Karlsruhe Institute of Technology

Certificateless Attribute-Based Server-Aided Cryptosystem for Substation Automation Systems (CASC-SAS)

Master's Thesis of

Moritz Gstür

At the KIT Department of Informatics
Institute for Automation and Applied Informatics (IAI)

First examiner:	Prof. Dr. Veit Hagenmeyer
Second examiner:	Prof. Dr. Achim Streit
First advisor:	Dr. Mohammed Ramadan
Second advisor:	Dr.-Ing. Ghada Elbez

01. August 2024 – 03. February 2025

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself. I have not used any other than the aids that I have mentioned. I have marked all parts of the thesis that I have included from referenced literature, either in their original wording or paraphrasing their contents. I have followed the by-laws to implement scientific integrity at KIT.

Karlsruhe, 03. February 2025

. . . .

(Moritz Gstür)

Abstract

Substation automation systems (SAS) increasingly rely on information and communication technology for monitoring and control. This leads to new challenges with regard to information security. Existing standards such as IEC 61850 and IEC 62351 do not sufficiently cover recent developments, including attribute-based access control (ABAC) and attribute-based public key cryptography (AB-PKC). Therefore, we propose a certificateless attribute-based server-aided cryptosystem for SAS, which integrates into the levels and busses of a newly constructed or retrofitted SAS to enhance its communication security. To protect a SAS against domain-typical adversarial attacks, our approach employs mandatory authentication, authorization, and access control for SAS communication.

Our certificateless attribute-based server-aided authentication approach provides algorithm-agnostic cryptographic protocols and services that serve as a foundation for other cybersecurity mechanisms in a SAS. With our approach we emphasize the advantages of PKC in a SAS, including lightweight and secure key distribution as well as malleability with regard to satisfied security requirements. Accordingly, to safeguard the authenticity, integrity, and non-repudiation of SAS communication, our approach uses authenticated message exchanges based on mandatory digital signatures and signature verification. Furthermore, as we tailored our approach for time-critical communication, it emphasizes the advantages of server-aided cryptography by providing a server-aided AB-PKC signature scheme. In addition to our authentication approach, we provide a server-aided attribute-based authorization and access control approach to prevent unauthorized access to SAS devices. We extend the concept of ABAC by introducing real-time attributes and time-dependent policy evaluation. To take the strict time and resource constraints of a SAS into account, SAS devices delegate the expressive and flexible yet computationally expensive ABAC to policy enforcement and decision points. Moreover, our approach provides evaluation strategies for different network traffic patterns to optimize the computation efficiency, power efficiency, and memory utilization.

To evaluate our approach, we conducted a theoretical and experimental evaluation based on a goal-question-metric approach. The evaluation covers security, performance, and compatibility aspects of our approach. For the experimentally performed evaluations, we provide a testbed implementation of the approach. Based on the implementation, we conducted a laboratory-based experimental demonstration of applicability using the GOOSE and SV protocol between an intelligent electronic device, a merging unit, and an I/O box. The results of the evaluation indicate that our approach is a viable solution to enhance the communication security in a newly constructed or retrofitted substation. The results also indicate, in accordance with the related literature, that the strict time constraints of the low latency communication in a SAS pose a key challenge for information security.

Zusammenfassung

Automatisierungssysteme digitaler Umspannwerke (SAS) nutzen zur Überwachung und Steuerung zunehmend Informations- und Kommunikationstechnologie. Dies führt zu neuen Herausforderungen in Bezug auf die Informationssicherheit. Bestehende Normen wie IEC 61850 und IEC 62351 decken jüngste Entwicklungen, einschließlich der attributbasierten Zugriffskontrolle (ABAC) und der attributbasierten Public-Key-Kryptographie (AB-PKC), nicht ausreichend ab. Daher schlagen wir ein zertifikatsloses, attributbasiertes, servergestütztes Kryptosystem für digitale Umspannwerke vor, das in neu gebaute oder nachgerüstete digitale Umspannwerke integriert wird. Um ein SAS gegen domänentypische Angriffe zu schützen, setzt unser Ansatz auf eine obligatorische Authentifizierung, Autorisierung und Zugriffskontrolle für die SAS-Kommunikation. Unser zertifikatsloser, attributbasierter, servergestützter Authentifizierungsansatz bietet algorithmenagnostische kryptographische Protokolle und Dienste, die als Grundlage für andere Cybersicherheitsmechanismen in einem SAS dienen. Mit unserem Ansatz heben wir die Vorteile von PKC in einem SAS hervor, einschließlich der leichtgewichtigen und sicheren Schlüsselverteilung sowie der Anpassungsfähigkeit in Hinblick auf zu erfüllende Sicherheitsanforderungen. Um die Authentizität, Integrität und Nichtabstreitbarkeit der SAS-Kommunikation zu gewährleisten, verwendet unser Ansatz einen authentifizierten Nachrichtenaustausch, der auf digitalen Signaturen basiert. Um die Vorteile serverbasierter Kryptographie zu unterstreichen, stellen wir zudem ein serverbasiertes AB-PKC-Signaturschema vor. Zusätzlich zu unserem Authentifizierungsansatz bieten wir einen servergestützten, attributbasierten Autorisierungs- und Zugriffskontrollansatz an, um unautorisierten Zugriff auf SAS-Geräte zu verhindern. Wir erweitern das Konzept von ABAC durch die Einführung von Echtzeit-Attributen und zeitabhängiger Richtlinienauswertung. Zudem delegieren SAS-Geräte die rechenintensive Zugriffskontrolle an Durchführungs- und Entscheidungspunkte. Darüber hinaus bietet unser Ansatz verschiedene Auswertungsstrategien für Zugriffsrichtlinien, um die Recheneffizienz, die Leistungseffizienz und die Speichernutzung für verschiedene Netzwerkverkehrsmuster zu optimieren. Um unseren Ansatz zu bewerten, führten wir eine theoretische und experimentelle Evaluation durch, die auf einem Ziel-Frage-Metrik-Ansatz basiert. Die Bewertung umfasst Sicherheits-, Leistungs- und Kompatibilitätsaspekte unseres Ansatzes. Für die experimentell durchgeführten Analyse stellen wir eine Implementierung des Ansatzes in Form einer Testumgebung bereit. Auf der Grundlage dieser Implementierung führten wir eine laborgestützte experimentelle Demonstration der Anwendbarkeit mit Umspannwerksequipment unter Verwendung des GOOSE- und SV-Protokolls durch. Die Ergebnisse der Evaluation zeigen, dass unser Ansatz eine Lösung zur Verbesserung der Kommunikationssicherheit in digitalen Umspannwerken darstellt. In Übereinstimmung mit den verwandten Arbeiten zeigen die Ergebnisse zudem, dass die strengen Zeitvorgaben für die Kommunikation in einem SAS eine große Herausforderung für die Informationssicherheit darstellen.

Contents

List of Figures

List of Tables

1 Introduction

Modern Operational Technology (OT) such as Industrial Control Systems (ICS) increasingly rely on Information and Communication Technology (ICT) for monitoring and control [1]. As a consequence, the resemblance of OT and Information Technology (IT) systems increases, as OT systems adopt IT technology. This development leads to new possibilities including the integration of distributed OT into Supervisory Control And Data Acquisition (SCADA) systems. Nevertheless, new challenges arise from the increased usage of ICT in OT systems.

According to Stouffer et al. [1], the typical long life cycle of OT systems and their unique requirements regarding performance, reliability, security, safety, privacy, and environmental impact have to be taken into account when designing, operating, and maintaining OT systems. In the following, we focus on the information security of OT systems. Although a variety of information security solutions exist for IT, migration of existing approaches to the OT domain may not be a viable solution due to the differing system characteristics, risks, and priorities. An example for the differing priorities are information confidentiality and access control. While the prevention of unauthorized access represents the core objective of IT security approaches, OT systems and especially OT-based critical infrastructure prioritize system availability and reliability.

In the energy-related sector, the infrastructure currently transforms from traditional top-down energy transmission and distribution systems to so-called smart grids with bidirectional data and energy flow [2]. In contrast to traditional energy grids, smart grids are adaptive, self-monitoring and self-healing infrastructures that enable pervasive control and monitoring of distributed heterogeneous grid participants. As illustrated in Figure 1.1, a smart grid interconnects not only producers, consumers, and control centers, but also integrates prosumers, substations, and other grid-related elements. The distribution of formerly centralized entities, such as power plants and control centers, necessitates not only changes in energy infrastructure but also leads to an increased reliance on communication solutions.

The IEC 61850 series provides standards for the communication networks of digital energy systems [3]. The goals of the IEC 61850 series are seamless communication and interoperability of systems in a smart energy grid. Although standards for the communication of digital energy systems are provided by the IEC 61850, information security is not an objective of these standards. To overcome this problem, the IEC 62351 standard series was created by the International Electrotechnical Commission. Part 6 of the IEC 62351

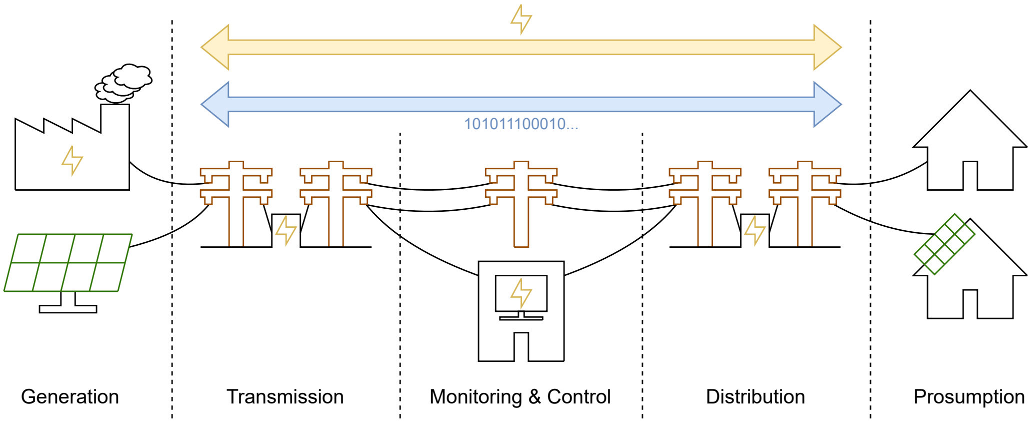


Figure 1.1: Bidirectional power and data distribution in a smart electricity grid.

series provides standardized security means for communication compliant to IEC 61850 [4]. Moreover, Part 8 of the IEC 62351 series provides a role-based access control concept for power systems [5].

