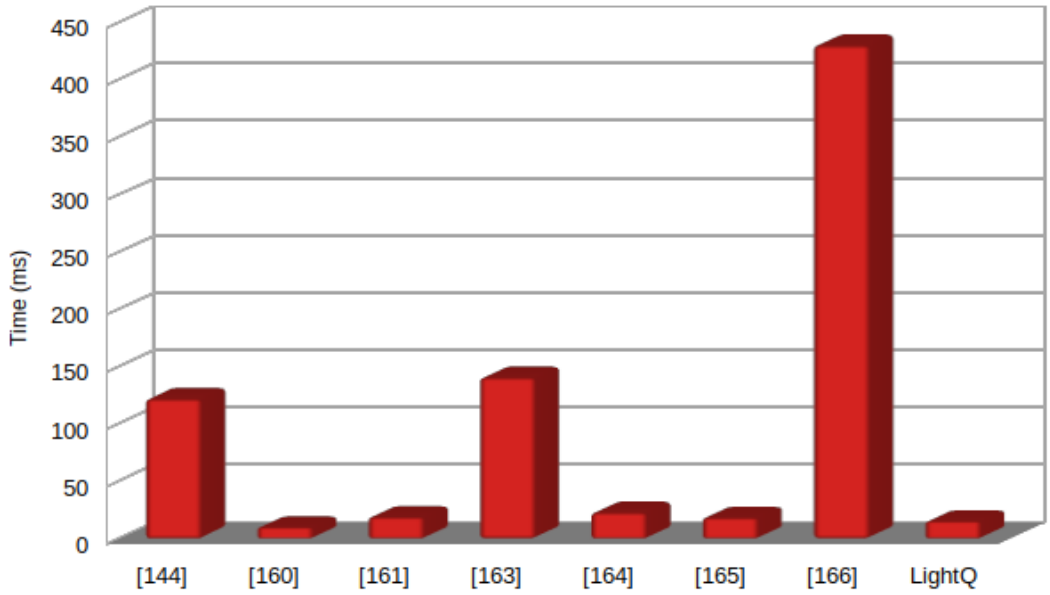


Figure 5.9: Computational cost.

requiring a total of 1760 bits.

In [144], two messages, $\{ID, ZR_{in}, R_{i1}, R_{si}, T_1, A_i\}$ and $\{ID, ZR_{jn}, R_{j1}, R_{sj}, T_2, A_j\}$, are used in the authentication phase. This includes eight ECC points, two timestamps, and two identities, requiring a total of 2752 bits. In [161], the authentication phase employs three messages: $M_1\{CA_H, R_H, C_1, t_1\}$, $M_2\{R_B, C_3, t_2\}$, and $M_3\{E_{Sk_x}(C_3||M)\}$. This includes five ECC points, two timestamps, and one symmetric ciphertext, requiring a total of 1792 bits. In [165] used three messages in the authentication phase $\{id_A, R_A, WT_A\}$ $\{id_B, R_B, V_B, WT_B\}$ and $\{id_A, V_A\}$ so four ECC points, three identities and two hash outputs in total 1984 bits. In [166], the authors used two messages in the authentication phase $\{ID_i, R_{in}, R_{i1}, R_{si}, T_1\}$ and $\{ID_j, R_{jn}, R_{j1}, h_j\}$; so, five ECC points, one timestamps and two identities and one hash output, in total 2016 bits.

Table 5.4 demonstrates that our scheme not only provides superior security features, but also incurs lower communication costs than other schemes. Furthermore, Table 5.2 presents an overview of the security vulnerabilities associated with these other schemes, and Figure 5.10 provides a comparative analysis of the communication costs.

Table 5.4: Communication cost comparison of LightQ scheme with recent works.

Ref	N message	Communication costs	Total (bits)
[144]	2	$8\|ECC\| + 2\|T\| + 2\|ID\|$	2752
[160]	2	$2\|ECC\| + 4\|H\| + 3\|T\|$	1760
[161]	3	$5\|ECC\| + 2\|T\| + \|C_s\|$	1792
[163]	2	$4\|ECC\| + 2\|T\| + 2\|ID\| + \|C_s\|$	1600
[164]	2	$6\|ECC\| + 2\|T\| + 2\|ID\|$	2112
[165]	3	$4\|ECC\| + 3\|ID\| + 2\|H\|$	1984
[166]	2	$5\|ECC\| + \|T\| + 2\|ID\| + \|H\|$	2016
LightQ	2	$2(\|SK\| + \|T\|)$	320

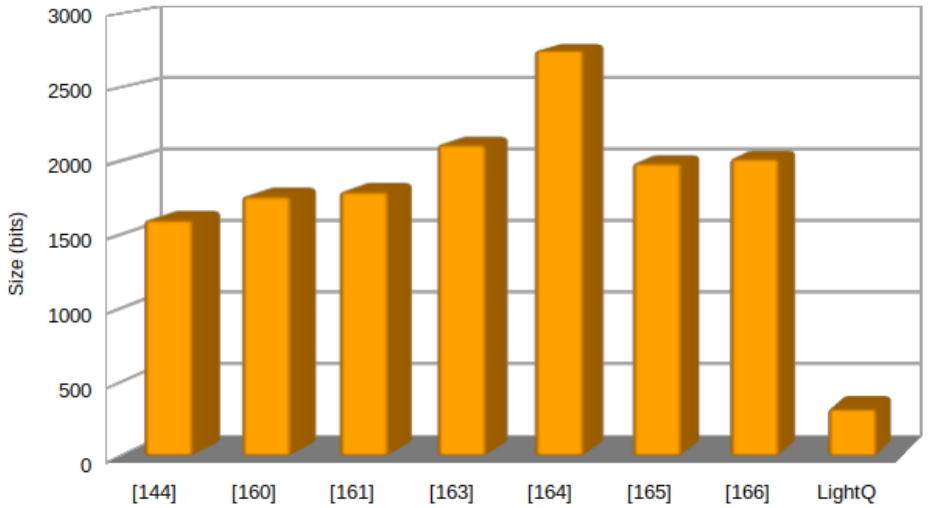


Figure 5.10: Communication cost.

5.6.4 Storage cost

This section evaluates the storage overhead of the proposed scheme, focusing on the cost of storing cryptographic elements in a single entity, which allows for a fair comparison with other schemes that do not store the key in both entities.

The results show that the proposed scheme offers better storage overhead, an important factor for resource-constrained devices like SMs and IoT nodes. A comparison is provided in Table 5.5, and a visual representation is shown in Figure 5.11. Efficient storage usage is crucial in environments with limited memory and processing power, making the minimization of storage overhead while maintaining adequate security a key consideration in cryptographic system design.

Table 5.5: Storage cost comparison of LightQ scheme with recent works.

Ref	Stored element	Storage cost (bits)
[144]	$\|H\| + \|ID\|$	320
[160]	$\|H\|$	256
[161]	$\|ECC\|$	320
[163]	$\|ID\| + \|ECC\|$	384
[164]	$\|H\| + \|ID\|$	320
[165]	$\|H\|$	256
[166]	$2\|ECC\| + \|ID\|$	704
LightQ	$\|SK\|$	128

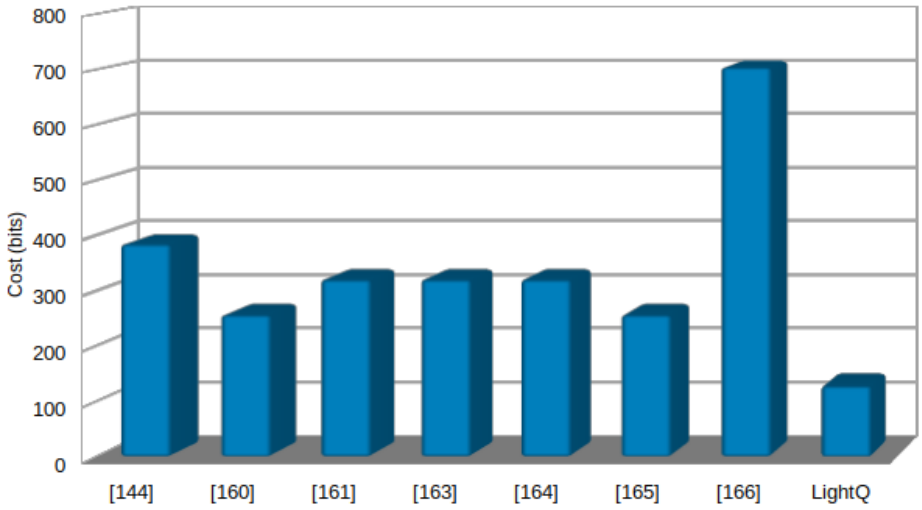


Figure 5.11: Storage cost.

5.7 Conclusion

In this chapter, we introduced a secure authentication scheme for communication between the BP and SP in an IoT-based EI environment, leveraging the GGH cryptosystem and QKD. QKD enables the generation of a highly secure shared secret key, providing resistance to quantum attacks and allowing for eavesdropping detection. However, due to the high cost and specialized hardware requirements of QKD, we limit its use to the key agreement phase. Once the shared key is established, the authentication process relies on the GGH cryptosystem, where each entity encrypts the secret key using the receiver's public key. Successful authentication is achieved when the decrypted key matches the pre-established one. This hybrid approach ensures strong security while maintaining efficiency in terms of computational, communication, and storage overheads.

In the next chapter, we build upon this work by integrating this authentication scheme with the blockchain-based solution proposed in Chapter 4. By combining the strengths of both approaches, we aim to develop a post-quantum blockchain solution that enhances security and resilience against emerging quantum threats in the EI.

Chapter 6

PQBlock: Secure and Efficient Mutual Authentication Protocol in IoT-based EI using Post-Quantum Blockchain

6.1 Introduction

This chapter presents a combined approach that integrates the two previous schemes, SemAuth described in Chapter 4 and LightQ discussed in Chapter 5. The integration leverages the advantages of both schemes, with one ensuring decentralization and the other providing security against quantum attacks.

The proposed solution aims to ensure the confidentiality, integrity, and availability of critical information, thereby strengthening the resilience of the energy grid. We validate the security of the proposed protocol using BAN logic and ProVerif tool [167].

6.2 System model

To address the specific requirements and challenges in the EI domain, the architecture proposed in [2] has been adapted and enhanced. While building upon the original framework, modifications and additions

have been made to cater to the unique needs of the system. The enhanced architecture is depicted in Figure 6.1. Table 6.1 illustrates the set of notations used in rest of the chapter.

Table 6.1: Key notations of PQBlock scheme.

Notations	Description
$\mathcal{H}(B)$	The Hadamard Ratio of the basis B.
L	Lattice of dimension n
SP_i	i^{th} Seller Prosumer
BP_j	j^{th} Buyer Prosumer
Sk_{SP_i}, Pk_{SP_i}	Private and Public key pairs of SP_i
Sk_{BP_j}, Pk_{BP_j}	Private and Public key pairs of BP_j
ID	Identifier
t_i	Timestamp
Δt	Maximum transmission delay
EM, EP	Energy Mount, and Price
$det(L)$	Determinant of L
ϵ	Cutoff value
e	Euler's number

- **Energy layer:** The energy layer plays a crucial role in the EI architecture, comprising the necessary physical infrastructure and components for energy generation, distribution, and consumption. It encompasses renewable energy sources, energy storage systems, power transmission and distribution networks, and SG technologies. The primary objective of this layer is to enable efficient energy generation, conversion, and distribution within the EI ecosystem [2]. It is important to emphasize that the IoT serves as the foundational framework supporting the overall architecture of the EI [168].

- **Decentralized control layer:** To enhance scalability, fault tolerance, and resilience, we propose the implementation of a decentralized control layer. This layer utilizes multiple distributed Registration Authorities (RAs) to decentralize control functions, reducing the risk of single points of failure and enabling efficient resource management [142]. Additionally, the architecture incorporates blockchain technology for secure data storage within the RAs. By leveraging the decentralized and tamper-resistant nature of the blockchain, it ensures data integrity, immutability, and security. This approach effectively mitigates the possibility of unauthorized modifications or data breaches, providing a robust and secure framework for storing and managing critical information in the EI ecosystem [14].

- **Secure communication model:** To facilitate seamless communication among the energy layer, decentralized control layer, and IoT devices, a secure communication model has been developed. This model incorporates robust authentication mechanisms, encryption protocols, and secure data transmission techniques to safeguard sensitive information and prevent unauthorized access [2].

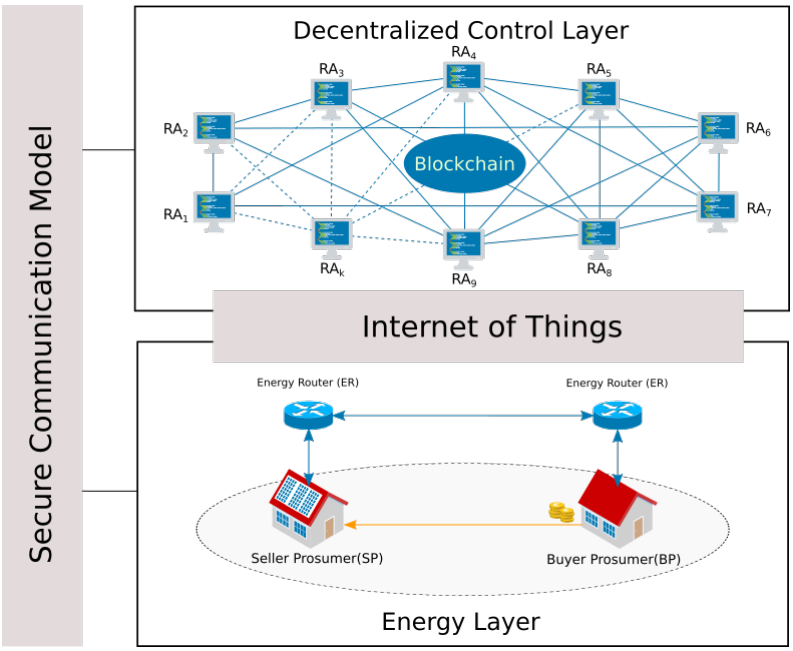


Figure 6.1: System model of PQBlock scheme.

6.3 Proposed scheme

The enhanced protocol builds upon the existing authentication schemes, LightQ and SemAuth, by integrating their strengths and introducing additional security enhancements and optimizations. Specifically, it adopts the decentralization features of SemAuth and the quantum-resistant properties of LightQ. The protocol is structured into three key phases: initialization, registration, and authentication. These phases collectively aim to establish a secure and resilient framework for verifying participant identities and enabling trustworthy energy transactions within the network.

6.3.1 Initialization

In the initialization phase of the protocol, the keys are generated using the GGH cryptosystem. The key generation process involves the following steps:

- Choosing a good basis: $v_1, ..., v_n$ and a bad basis: $w_1, ..., w_n$ for the lattice L.
- Keeping the good basis as the secret key $Sk = v_1, ..., v_n$.
- Publishing the public key $Pk = w_1, ..., w_n$.

The process of choosing the good and bad basis is described in Algorithm 5. In each iteration, a basis: $x_1, x_2, ..., x_n$ is selected, and the Hadamard ratio is computed to determine if it qualifies as a good basis. If the basis is deemed good, it is chosen as the private key Sk; otherwise, it becomes the public key Pk.