The focus of this thesis is on the communication aspects of smart grids. To enhance the communication security and overcome the limitations of existing standards, we propose an approach that can be integrated into smart electricity grids. The field of application of the approach proposed in this thesis is known as a Substation Automation System (SAS). A SAS represents the entirety of communication and control equipment of a substation [6]. A substation is a facility of a high-voltage electricity grid connecting power transmission and distribution lines that use different voltage levels [7]. A substation and its SAS represent a specific type of ICS. The tasks of a SAS are time-critical and have to be executed reliably, as the electricity sector and its substations are critical infrastructures.

1.1 Objective

Although standards regarding the communication networks of smart grid systems are widely accepted and utilized, information security continues to confront unresolved challenges. Historical evidence indicates that economically or politically motivated adversaries pose a risk to OT systems, including energy-related systems. The Communications Security Establishment Canada [8] published a list of 28 OT-related cybersecurity incidents between 2010 and 2020, including incidents in energy-related sectors. These incidents comprise 13 state-sponsored incidents, 13 cybercrime incidents, and two incidents perpetrated by thrill-seeking individuals. The state-sponsored incidents include the Stuxnet malware deployed in Iranian nuclear power and enrichment facilities in 2010 [9], the Shamoon malware used against Saudi Aramco in 2012 [10], the Blackenergy malware used to attack Ukrainian power distribution systems in 2015 [11], the Industroyer/CrashOverride malware used to shut down remote terminal units of a Ukrainian power transmission facility in 2016 [12, 13], and the Triton/Trisis malware used to attack Triconex Safety Instrumented System (SIS) controllers in 2017 [14].

Despite the existence of standards for communication and information security including the IEC 61850 and 62351, there are remaining challenges in order to secure SAS communication. This thesis focuses on these remaining challenges to enhance the information security of SAS communication. As stated by Ishchenko and Nuqui [15], these challenges include, among others, ensuring the integrity and authenticity of substation control and protection communication without compromising the time criticality. For this purpose, cryptographic signature and verification approaches can be employed in the SAS environment. According to Elbez et al. [16], the strict time constraints of the low latency communication in substations are key factors for the information security. Accordingly, Public Key Cryptography (PKC), which was formerly specified by the IEC 62351 standards, seemed to be inappropriate due to computational complexity and latency.

Due to an increase in processing performance of IT and OT devices nowadays, this thesis examines the applicability of effective and efficient PKC in substations. For this purpose, this thesis proposes new cryptographic and cybersecurity approaches for authentication, authorization, and access control. Moreover, the thesis discusses the employment of speedup techniques to enable the usage of secure PKC in time-critical OT systems. Therefore, the following research questions are going to be answered in the course of this thesis:

RQ1 How can expressive and flexible yet computationally expensive access control approaches such as Attribute-Based Access Control (ABAC) be employed to enable prevention of unauthorized access, enable the Separation of Duties (SoD), and ensure the Principle of Least Privilege (PoLP) in a time-critical SAS environment?

RQ2 How can a secure and lightweight PKC approach be designed and implemented, that is able to ensure the authenticity, integrity, and non-repudiation of communication in a time-critical SAS environment?

RQ3 How can authentication, authorization, and access control be integrated into a malleable, scalable, and lightweight cryptosystem for time-critical SAS communication?

1.2 Contribution

With the aim of providing means to enhance the information security in a SAS, we propose a **C**ertificateless **A**ttribute-Based **S**erver-Aided **C**ryptosystem for **S**ubstation **A**utomation **S**ystems (CASC-SAS). The main objective of the proposed approach is to provide secure protocols, algorithms, and schemes for SAS communication. The provided protocols, algorithms, and schemes aim to satisfy SAS security requirements such as integrity, authenticity, access control, and non-repudiation. Furthermore, the approach takes the specific characteristics, risks, and priorities of OT, ICS, and SAS into account. To address the aforementioned objectives and considerations, this thesis comprises the following contributions:

- Identification of security, safety, availability, performance, and compatibility requirements of the proposed approach, and development of a system model, which represents the corresponding field of application.

- Design of a server-aided attribute-based authorization and access control approach, which relies on speedup techniques such as access decision caching and policy evaluation precomputation.
- Design of a certificateless attribute-based server-aided authentication approach, which provides algorithm-agnostic cryptographic protocols and services as well as an AB-PKC signature scheme.
- Design of a certificateless attribute-based server-aided cryptosystem for SAS, which integrates authentication, authorization, and access control into a dual-path four-layered system architecture.
- Implementation of the proposed approach using high-level programming languages, and deployment of the implementation to a test bed that mimics the behavior of an interconnected OT system.
- Security evaluation to prove the security characteristics of the approach.
- Performance evaluation to demonstrate the applicability of the approach in an OT environment with strict time and resource constraints.
- Compatibility evaluation to demonstrate the feasibility of the approach for the construction and retrofitting of a SAS.

1.3 Structure

The following section presents the structure of this thesis. The structure consists of six chapters and is illustrated in Figure 1.2.

Chapter 1 serves to motivate communication security in OT and SAS. In addition, the chapter presents the research questions and outlines the objective and contributions of the proposed approach.

Chapter 2 presents the fundamental concepts upon which this thesis and its proposed approach are based. Among other concepts, it introduces the fundamentals of OT, ICS, information security, system safety, access control, and cryptography.

Chapter 3 presents a review of the existing literature and offers a delineation between the literature and the proposed approach.

Chapter 4 defines the proposed SAS security approach, including its system model, requirements, potential adversarial attacks, security policies, security architecture, and realization. Furthermore, this chapter elucidates the components, algorithms, schemes, and protocols of the proposed security approach.

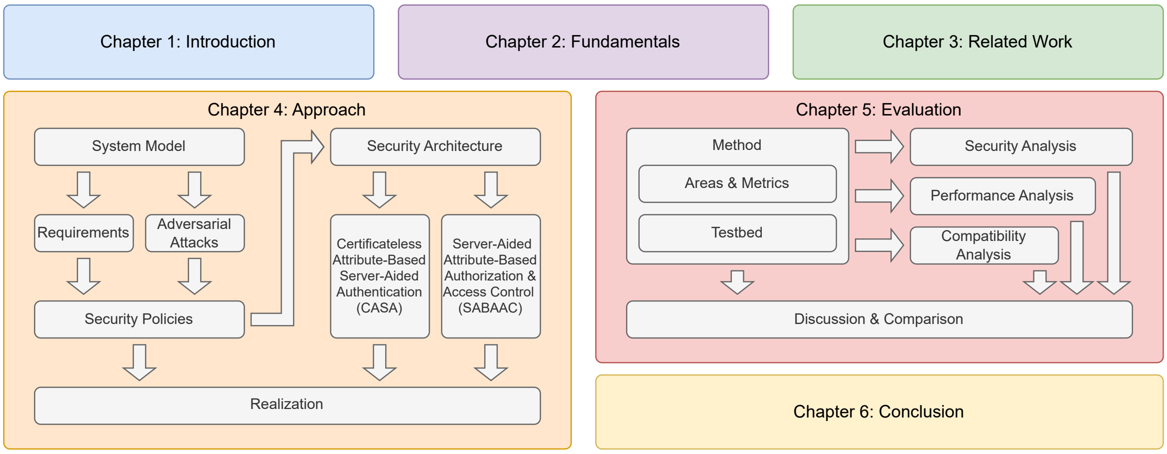


Figure 1.2: Structure of the thesis consisting of six interrelated chapters.

Chapter 5 presents a comprehensive evaluation of the proposed approach, encompassing security, performance, and compatibility considerations. Furthermore, the chapter discusses the results of the evaluation, contextualizes the approach within the existing literature by comparing it to related approaches, and describes the limitations and constraints inherent to the evaluation and proposed approach.

In conclusion, chapter 6 provides insight into prospective future research and presents a summary of the thesis.

2 Fundamentals

The purpose of this chapter is to introduce, define, and describe the fundamental terms and concepts of this thesis. Moreover, this chapter provides an introduction into the foundational literature. The terms and concepts defined within this chapter are assumed to be known in the following chapters.

At the beginning of this chapter, in section 2.1 and section 2.2, the concepts of information technology, operational technology, and industrial control systems are introduced. Moreover, this chapter defines the terms information security in section 2.3 and safety in section 2.4 for the scope of the thesis. Furthermore, this chapter provides an introduction for access control including five access control models. The introduction of access control can be found in section 2.5. At the end of this chapter, in section 2.6, an overview of secret key cryptography and public key cryptography is provided.

2.1 Information Technology (IT) & Operational Technology (OT)

The term Information Technology (IT) encompasses the technological concepts and systems required to create, process, store, present, and communicate information. In the scope of IT, information is an abstract concept which is represented by so-called data or data objects [17]. The meaning of data is assigned to a data object by using a specific information interpretation rule. Data objects can be distinguished based on their abilities by being either passive or active. Passive data objects can only represent information for storage, whereas active data objects can store and process information.

As stated by Eckert [17], an IT system is a dynamic technical system which is able to process and store information. An IT system is part of a sociotechnical system and provides information-based services to more abstract social, economical or political structures. Moreover, the users of an IT system may have different goals, levels of experience, and technical know-how.

When shifting the scope from abstract information storage and processing to the interaction with the physical world, the term Operational Technology (OT) arises. As a consequence, OT describes the application and interaction of information storage and processing procedures in a physical environment. According to Stouffer et al. [1], OT encompasses systems and devices interacting with the physical environment directly or through managed devices. The systems and devices interact with the physical environment by detecting changes through monitoring or by causing changes through control of devices or processes. In the context of

OT systems the term process refers to the part of a system producing an output, whereas a controller represents a part of a system that maintains the conformance with specifications. Besides the Industrial Control Systems (ICS), further discussed in section 2.2, other examples of OT systems are building automation systems and transportation systems.

Although the evolution from analog systems to OT systems by inserting IT into existing physical systems might provide new functionality and enhance system parameters like costs or performance, new challenges may arise [1]. Especially the typical long life cycle of OT systems and their unique requirements regarding performance, reliability, security, safety, privacy, and environmental impact have to be taken into account when designing, operating, and maintaining OT systems. In the following, the thesis focuses on the security implications as well as the design and implementation of secure OT systems.

2.2 Industrial Control System (ICS)

The term ICS encompasses different types of control systems consisting of monitoring, control and network components acting together to achieve an industrial objective [18]. In the scope of the thesis, an ICS represents a specific type of OT system that gathers, processes, and stores information while interacting with a physical environment to achieve an industrial objective. According to Stouffer et al. [18], the control in an ICS can be partially or fully automated. Moreover, an ICS can be configured to operate in three different modes:

1. Manual Mode: The ICS is completely controlled by humans.
2. Open-Loop Control Mode: The output of the system process is controlled by established settings rather than process feedback.
3. Closed-Loop Control Mode: The ICS uses the process output as feedback to achieve the control objective.

2.2.1 Architectures

ICS as well as generic control systems consisting of multiple interconnected components can be classified regarding their control system architecture. According to Galloway and Hancke [19], an ICS architecture or architecture of an ICS network is typically deeper regarding the levels of hierarchy than a company network. Moreover, the technologies including devices as well as the communication links and protocols in an ICS network are often heterogeneous.

In the following sections, the main types of control system architectures and topologies are presented. While these architectures introduce different and partially incompatible concepts, the approaches can be complementing when used on different levels of hierarchy of a complex ICS network [1].

2.2.1.1 Supervisory Control & Data Acquisition (SCADA)

Supervisor Control And Data Acquisition (SCADA) is a type of control system architecture. As stated by Bailey and Wright [20], SCADA refers to a combination of telemetry and data acquisition. The objective of SCADA is to collect data of a remote process, transfer it to a central site, process and analyze the data, and present it to a human operator via Human Machine Interfaces (HMI). Moreover, SCADA enables sending control actions back to the remote process.

The collection of data from devices of a remote process and the delivery of control actions back to the remote devices requires a communication path between the central and remote site [1]. Within the scope of OT and ICS, the central site is referred to as control center and the remote site is referred to as field or field site. Specialized network components at the field site enable remote devices to communicate with the control center via telecommunication technologies. These specialized network components at the field are referred to as gateways or Remote Terminal Units (RTU). The RTUs communicate with a device at the control center also known as Master Terminal Unit (MTU). The network components of an ICS network are further discussed in subsection 2.2.2. Examples for telecommunication technologies used for the communication are Wide Area Networks (WAN), satellite, cellular, and radio technology.

Although the SCADA approach not necessarily requires a communication network to exist but rather works via direct connection between remote devices and the central site, modern SCADA systems rely on bus-based field networks or Ethernet-based solutions [20]. As a consequence, according to Bailey and Wright, the benefits of modern SCADA approaches are minimal required wiring, plug-and-play installation and replacement of devices, remote access to data from anywhere, easier large-scale data storage, and higher flexibility for visualization and incorporation of real data simulations. The disadvantages of modern SCADA approaches are the higher complexity of components, the functional limitations induced by the network components, the requirement of better trained employees, the higher reliance on communication networks, and the high prices of intelligent field equipment.

Stouffer et al. [1] further described four basic communication topologies for modern SCADA networks that were initially introduced by the American Gas Association [21]. The four topologies introduced are point-to-point, series, series-star, and multi-drop. The point-to-point topology connects each field device using an individual communication channel. The series, series-star, and multi-drop topologies use daisy-chaining and switching to connect multiple devices using a single shared channel. The sharing of a single channel among multiple devices increases the efficiency and operation complexity, while it decreases the costs and system complexity.

2.2.1.2 Distributed Control System (DCS)

A Distributed Control System (DCS) is a control system architecture without centralized remote control of the field site [1]. Instead of controlling the field site remotely from a control center, a DCS realizes supervisory control of multiple process sub-systems at the field site. Therefore, a DCS is typically implemented for the control of a process and its sub-processes within the same geographic location.

As stated by Galloway and Hancke [19], a DCS is a process-driven system rather than an event-driven system like SCADA. The objective of a DCS is the control of integrated systems that are closely located, whereas SCADA focuses on independent systems with large geographical extent. Due to the small geographical area and high interconnection within a DCS, the communication with control devices is more reliable and less prone to issues based on the data quality.

2.2.1.3 Programmable Logic Controller (PLC)

A Programmable Logic Controller (PLC) is a control system component responsible for locally managing and controlling a certain process [1]. Therefore, a PLC may represent the primary controller in a PLC-based topology for small OT systems. Moreover, PLCs can be used as building blocks to realize more complex hierarchical topologies like SCADA or DCS. In the latter case, a PLC may integrate or use the services and communication abilities of a RTU, as further discussed in subsection 2.2.2.

PLCs can provide fixed functionality or be modular. Fixed functionality PLCs may also be programmable but limited to certain inputs and outputs, processing abilities, or communication abilities. According to Galloway and Hancke [19], modularity of PLCs eases maintenance and grants more flexibility for the installation. A modular PLC generally consists of a power supply, processing modules, input and output modules, and communication modules.

2.2.2 Network Components

An ICS consists of different components providing functionalities for the monitoring and control of industrial processes [1]. As mentioned above, the field devices of an ICS interact with a physical environment to achieve an industrial control objective. These field devices include different types of sensors and actuators.

As discussed in subsection 2.2.1, ICS network architectures with certain topologies integrate multiple devices into a single complex centralized or distributed ICS system achieving a control objective. To integrate field devices like sensors and actuators into an ICS network, specialized network devices provide communication services. These provided services enable communication between devices at the same field site or to remote devices like a SCADA MTU.

2.2.2.1 Remote Terminal Unit (RTU)

A RTU is a network device at the field site that forwards information from connected field devices to other network devices and vice versa [1]. As a consequence, an RTU acts as a gateway between field devices and network devices at a higher level of the network hierarchy. In other words, an RTU provides an interface for the physical environment to an ICS based on SCADA or DCS. According to Galloway and Hancke [19], an RTU is usually a special type of PLC.

2.2.2.2 Intelligent Electronic Device (IED)

An Intelligent Electronic Device (IED) is a network device with one or more processors capable of sending data to external sources or receiving data [21]. As stated by Stouffer et al. [1], an IED provides a direct interface for controlling and monitoring of field devices to a supervisory controller. Moreover, an IED can be distinguished from an RTU as it is able to act without direct instructions of a supervisory controller.

According to Stouffer et al. [1], the control timing requirements have to be considered when designing OT systems. Therefore, automated control devices are required to perform necessary control actions as human operators may not be reliable, consistent or fast enough. Especially in an ICS with large geographical extent, it may be required to perform computations close to the field devices to reduce or avoid communication latency. IEDs can provide the computational performance and features required to realize time-constrained control functionality at the field.

2.3 Security

Eckert [17] states that security is a characteristic of an IT system. A secure IT system does not allow any system states leading to unauthorized information extraction or manipulation. Security in the scope of computer systems is also referred to as information security or IT security.

Within the scope of OT, the risks and priorities differ from IT systems [1]. While security approaches for IT systems were developed and refined over the years, OT systems were often isolated and widely used proprietary solutions. As modern OT systems increasingly integrate IT technology for connectivity and remote access, proprietary solutions get replaced with widely available solutions. This leads to less isolation and a requirement for OT security solutions. According to Stouffer et al. [1], precautions have to be taken when introducing OT security solutions resembling IT solutions due to the differing requirements of OT systems. Stouffer et al. state that considerations for OT security have to include the special requirements regarding timeliness, performance, constrained resources, availability, communication protocols, and risk management. Moreover, they mention the physical effect an OT system has on its environment, its typically longer component lifecycle including the differing change management, and the geographical distribution of physical components.

2.3.1 Subject & Object

Within the scope of information security, the entities of a system are either referred to as subjects or objects [22]. A subject of a system is an active entity that represents an individual, process, or device causing information to flow among objects or changing the system state. On the other hand, an object is a passive entity of a system representing devices, files, records, or programs. In other words, an object is an entity used to store, access, and process information.