Algorithm 5 Initialization

Input
Lattice L
Output
$Sk; Pk$

$v_1, v_2, ..., v_n = \text{NULL};$
$w_1, w_2, ..., w_n = \text{NULL};$
$Sk = v_1, v_2, ..., v_n;$
$Pk = w_1, w_2, ..., w_n;$

while $(Sk = \text{NULL})$ or $(Pk = \text{NULL})$ **do**
  if $(\frac{\det(L)}{|x_1| \cdot |x_2| \cdot ... \cdot |x_n|})^{1/n} \leq 0.5$ **then**
    $w_1, w_2, ..., w_n \leftarrow x_1, x_2, ..., x_n;$
    $Pk = w_1, w_2, ..., w_n;$
  else
    $v_1, v_2, ..., v_n \leftarrow x_1, x_2, ..., x_n;$
    $Sk = v_1, v_2, ..., v_n;$
  end if
end while

6.3.2 Registration

During the registration process, both the SP_i and the BP_j submit their information to the RA_k. This information includes their public keys Pk, signatures Sig, energy quantities, requested energy amounts EM, and prices EP. The RA_k diligently verifies the accuracy and authenticity of this information.

Upon successful verification, the RA_k incorporates the registered data into the blockchain database. This decentralized ledger guarantees transparency and immutability of the transaction records. The RA_k then issues the necessary verification, certifying the validity and authenticity of the registered information. This verification acts as a trusted confirmation for all participants involved, instilling confidence and upholding the integrity of the energy trading process. The detailed process is outlined below:

1. The SP_i generates a random value, as here ID_{SP_i}, and applies Babia's algorithm 1 using a good basis Sk_{SP_i} to compute a vector s from the set L that closely approximates ID_{SP_i}. Write $s = a_1 \cdot w_1 + a_2 \cdot w_2 + ... + a_n \cdot w_n$, then defined the signature of SP_i as: $Sig_{SP_i} = (a_1, a_2, ..., a_n)$, SP_i then sends the message $M_1 = \{Pk_{SP_i}, EM, EP, Sig_{SP_i}, ID_{SP_i}, t_1\}$ to the corresponding RA_k.

2. RA_k invokes the *Registration* function (illustrated in Algorithm 6). Initially, the *Registration* function checks whether the SP's record already exists in the blockchain. If the data exists, the request is declined.

3. Otherwise, the RA_k invoke *SignatureVerify* (illustrated in Algorithm 7) to check the validity of the signature. If the verification succeeds, the SP's information is added as a blockchain record. Subsequently, a consensus process is initiated with other RAs.

4. The registration process of BP_j is similar to SP_i. Therefore, the steps are omitted here.

Algorithm 6 Registration

Input
Prosumer information: Pk, EM, EP, Sig, ID
Output
Registration status

```
if Prosumer record exists in the blockchain then
    Decline the request
else
    Verify the validity of Sig using SignatureVerify
    if Verification succeeds then
        Add Prosumer information as a blockchain record

        Initiate a consensus process with other RAs
    else
        Decline the request
    end if
end if
```

Algorithm 7 SignatureVerify

Input
$Sig = (a_1, a_2, ..., a_n)$; $Pk = w_1, w_2, ..., w_n$; Challenge c
Output
Signature valid or invalid
1: $s = a_1 \cdot w_1 + a_2 \cdot w_2 + ... + a_n \cdot w_n$
2: $\epsilon \approx \frac{n(det(L))^{\frac{1}{n}}}{\sqrt{2\pi e}}$
3: **if** $\|s - c\| \leq \epsilon$ **then**
4: Signature valid
5: **else**
6: Signature invalid
7: **end if**

6.3.3 Mutual authentication

When the BP_j selects an SP_i from the blockchain record for energy purchase, establishing mutual authentication is crucial to ensure secure transactions. The authentication process, illustrated in Figure 6.2, involves the following steps:

1. BP_j initiates the authentication process by generating a random challenge value, c_1, and encrypting it using the public key of SP_i, denoted as Pk_{SP_i}. The encrypted challenge value is represented as $E_1 = Encrypt(c_1, Pk_{SP_i})$ (Refer to Algorithm 3 for the encryption process). Subsequently, BP_j sends the message $M_1 = \{E_1, t_1\}$ to SP_i, where t_1 represents the timestamp of the message.

2. The SP_i checks the freshness of M_1 by comparing the timestamp t_1 with a predefined threshold Δt. If the freshness condition holds ($t_1 - t_1' \leq \Delta t$), the SP_i proceeds with the authentication process. The SP_i decrypts the received message M_1 using its secret key Sk_{SP_i}, obtaining $c_1' = Decrypt(E_1', Sk_{SP_i})$ (see Algorithm 4). Next, the SP_i computes the signature $s_1 = \text{Sig}(c_1', Sk_{SP_i})$ and generates a random challenge value c_2. The SP_i encrypts c_2 using the BP_j's public key, resulting in $E_2 = Encrypt(c_2, Pk_{BP_j})$. The SP_i constructs the message $M_2 = \{E_2, t_2, s_1\}$ and sends it to the BP_j.

3. BP_j verifies the correctness of the received signature s_1 using the *SignatureVerify* function. If the signature is valid and matches the challenge value, the authentication of the SP_i is considered successful. BP_j then decrypts the received challenge value, obtaining $c_2' = Decrypt(E_2', Sk_{BP_j})$. BP_j computes the signature $s_2 = \text{Sig}(c_2', Sk_{BP_j})$ and constructs the message $M_3 = \{s_2, t_3\}$, which is sent

to the SP_i.

4. The SP_i verifies the correctness of the received signature s_2 using the *SignatureVerify* function. If the signature is valid and matches the challenge value, the authentication of the BP_j is considered successful.

5. Both the SP_i and BP_j compute the session key as follows: $Ssk = (c_1 \oplus ID_{SP_i} || c_2 \oplus ID_{BP_j})$.

By following this mutual authentication process, both the BP_j and SP_i validate each other's identities, ensuring trust and security during energy transactions. The computation of a session key further strengthens the security of the communication between the two parties.

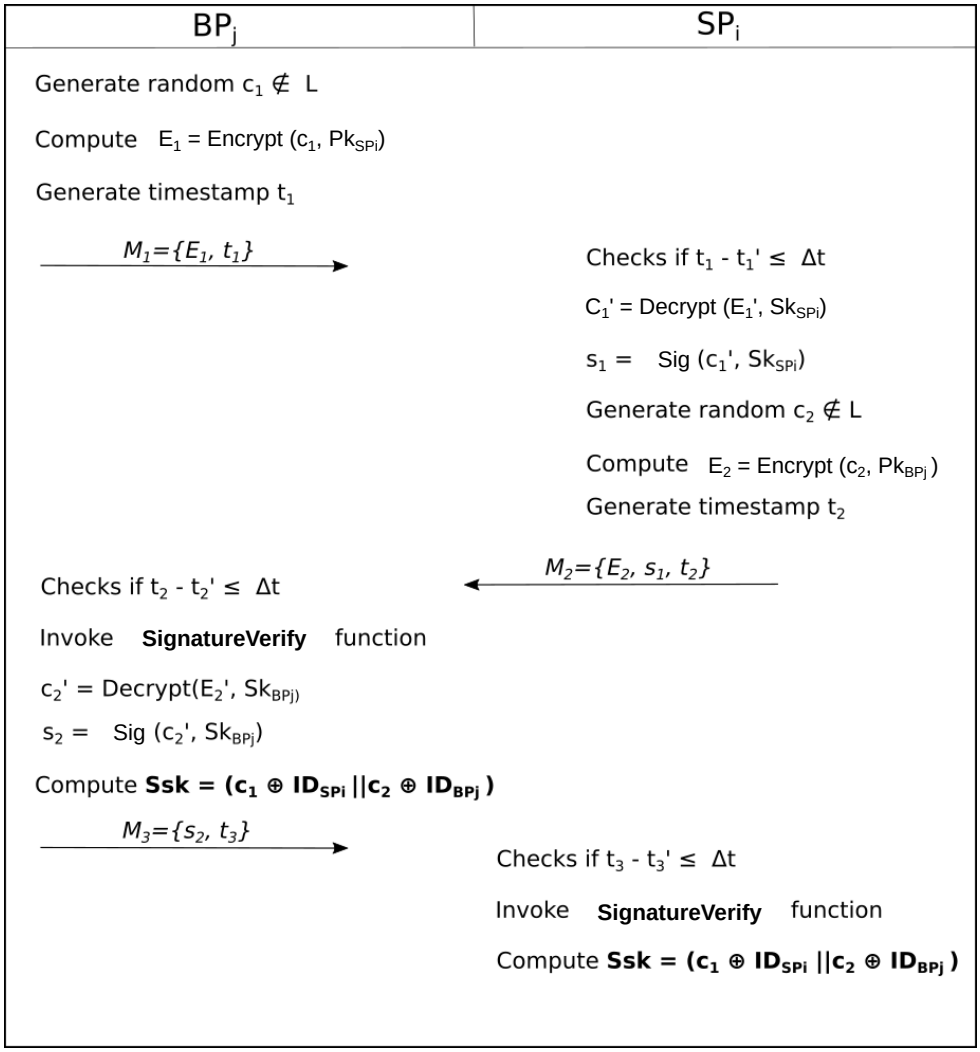


Figure 6.2: Authentication phase of PQBlock scheme.

6.4 Security evaluation

A comprehensive security analysis is carried out for the enhanced authentication protocol. Initially, an informal security analysis is conducted to demonstrate that the proposed system provides robust protection against common security threats. This is followed by a formal validation using BAN logic and the ProVerif tool. The analysis confirms that the protocol supports secure mutual authentication and session key agreement while ensuring resilience against various active and passive attacks, including replay and MITM attacks.

6.4.1 Informal security analysis

- **Quantum attack:** The enhanced protocol incorporates the GGH cryptosystem for key generation, which enhances its resistance against potential quantum attacks. By utilizing the lattice structure and the good basis for encryption and decryption operations, the protocol leverages the security properties of the GGH cryptosystem. This choice of cryptographic algorithm strengthens the protocol's security against attacks from quantum computers.

- **MITM attack:** To protect against MITM attacks, the enhanced protocol includes a mutual authentication mechanism during the authentication phase. Both the SP and the BP generate and exchange challenge values c_1 and c_2, encrypting them with the respective public keys. By verifying the received challenges and signatures, the protocol ensures the authenticity of both entities, mitigating the risk of MITM attacks.

- **Mutual authentication:** The enhanced protocol implements a mutual authentication process during the authentication phase. Both the SP and BP generate challenge values, exchange and decrypt them using their secret keys, and verify the received signatures. This mutual authentication mechanism ensures that both entities authenticate each other's identities before proceeding with the energy transaction, enhancing the overall security of the protocol.

- **Replay attack:** In order to mitigate replay attacks, our protocol incorporates the use of timestamps to verify the authenticity and freshness of transmitted data. Each party in the protocol checks the freshness of the received timestamp by comparing it to the most recently received one. Specifically,

we verify that the time difference between the current timestamp t_1 and the previously received timestamp t_1' is less than or equal to a predefined threshold Δt. This enhancement greatly enhances the protocol's resistance against replay attacks, ensuring the integrity and freshness of the exchanged messages.

- **DoS attack:** The distributed architecture of blockchain offers a significant advantage in mitigating DoS attacks. By storing authentication records across multiple nodes in the network, the protocol becomes more resilient to attacks that aim to disrupt the authentication process. This distributed approach makes it challenging for attackers to target a single point of vulnerability, as authentication records are spread across multiple nodes. As a result, the blockchain-based system enhances the security and reliability of the authentication process, protecting it from DoS attacks.

- **Decentralization:** The enhanced protocol incorporates a blockchain-based decentralized ledger for registering and verifying participants' information. This decentralized approach enhances the security and transparency of the authentication process, as it guarantees the immutability and transparency of transaction records. By distributing authority and control across multiple entities or nodes, the protocol reduces the risk of a single point of failure or compromise, thereby strengthening its overall security posture.

6.4.2 Formal security analysis

- **BAN logic:** Our protocol aims to fulfill the following goals, as determined through the application of BAN logic [132], a widely recognized formal process for validating security protocols [169]. The specific notations and rules of BAN logic are detailed in Section 3.2.1.

  G_1: $SP_i \mid\equiv (SP_i \overset{Ssk}{\longleftrightarrow} BP_j)$

  G_2: $BP_j \mid\equiv (SP_i \overset{Ssk}{\longleftrightarrow} BP_j)$

  G_3: $SP_i \mid\equiv BP_j(SP_i \overset{Ssk}{\longleftrightarrow} BP_j)$

  G_4: $BP_j \mid\equiv SP_i(SP_i \overset{Ssk}{\longleftrightarrow} BP_j)$

  First, we transform the proposed protocol into an idealized form:

  M_1: $BP_j \longrightarrow SP_i : \{c_1\}_{Pk_{SP_i}}$

M_2: $SP_i \longrightarrow BP_j : \{\{c_2\}_{Pk_{BP_j}}, \{c_1\}_{Sk_{SP_i}}\}$

M_3: $BP_j \longrightarrow SP_i : \{c_2\}_{Sk_{BP_j}}$

Second, we make the following assumptions about the initial state of the protocol for analysis:

A_1: $SP_i \mid\equiv \#(Sk_{SP_i})$

A_2: $BP_j \mid\equiv \#(Sk_{BP_j})$

$A3$: $BP_j \mid\equiv \#(c_1)$

A_4: $SP_i \mid\equiv \#(c_2)$

A_5: $BP_j \mid\equiv \overset{Pk_{SP_i}}{\longmapsto} SP_i$

A_6: $SP_i \mid\equiv \overset{Pk_{BP_j}}{\longmapsto} BP_j$

Third, we analyze the idealized form of the proposed protocol based on the BAN logic rules and the given assumptions. The main proofs are as follows:

1. M_3: $BP_j \longrightarrow SP_i : \{c_2\}_{Sk_{BP_j}}$. According to the seeing rule, we obtain: S_1: $SP_i \triangleleft (\{c_2\}_{Sk_{BP_j}})$. Based on A_6, S_1, and R_4 rule, we derive: S_2: $SP_i \mid\equiv BP_j \mid\sim c_2$. Using A_4, S_2, and R_3 rule, we deduce: S_3: $SP_i \mid\equiv BP_j \mid\equiv c_2$. Here, c_2 represents the necessary parameter of the session key in the proposed protocol. Using A_4, S_3, and R_1 rule, we obtain: G_1: $SP_i \mid\equiv (SP_i \overset{Ssk}{\longleftrightarrow} BP_j)$ With A_4, G_1, and R_3 rule, we conclude: G_3: $BP_j \mid\equiv SP_i(SP_i \overset{Ssk}{\longleftrightarrow} BP_j)$.
2. M_2: $SP_i \longrightarrow BP_j : \{c_1\}_{Sk_{SP_i}}$. According to the seeing rule, we have: S_1: $BP_j \triangleleft (\{c_1\}_{Sk_{SP_i}})$. Using A_5, S_1, and R_6 rule, we derive: S_2: $BP_j \mid\equiv SP_i \mid\sim c_1$. Applying A_3, S_2, and R_3 rule, we deduce: S_3: $SP_i \mid\equiv BP_j \mid\equiv c_1$. Here, c_1 represents the necessary parameter of the session key in the proposed protocol. Using A_4, S_3, and R_1 rule, we obtain: G_2: $BP_j \mid\equiv (SP_i \overset{Ssk}{\longleftrightarrow} BP_j)$. With A_3, G_2, and R_3 rule, we conclude: G_4: $BP_j \mid\equiv SP_i(SP_i \overset{Ssk}{\longleftrightarrow} BP_j)$.