2.3.2 Objective

As state by the National Security Agency [23], a security objective is a statement of intent to counter a given threat or enforce a given organizational security policy. In other words, security objectives define the security requirements of a system. The security objectives of a system are referred to as security goals or protection goals. As stated by Eckert [17], literature typically mentions three main security goals for IT systems. These goals are referred to as CIA which stands for confidentiality, integrity, and availability. The relative importance of a specific security goal depends on the concrete system and its environment. Therefore, within the scope of IT systems confidentiality and integrity may be more important than availability. The six security goals described by Eckert including CIA are discussed in the following sections.

According to Stouffer et al. [1], the characteristics of an OT system may differ from the characteristics of an IT system. As a consequence, the relative importance of specific security goals may differ. Especially if the operation of an OT system has an impact on human health and safety or may cause environmental damage, the security goals integrity and availability may be prioritized over confidentiality of information.

2.3.2.1 Confidentiality

A system has the characteristic of confidentiality if it prevents unauthorized access or extraction of information [17]. To prohibit direct unauthorized access of sensitive information, encryption techniques and access control as described in section 2.5 are used.

Moreover, besides preventing the direct access of information in an unauthorized manner, a system must be protected against leakage of data. This leakage can occur if multiple programs or processes communicate to provide a certain service. According to Lampson [24], a program that is unable to leak data is called confined. The corresponding problem is referred to as confinement problem.

2.3.2.2 Integrity

A system has the characteristic of integrity if the system prevents undetected unauthorized or accidental manipulation of data [17]. If a manipulation cannot be prevented due to the environment, for example when data is exchanged using a shared network, the manipulation has to be detected by the system. As a consequence, a system with integrity always detects manipulation and never processes manipulated data. To detect manipulation, cryptographic hash functions can be used to verify the integrity of data.

2.3.2.3 Availability

A system satisfies the conditions of availability, if authenticated and authorized access to the services and data provided by the system is possible at any time [17]. An available system has to prevent accidentally and maliciously caused discontinuities and disturbances.

2.3.2.4 Authenticity

Authenticity is a characteristic of data objects or entities accessing data objects [17]. A data object or subject is authentic, if it is genuine and trustworthy. The authenticity of a subject can be proven using its unique identity and certain characteristics. The characteristics to prove the trustworthiness of a subject may include credentials like username and password or biometric information. The authenticity of a data object can be proven by verifying the corresponding source and originator.

2.3.2.5 Non-Repudiation

A system ensures non-repudiation by making it impossible for a subject or author of data to dispute its authorship [17]. Non-repudiation can be realized within a system using digital signatures and mechanisms to audit and log user activity.

2.3.2.6 Privacy

The term privacy describes the ability of a person to control the usage of personal information [17]. Moreover, privacy requires special mechanisms for protection of personal information to prevent unauthorized access and fraudulent use. Besides techniques to ensure confidentiality and integrity, data anonymization and pseudonymization can be used.

According to Eckert [17], the term anonymization comprises techniques to change personal data in a certain way to make it impossible to infer the identity of a person from the personal data. Pseudonymization is a weaker form of anonymization allowing the processing of personal data as long as the identity of a person cannot be inferred from the personal data directly without the use of additional information.

2.3.3 Level & Category

The security level and security category represent a characteristic of data objects and subjects denoting their degree of sensitivity [25]. A security level represents a hierarchical or ordered sensitivity, whereas the security category defines a non-hierarchical group to assign degrees of sensitivity to objects and subjects. As stated by Stine et al. [25], the degree of sensitivity is a measure of importance of information assigned by its owner. As a consequence, the degree of sensitivity denotes its need for protection.

The security label is the concrete attribute associated with an object or subject indicating its security level or categories [22]. In other words, each object or subject within the system is labeled according to its security level or categories. The security labels of a subject are referred to as clearances, whereas the security labels of an object are referred to as classifications [26].

2.3.4 Policy

According to Anderson et al. [27], a security policy is a set of documents or a high-level specification stating the security goals and properties to be achieved by the security mechanisms of a system. In other words, a security policy is a set of criteria for the provision of security capabilities and functions to support one or more security objectives [22]. As a consequence, a security policy defines the conditions under which a system grants or denies the access to an object for a specific subject.

2.4 Safety

While information security as described in section 2.3 serves the purpose of avoiding unauthorized access and manipulation of the system, the consequences for the environment due to an erroneous state of the system are not considered. Therefore, safety represents a characteristic of an IT system that is present if the system cannot transition into a functionally invalid state under possible operating conditions [17]. As a consequence, a safe system does not pose a threat to its physical environment including its human operators.

As an OT system may be able to directly interact with its physical environment, safety requirements have to be considered in the OT system design [1]. OT systems have to detect unsafe states and trigger actions to transition into safe states. Moreover, the impact of failures has to be considered and solutions to continue operations may be required. To continue operations, redundancy or the ability to operate in a degraded state can be used. Besides automatic procedures, human oversight and manual supervisory control are essential for safety-critical processes.

2.5 Access Control

As stated by the National Institute of Standards and Technology [28], Access Control (AC) is the process of granting and denying specific requests to logical or physical services and resources. Based on the type of service or resource guarded by the access control, two types of access control can be distinguished. Physical access control supervises access requests of subjects to specific physical facilities like federal buildings or military establishments. Logical access control monitors and controls the access and usage of information and related information processing services. Within the scope of the thesis, the term access control is going to be used to describe logical access control for IT and OT systems.

As stated by Hu et al. [29], logical access control protects objects like data, services, executable applications, or network devices from unauthorized operations. An operation is performed by a subject on a specific object. Operations include access, utilization, manipulation, and deletion of objects. An operation may also be referred to as action. To protect an object, the owners of the objects establish access control policies. These policies describe which subjects may perform certain operations on a specific object.

The policies are enforced by logical components referred to as Access Control Mechanisms (ACM). Hu et al. [29] state that the ACM receives the access request from the subject, decides whether the request should be granted or denied, and enforces the decision taken. The ACM takes the decision based on a framework called access control model. The access control model defines the functionalities and environment including subjects, objects, and rules for the ACM to take and enforce a decision. In the following sections, five different access control models are introduced. The access control models presented differ regarding their applicability and flexibility. Moreover, each model has specific advantages and disadvantages.

2.5.1 Discretionary Access Control (DAC)

Discretionary Access Control (DAC) is an object-based access control model [17]. An owner has to monitor and control the access of other subjects to its own objects. The owner grants or denies the access to its own objects individually. Dependencies between objects have to be considered and solved for each object manually which may lead to inconsistencies.

The Task Force Interagency Working Group [22] defines DAC as an access control policy that enables a subject, that has been granted access to information, to pass the information and its own privileges to other subjects. Moreover, a subject may choose the security attributes of newly created objects, change security attributes, and change rules governing access control.

2.5.2 Mandatory Access Control (MAC)

Mandatory Access Control (MAC) is a system-based access control model [17]. MAC specifies system-wide or global access policies. MAC can complement DAC and vice-versa. If the DAC grants access and the MAC does not, the access request is denied. Moreover, if the MAC grants access to an object the DAC can further restrict the access.

The Task Force Interagency Working Group [22] defines MAC as an access control policy uniformly enforced over all subjects and objects within a system. MAC is considered a non-discretionary access control prohibiting and preventing authorized subjects from passing information and privileges to unauthorized subjects.

2.5.3 Identity-Based Access Control (IBAC)

Identity-Based Access Control (IBAC) is a user-centric access control model employing mechanisms that use the identities of subjects to take authorization decisions [29]. In other words, IBAC represents an access control assigning access authorizations to objects based on the user identity [26].

An example of an IBAC mechanism capturing the identities of subjects and their access privileges is an access control list (ACL) [29]. Each object is associated with an ACL containing privileges assigned to each subject and a representation of a subject identity like credentials. If a subject requests access to a specific object and the presented identity matches the ACL entry, the request is granted or denied as indicated by the ACL entry. As a consequence, an ACL makes authorization decision statically based on its entries prior to the access request. The static behavior of ACLs leads to the disadvantage that the entries have to be reevaluated and revoked regularly to avoid users accumulating privileges.

2.5.4 Role-Based Access Control (RBAC)

The Role-Based Access Control (RBAC) is a task-centric or responsibility-centric access control model [17]. Instead of assigning privileges to each subject individually, roles for different tasks or responsibilities within the system are created. These roles are assigned to subjects explicitly and subjects inherit the privileges of their roles. As stated by Hu et al. [29], a role can be seen as a subject attribute evaluated by the ACM to take an access decision.

According to the Task Force Interagency Working Group [22], a role may apply to a single or multiple individuals. The privileges of a role reflect the permissions an individual requires within an organization and may be inherited through a role hierarchy.

2.5.5 Attribute-Based Access Control (ABAC)

The Attribute-Based Access Control (ABAC) is an access control model enabling access decisions based on attributes associated with subjects, objects, actions, and the environment of a system [22]. In other words, in ABAC an access request of a subject to perform operations on objects is decided based on assigned attributes of the subject and object, environment conditions, and a set of policies [29]. As a consequence, ABAC is also referred to as aspect-based access control [30] or policy-based access control.

Within the context of ABAC, an attribute is a characteristic containing information in the form of a name-value pair [29]. A subject attribute describes the characteristics of a person or non-person entity like identity, clearance, or department. An object attribute describes the resource for which the access is requested, including the object classification, type, or owner. An operation or action attribute describes the function performed on an object by a subject. Operations include create, read, update, delete, or execute. The environment conditions or environment attributes describe the context of an access request. Environment conditions include dynamic characteristics like time of the day, day of the week, and request location of the subject.