- **ProVerif:** By employing ProVerif, we analyze the proposed scheme and closely observe the process to ensure its security. The verification results obtained using ProVerif are presented in Figure 6.3, conclusively demonstrating the security of our enhanced protocol.

```
Completing equations...
-- Process 1-- Query not attacker(IDi[]) in process 1
Translating the process into Horn clauses...
Completing...
Starting query not attacker(IDi[])
RESULT not attacker(IDi[]) is true.
-- Query not attacker(IDj[]) in process 1
Translating the process into Horn clauses...
Completing...
Starting query not attacker(IDj[])
RESULT not attacker(IDj[]) is true.
-- Query not attacker(Ssk[]) in process 1
Translating the process into Horn clauses...
Completing...
Starting query not attacker(Ssk[])
RESULT not attacker(Ssk[]) is true.
-- Query inj-event(endEntitySPi(id)) ==> inj-
event(beginEntitySPi(id)) in process 1
Translating the process into Horn clauses...
Completing...
Starting query inj-event(endEntitySPi(id)) ==> inj-
event(beginEntitySPi(id))
RESULT inj-event(endEntitySPi(id)) ==> inj-
event(beginEntitySPi(id)) is true.
-- Query inj-event(endEntityBPj(id)) ==> inj-
event(beginEntityBPj(id)) in process 1
Translating the process into Horn clauses...
Completing...
Starting query inj-event(endEntityBPj(id)) ==> inj-
event(beginEntityBPj(id))
RESULT inj-event(endEntityBPj(id)) ==> inj-
event(beginEntityBPj(id)) is true.
```

Figure 6.3: The results of PQBlock scheme analysis using ProVerif.

6.5 Comparative analysis

A comparative analysis is provided to evaluate the proposed solution against existing approaches based on security features, computational cost, and communication cost. The aim is to highlight the strengths of the scheme, especially its capability to address a broad spectrum of security threats while maintaining operational efficiency.

6.5.1 Security features

Table 6.2 presents a comparison of various schemes based on their support for different security features. The schemes in [142, 144, 146, 147, 149] are evaluated according to their resilience against specific attacks and the authentication methods they employ. The proposed approach supports all listed security features, including protection against MITM attacks, DoS attacks, replay attacks, and others. Additionally, it addresses the potential threat posed by quantum attacks.

Table 6.2: Comparing security features of PQBlock scheme with recent works.

Ref	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
[142]	✓	✓	✓	×	✓	×	×	✓	✓	✓	×
[144]	✓	✓	✓	✓	✓	✓	✓	×	×	×	×
[146]	✓	N/A	✓	✓	✓	✓	✓	×	×	✓	×
[147]	✓	N/A	N/A	✓	✓	✓	✓	×	×	✓	×
[149]	✓	N/A	✓	✓	✓	✓	✓	×	×	✓	×
PQBlock	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

F1: MITM attack, **F2:** DoS attack, **F3:** Replay attack, **F4:** Impersonation attack, **F5:** Mutual authentication, **F6:** Session hijacking, **F7:** Information disclosure, **F8:** Batch verification, **F9:** RA decentralization, **F10:** Support blockchain-based solution, **F11:** Quantum attack.

6.5.2 Computational cost

The computational costs of the enhanced cryptographic scheme are compared with those of existing protocols. The analysis focuses specifically on the authentication phase and session key agreement, which represent the core components of the improved protocol. Execution times for cryptographic operations are measured using standard metrics commonly adopted in related studies [142, 144, 146, 147, 149].

The computational costs associated with each protocol are outlined in Table 6.3, which provides an overview of the required computational resources for both the BP_j (or SM_i) and the SP_i (or UC_j) for each protocol. The comparative results are depicted in Figure 6.4. While our protocol may not exhibit the lowest computational cost, it offers the highest level of security features.

Table 6.3: Calculation of the computational cost.

Ref	BP_j/SM_i	SP_i/UC_j
[142]	$2T_h + 4T_m + T_a$	$2T_h + 6T_m + T_a$
[144]	$4T_h + 6T_m + T_b$	$5T_h + 4T_m + T_b$
[146]	$15T_h + 2T_m + T_a$	$15T_h + 2T_m + T_a$
[147]	$7T_h + 4T_m + T_a$	$7T_h + 4T_m + T_a + 2T_{se/d}$
[149]	$8T_h + 4T_m + T_a$	$15T_h + 4T_m + 6T_a$
PQBlock	$2T_{ae/d}$	$2T_{ae/d}$

6.5.3 Communication cost

Various parameters and their corresponding sizes in bits are considered. The comparison of communication costs among different schemes is presented in Table 6.4, which provides the bit sizes for both

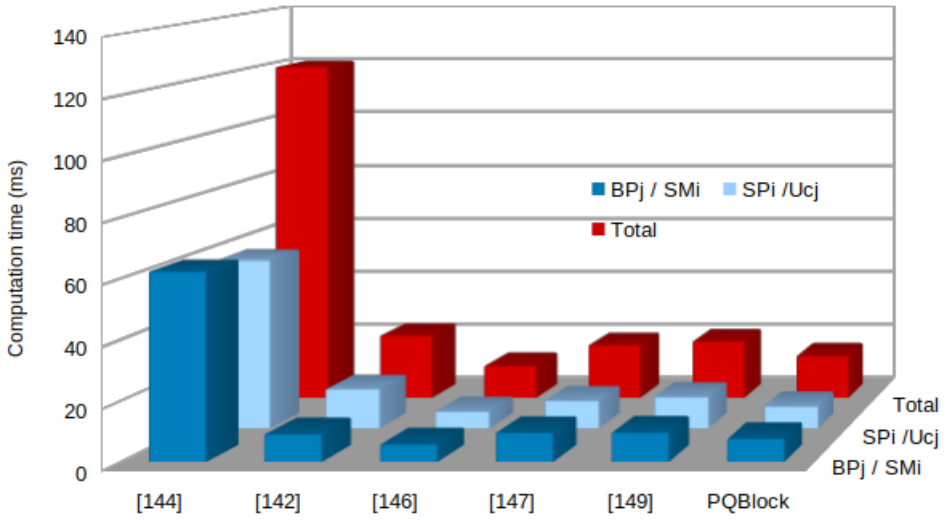


Figure 6.4: Comparing computational costs of PQBlock scheme with recent works.

BP_j/SM_i and SP_i/UC_j communication, as well as the total cost.

From the results in Table 6.4, it is evident that our proposed PQBlock scheme achieves a significant reduction in communication overhead compared to existing works. Specifically, PQBlock requires only 608 bits in total, which is considerably lower than other schemes, such as [146] with 3616 bits and [144] with 2752 bits. This reduction in communication cost enhances the efficiency of the proposed scheme.

Table 6.4: Comparison of communication costs of PQBlock scheme with recent works.

Ref	BP_j/SM_i	SP_i/UC_j	Total cost
[142]	1018 bits	928 bits	1964 bits
[144]	1376 bits	1376 bits	2752 bits
[146]	1760 bits	1856 bits	3616 bits
[147]	928 bits	800 bits	1728 bits
[149]	832 bits	992 bits	1824 bits
PQBlock	320 bits	288 bits	608 bits

The efficiency gain can be attributed to the optimized cryptographic operations and lightweight authentication mechanisms employed in PQBlock. By reducing the size of transmitted messages, our approach minimizes bandwidth consumption and decreases latency, which is crucial for real-time energy trading and secure data exchange in EI systems. Moreover, despite achieving lower communication overhead, our scheme maintains strong security guarantees, as demonstrated in the security feature comparison.

6.6 Conclusion

In this chapter, we proposed a mutual authentication protocol to enhance the security and resilience of EI. By integrating blockchain and the post-quantum cryptosystem GGH, our approach addresses vulnerabilities in traditional cryptographic methods, particularly their susceptibility to quantum attacks and the risks associated with centralization.

Through a detailed security analysis using BAN logic, ProVerif, and informal verification methods, we demonstrated the robustness of our scheme against various attacks. Additionally, a performance evaluation and comparative analysis highlighted the efficiency of our protocol in terms of communication and computational costs. The results confirm that our solution offers a secure and scalable authentication mechanism tailored for EI.

While this chapter focuses on securing authentication in the EI, another critical challenge remains: intrusion detection to resist unauthorized access attempts. In the next chapter, we address this gap by employing DL and XAI to develop an IDS. This approach enhances cybersecurity in the EI by enabling intelligent threat detection while ensuring transparency and interpretability in decision-making.

Chapter 7

XDetect: An Explainable CNN-based Intrusion Detection System for Enhanced Smart Grid Security

7.1 Introduction

The SG serves as the foundational building block of the EI, playing a central role in transforming electrical infrastructure. By integrating advanced communication technologies with traditional power grid components, the SG enables more efficient and intelligent management of energy generation, transmission, and distribution [3]. As discussed in previous chapters, the EI builds upon this evolution to create an interconnected ecosystem where producers, consumers, and operators can securely and dynamically exchange energy.

In this context, IDSs play a crucial role in proactively identifying security threats. However, understanding and interpreting how such systems make decisions is still somewhat limited. Many existing IDS techniques focus on achieving high decision accuracy but often overlook the important aspect of explainability. This lack of transparency can decrease trust and slow down adoption, especially in use cases and applications where it is critical to understand what triggers alerts. The complexity of modern IDSs, particularly those that use DL, makes it harder to understand their decision-making processes, which

can reduce user trust. Moreover, regulations that require explainable AI emphasize the need for models that can explain their predictions. For instance, regulations such as the European Union's General Data Protection Regulation (GDPR) prioritize transparent AI systems, granting individuals the right to receive explanations for algorithmic decisions that affect them [170]. Similarly, proposed legislation like the Algorithmic Accountability Act in the United States aims to ensure fairness and transparency in automated decision-making processes, requiring companies to provide explanations for significant decisions [171].

Building on this foundation, our approach proposes an integrated solution that combines security, artificial intelligence, and transparency to strengthen the resilience of the SG and, by extension, the EI.

In this chapter, we propose a novel DL model capable of accurately classifying ten different types of attacks based on system behavior to detect potential intrusions targeting the SG system. Specifically, we use CNNs for this task. CNNs are highly effective in identifying complex patterns and features from intricate data, such as network traffic [172]. Given that DL models are often considered black boxes, we employ XAI, specifically the SHAP algorithm, to interpret the model's decisions. This use of XAI provides insights into why and how the model makes its decisions, aiming to improve transparency and reliability. By leveraging SHAP, we enhance the model's reliability through increased transparency, trust, error detection, model improvement, compliance, user feedback, and defense against adversarial attacks.

7.2 Overview of CNN and SHAP

This section presents a summary of two key components used: CNNs, a class of deep learning models widely employed for pattern recognition, and SHAP, an XAI technique that enhances interpretability by quantifying the contribution of individual features in model predictions.

7.2.1 Convolutional Neural Network (CNN)

A CNN is a class of deep neural network well-known for its performance in computer vision tasks. The structure of a CNN generally consists of three main types of layers [173]:

- Convolutional layers, which are central to distinguishing CNNs from other types of neural networks, use convolutional filters to extract key features from the dataset.

- Pooling layers, crucial for reducing the spatial dimensions of the feature maps produced by convolutional layers.
- Fully connected layers, which transform the output from the final convolutional or pooling layer into a flat vector.

7.2.2 SHapley Additive exPlanations (SHAP)

ML models often operate as black boxes, with decisions or predictions that are opaque to users, leaving them unclear about how or why specific decisions were made. In this context, XAI acts as a bridge between human users and AI systems by making the decision-making process interpretable and transparent. SHAP is a widely utilized XAI tool that interprets ML models using a technique derived from game theory [107]. The SHAP explains the models using Shapley value which represents the contribution of the feature j calculated as follows [122]:

$$\phi(j) = \sum_{S \subseteq F \setminus \{j\}} \left(\frac{|S|!(|F| - |S| - 1)!}{|F|!} \right) \cdot (v(S \cup \{j\}) - v(S)) \tag{7.1}$$

- $\phi(j)$: The Shapley value for feature j, representing the contribution of feature j to the prediction.
- $S \subseteq F \setminus \{j\}$: All possible subsets of the feature set F excluding the feature j.
- $|S|$: The number of features in the subset S.
- $|F|$: The total number of features.
- $v(S)$: The value function, which represents the prediction made by the model when only the features in the subset S are considered.
- $v(S \cup \{j\})$: The value function when the feature j is added to the subset S.

Let j denote a feature from a given data point, P represent all the features in the dataset, and S be a subset of features from P excluding feature j. $v(x)$ represents the contribution of subset x.

7.3 Proposed framework

This section presents the workflow of the proposed XDetect framework as illustrated in Figure 7.1. The main steps are: dataset description, data pre-processing, model building, and model explanation using local and global interpretation. These steps are detailed as follows:

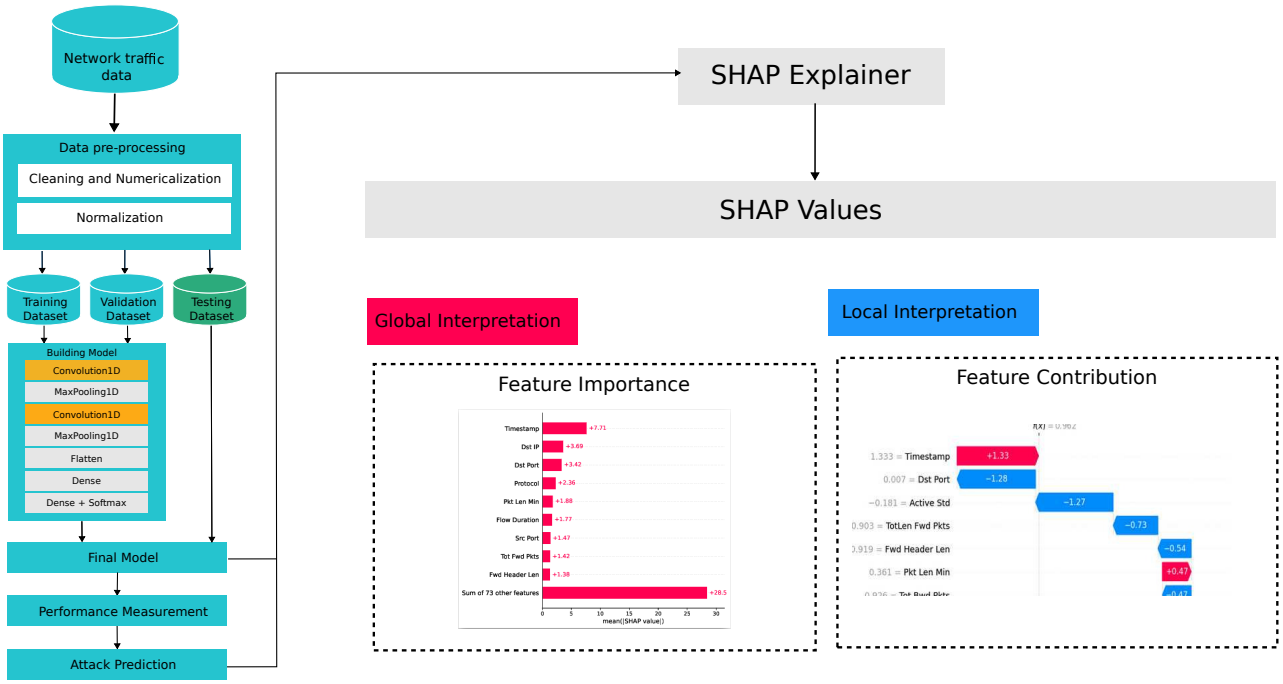


Figure 7.1: Flowchart of the XDetect framework.