A policy represents a rule based on which an access decision is taken for specific attributes [29]. As a consequence, a policy can be seen as a relationship between subject, object, environment, and operation attributes describing under which circumstances the ACM grants or denies an access request.

According to Hu et al. [29], RBAC and IBAC represent special cases of ABAC regarding their attributes used. Furthermore, ABAC is capable of enforcing DAC as well as MAC concepts. An advantage of ABAC compared to different access control models is the higher flexibility regarding multifactor policy expression. Moreover, ABAC can take access control decisions based on ad-hoc knowledge and knowledge from separate infrastructure. This is possible due to ABAC taking decisions at request time by evaluating policies instead of static decision-making as found in IBAC and RBAC. As a consequence, pre-provisioning of requesting subjects in a multi-organization environment can be avoided.

2.5.6 NIST Recommendations

The National Institute of Standards and Technology (NIST) provides recommendations and guidance for authentication, authorization, and access control in ICS [18] and OT systems [1]. With regard to authentication, NIST recommends considering the identity management lifecycle in OT environments. This lifecycle includes the issuance, update, and revocation of authentication credentials. Furthermore, NIST recommends the consideration of centralized identity management and authentication to improve management and monitoring. NIST mentions Active Directory (AD) and Lightweight Directory Access Protocol (LDAP) as centralization supporting network technologies. Nevertheless, NIST points out that authentication might not be advisable if it has an impact on performance, reliability, and safety. This is especially the case in emergency situations in which authentication may impede procedures and result in negative consequences for system safety.

With regard to authorization and access control, NIST emphasizes the importance of considering physical and logical means for OT security. NIST states that organizations are not limited to a single access control approach, but may rather employ different approaches resulting in higher effectiveness and efficiency. A combination of ACLs, RBAC, and ABAC is mentioned as an example for achieving the access control requirements of an organization.

NIST recommends considering logical access control to minimize errors and costs of maintaining access privileges. It is recommended that approaches support the Principle of Least Privilege (PoLP) and Separation of Duties (SoD). Furthermore, NIST recommends solutions that incorporate credential management, authentication, authorization, access control, and system monitoring. These solutions represent secure platforms enabling the access to OT devices. Solutions that verify the identity of individuals or devices before granting access are recommended, as they lead to lower access and command processing latencies. Moreover, solutions should be highly reliable and designed to reduce the impact on OT operations and safety.

2.6 Cryptography

Cryptography is a scientific discipline concerned with the study of methodologies, algorithms, schemes, and protocols for the encryption and verification of information [31, 32, 26]. In other words, cryptography provides means to prevent unauthorized access and to enable the verification of information. The objective of cryptography is to satisfy specific security goals, including the assurance of confidentiality, integrity, authenticity, and non-repudiation.

Cryptographic algorithms are well-defined computational procedures that transform a variable input into an output [32]. The input of an algorithm comprises a cryptographic key that determines the algorithm's operation. Algorithms are classified based on their complexity and degree of distribution. Cryptographic primitives represent low-level cryptographic algorithms. The purpose of primitives is to act as building blocks for more complex algorithms. A cryptographic scheme represents a set of unambiguously specified transformations providing a cryptographic service. Accordingly, schemes are more abstract or higher-level constructs than primitives. Cryptographic protocols specify the information exchange between communicating entities. For this purpose, protocols define the message order and data structures for exchanged information. Consequently, protocols are more abstract or higher-level constructs than schemes.

A cryptographic system, also referred to as cryptosystem, is a set of cryptographic algorithms [33]. Moreover, a cryptosystem comprises sets of valid inputs and outputs as well as required cryptographic keys [17]. The goal of a cryptosystem is to provide specific cryptographic services such as encryption or verification. Verification describes the process of proving the integrity, authenticity, or non-repudiation of information [34]. The verification of information is based on a so-called tag or signature created by a signature algorithm. Encryption describes the process of transforming plain information into an unintelligible form to maintain its secrecy [31, 34]. The inverse process of encryption is referred to as decryption. The intelligible plain information is referred to as plaintext. The unintelligible or encrypted form of information is referred to as ciphertext.

Two important principles for the design of cryptosystems were formulated by Kerckhoffs and Shannon. As stated by Kerckhoffs, the cryptosystem must not require secrecy and must be able to be known by the adversary without inconvenience [35]. According to Shannon, it shall be assumed that the adversary knows the system being used [36].

2.6.1 Secret Key Cryptography (SKC)

Secret Key Cryptography (SKC), also referred to as symmetric cryptography, relies on algorithms which use the same key for a cryptographic operation and its inverse operation [32, 17]. In other words, the same so-called secret key is used for encryption and decryption, or signing and verification. Consequently, the key must be kept secret to satisfy the security objectives.

To encrypt and decrypt information, an entity uses a secret key [34]. Sender and receiver of confidential messages must agree upon a common secret key prior to the exchange of messages. The secret key has to be exchange via a secure communication channel or with the assistance of a secure key exchange protocol. For the purpose of verification, Message Authentication Codes (MAC) are used in SKC. Therefor, the sender computes a MAC tag for a specific message using the secret key. The MAC tag is then appended to the message. The receiver is able to prove the authenticity and integrity of the message by verifying the appended MAC tag using the secret key.

Consider, for instance, two subjects called Alice and Bob, who wish to exchange confidential messages via an unsecure communication channel. At the beginning, Alice and Bob agree upon a common secret key s with the assistance of a secure key exchange protocol. Subsequently, Alice uses s to encrypt a plaintext message m and transmits the ciphertext message c to Bob via the unsecure communication channel. Upon receipt, Bob retrieves m by using s to decrypt c.

Symmetric cryptography has advantages in comparison with asymmetric cryptography [32]. Firstly, symmetric-key algorithms are faster than asymmetric-key algorithms. Secondly, for a given level of security, symmetric cryptographic keys are shorter. This reduces the memory and bandwidth requirements for key storage and transmission.

2.6.2 Public Key Cryptography (PKC)

Public Key Cryptography (PKC), also referred to as asymmetric cryptography, relies on algorithms which use a pair of two related keys for a cryptographic operation and its inverse operation [32, 26, 17]. This pair of related keys consists of a private key and a public key. The private key must be kept secret. The public key may be shared without consequences for security, as long as its authenticity and integrity is ensured. Although the two keys are related, the private key cannot be efficiently derived from the public key.

To encrypt a plaintext, a sending entity uses the public key of a receiving entity [34]. Only the receiving entity, whose public key was used for encryption, is able to decrypt the ciphertext using its own private key. For the purpose of verification, so-called digital signatures are used in PKC. Therefor, a sending entity computes a digital signature for a specific message using its own private key. The digital signature is then appended to the message. The receiver is able to prove the authenticity and integrity of the message by verifying the appended digital signature using the sender's public key.

Consider, for instance, two subjects called Alice and Bob, who wish to exchange confidential messages via an unsecure communication channel. At the beginning, Alice generates a private key s_{Alice} and a corresponding public key p_{Alice}, and transmits p_{Alice} to Bob. Alice's private key s_{Alice} is never exchanged. Subsequently, Bob uses p_{Alice} to encrypt a plaintext message m for Alice. The ciphertext message c_{Alice} is then transmitted to Alice via the unsecure communication channel. Since Bob encrypted the message with p_{Alice}, only Alice is able to decrypt the message using the corresponding private key s_{Alice}. Upon receipt, Alice retrieves m by using s_{Alice} to decrypt c_{Alice}.

PKC offers the following advantages over SKC [32, 17]: Firstly, PKC does not require a secure channel to exchange keys. Secondly, the overall number of required keys using PKC is lower. Moreover, the number of keys scales linear with the number of communication entities. For example in a network with n entities, n key pairs or $2n$ keys have to be established. In the same network, pairwise symmetric cryptography would require $n(n-1)/2$ keys.

2.6.2.1 Identity-Based Public Key Cryptography

The concept of identity-based cryptosystems and schemes was initially proposed by Shamir [37]. Identity-Based Public Key Cryptography (ID-PKC) approaches allow the derivation of public keys from subject identities [37, 34]. Therefore, no certificates are required to verify that a public key belongs to a certain subject, since the subject identity is used to derive the public key. In ID-PKC the existence of a public key does not depend on the existence of a corresponding private key. In other words, the public key is not derived from the private key and may therefore exist before the private key.

ID-PKC employs a Trusted Third Party (TTP) called Private Key Generator (PKG) to generate private keys for entities [38, 34]. The PKG generates a system-wide key pair consisting of the master public key and master secret key. The master public key is known to all entities.

The master secret key is only known to the PKG. If an entity wishes to obtain its own private key, it has to prove its identity to the PKG. If the identity is proven to the PKG successfully, the PKG generates a private key based on the entity's identity and the master secret key.

Consider, for instance, two subjects called Alice and Bob, who wish to exchange confidential messages via an unsecure communication channel. Alice obtains Bob's public key p_{Bob} from Bob's identity, e.g., an email address, and the master public key. Subsequently, Alice uses p_{Bob} to encrypt a plaintext message m for Bob and transmits the ciphertext message c_{Bob} via an unsecure communication channel. Upon receipt, Bob obtains the private key s_{Bob} from the PKG via a secure communication channel and decrypts c_{Bob} to retrieve m.

Since the PKG is able to generate private keys for arbitrary identities, ID-PKC leads to the key escrow problem [38]. This allows a misbehaving PKG to decrypt confidential messages or forge subject's signatures. Key escrow is a mechanism that enables a TTP, e.g., a company, to retain the private component of a key pair [39, 34]. The goal of key escrow is to recover encrypted information, even if the encrypting entity and the corresponding secret key is not available.

2.6.2.2 Certificateless Public Key Cryptography

Certificateless Public Key Cryptography (CL-PKC) can be seen as an intermediate approach between ID-PKC and certificate-based PKC approaches such as Public Key Infrastructure (PKI) [38]. Certificate-based PKC approaches such as PKI use a TTP called Certificate Authority (CA) to issue, store, and revoke digital certificates [17]. These digital certificates are used to verify that a certain public key belongs to a specific subject which holds the corresponding private key.

CL-PKC approaches make use of a TTP called Key Generating Center (KGC) to generate partial private keys for entities based on the entity's identity and a master key [38]. To obtain the private key, the entity combines the partial private key with a secret value. The generated private key is never shared with the KGC or other entities. Consequently, CL-PKC neither suffers from the key escrow problem nor requires a secure communication channel for the key distribution.

To obtain the public key, the entity generates it based on public parameters and the secret value [38]. After the generation, the public key may be shared with other entities directly or via public directories. As with ID-PKC, the public key is not derived from the private key and may therefore exist prior to it. The only restriction is that the public key and the private key must use the same secret value. However, CL-PKC is not identity-based, as the public key is not solely based on the identity of an entity.

2.6.2.3 Attribute-Based Public Key Cryptography

Attribute-Based Public Key Cryptography (AB-PKC) is a generalization of the ID-PKC concept [40, 41, 42]. Attribute-Based Encryption (ABE) combines the principles of ABAC with the concept of PKC. Therefor, attribute-based policies are integrated into cryptographic algorithms in the form of access structures and attributes. This integration enables encryption based on arbitrary attributes and provides fine-grained access control via attribute-based decryption. Communication complexity for key distribution in ID-PKC scales linear with the number of communication entities, i.e., users or devices. In ABE the communication complexity scales linear with the number of attributes. As with the concept of ABE, Attribute-Based Signatures (ABS) enable the integration of attributes into signing and verification algorithms [43, 44].

ABE approaches are classified as either Key-Policy ABE (KP-ABE) or Ciphertext-Policy ABE (CP-ABE), depending on whether the access structure is associated with a key or a ciphertext [41, 45, 42]. In KP-ABE the access structure is associated with a key. The ciphertext is labeled with a set of attributes. A user's secret key is able to decrypt the ciphertext if the attributes of the ciphertext satisfy the key-associated access structure. Consequently, a data owner cannot control who is able to access the data and has to trust a TTP to issue appropriate keys [45]. In CP-ABE the access structure is associated with a ciphertext, and keys are associated with a set of attributes. A user's secret key is able to decrypt the ciphertext if the key-associated attributes satisfy the ciphertext's access structure. Accordingly, each data owner manages the access control policies for its own data, which makes CP-ABE more flexible and scalable than KP-ABE.

3 Related Work

In the following section, the related work of the thesis is presented. The introduced related work serves as a foundation for the proposed approach presented in chapter 4. Moreover, similarities and differences of the related work and the proposed approach are discussed.

The related work consists of two parts. The first part of the related work introduces means to secure the communication, i.e., frame or packet exchange between devices in a SAS. For this purpose, we discuss approaches which safeguard the confidentiality, authenticity, integrity, and non-repudiation of SAS communication. The second part of the related work is dedicated to the prevention of unauthorized access. Accordingly, we discuss approaches which enable access control and satisfy the Principle of Least Privilege (PoLP) and Separation of Duties (SoD).

3.1 Secure Communication in Substations

An authenticated communication approach for network packets between IEDs and merging units is presented by Ishchenko and Nuqui [15]. They identified the lack of security in existing IEC 61850 substations and ICSs in general as a key weakness. To mitigate this weakness, Ishchenko and Nuqui present retrofitting of substations as a viable solution For this purpose, they introduce a system and bump-in-the-wire device called security filter as an add-on device between IEDs and Ethernet-based communication busses using the Generic Object Oriented Substation Event (GOOSE) or Sampled Values (SV) protocol. Security filter appends Message Authentication Code (MAC) tags to outgoing messages of the IEDs and verifies incoming MAC tags. As a consequence, the communication busses are secured against unauthenticated messages achieving the security goals integrity and authenticity. Moreover, the security filter approach uses a timestamp to avoid replay attacks.

To achieve interoperability with legacy communication systems and compatibility with different substation automation systems, the authors introduce a multimode operation design for the security filter. The multimode operation design consists of three operation modes. In filtering mode the security filter verifies all incoming packets, blocks compromised packets after exceeding a certain threshold, and tags all outgoing packets. Moreover, the security filter alarms the IED about compromised packets. In supervisory mode the security filter tags selected packets based on a specific rate of packets, verifies tagged packets only, and blocks and alarms when the number of compromised packets exceeds the threshold. Consequently, supervisory mode leads to a reduced computational effort. The last mode is called advisory mode. In advisory mode the security filter selectively tags and verifies packets based on a specific rate of packets but only triggers alarms and does not block packets after the threshold of compromised packets is reached. Additionally, the operation of the security filter can be disabled in case of internal errors allowing all packets to pass through. Ishchenko and Nuqui showed that the security filter is able to meet the IEC 61850 performance requirements of GOOSE and SV [3] using a Hash Message Authentication Code (HMAC) and Galois Message Authentication Code (GMAC) algorithm even on commodity of-the-shelf ARM hardware.

This thesis introduces an approach similar to the security filter approach presented by Ishchenko and Nuqui. The architecture and security procedures of the proposed approach are inspired by the security filter. The concept of authenticated communication is proposed as a foundation for secure communication in substations. Our approach aims to extend the employed access control from identity-based to attribute-based authorization. As a consequence, more complex access control policies can be established within a substation or ICS in general.

A review of IEC 62351 security recommendations with regard to message authentication and a comparison of viable authentication approaches for IEC 61850 substations is presented by Elbez et al. [16]. As stated by the authors, ensuring integrity and authenticity of substation communication is critical. As with the approach presented by Ishchenko and Nuqui [15], the authors focus on Ethernet-based substation communication using the GOOSE protocol. To ensure integrity and authenticity of substation communication, the authors present two authentication approach for GOOSE messages. Firstly, the authors present the digital signature authentication approach specified by IEC 62351 [4]. This approach is based on asymmetric cryptography using the RSA Probabilistic Signature Scheme with Appendix (RSASSA-PSS) algorithm. Secondly, the authors present a keyed HMAC. The HMAC approach is based on symmetric cryptography and uses a shared secret for signing and verification of GOOSE messages. According to the authors, the HMAC approach requires less computation time. On the one hand, this leads to HMAC being a more viable solution for message authentication under strict timing constraints. On the other hand, a prior key exchange is required to establish the shared secret for the GOOSE provider and each subscriber. Elbez et al. identify the performance of the presented authentication approaches as key factor for GOOSE communication. As a consequence, the authors implemented the authentication approaches and compared the computational times. In addition to the presented implementations, computation times from three other papers were taken into account. According to Elbez et al., the presented computational times show that asymmetric cryptography solutions based on RSA and RSASSA-PSS are not suitable for the timing constraints of GOOSE messages. However, the authentication time of the HMAC approach is of the order of microseconds. Consequently, as stated by the authors, HMAC is a viable approach for the authentication and integrity protection of GOOSE messages.

This thesis backs the results of the authors by evaluating different cryptographic algorithms for low-latency communication in substations. Our approach is based on an algorithm-agnostic cryptography concept. The algorithm-agnostic approach is inspired by the Transport Layer Security (TLS) protocol [46]. As a consequence, our approach is based on the idea that not a single cryptographic algorithm should be used, but rather different cryptographic algorithms and schemes might be optimal solutions for different communications in a SAS. Moreover, our approach emphasizes the advantages of PKC in a SAS, including lightweight and secure key distribution as well as malleability with regard to satisfied security requirements.

An authentication and encryption approach for substation communication using the protocols GOOSE and SV is presented by Rodriguez et al. [47]. The authors state that GOOSE and SV messages are sensitive to not only availability and integrity but also confidentiality threats. Therefore, the authors present a hardware architecture for the encryption and authentication of GOOSE and SV packets at wire-speed conforming to IEC 62351:2020 [4]. The hardware architecture consists of six sections for packet processing that can be implemented using FPGAs. According to Rodriguez et al., the architecture design follows three main guidelines to face challenges within substations. Firstly, the architecture has to be modular to support future revisions of standards, algorithms, and protocols. Secondly, the architecture has to have high performance by making use of techniques like parallelization and pipelining. Lastly, the implementation in substation systems must be viable with regard to required area usage and computing power. The authors conducted the evaluation of the presented architecture using simulation-based and hardware-based timing results. As stated by the authors, the hardware implementation is able to process GOOSE and SV packets with a fixed latency in the order of microseconds. Consequently, the authors state that the presented hardware architecture is able to provide integrity and confidentiality without exceeding the maximum delivery time of three milliseconds introduced by IEC 61850 for GOOSE and SV packets [3].

Besides securing the intra-substation communication based on the GOOSE and SV protocol, the thesis extends the idea of providing integrity, authenticity, and non-repudiation to inter-substation and remote communication. To achieve flexibility and interoperability with regard to different ICS environments including different protocols and algorithms used, our approach is software-based rather than hardware-based. Furthermore, our approach does not rely on a symmetric-key algorithms, but on asymmetric-key algorithms. This is possible due to an increase in processing performance of IT and OT devices nowadays.