7.3.1 Dataset description

The dataset used in this study was obtained from [174], specifically the Distributed Network Protocol 3 (DNP3) intrusion detection dataset[1]. This dataset includes network traffic collected from both TCP/IP and DNP3 flows. The testbed used to generate this dataset comprises eight entities acting as DNP3 outstations, such as Intelligent Electronic Devices (IEDs) and Remote Terminal Units (RTUs). Additionally, there is a workstation serving as the master station, functioning similarly to a Master Terminal Unit (MTU). Communication between the DNP3 outstations and the master station was facilitated using opendnp3 protocol. The dataset consists of 84 features and includes 10 categories of attacks, summarized in Table 7.1.

[1] https://ieee-dataport.org/documents/dnp3-intrusion-detection-dataset

Table 7.1: Attack classes of DNP3 intrusion detection dataset.

Label	**Description**
STOP_APP	Step Application Attack
DNP3_INFO	Info Attack
DISABLE_UNSOLICITED	Disable Unsolicited Message Attack
MITM_DOS	Man-in-the-middle and Denial-of-service Attack
ARP_POISONING	Address Resolution Protocol Poisoning
WARM_RESTART	Warm Restart Attack
REPLAY	Replay Attack
NORMAL	Normal traffic
DNP3_ENUMERATE	Enumerate Attack
INIT_DATA	Data Initialisation Attack
COLD_RESTART	Cold Restart Attack

7.3.2 Data pre-processing

Pre-processing the dataset is a critical step that significantly impacts the performance of the model. In our pre-processing workflow, we follow the steps below:

- **Data cleaning:** We proceed by eliminating rows containing infinite values. Furthermore, we exclude the 'Flow ID' feature from consideration, as it does not contribute to the outcome of the model.
- **Numerical representation of data:** Next, we convert the 'Timestamp' and 'IP address' columns into numerical types, as numerical representation of data is required before training the model.
- **Data normalization:** Finally, we normalize the dataset using "StandardScaler" and "LabelEncoder" functionalities from the Sklearn library. This normalization process ensures that all features are on a comparable scale, with a mean of 0 and a standard deviation of 1, which is essential for the accurate training and evaluation of the model.

7.3.3 Model Building

In our proposed CNN model, outlined in Table 7.2, we designed a multi-layer architecture specifically for our IDS. The model starts with two convolutional layers 'conv1d' and 'conv1d_1' that effectively extract key features from the input data. These are followed by two max pooling layers 'max_pooling1d' and

'max_pooling1d_1' which reduce the dimensionality of the data, enhancing processing efficiency and reducing overfitting. The data is then flattened into a 1216-size vector and processed through a dense layer 'dense' that prepares it for classification into 11 distinct categories by the final dense layer 'dense_1'. Our model employs the ReLU activation function for efficient non-linear processing and the softmax function for final output classification, ensuring a probabilistic distribution across the classes. We use the Adam optimizer with a default learning rate for optimal balance in training speed and accuracy, and the SparseCategoricalCrossentropy loss function for precise predictions. Additionally, a ModelCheckpoint is used to monitor validation accuracy, preserving the best model state to ensure robust intrusion detection performance. To train our model we split the dataset as follows: 80% for training and 20% for testing, with 20% of the training dataset allocated for validation.

Table 7.2: CNN model summary.

Layer (type)	**Output shape**	**Number of parameters**
conv1d (Conv1D)	(None, 80, 32)	128
max_pooling1d (MaxPooling1D)	(None, 40, 32)	0
conv1d_1 (Conv1D)	(None, 38, 64)	6,208
max_pooling1d_1 (MaxPooling1D)	(None, 19, 64)	0
flatten (Flatten)	(None, 1216)	0
dense (Dense)	(None, 64)	77,888
dense_1 (Dense)	(None, 11)	715
Total parameters: 254,819 (995.39 KB) **Trainable parameters:** 84,939 (331.79 KB) **Non-trainable parameters:** 0 (0.00 B) **Optimizer parameters:** 169,880 (663.60 KB)		

7.4 Performance evaluation and analysis

The proposed model was executed on a workstation equipped with an Intel Core i9-13900KF CPU, 128 GB of RAM, and Nvidia GeForce RTX 3060 12G graphics card. Additionally, the experiments were conducted on a system running Windows 11, utilizing Anaconda Jupyter Notebook, Python 3.10, and TensorFlow 2.16 environments.

7.4.1 Evaluation metrics

In assessing our model's performance, we rely on several key metrics. These metrics are based on the concepts of True Positives (TP), where the model correctly predicts positive cases; True Negatives (TN), where it correctly identifies negative cases; False Positives (FP), where it incorrectly predicts positive cases; and False Negatives (FN), where it misses positive cases [175].

Precision, expressed as $Precision = \frac{TP}{TP+FP}$, measures the accuracy of positive predictions. Recall, defined as $Recall = \frac{TP}{TP+FN}$, evaluates the model's ability to capture all relevant instances. The F1 Score, given by $F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall}$, strikes a balance between precision and recall. Finally, Accuracy, indicating the proportion of correct results among all cases examined, is calculated as $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$.

7.4.2 Model performance

The confusion matrix presented in Figure 7.2 effectively demonstrates the performance of our model in a multi-class classification task.

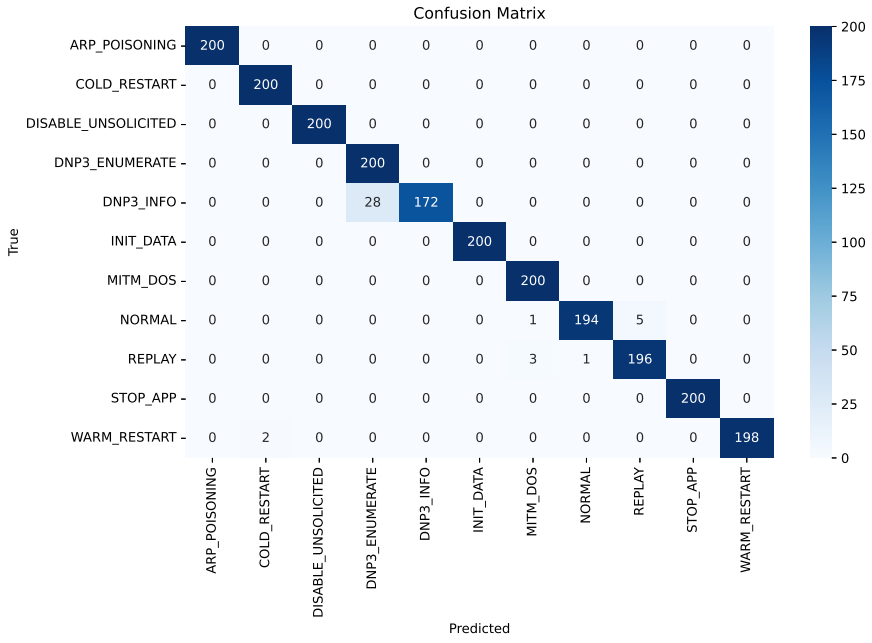


Figure 7.2: Confusion matrix of CNN model.

It shows that our model excels in identifying most classes accurately, as evidenced by the substantial values along the matrix diagonal. Nonetheless, there are noticeable misclassifications in specific classes

such as 'DNP3_INFO', indicating potential areas for model refinement. This matrix forms a part of our comprehensive evaluation, where our CNN is benchmarked against various ML and DL models, including k-nearest neighbors algorithm (KNN), decision tree (DT), support vector classifier (SVC), random forest (RF), logistic regression, and deep neural network (DNN), as detailed in Table 7.3. Our CNN distinctly outperforms the other models, achieving an accuracy, precision, recall, and F1-score of 0.988, thus providing a robust framework for comparing diverse classification strategies within our study.

The superiority of our CNN can be attributed to several factors. Firstly, its ability to automatically learn hierarchical features from raw data contributes to its effectiveness, even in tabular datasets like ours. Additionally, the use of convolutional layers enables the CNN to capture complex relationships and patterns within the input data, enhancing its discriminative power. Moreover, the inherent architecture of CNNs, which includes both convolutional and pooling layers, allows for effective feature extraction and abstraction, leading to superior performance in classification tasks.

Table 7.3: Experimental results for multi-class classification.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.973	0.973	0.973	0.973
DNN	0.970	0.970	0.970	0.970
SVC	0.831	0.843	0.831	0.825
RF	0.952	0.964	0.952	0.947
Logistic regression	0.938	0.940	0.938	0.936
DT	0.895	0.851	0.895	0.865
CNN (our)	**0.988**	**0.988**	**0.988**	**0.988**

7.4.3 Model interpretation

Model interpretation can be divided into two main approaches: global interpretation, which provides a general understanding of the model's behavior on the dataset, and local interpretation, which explains individual predictions. In this section, we highlight the importance of different features and their contribution to model decisions.

7.4.3.1 Global interpretation

The most traditional way to understand how a ML models work is to list the variable importance ranking (see Figure 7.3). In this instance, we use SHAP values to sort each predictor according to the impact it may have over the final outcome. Among all the variables we considered, Timestamp, Dst IP, Dst Port, Protocol, Pkt Len Min, Flow Duration, Src Port, Tot Fwd Pkts, and Fwd Header Len appear to be the dominating ones.

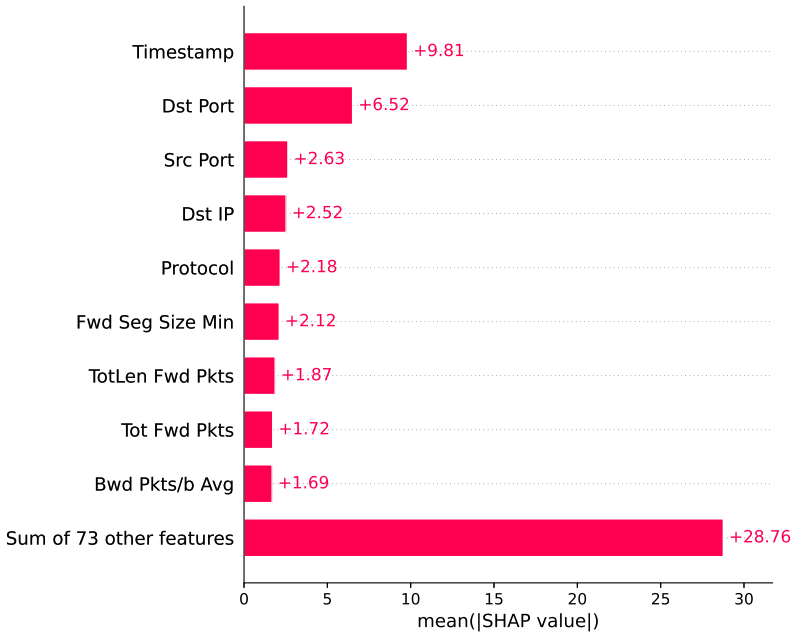


Figure 7.3: Feature importance.

The Timestamp feature has the highest mean SHAP value (+9.81), indicating that it has the most substantial average impact on increasing the model's output. This suggests that the model heavily relies on the time aspect, which might be crucial in contexts where temporal dynamics influence the outcome. Features such as Dst IP, Dst Port, and Protocol also show significant positive contributions. These network-related features suggest that the model considers both the network source/destination and the type of communication protocol as important predictors, which is typical in network traffic analysis and cybersecurity applications. Other important features like Pkt Len Min, Flow Duration, and Tot Fwd Pkts indicate that the model pays attention to the characteristics of the data packets and the session details. These features are essential for understanding the behavior of data flows across a network.

7.4.3.2 Local interpretation

As the global analysis does not take into account if the features have a positive or negative impact on the model predictions, we used the analysis of local instance-wise effects.

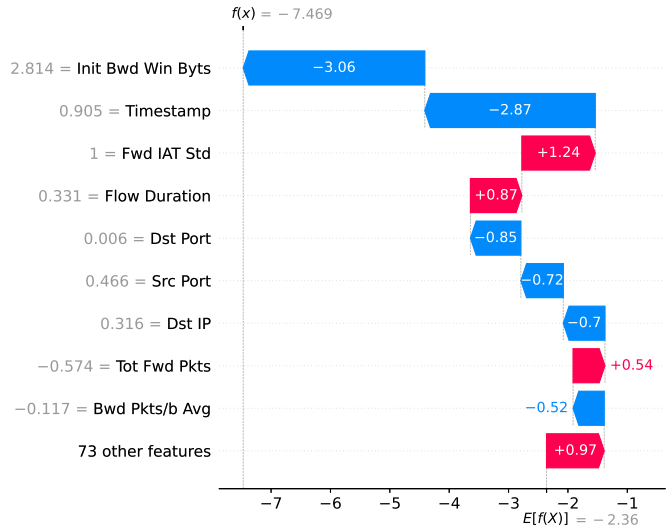


Figure 7.4: ARP Poisoning.

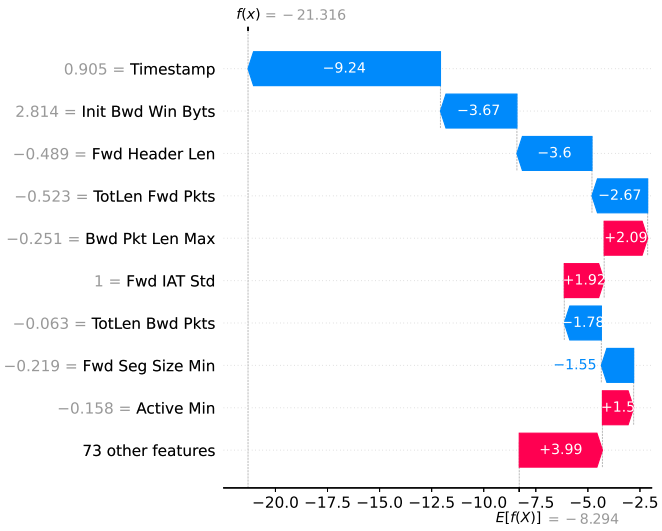


Figure 7.5: Cold Restart Attack.

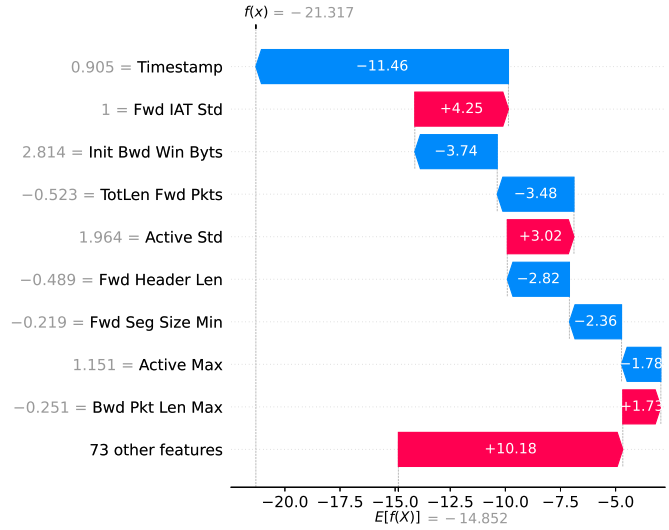


Figure 7.6: Disable Unsolicited Message Attack.

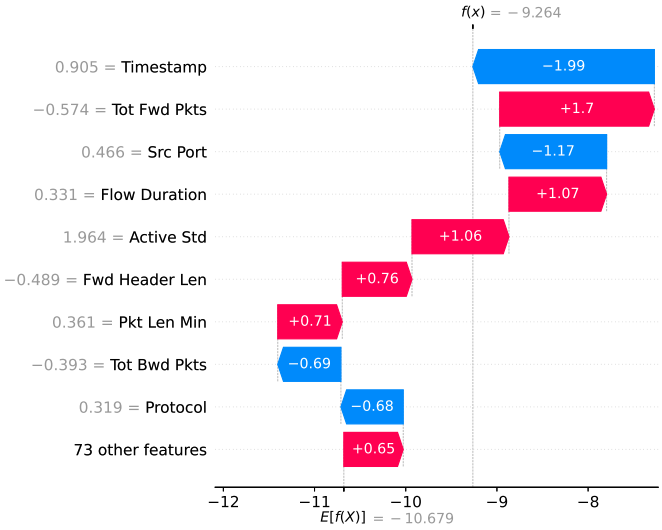


Figure 7.7: Enumerate Attack.

The SHAP plots in Figures 7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 7.10, 7.11, 7.12, 7.13, and 7.14 corresponding to the target class with a prediction score $f(x) = 17.212$, the analysis reveals how individual feature values are contributing to this particularly high likelihood of predicting a replay attack, various network traffic features distinctly influence the model's decision-making process. The 'Active Std', representing the

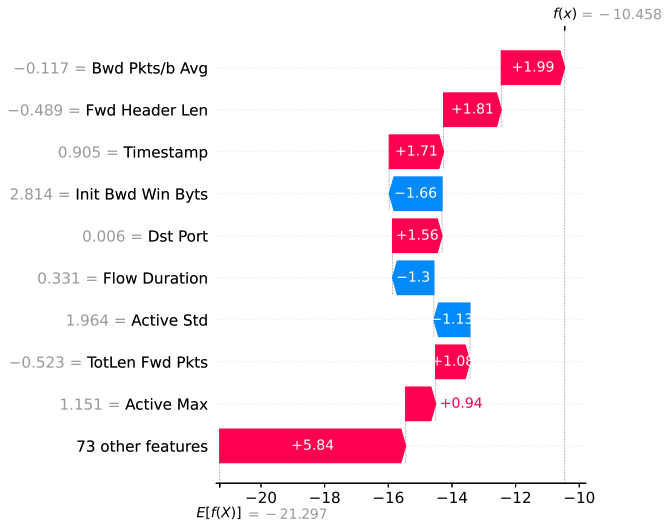


Figure 7.8: Info Attack.

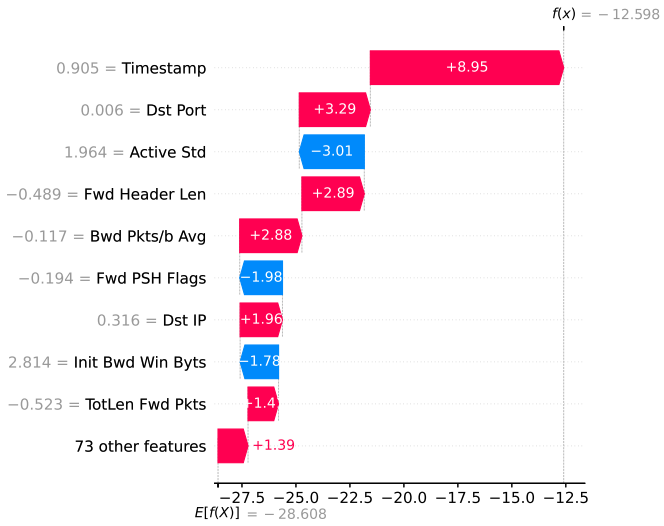


Figure 7.9: Data Initialization Attack.

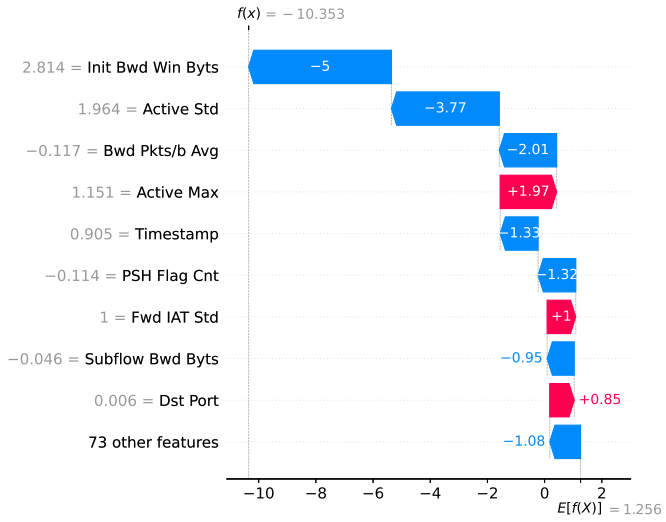


Figure 7.10: MITM DoS Attack.

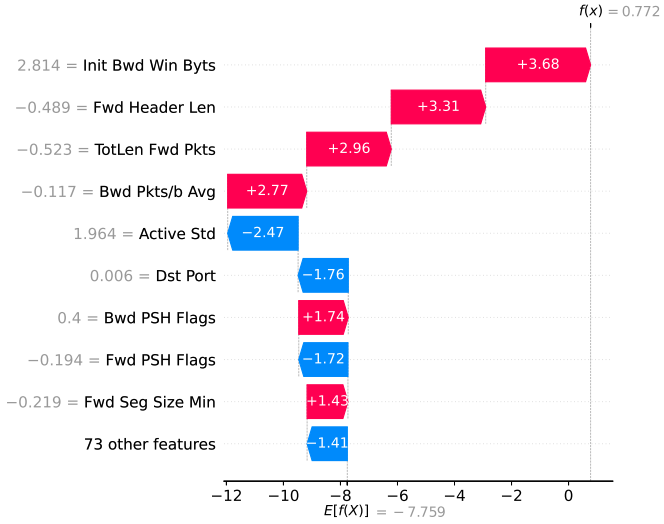


Figure 7.11: Normal traffic.

standard deviation of the duration for which a flow remained active before transitioning to an idle state, shows a significant negative impact, suggesting that high variability in activity timing is more typical of normal behavior rather than a replay attack. Conversely, 'Init Bwd Win Byts', which measures the total bytes sent in the initial window in the backward direction, along with 'Dst Port', the destination TCP/UDP port, both have substantial positive contributions, indicating their relevance in identifying replay attack patterns. Meanwhile, 'Flow Duration', the time span of the flow in microseconds, decreases the likelihood of an attack when prolonged, reflecting that shorter flows might be suspicious.

Additionally, 'Active Max', the maximum time a flow was active before turning idle, also positively im-

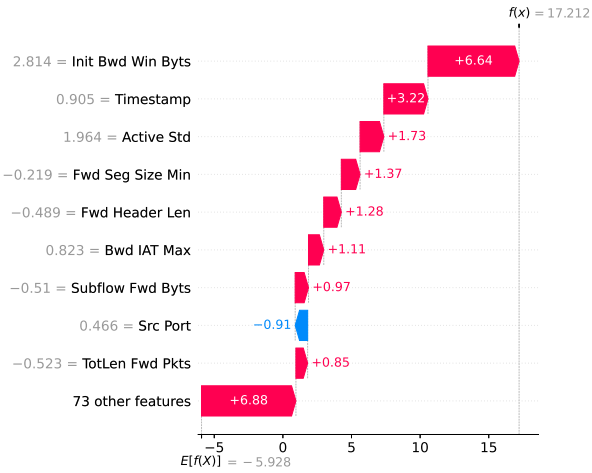


Figure 7.12: Replay Attack.

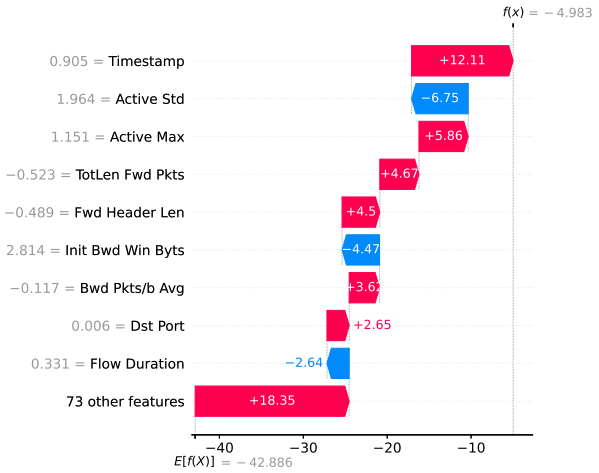


Figure 7.13: Step Application Attack.

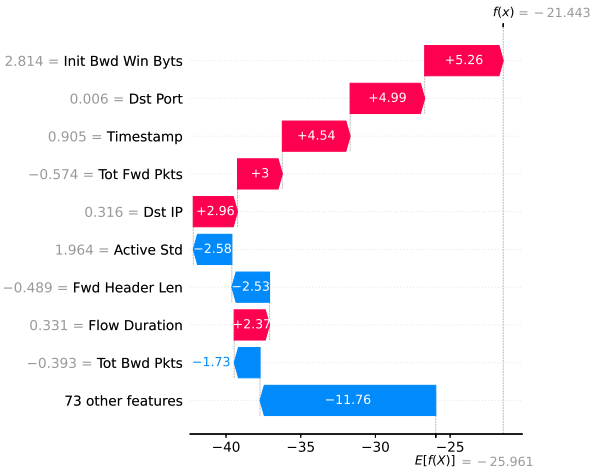


Figure 7.14: Warm Restart Attack

pacts the prediction, aligning with attack behavior where active durations are generally longer. Other influential features include the 'Dst IP', the destination IP address, and 'Pkt Len Min', the minimum packet length, each adding to the probability of an attack due to their specific values in the context of network security threats. On the other hand, 'Fwd IAT Std' and 'Bwd IAT Tot', representing the standard deviation and total time between packets sent in the forward and backward directions, respectively, show varied impacts, underscoring the complex role of timing in network traffic for security analysis. This comprehensive analysis highlights the nuanced interplay of key network features in the model's predictive framework, essential for understanding and mitigating potential security vulnerabilities.

7.4.4 Discussion

From the global interpretation, we can conclude that our model effectively leverages key features to detect network intrusions in the SG. The global interpretation shows that our model heavily relies on temporal information (Timestamp) and network-related features (Dst IP, Dst Port, Protocol) as primary indicators of potential attacks. This is typical in network traffic analysis and aligns with common cybersecurity practices.

The local interpretation further confirms the model's robustness by revealing how individual features contribute to specific predictions. For instance, in identifying replay attacks, features such as Active Std, Init Bwd Win Byts, and Dst Port play crucial roles. The model's ability to distinguish between normal and malicious behavior based on these features indicates its strong predictive power and reliability.

7.5 Conclusion

In this chapter, we introduced an advanced IDS that leverages a CNN model integrated with XAI techniques to enhance the security and interpretability of SGs. By combining deep learning with explainability, our approach improves transparency, fostering trust in automated threat detection systems. The proposed model not only demonstrated high accuracy in identifying cyber threats but also exhibited reduced training times, making it a practical and efficient solution for real-time deployment. These advancements contribute significantly to strengthening the cyber resilience of SGs, ensuring robust protection against evolving threats while maintaining system reliability and operational efficiency.

General conclusion

In this thesis, we have explored existing security solutions in EI by establishing a clear classification. We created a global map to identify key challenges, aiming to address them systematically. In this context, we proposed four contributions summarized in Figure 8.1.

In the first contribution, we integrated blockchain and traditional cryptography. While building on an existing protocol, our improved version addresses its security vulnerabilities while maintaining lightweight communication, computation, and storage costs. To validate real-world feasibility, we implemented the protocol using Hyperledger Fabric for blockchain deployment and Caliper for performance benchmarking. The results demonstrate the enhanced protocol's high performance. However, it still does not fully address all security issues.

For the second contribution, we combined PQC with QKD. QKD mitigates quantum attacks by leveraging physical laws to detect message interception during symmetric key exchange, while PQC establishes secure session keys for encryption and decryption. Though AVISPA validates its theoretical security, QKD's non-lightweight nature limits its use to key exchange. Practical deployment remains constrained by the current absence of scalable quantum infrastructure.

The third contribution merges the first two solutions. By omitting QKD (due to its complexity) and replacing traditional cryptography with PQC in blockchain, we resolved vulnerabilities from the first protocol.

The fourth contribution focuses on detecting suspicious activities using a deep learning model to filter traffic, augmented with XAI for transparent decision-making. This adds an additional security layer. However, the absence of datasets integrating cyber and physical layer data remains a limitation.

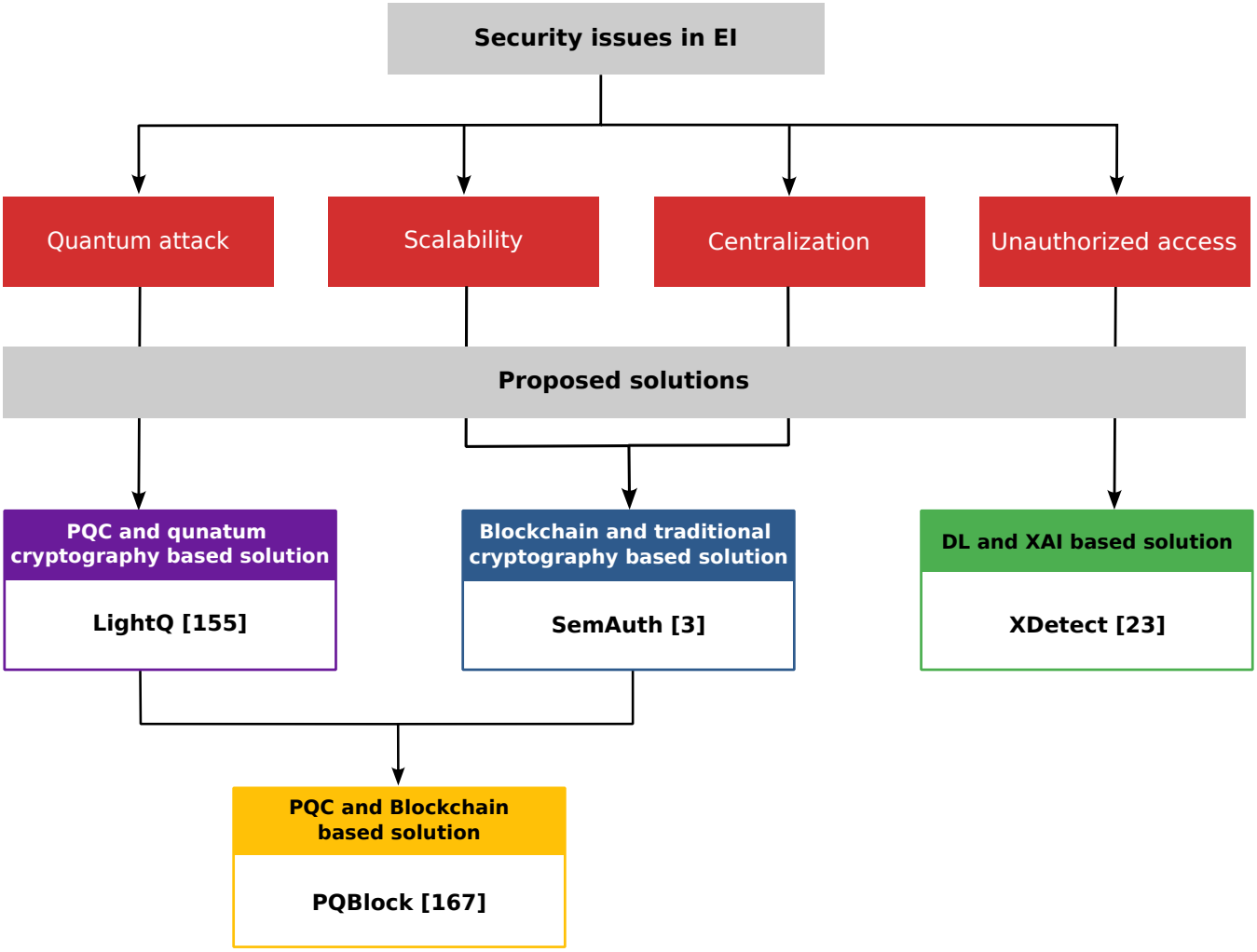


Figure 8.1: Contributions summary.