According to Hong et al. [48], new technologies in substations lead to benefits including enhanced reliability, interoperability, and reduced engineering effort and costs. Besides the benefits, new technologies introduce vulnerabilities that may result in security breaches. As an example, the authors mention unauthorized remote access to substations through misconfigured security devices, such as firewalls. Moreover, the authors state that an adversary might not only intrude the substation from outside but also from the inside. From inside the substation, an adversary may inject false measurements into the process bus or gain access to the station bus to inject forged control signals or change the configuration of devices like IEDs. To protect substations against attacks, Hong et al. present a domain-based collaborative mitigation approach. According to the authors, the goal of the approach is to enable substation devices to collaboratively defend against attacks. For this purpose, the authors propose a distributed security domain layer. The proposed approach can be employed independently or can complement existing information and communication technology

(ICT) security approaches. As stated by the authors, ICT-based security approaches such as firewalls and intrusion detection systems rely exclusively on ICT domain knowledge, whereas the proposed approach relies on knowledge of the power system domain. As a consequence, new types of attacks as well as errors caused by substation operators can be detected and mitigated. Hong et al. presented three attack scenarios that can be mitigated using the presented domain-based collaborative approach. The presented attack scenarios are an accidental or malicious IED configuration change, false sensor data injection, and false device command injection. Collaborating devices can block these attacks by validating sensor data and configuration changes based on measurements and metrics as well as predicting consequences of control actions.

The approach presented in the thesis is inspired by the usage of domain-specific knowledge to detect and block attacks. Our approach uses available domain-specific knowledge to design and implement a substation-specific cryptosystem. Moreover, the incremental framework of our approach for the system design, threat analysis, and mitigation strategy design is based on the research framework presented by Hong et al.

3.2 Access Control in Substations

An access control approach driven by ABAC policies for smart grid systems including substations is presented by Ruland and Sassmannshausen [49]. As stated by the authors, communication security enables information confidentiality and integrity but does not protect against internal attacks. As a consequence, the authors present an access control approach to protect devices from unauthorized access. The presented access control approach is realized in the form of an access control firewall. The approach is based on an architecture that implements a split station bus. The split station bus serves the purpose of controlling access requests from devices of the outer bus to devices connected to the inner bus. The access control firewall connects the outer and inner station bus by processing access requests of connected devices. Devices connected to the outer station bus include Human Machine Interfaces (HMI), station computers, and WAN gateways. The inner station bus connects IEDs and enables low-latency GOOSE or Generic Substation State Events (GSSE) communication between them. The access control firewall enforces access request decisions based on ABAC policies. The ABAC policies used in the presented approach are defined, communicated, and evaluated using the eXtensible Access Control Markup Language (XACML) standard [50]. According to Ruland and Sassmannshausen, the access request decisions are made by a Policy Decision Point (PDP) that can either be part of the access control firewall or be implemented as an external server on the outer station bus.

The approach presented in the thesis employs ABAC similarly to the access control approach presented by Ruland and Sassmannshausen. Besides employing ABAC to secure the communication between devices on the station bus, our approach controls access requests to any device within the substation that requires access control. For this purpose, a distributed ABAC firewall is used instead of a single firewall. As a consequence, the firewall does not represent a communication bottleneck or single point of failure of an ICS in our approach.

A real-time capable ABAC approach is presented by Burmester et al. [51]. The presented approach identifies the requirements of cyber-physical systems including confidentiality, integrity, and availability. In particular, according to the authors, employing ABAC in real-time availability scenarios can be challenging due to the dynamic and large event space determining the attribute values. In these real-time availability scenarios, events threatening the system state might not be addressed within strict time limits if attribute values are not available in time. For this purpose, the authors propose an extended ABAC model that is based on real-time attributes to support availability within the strict time constraints of cyber-physical systems. A real-time attribute represents an attribute whose value is time-dependent. The availability of a time-dependent attribute can be expressed with an availability label that is dynamically determined based on user and system events as well as the context of the requested service. An availability label is referred to as priority if it is associated to a subject attribute, congestion for an object attribute, and criticality for an environment attribute. The authors demonstrate the real-time ABAC approach for IP multicast in Trusted Computing (TC) compliant networks. Therefor, the authors propose a congestion control algorithm based on the availability labels. The proposed algorithm guarantees that high priority packets are delivered timely. In case of a congestion, lower priority packets may be buffered or dropped to support the real-time requirement of high priority packets. As stated by the authors, the extended ABAC model is applicable to substation automation systems and medical cyber-physical systems.

The access control policy classification and evaluation presented in the thesis are inspired by the authors' concept of real-time attributes. In our approach, real-time attributes together with static attributes form dynamic and static ABAC policies, i.e., access control policies whose evaluation does or does not rely on time-variable subject, object, environment, or action attributes. Consequently, while relying on a similar concept of real-time attributes, the approaches differ in their utilization of these attributes.

An IEC 61850 and IEC 62351 compliant RBAC approach for substations is presented by Lee et al. [52]. According to the authors, data collection and analysis are key drivers in smart grids leading to an increased requirement for data security and access control of substation devices. To address requirements such as confidentiality and integrity, the authors propose an RBAC approach based on IEC 62351 [5] using XACML [50]. As stated by the authors, the communication within substations can either be classified as session-based TCP/IP client-server communication or Ethernet-based publisher-subscriber communication. The presented approach focuses on session-based access control for TCP/IP communication on the station bus of substations. As a consequence, the presented RBAC approach can be employed to process Manufacturing Message Specification (MMS) communication between IEDs and devices at station level. The main contribution of Lee et al. is an implementation of the presented RBAC approach. The presented implementation relies on a role-based client-server architecture. The architecture consists of two interconnected client-server pairs, namely an IEC 61850 client and server as well as a RBAC client and server. The IEC client sends a request including the client's role to the corresponding IEC server. The IEC server responds to permitted IEC client requests. Moreover, the IEC server acts as a Policy Enforcement Point (PEP) by delegating requests to an RBAC client. The RBAC client transforms an IEC request into an XACML request and sends it to the RBAC server for an access request decision. The RBAC server serves the purpose of making access request decisions by evaluating access control policies. An IED of a substation incorporates an IEC 61850 server and RBAC client. The implementation demonstrates the feasibility of RBAC for substations as specified by IEC 62351 [5]. Furthermore, as stated by the authors, the presented implementation is capable of processing and responding to MMS requests within the 500 millisecond time requirement for type 3 messages (low speed messages) specified by IEC 61850-5 [3].

Instead of exclusively relying on roles, the approach presented in the thesis employs ABAC to enable the usage of fine-grained and flexible attribute-based access policies. Moreover, the goal of the proposed approach is to secure any communication within substations including type 1 messages (fast messages) and type 2 messages (medium speed messages) as described by IEC 61850-5 [3].

A distributed RBAC approach for subscription-based remote network services is presented by Ma and Woodhead [54, 53]. According to the authors, identity management for IBAC is a significant challenge for resource providers and subscribing institutions due to the high number of potential users in subscribing institutions. Furthermore, traditional RBAC approaches require a centralized administration of roles, users, and resources by a single organization. As a consequence, traditional RBAC and IBAC approaches do not work well in multi-organization distributed systems such as subscription-based remote network services. For this reason, Ma and Woodhead propose an approach called Distributed Role-based Access Control (DRBAC). DRBAC is a distributed authentication and role-based authorization framework. As stated by the authors, the distributed authentication is realized by delegating the authentication of users to the corresponding subscribing institutions by issuing authentication delegation certificates. The subscribing institutions use their existing authentication infrastructure to authenticate users and create digitally signed Service Access Tickets (SAT). The resource provider is able to use the SAT to verify the legitimacy of requests. The role-based authorization approach of the DRBAC framework extends traditional RBAC by adding the concept of distributed roles shared by the resource provider and resource subscribers. The resource provider specifies the distributed roles and exports them to the subscribing institutions via distributed role certificates. The resource subscribers map their local roles to distributed roles to indirectly associate individual subjects with distributed roles. Therefore, distributed roles represent a middle layer in the DRBAC framework to abstract from subscriber-specific local roles and individual subject identities. As a consequence, DRBAC enables access control policies associated with distributed roles rather than subject identities, which leads to an increase in scalability and manageability of access control. The DRBAC policies are realized in the form of authorization policy certificates. Each DRBAC policy is associated to a certain distributed role and contains a domain-dependent resource operation permission. Moreover, the authors state that their DRBAC approach supports temporal, contextual, or cardinality constraints to enhance the semantic expressiveness of access control and enable the definition of higher-level organizational policies.

The authentication and authorization approach employed in the thesis is inspired by the concept of delegation presented by Ma and Woodhead. Ma and Woodhead illustrate the concept of delegation within the context of a subscription-based remote network service environment. Our approach entails the utilization of authentication and authorization delegation in substations. Moreover, our approach elevates the degree of abstraction of the presented delegation concept by decoupling it from the concrete access control model used. Our approach realizes authorization delegation via PDPs that make access control decisions for resource requests in place of other devices. Furthermore, authentication delegation is used for external resource requests to increase scalability and manageability.