Future perspectives

While this thesis advances security solutions for EI, several avenues for future work remain. A primary limitation is the absence of datasets integrating cyber and physical layers of EI. Addressing this gap will require developing dedicated testbeds to collect real-world cyber-physical data, enabling robust training and evaluation of IDS.

Scalability also remains a critical challenge for blockchain-based solutions despite their demonstrated performance. Advanced techniques such as sharding, sidechains, or layer-2 protocols could improve efficiency for large-scale deployments by partitioning blockchain networks.

The integration of deep learning with XAI—using methods like SHAP and LIME—has enhanced IDS

transparency. Future efforts will refine these models through advanced feature selection and testing on diverse datasets, improving detection accuracy while providing granular insights into decision-making processes. This is essential for fostering trust and enabling practical deployment.

Finally, integrating these protocols with broader EI applications, such as blockchain-based energy trading systems, will strengthen infrastructure resilience. A holistic approach will ensure interoperability and resistance to both cyber and physical threats.

List of publications

Journal Articles

- **Benrebbouh, C.**, Mansouri, H., Cherbal, S., & Pathan, A. S. K. (2024). "Enhanced secure and efficient mutual authentication protocol in iot-based energy internet using blockchain, " *Peer-to-Peer Networking and Applications*, 17(1), 68-88.
- **Benrebbouh, C.**, Mansouri, H., Cherbal, S., & Pathan, A. S. K. (2023). "A lightweight security scheme to defend against quantum attack in IoT-based energy internet" *International Journal of Sensor Networks*, 43(1), 13-26.
- **[Submitted] Benrebbouh, C.**, Mansouri, H., Cherbal, S., Messai, M. L. & Pathan, A. S. K., "A Survey of Quantum and Blockchain Security Solutions for IoT-based Energy Internet. "
- **[In press]** Angague, Y., Sahraoui, H., **Benrebbouh, C.**, Mansouri, H. & Pathan, A. S. K., "A secure consensus mechanism for IoT-based energy internet using post-quantum blockchain, " *International Journal of Computational Science and Engineering.*

Conference Proceedings

- **Benrebbouh, C.**, Mansouri, H., Cherbal, S., Djahel, S., & Arrar, D. (2024, November). "An Explainable CNN-based Intrusion Detection System for Enhanced Smart Grid Security," *In 2024 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)* (pp. 1-7). IEEE.

- **Benrebbouh, C**., Mansouri, H., & Cherbal, S. (2023, October). "Enhancing Security and Authentication in IoT-based Energy Internet using Post-Quantum Blockchain," *In 2023 5th International Conference on Pattern Analysis and Intelligent Systems (PAIS)* (pp. 1-8). IEEE.

- **Benrebbouh, C.**, Cherbal, S., Mansouri, H., & Pathan, A. S. K. (2022, September). "Future security issues in internet of energy, " *In 2022 4th International Conference on Advanced Science and Engineering (ICOASE)* (pp. 107-112). IEEE.

Bibliography

[1] Yan Li, Guiwen Wang, Long Yang, Yuting Deng, Beibei Shi, Nan Li, Rong Kang, Yating Yang, and Tingting Yang. Can the energy internet achieve carbon reduction? *Frontiers in Energy Research*, 12:1341542, 2024.

[2] Abubakar Sadiq Sani, Dong Yuan, Jiong Jin, Longxiang Gao, Shui Yu, and Zhao Yang Dong. Cyber security framework for internet of things-based energy internet. *Future Generation Computer Systems*, 93:849–859, 2019.

[3] Chahrazed Benrebbouh, Houssem Mansouri, Sarra Cherbal, and Al-Sakib Khan Pathan. Enhanced secure and efficient mutual authentication protocol in iot-based energy internet using blockchain. *Peer-to-Peer Networking and Applications*, 17(1):68–88, 2024.

[4] Chahrazed Benrebbouh, Sarra Cherbal, Houssem Mansouri, and Al-Sakib Khan Pathan. Future security issues in internet of energy. In *2022 4th International Conference on Advanced Science and Engineering (ICOASE)*, pages 107–112. IEEE, 2022.

[5] Mohammad Ghiasi, Moslem Dehghani, Taher Niknam, and Abdollah Kavousi-Fard. Investigating overall structure of cyber-attacks on smart-grid control systems to improve cyber resilience in power system. *Network*, 1(1):1–6, 2020.

[6] Sanaz Amanlou, Mohammad Kamrul Hasan, Umi Asma Mokhtar, Khalid Mahmood Malik, Shayla Islam, Sheroz Khan, Muhammad Attique Khan, and Muhammad Asghar Khan. Cybersecurity challenges in smart grid systems: Current and emerging attacks, opportunities, and recommendations. *IEEE Open Journal of the Communications Society*, 2025.

[7] Md Habib Ullah, Rozhin Eskandarpour, Honghao Zheng, and Amin Khodaei. Quantum computing for smart grid applications. *IET Generation, Transmission & Distribution*, 16(21):4239–4257, 2022.

[8] Sarmadullah Khan, Rafiullah Khan, and Ali Hilal Al-Bayatti. Secure communication architecture for dynamic energy management in smart grid. *IEEE Power and Energy Technology Systems Journal*, 6(1):47–58, 2019.

[9] Priyanka Mishra and Ghanshyam Singh. Energy management systems in sustainable smart cities based on the internet of energy: A technical review. *Energies*, 16(19):6903, 2023.

[10] Hamza El Hafdaoui and Ahmed Khallaayoun. Internet of energy (ioe) adoption for a secure semi-decentralized renewable energy distribution. *Sustainable Energy Technologies and Assessments*, 57:103307, 2023.

[11] Mir Hamid Taghavi, Peyman Akhavan, Rouhollah Ahmadi, and Ali Bonyadi Naeini. Identifying key components in implementation of internet of energy (ioe) in iran with a combined approach of meta-synthesis and structural analysis: A systematic review. *Sustainability*, 14(20):13180, 2022.

[12] Bassam Zafar and Sami Ben Slama. Energy internet opportunities in distributed peer-to-peer energy trading reveal by blockchain for future smart grid 2.0. *Sensors*, 22(21):8397, 2022.

[13] Charithri Yapa, Chamitha De Alwis, Uditha Wijewardhana, and Madhusanka Liyanage. Utilization of a blockchain-based reputation management system for energy trading in smart grid 2.0. In *2023 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6. IEEE, 2023.

[14] Charithri Yapa, Chamitha de Alwis, and Madhusanka Liyanage. Can blockchain strengthen the energy internet? *Network*, 1(2):95–115, 2021.

[15] K Parvin, MA Hannan, Looe Hui Mun, MS Hossain Lipu, Maher GM Abdolrasol, Pin Jern Ker, Kashem M Muttaqi, and ZY Dong. The future energy internet for utility energy service and demand-side management in smart grid: Current practices, challenges and future directions. *Sustainable Energy Technologies and Assessments*, 53:102648, 2022.

[16] Akhil Joseph and Patil Balachandra. Smart grid to energy internet: A systematic review of transitioning electricity systems. *IEEE Access*, 8:215787–215805, 2020.

[17] Jeremy Rifkin, M Carvalho, A Consoli, and M Bonifacio. Leading the way to the third industrial revolution. *European Energ Rev*, 1:4–18, 2008. Special edition.

[18] Jeremy Rifkin. *The Third Industrial Revolution: How Lateral Power Is Transforming Energy, the Economy, and the World*. Palgrave Macmillan, New York, first edition, 2011. ISBN: 978-0-230-11521-7.

[19] Lefteri H Tsoukalas and Rong Gao. Inventing energy internet the role of anticipation in human-centered energy distribution and utilization. In *2008 SICE Annual Conference*, pages 399–403. IEEE, 2008.

[20] Lefteri H Tsoukalas and Rong Gao. From smart grids to an energy internet: Assumptions, architectures and requirements. In *2008 Third international conference on electric utility deregulation and restructuring and power technologies*, pages 94–98. IEEE, 2008.

[21] Ilias Laroussi, Liu Huan, and Zhao Xiusheng. How will the internet of energy (ioe) revolutionize the electricity sector? a techno-economic review. *Materials Today: Proceedings*, 72:3297–3311, 2023.

[22] Ersan Kabalci and Yasin Kabalci. Introduction to smart grid architecture. In Ersan Kabalci and Yasin Kabalci, editors, *Smart Grids and Their Communication Systems*, chapter 1, pages 3–45. Springer, Singapore, 2019.

[23] Chahrazed Benrebbouh, Houssem Mansouri, Sarra Cherbal, Soufiene Djahel, and Djihad Arrar. An explainable cnn-based intrusion detection system for enhanced smart grid security. In *2024 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pages 1–7. IEEE, 2024.

[24] Ines Adjeroud, Sarra Cherbal, Chahrazed Benrebbouh, and Hamza Baaraoui. Authentication scheme based on blockchain and proof-of-work for iot. In *2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pages 1–8. IEEE, 2024.

[25] I Sami, Z Ullah, K Salman, I Hussain, SM Ali, B Khan, CA Mehmood, and U Farid. A bidirectional interactive electric vehicles operation modes: Vehicle-to-grid (v2g) and grid-to-vehicle (g2v) variations within smart grid. In *2019 international conference on engineering and emerging technologies (ICEET)*, pages 1–6. IEEE, 2019.

[26] Sayed Saeed Hosseini, Ali Badri, and Masood Parvania. The plug-in electric vehicles for power system applications: The vehicle to grid (v2g) concept. In *2012 IEEE International Energy Conference and Exhibition (ENERGYCON)*, pages 1101–1106. IEEE, 2012.

[27] Muqit Farhan, Tanzim N Reza, Faisal R Badal, Md R Islam, SM Muyeen, Z Tasneem, Md Mehedi Hasan, Md F Ali, Md H Ahamed, SH Abhi, et al. Towards next generation internet of energy system: Framework and trends. *Energy and AI*, 14:100306, 2023.

[28] GJRE Dileep. A survey on smart grid technologies and applications. *Renewable energy*, 146:2589–2625, 2020.

[29] R Sujeetha, Himanshu Das, Tanish Dhelawat, and Mohammad Tanveer. Cyber-space and its menaces. In *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–5. IEEE, 2019.

[30] Muhammed Zekeriya Gunduz and Resul Das. Cyber-security on smart grid: Threats and potential solutions. *Computer networks*, 169:107094, 2020.

[31] Daniele Antonioli, Nils Ole Tippenhauer, and Kasper Rasmussen. Bias: Bluetooth impersonation attacks. In *2020 IEEE symposium on security and privacy (SP)*, pages 549–562. IEEE, 2020.

[32] Carlisle Adams. Replay attack. In Sushil Jajodia, Pierangela Samarati, and Moti Yung, editors, *Encyclopedia of Cryptography, Security and Privacy*, pages 2097–2097. Springer Nature Switzerland, Cham, 2025.

[33] Yue Zhang, Jian Weng, Rajib Dey, and Xinwen Fu. Bluetooth low energy (ble) security and privacy. In Xuemin (Sherman) Shen, Xiaodong Lin, and Kuan Zhang, editors, *Encyclopedia of Wireless Networks*, pages 1–12. Springer, Cham, 2019.

[34] Rick Hofstede, Mattijs Jonker, Anna Sperotto, and Aiko Pras. Flow-based web application brute-force attack and compromise detection. *Journal of network and systems management*, 25:735–758, 2017.

[35] Mingfang Li and Zheng Dou. Active eavesdropping detection: a novel physical layer security in wireless iot. *EURASIP Journal on Advances in Signal Processing*, 2023(1):119, 2023.

[36] Minghan Chen, Fangyan Dai, Bingjie Yan, and Jieren Cheng. Encryption algorithm for tcp session hijacking. In *International Conference on Artificial Intelligence and Security*, pages 191–202. Springer, 2020.

[37] Chien-Ding Lee and Tzung-Her Chen. New secure and practical e-mail protocol with perfect forward secrecy. *Symmetry*, 13(7):1144, 2021.

[38] Shuishuai Xu, Xindong Liu, Mimi Ma, and Jianhua Chen. An improved mutual authentication protocol based on perfect forward secrecy for satellite communications. *International Journal of Satellite Communications and Networking*, 38(1):62–73, 2020.

[39] Mohamed Taoufiq Damir, Tommi Meskanen, Sara Ramezanian, and Valtteri Niemi. A beyond-5g authentication and key agreement protocol. In *International Conference on Network and System Security*, pages 249–264. Springer, 2022.

[40] Jaegeun Moon, Im Y Jung, and Jong Hyuk Park. Iot application protection against power analysis attack. *Computers & Electrical Engineering*, 67:566–578, 2018.

[41] Yu-Long Gao, Xiu-Bo Chen, Yu-Ling Chen, Ying Sun, Xin-Xin Niu, and Yi-Xian Yang. A secure cryptocurrency scheme based on post-quantum blockchain. *IEEE Access*, 6:27205–27213, 2018.

[42] Nimish Mishra, SK Hafizul Islam, and Sherali Zeadally. A survey on security and cryptographic perspective of industrial-internet-of-things. *Internet of Things*, 25:101037, 2024.

[43] M Kokila and Srinivasa Reddy. Authentication, access control and scalability models in internet of things security-a review. *Cyber Security and Applications*, 2024. Early access: https://doi.org/10.1016/j.csa.2024.100057.

[44] Muhammad Ajmal Azad, Sidrah Abdullah, Junaid Arshad, Harjinder Lallie, and Yussuf Hassan Ahmed. Verify and trust: A multidimensional survey of zero-trust security in the age of iot. *Internet of Things*, 27:101227, 2024.

[45] Sarra Cherbal and Rania Benchetioui. Scpuak: Smart card-based secure protocol for remote user authentication and key agreement. *Computers and Electrical Engineering*, 109:108759, 2023.

[46] Yasmine Harbi, Zibouda Aliouat, Saad Harous, Abdelhak Bentaleb, and Allaoua Refoufi. A review of security in internet of things. *Wireless Personal Communications*, 108:325–344, 2019.

[47] Abidemi Emmanuel Adeniyi, Rasheed Gbenga Jimoh, and Joseph Bamidele Awotunde. A systematic review on elliptic curve cryptography algorithm for internet of things: Categorization, application areas, and security. *Computers and Electrical Engineering*, 118:109330, 2024.

[48] Wenlong Zhu, Xuexiao Chen, and Linmei Jiang. A secure and efficient authentication key agreement scheme for industrial internet of things based on edge computing. *Alexandria Engineering Journal*, 101:52–61, 2024.

[49] Shtwai Alsubai, Abdullah Alqahtani, Harish Garg, Mohemmed Sha, and Abdu Gumaei. A blockchain-based hybrid encryption technique with anti-quantum signature for securing electronic health records. *Complex & Intelligent Systems*, 2024. Early access: https://doi.org/10.1007/s40747-024-01477-1.

[50] Tianchen Ma. White-box schnorr signature for internet of things security. In *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pages 1939–1942. IEEE, 2020.

[51] Yuqing Xu, Guangxia Xu, Yong Liu, Yuan Liu, and Ming Shen. A survey of the fusion of traditional data security technology and blockchain. *Expert Systems with Applications*, 2024. Early access: https://doi.org/10.1007/s40747-024-01477-1.

[52] Dingyi Shui, Yong Xie, Libing Wu, Yining Liu, and Xing Su. Lightweight three-party key agreement for v2g networks with physical unclonable function. *Vehicular Communications*, 47:100747, 2024.

[53] Girraj Kumar Verma, BB Singh, and Harendra Singh. Provably secure certificate-based proxy blind signature scheme from pairings. *Information Sciences*, 468:1–13, 2018.

[54] Girraj Kumar Verma, Prosanta Gope, and Neeraj Kumar. Pf-da: Pairing free and secure data aggregation for energy internet-based smart meter-to-grid communication. *IEEE Transactions on Smart Grid*, 13(3):2294–2304, 2021.

[55] Akber Ali Khan, Vinod Kumar, Musheer Ahmad, Brij B Gupta, Musheer Ahmad, and Ahmed A Abd El-Latif. A secure and efficient key agreement framework for critical energy infrastructure using mobile device. *Telecommunication Systems*, 78:539–557, 2021.

[56] Akber Ali Khan, Vinod Kumar, Musheer Ahmad, and Srinivas Jangirala. A secure and energy efficient key agreement framework for vehicle-grid system. *Journal of Information Security and Applications*, 68:103231, 2022.

[57] Masoumeh Safkhani, Saru Kumari, Mohammad Shojafar, and Sachin Kumar. An authentication and key agreement scheme for smart grid. *Peer-to-Peer Networking and Applications*, 15(3):1595–1616, 2022.

[58] Hanyu Rao, Qianqian Ma, Dong Mao, Chen Zhang, and Zhongyuan Qin. Multi-dimensional user data security aggregation in energy internet. In *2022 IEEE 10th International Conference on Information, Communication and Networks (ICICN)*, pages 217–222. IEEE, 2022.

[59] Dong Mao, Junlun Wu, Hanyu Rao, Chen Zhang, Shan Chen, and Zhongyuan Qin. An identity based key update scheme for energy internet edge devices. In *2022 IEEE 10th International Conference on Information, Communication and Networks (ICICN)*, pages 236–241. IEEE, 2022.

[60] Samiulla Itoo, Lalit Kumar Som, Musheer Ahmad, Ram Baksh, and Faheem Syeed Masoodi. A robust ecc-based authentication framework for energy internet (ei)-based vehicle to grid communication system. *Vehicular Communications*, 41:100612, 2023.

[61] Akber Ali Khan, Vinod Kumar, Musheer Ahmad, Saurabh Rana, and Dheerendra Mishra. Palk: Password-based anonymous lightweight key agreement framework for smart grid. *International Journal of Electrical Power & Energy Systems*, 121:106121, 2020.

[62] Jay Gambetta. Ibm's roadmap for scaling quantum technology. https://www.ibm.com/quantum/blog/ibm-quantum-roadmap, 2020. Accessed: 18-03-2024.

[63] Hadi Gharavi, Jorge Granjal, and Edmundo Monteiro. Post-quantum blockchain security for the internet of things: Survey and research directions. *IEEE Communications Surveys & Tutorials*, 26(3):1748–1774, 2024.

[64] Ezhil E Nithila, A Rosi, et al. A survey about post quantum cryptography methods. *EAI Endorsed Transactions on Internet of Things*, 10:e5, 2024.

[65] Xiaoyun Wang, Guangwu Xu, and Yang Yu. Lattice-based cryptography: A survey. *Chinese Annals of Mathematics, Series B*, 44(6):945–960, 2023.

[66] NIST website. Post-quantum cryptography. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022, 2024. Accessed: 19-03-2024.

[67] Hee-Yong Kwon, Indra Bajuna, and Mun-Kyu Lee. Compact hybrid signature for secure transition to post-quantum era. *IEEE Access*, 12:39417–39429, 2024.

[68] Dequan Gao, Yaofu Cao, Ziyan Zhao, Pengcheng Ni, Hao Qin, and Zhiyuan Ye. Test analysis of practical quantum vpn gateway for electric power telecommunication security in energy internet. In *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pages 1–6. IEEE, 2017.

[69] Gengtao Jia, Weidong Ni, and Jiawei Wu. Research and applications of key technologies of quantum secure communication in energy internet. In *2019 4th International Conference on Intelligent Green Building and Smart Grid (IGBSG)*, pages 54–60. IEEE, 2019.

[70] Vinícius Lagrota Rodrigues da Costa, Julio López, and Moisés Vidal Ribeiro. A system-on-a-chip implementation of a post-quantum cryptography scheme for smart meter data communications. *Sensors*, 22(19):7214, 2022.

[71] Kumar Prateek, Meghashrita Das, Sairaaj Surve, Soumyadev Maity, and Ruhul Amin. Q-secure-p^2-sma: Quantum-secure privacy-preserving smart meter authentication for unbreakable security in smart grid. *IEEE Transactions on Network and Service Management*, 21(5):5149–5163, 2024.

[72] Peng Wang, Tao Xiang, Xiaoguo Li, and Hong Xiang. Access control encryption without sanitizers for internet of energy. *Information Sciences*, 546:924–942, 2021.

[73] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf, 2008. Accessed: 15-01-2024.

[74] Charithri Yapa, Chamitha De Alwis, Madhusanka Liyanage, and Janaka Ekanayake. Survey on blockchain for future smart grids: Technical aspects, applications, integration challenges and future research. *Energy Reports*, 7:6530–6564, 2021.

[75] Arun Sekar Rajasekaran, Maria Azees, and Fadi Al-Turjman. A comprehensive survey on blockchain technology. *Sustainable Energy Technologies and Assessments*, 52:102039, 2022.

[76] Oluwafemi Akanfe, Diane Lawong, and H Raghav Rao. Blockchain technology and privacy regulation: Reviewing frictions and synthesizing opportunities. *International Journal of Information Management*, 76:102753, 2024.

[77] Nils M Denter, Fabian Seeger, and Martin G Moehrle. How can blockchain technology support patent management? a systematic literature review. *International Journal of Information Management*, 68:102506, 2023.

[78] Guocheng Zhu, Debiao He, Haoyang An, Min Luo, and Cong Peng. The governance technology for blockchain systems: a survey. *Frontiers of Computer Science*, 18(2):182813, 2024.

[79] Sami Bettayeb, Mohamed-Lamine Messai, and Sofiane Mounine Hemam. A smart contract-based blockchain solution for key revocation in iot networks. Research Square, 2024. Preprint: https://doi.org/10.21203/rs.3.rs-3861364/v1.

[80] Roy Lai and David LEE Kuo Chuen. Blockchain – from public to private. In David LEE Kuo Chuen and Robert Deng, editors, *Handbook of Blockchain, Digital Finance, and Inclusion*, chapter 7, pages 145–177. Academic Press, 2018.

[81] Hamed Al-Shaibani, Noureddine Lasla, and Mohamed Abdallah. Consortium blockchain-based decentralized stock exchange platform. *IEEE Access*, 8:123711–123725, 2020.

[82] Jiarui Zhang. A multi-transaction mode consortium blockchain. *International Journal of Performability Engineering*, 14(4):765, 2018.

[83] Ziad Hussein, May A Salama, and Sahar A El-Rahman. Evolution of blockchain consensus algorithms: a review on the latest milestones of blockchain consensus algorithms. *Cybersecurity*, 6(1):30, 2023.

[84] Kesara Wimal and Geethapriya Liyanage. Towards true decentralization: Development, testing and evaluation of a novel blockchain consensus protocol. In *International Conference on Asia Pacific Advanced Network*, pages 86–99. Springer, 2023.

[85] Tengfei Xue, Yuyu Yuan, Zahir Ahmed, Krishna Moniz, Ganyuan Cao, and Cong Wang. Proof of contribution: A modification of proof of work to increase mining efficiency. In *2018 IEEE 42nd annual computer software and applications conference (COMPSAC)*, volume 1, pages 636–644. IEEE, 2018.

[86] Fahad Saleh. Blockchain without waste: Proof-of-stake. *The Review of financial studies*, 34(3):1156–1190, 2021.

[87] Xinxin Fan and Qi Chai. Roll-dpos: a randomized delegated proof of stake scheme for scalable blockchain-based internet of things systems. In *Proceedings of the 15th EAI international conference on mobile and ubiquitous systems: computing, networking and services*, pages 482–484. ACM, 2018.

[88] Kausthav Pratim Kalita, Jerry Casper Kharbhih, Debojit Boro, and Dhruba Kumar Bhattacharyya. An enhanced blockchain consensus mechanism using proof-of-work and proof-of-stake. In *International Conference on Emerging Global Trends in Engineering and Technology*, pages 501–511. Springer, 2022.

[89] Cheng You, Yanjia Qin, Qi Chen, Chang Chen, and Jiahui Huang. Hadpos: Improvement of dpos consensus mechanism based on heat attenuation. *IT Professional*, 25(1):40–51, 2023.

[90] Kostis Karantias, Aggelos Kiayias, and Dionysis Zindros. Proof-of-burn. Cryptology ePrint Archive, 2019. Preprint: https://eprint.iacr.org/2019/1096.

[91] Wai Yan Maung Maung Thin, Naipeng Dong, Guangdong Bai, and Jin Song Dong. Formal analysis of a proof-of-stake blockchain. In *2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS)*, pages 197–200. IEEE, 2018.

[92] Sarwar Sayeed and Hector Marco-Gisbert. Proof of adjourn (poaj): A novel approach to mitigate blockchain attacks. *Applied Sciences*, 10(18):6607, 2020.

[93] Xiaolian Chen, Xiao Hu, Yang Li, Xue Gao, and Dawei Li. A blockchain based access authentication scheme of energy internet. In *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pages 1–9. IEEE, 2018.

[94] Xin Lu, Lingyun Shi, Zhenyu Chen, Xunfeng Fan, Zhitao Guan, Xiaojiang Du, and Mohsen Guizani. Blockchain-based distributed energy trading in energy internet: An sdn approach. *IEEE access*, 7:173817–173826, 2019.

[95] Xin Lu and Zhitao Guan. A blockchain-based trading matching scheme in energy internet. In *Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure*, pages 142–150, 2020.

[96] Shenghong Ding, Jun Zeng, Zongkang Hu, and Yang Yang. A peer-2-peer management and secure policy of the energy internet in smart microgrids. *IEEE Transactions on Industrial Informatics*, 18(8):5689–5697, 2021.

[97] Haiyan Wang, Wei Wang, Liyang Liu, Chuan Long, Jun Wei, and Tinghu Zhu. Electricity decentralized transaction framework of community energy internet cluster based on blockchain. In *2021 International Conference on Intelligent Technology and Embedded Systems (ICITES)*, pages 164–170. IEEE, 2021.

[98] Guanlin Si, Yue Zhang, Yue Sun, and Wei Chen. Blockchain-based privacy protection scheme for smart park multi-energy fusion system. In *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, volume 4, pages 1011–1016. IEEE, 2021.

[99] Khawla Hassan, Fatima Dakalbab, Manar Abu Talib, Chaouki Ghenai, Qassim Nasir, and Maamar Bettayeb. Blockchain networks for solar pv electric vehicles charging station to support and foster clean energy transition. In *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*, pages 1–7. IEEE, 2022.

[100] Donglan Liu, Xin Liu, Rui Wang, Hao Zhang, Fangzhe Zhang, Lili Sun, Honglei Yao, and Hao Yu. A

multi-blockchain-based cross-domain authentication and authorization scheme for energy internet. *Wireless Communications and Mobile Computing*, 2023(1):4778967, 2023.

[101] Appasani Prem Sai and Aashish Kumar Bohre. Energy trading in the internet of energy using ethereum smart contracts and smart energy meters. In *2023 3rd International Conference on Intelligent Technologies (CONIT)*, pages 1–6. IEEE, 2023.

[102] Fatima Hussain, Rasheed Hussain, Syed Ali Hassan, and Ekram Hossain. Machine learning in iot security: Current solutions and future challenges. *IEEE Communications Surveys & Tutorials*, 22(3):1686–1721, 2020.

[103] Erdal Ozdogan. A comprehensive analysis of the machine learning algorithms in iot ids systems. *IEEE Access*, 12:46785–46811, 2024.

[104] Asmaa Halbouni, Teddy Surya Gunawan, Mohamed Hadi Habaebi, Murad Halbouni, Mira Kartiwi, and Robiah Ahmad. Machine learning and deep learning approaches for cybersecurity: A review. *IEEE Access*, 10:19572–19585, 2022.

[105] Adarsh Prasad Behera, Satya Prakash, Siddhant Khanna, Shivangi Nigam, and Shekhar Verma. Cnn-based metrics for performance evaluation of generative adversarial networks. *IEEE Transactions on Artificial Intelligence*, 5(10):5040–5049, 2024.

[106] Abdulrahman Takiddin, Muhammad Ismail, and Erchin Serpedin. Robust data-driven detection of electricity theft adversarial evasion attacks in smart grids. *IEEE Transactions on Smart Grid*, 14(1):663–676, 2022.

[107] Huanjing Wang, Qianxin Liang, John T Hancock, and Taghi M Khoshgoftaar. Feature selection strategies: a comparative analysis of shap-value and importance-based methods. *Journal of Big Data*, 11(1):44, 2024.

[108] Noshina Tariq, Amjad Alsirhani, Mamoona Humayun, Faeiz Alserhani, and Momina Shaheen. A fog-edge-enabled intrusion detection system for smart grids. *Journal of Cloud Computing*, 13(1):43, 2024.

[109] Edosa Osa, Patience E Orukpe, and Usiholo Iruansi. Design and implementation of a deep neural network approach for intrusion detection systems. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, 7:100434, 2024.

[110] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Cicids2017: An intrusion detection evaluation dataset. https://www.unb.ca/cic/datasets/ids-2017.html, 2017.

[111] Amina Khacha, Rafika Saadouni, Yasmine Harbi, Chirihane Gherbi, Saad Harous, and Zibouda Aliouat. Robust intrusion detection for iot networks: an integrated cnn-lstm-gru approach. In *2023 International Conference on Networking and Advanced Systems (ICNAS)*, pages 1–6. IEEE, 2023.

[112] Rafika Saadouni, Amina Khacha, Yasmine Harbi, Chirihane Gherbi, Saad Harous, and Zibouda Aliouat. Secure iiot networks with hybrid cnn-gru model using edge-iiotset. In *2023 15th International Conference on Innovations in Information Technology (IIT)*, pages 150–155. IEEE, 2023.

[113] Kashif Mahmood, Jinshu Hu, and Guojun Wang. Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot. https://ieee-dataport.org/documents/edge-iiotset-new-comprehensive-realistic-cyber-security-dataset-iot-and-iiot, 2022.

[114] Hakan Can Altunay and Zafer Albayrak. A hybrid cnn+ lstm-based intrusion detection system for industrial iot networks. *Engineering Science and Technology, an International Journal*, 38:101322, 2023.

[115] Nour Moustafa and Jill Slay. Unsw-nb15: A comprehensive data set for network intrusion detection systems. https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/, 2015.

[116] Murat Altunay and Gunes Kurt. X-iiotid: A hybrid dataset for industrial iot intrusion detection. https://github.com/altunaymurat/X-IIoTID, 2023.

[117] Thi-Thu-Huong Le, Haeyoung Kim, Hyoeun Kang, and Howon Kim. Classification and explanation for intrusion detection system based on ensemble trees and shap method. *Sensors*, 22(3):1154, 2022.

[118] Mohanad Sarhan, Siamak Layeghy, Nour Moustafa, and Marius Portmann. Nf-bot-iot-v2 dataset. https://www.cyber.uq.edu.au/project/machine-learning-based-nids-datasets, 2021.

[119] Mohanad Sarhan, Siamak Layeghy, Nour Moustafa, and Marius Portmann. Nf-ton-iot-v2 dataset. https://www.cyber.uq.edu.au/project/machine-learning-based-nids-datasets, 2021.

[120] Ha Le, Duc Pham, Dipankar Sahoo, and Steven C. H. Hoi. Iotds20: A dataset for iot intrusion detection. https://github.com/dhphust/IOTDS20, 2020.

[121] Bhawana Sharma, Lokesh Sharma, Chhagan Lal, and Satyabrata Roy. Anomaly based network intrusion detection for iot attacks using deep learning technique. *Computers and Electrical Engineering*, 107:108626, 2023.

[122] Bhawana Sharma, Lokesh Sharma, Chhagan Lal, and Satyabrata Roy. Explainable artificial intelligence for intrusion detection in iot networks: A deep learning based approach. *Expert Systems with Applications*, 238:121751, 2024.

[123] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. Nsl-kdd dataset. https://www.unb.ca/cic/datasets/nsl.html, 2009.

[124] Remah Younisse, Ashraf Ahmad, and Qasem Abu Al-Haija. Explaining intrusion detection-based convolutional neural networks using shapley additive explanations (shap). *Big Data and Cognitive Computing*, 6(4):126, 2022.

[125] Salvatore J. Stolfo, Wenke Lee Fan, Andreas Prodromidis, and Philip K. Chan. Kdd cup 1999 dataset. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999.

[126] Zhitao Guan, Xin Lu, Naiyu Wang, Jun Wu, Xiaojiang Du, and Mohsen Guizani. Towards secure and efficient energy trading in iiot-enabled energy internet: A blockchain approach. *Future Generation Computer Systems*, 110:686–695, 2020.

[127] Rahul Saha, Gulshan Kumar, G Geetha, Mamoun Alazab, Reji Thomas, Mritunjay Kumar Rai, Joel JPC Rodrigues, et al. The blockchain solution for the security of internet of energy and electric vehicle interface. *IEEE Transactions on Vehicular Technology*, 70(8):7495–7508, 2021.

[128] Qiaolian Zhang, Fenhua Bai, Zhuo Yu, Yingli Liu, Tao Shen, Anke Xie, and Lin Huang. Editable and verifiable anonymous authentication incorporating blockchain in the internet of energy. *Electronics*, 11(13):1992, 2022.

[129] Houpeng Hu, Jiaxiang Ou, Bin Qian, Yi Luo, Peilin He, Mi Zhou, and Zerui Chen. A practical anonymous voting scheme based on blockchain for internet of energy. *Security and Communication Networks*, 2022(1):4436824, 2022.

[130] Yanchi Chen, Haoxiang Luo, Qi Huang, and Jian Luo. Bcacp-ioe: A novel blockchain-based security access control protocol for internet of energy. In *2023 6th International Conference on Information Communication and Signal Processing (ICICSP)*, pages 711–716. IEEE, 2023.

[131] Jiansheng Zhang, Yang Xin, Yuyan Wang, Xiaohui Lei, and Yixian Yang. A secure energy internet scheme for iov based on post-quantum blockchain. *Computers, Materials & Continua*, 75(3):6323–6336, 2023.

[132] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems (TOCS)*, 8(1):18–36, 1990.

[133] Huanhuan Ma, Chenyu Wang, Guosheng Xu, Qiang Cao, Guoai Xu, and Li Duan. Anonymous authentication protocol based on physical unclonable function and elliptic curve cryptography for smart grid. *IEEE Systems Journal*, 17(4):6425–6436, 2023.

[134] Danny Dolev and Andrew Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, 1983.

[135] Alessandro Armando, David Basin, Yohan Boichut, Yannick Chevalier, Luca Compagna, Jorge Cuéllar, P Hankes Drielsma, Pierre-Cyrille Héam, Olga Kouchnarenko, Jacopo Mantovani, et al. The avispa tool for the automated validation of internet security protocols and applications. In *Computer Aided Verification: 17th International Conference, CAV 2005, Edinburgh, Scotland, UK, July 6-10, 2005. Proceedings 17*, pages 281–285. Springer, 2005.

[136] José L Hernández-Ramos, M Victoria Moreno, Jorge Bernal Bernabé, Dan García Carrillo, and Antonio F Skarmeta. Safir: Secure access framework for iot-enabled services on smart buildings. *Journal of Computer and System Sciences*, 81(8):1452–1463, 2015.

[137] LF Decentralized Trust. Hyperledger fabric. https://www.hyperledger.org/use/fabric, 2022. Accessed: 10-03-2023.

[138] Ashwani Kumar. *Hyperledger Fabric In-Depth: Learn, Build and Deploy Blockchain Applications Using Hyperledger Fabric*. BPB Publications, New Delhi, India, first edition, 2020. ISBN = 978-938-932-822-6.

[139] Veneta Aleksieva, Hristo Valchanov, and Anton Huliyan. Implementation of smart-contract, based on hyperledger fabric blockchain. In *2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA)*, pages 1–4. IEEE, 2020.

[140] Rebecca Yang, Ron Wakefield, Sainan Lyu, Sajani Jayasuriya, Fengling Han, Xun Yi, Xuechao Yang, Gayashan Amarasinghe, and Shiping Chen. Public and private blockchain in construction business process and information integration. *Automation in construction*, 118:103276, 2020.

[141] Caliper banchmark. https://hyperledger.github.io/caliper/. Accessed: 2023-03-10.

[142] Weizheng Wang, Huakun Huang, Lejun Zhang, and Chunhua Su. Secure and efficient mutual authentication protocol for smart grid under blockchain. *Peer-to-Peer Networking and Applications*, 14:2681–2693, 2021.

[143] Dipanwita Sadhukhan, Sangram Ray, Mohammad S Obaidat, and Mou Dasgupta. A secure and privacy preserving lightweight authentication scheme for smart-grid communication using elliptic curve cryptography. *Journal of Systems Architecture*, 114:101938, 2021.

[144] Tsu-Yang Wu, Yu-Qi Lee, Chien-Ming Chen, Yuan Tian, and Najla Abdulrahman Al-Nabhan. An enhanced pairing-based authentication scheme for smart grid communications. *Journal of Ambient Intelligence and Humanized Computing*, 15:165, 2021.

[145] Qing Fan, Jianhua Chen, Lazarus Jegatha Deborah, and Min Luo. A secure and efficient authentication and data sharing scheme for internet of things based on blockchain. *Journal of Systems Architecture*, 117:102112, 2021.

[146] Basudeb Bera, Sourav Saha, Ashok Kumar Das, and Athanasios V Vasilakos. Designing blockchain-based access control protocol in iot-enabled smart-grid system. *IEEE Internet of Things Journal*, 8(7):5744–5761, 2020.

[147] Amina Zahoor, Khalid Mahmood, Salman Shamshad, Muhammad Asad Saleem, Muhammad Faizan Ayub, Mauro Conti, and Ashok Kumar Das. An access control scheme in iot-enabled smart-grid systems using blockchain and puf. *Internet of Things*, 22:100708, 2023.

[148] Akhtar Badshah, Muhammad Waqas, Ghulam Abbas, Fazal Muhammad, Ziaul Haq Abbas, S Vimal, and Muhammad Bilal. Lake-bsg: Lightweight authenticated key exchange scheme for blockchain-enabled smart grids. *Sustainable Energy Technologies and Assessments*, 52:102248, 2022.

[149] Ashish Tomar and Sachin Tripathi. Blockchain-assisted authentication and key agreement scheme for fog-based smart grid. *Cluster Computing*, 25:451–468, 2022.

[150] KiSung Park, JoonYoung Lee, Ashok Kumar Das, and Youngho Park. Bpps: Blockchain-enabled privacy-preserving scheme for demand-response management in smart grid environments. *IEEE Transactions on Dependable and Secure Computing*, 20(2):1719–1729, 2022.

[151] Sudeep Tanwar, Karan Parekh, and Richard Evans. Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *Journal of Information Security and Applications*, 50:102407, 2020.

[152] Leibo Liu, Shaojun Wei, Jianfeng Zhu, and Chenchen Deng. Future application prospects. In *Software Defined Chips*, pages 279–318. Springer Nature, Singapore, 2023.

[153] Xiaohui Li, Dexin Zhu, Jianan Wu, Huan Wang, Lifeng Yang, and Lijun Song. A quantum key injection scheme for mobile terminals based on commercial quantum key distribution. *International Journal of Sensor Networks*, 38(2):132–141, 2022.

[154] Shalini Subramani and Santhosh Kumar Svn. Review of security methods based on classical cryptography and quantum cryptography. *Cybernetics and Systems*, 56(3):302–320, 2025.

[155] Chahrazed Benrebbouh, Houssem Mansouri, Sarra Cherbal, and Al-Sakib Khan Pathan. A lightweight security scheme to defend against quantum attack in iot-based energy internet. *International Journal of Sensor Networks*, 43(1):13–26, 2023.

[156] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings 17*, pages 112–131. Springer, 1997.

[157] Jeffrey Hoffstein, Jill Pipher, Joseph H Silverman, and Joseph H Silverman. *An introduction to mathematical cryptography*, volume 1. Springer, New York, USA, first edition, 2008. ISBN: 978-0-387-77994-2.

[158] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical computer science*, 560:7–11, 2014.

[159] Chankyun Lee, Ilkwon Sohn, and Wonhyuk Lee. Eavesdropping detection in bb84 quantum key distribution protocols. *IEEE Transactions on Network and Service Management*, 19(3):2689–2701, 2022.

[160] Neeraj Kumar, Gagangeet Singh Aujla, Ashok Kumar Das, and Mauro Conti. Eccauth: A secure authentication protocol for demand response management in a smart grid system. *IEEE Transactions on Industrial Informatics*, 15(12):6572–6582, 2019.

[161] Dipanwita Sadhukhan, Sangram Ray, Mohammad S Obaidat, and Mou Dasgupta. A secure and privacy preserving lightweight authentication scheme for smart-grid communication using elliptic curve cryptography. *Journal of Systems Architecture*, 114:101938, 2021.

[162] Dipanwita Sadhukhan, Sangram Ray, GP Biswas, Muhammad Khurram Khan, and Mou Dasgupta. A lightweight remote user authentication scheme for iot communication using elliptic curve cryptography. *The Journal of Supercomputing*, 77:1114–1151, 2021.

[163] Qing Fan, Jianhua Chen, Lazarus Jegatha Deborah, and Min Luo. A secure and efficient authentication and data sharing scheme for internet of things based on blockchain. *Journal of Systems Architecture*, 117:102112, 2021.

[164] Khalid Mahmood, Shehzad Ashraf Chaudhry, Husnain Naqvi, Saru Kumari, Xiong Li, and Arun Kumar Sangaiah. An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. *Future Generation Computer Systems*, 81:557–565, 2018.

[165] Dariush Abbasinezhad-Mood and Morteza Nikooghadam. Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended chebyshev chaotic maps. *IEEE Transactions on Industrial Informatics*, 14(11):4815–4828, 2018.

[166] Yuwen Chen, José-Fernán Martínez, Pedro Castillejo, and Lourdes López. A bilinear map pairing based authentication scheme for smart grid communications: Pauth. *IEEE Access*, 7:22633–22643, 2019.

[167] Chahrazed Benrebbouh, Houssem Mansouri, Sarra Cherbal, and Lemia Louail. Enhancing security and authentication in iot-based energy internet using post-quantum blockchain. In *2023 5th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pages 1–8. IEEE, 2023.

[168] Arwa Alrawais, Abdulrahman Alhothaily, Chunqiang Hu, and Xiuzhen Cheng. Fog computing for the internet of things: Security and privacy issues. *IEEE Internet Computing*, 21(2):34–42, 2017.

[169] Patil Rachana Yogesh et al. Formal verification of secure evidence collection protocol using ban logic and avispa. *Procedia Computer Science*, 167:1334–1344, 2020.

[170] Tech Target Network. Ai and gdpr: How is ai being regulated? https://www.techtarget.com/searchdatabackup/feature/AI-and-GDPR-How-is-AI-being-regulated, 2024. Accessed: 24-05-2024.

[171] Data Guidance. International: The interplay between the ai act and the gdpr - ai series part 1. https://www.dataguidance.com/opinion/international-interplay-between-ai-act-and-gdpr-ai, 2023. Accessed: 24-05-2024.

[172] Vanlalruata Hnamte and Jamal Hussain. Dependable intrusion detection system using deep convolutional neural network: A novel framework and performance evaluation approach. *Telematics and Informatics Reports*, 11:100077, 2023.

[173] Mohaimenul Azam Khan Raiaan, Sadman Sakib, Nur Mohammad Fahad, Abdullah Al Mamun, Md Anisur Rahman, Swakkhar Shatabda, and Md Saddam Hossain Mukta. A systematic review of hyperparameter optimization techniques in convolutional neural networks. *Decision Analytics Journal*, 11:100470, 2024.

[174] Panagiotis Radoglou-Grammatikis, Vasiliki Kelli, Thomas Lagkas, Vasileios Argyriou, and Panagiotis Sarigiannidis. Dnp3 intrusion detection dataset. IEEE Dataport: https://dx.doi.org/10.21227/s7h0-b081, 2022. Accessed: 24-03-2024.

[175] Lotfi Mhamdi and Mohd Mat Isa. Securing sdn: Hybrid autoencoder-random forest for intrusion detection and attack mitigation. *Journal of Network and Computer Applications*, 225:103868, 2024.