A rule-based RBAC policy enforcement approach for smart grid systems is presented by Alcaraz et al. [55]. According to the authors, the presented approach integrates into a smart grid system with supernode networking architecture. As stated by Samuel et al. [56], supernodes are servers at fixed locations responsible for handling data flows of a set of subscribers. In other words, supernodes represent proxies enabling peer-to-peer connections between devices of dynamic and heterogeneous networks. The policy enforcement approach presented by Alcaraz et al. consists of three execution phases. The first phase is dedicated to the authentication. During the authentication phase a subject authenticates itself at an identity server within its own infrastructure. In case of a successful authentication, the identity server provides the subject with an authentication token. During the second phase the authorization takes place. To acquire an authorization token, a PEP provides the authentication token and the desired type of action on the target object to a PDP. The PDP of the presented approach consists of a validation manager and a Policy Decision Manager (PDM). The former one validates the authentication token as well as roles and permissions associated with the requesting subject, whereas the latter one evaluates the access request and creates the authorization token if it grants the request. Moreover, the presented PDM is based on a rule-based expert system and a context manager for the analysis of the subject, target object, and context of the request. The last phase of the presented approach is referred to as interoperability. During the interoperability phase the PEP transparently applies the security policies as indicated by the authorization token and performs the action requested by the subject.

The approach presented in the thesis relies on a system model with an architecture similar to the approach presented by Alcaraz et al. Moreover, the two approaches resemble in their usage of authentication and authorization delegation as well as their awareness regarding the request context realized via PDP decision-making. Nevertheless, the approaches differ in their degree of dependence on specific access control models. The approach presented by Alcaraz et al. depends on the subject-role associations of RBAC for decision-making as specified by IEC 62351-8 [5]. The proposed approach supports more fine-grained and flexible ABAC policies.

An RBAC-based access control approach using Privilege Management Infrastructure (PMI) for IEC 61850 substations is presented by Liu et al. [57]. The presented access control system is realized in the form of a so-called access security agent component. According to the authors, the access security agent handles the authentication of subjects, parses role-based privileges from subject attribute certificates, provides certificate storage, and performs cryptographic computing. Besides the access control system architecture, the authors provide a single Round-Trip Time (1-RTT) authentication and attribute certificate exchange protocol relying on symmetric as well as asymmetric cryptography. Moreover, the authors present an algorithm for access privilege parsing to retrieve roles and access policies from attribute certificates. In the presented access control approach the parsed role-based access policies are used to establish identity-based access control matrices. An access control matrix of the presented approach controls the access to logical nodes of a substation IED. For this purpose, an access control matrix associates subject identities with permitted operations for each individual data object.

In contrast to the approach presented by Liu et al., our approach relies on attribute-based authorization and access control. While both approaches aim to secure IEC 61850 substation communication, our approach is based on a certificateless communication concept. This concept enables the server-aided evaluation and enforcement of expressive and flexible yet computationally expensive ABAC policies. Moreover, we emphasize the importance of securing not only bay level but also station and process level devices of a SAS.

4 Approach

In the following section, we introduce our proposed security approach for substation automation systems. With the aim of securing the time-critical communication between resource-constrained devices in a time-variable environment, we propose a **C**ertificateless **A**ttribute-Based **S**erver-Aided **C**ryptosystem for **S**ubstation **A**utomation **S**ystems (CASC-SAS). The CASC-SAS cryptography and cybersecurity approach is able to prevent and mitigate cyberattacks by providing security schemes and mechanisms, and enforcing mandatory communication policies. The goal of the approach is the enhancement of SAS security by providing secure authentication, authorization, and attribute-based access control for time-critical SAS communication.

The CASC-SAS approach comprises two core concepts. The first core concept of the approach is the **C**ertificateless **A**ttribute-Based **S**erver-Aided **A**uthentication (CASA). This concept represents the foundation of the CASC-SAS approach. The concept provides cryptographic algorithms and schemes for authentication, including a PKC signature scheme for key generation, signing, and verification. Communicating SAS devices as well as more abstract cybersecurity services can rely on the provided communication integrity, authenticity, and non-repudiation. The CASA concept is further discussed in section 4.6. The second core concept of the approach is the **S**erver-Aided **A**ttribute-**B**ased **A**uthorization and **A**ccess **C**ontrol (SABAAC). This concept provides mechanisms to enable attribute-based authorization and ABAC for time-critical SAS communication. Accordingly, this concept represents cybersecurity means to provide access control, PoLP, and SoD. For this purpose, the concept relies on authentication services provided by CASA. The SABAAC concept is further discussed in section 4.7.

In the following sections, we introduce and discuss our proposed approach. At the beginning of this chapter, in section 4.1, we discuss the field of application of the proposed approach by introducing a system model. Based on the presented system model, we define the requirements of the proposed approach in section 4.2. In section 4.3 we address potential adversarial attacks, for which the approach must provide mitigation strategies. To satisfy the aforementioned requirements and mitigate adversarial attacks, the CASC-SAS approach enforces security policies, which are discussed in section 4.4. Subsequently, in section 4.5, the dual-path four-layered security architecture of CASC-SAS is defined. The two main CASC-SAS concepts, its cryptography approach CASA and its authorization and access control approach SABAAC, are introduced in section 4.6 and section 4.7. Finally, in section 4.8, we present the realization of the CASC-SAS approach.

4.1 System Model

In the following sections, we introduce the system model of the CASC-SAS approach. The system model serves the purpose of delimiting the scope and area of application of the proposed approach.

The area of application of the proposed approach consists of ICSs in the power system domain. More specifically, the proposed approach is tailored to the communication and control systems of substations in the electricity grid. The communication and control equipment of an ICS is referred to as secondary equipment. The entirety of secondary equipment of a substation is referred to as SAS [6]. Although the proposed approach is tailored to the power system domain and substation environment, its main concepts may also be applied to other ICSs with similar requirements and constraints.

4.1.1 Architecture

The architecture of the presented system model is based on the IEC 61850 standards [3]. The presented system model architecture consists of four layers called network, station, bay, and process level. The process, bay, and station level represent the internal layers of a SAS architecture. The SAS architecture containing the three internal layers as well as the station and process bus is shown in Figure 4.1. The shown busses are further discussed in subsection 4.1.2. The network level represents a SAS-external layer to integrate multiple SAS instances and supervisory controllers into a comprehensive power system. Each of the four layers consists of different devices and provides different control and automation functions:

1. Process Level: The process level provides functions to interact with the physical process via sensors and actuators. As a consequence, SAS devices located at the process level provide interfaces to the physical process. In other words, devices located at the process level transform analog measurements or control signals into digital values and vice versa. Devices restricted to the transformation and provision of measurement and control values are referred to as Merging Units (MU). Moreover, IEDs can be employed at the process level to combine MU functions with higher-level functions such as protection or communication tasks.

2. Bay Level: The bay level provides common functions of so-called bays of a SAS. As stated by the International Electrotechnical Commission [3], a bay represents a closely connected subpart of a substation with common functionality. The devices at bay level supervise the operation of lower-level devices of a SAS bay. Consequently, a supervising bay level device is referred to as bay controller or bay protection.

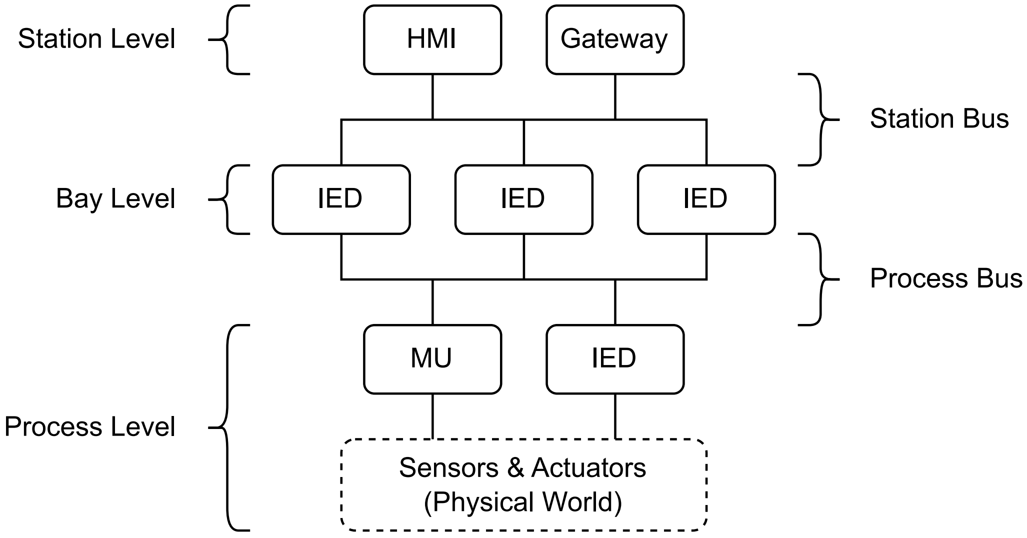


Figure 4.1: Internal three-layered architecture of a SAS.

3. Station Level: The station level provides functions related to the substation as a whole. Therefore, the station level comprises devices required for on-site and remote monitoring and control of the substation. Devices at the station level include Human Machine Interfaces (HMI) for substation operators as well as Wide Area Network (WAN) gateways like SCADA RTUs.
4. Network Level: The network level provides higher-level functions exceeding the scope of a single SAS. The network level devices include supervisory monitoring and control devices like SCADA MTUs.

4.1.2 Communication

In the following, we discuss the communication between devices of the presented system model. For this purpose, we identify different communication characteristics based on which communication relationships and messages can be classified. Moreover, we define three messages types for time-critical ICS and SAS communication. Furthermore, we discuss the bus-based device interactions occurring in the above-mentioned four layer system model.

4.1.2.1 Classification Characteristics

The communication relationships between devices within a SAS can be classified using different communication characteristics. In the following sections, classifications based on topology, continuity, and latency are further discussed.

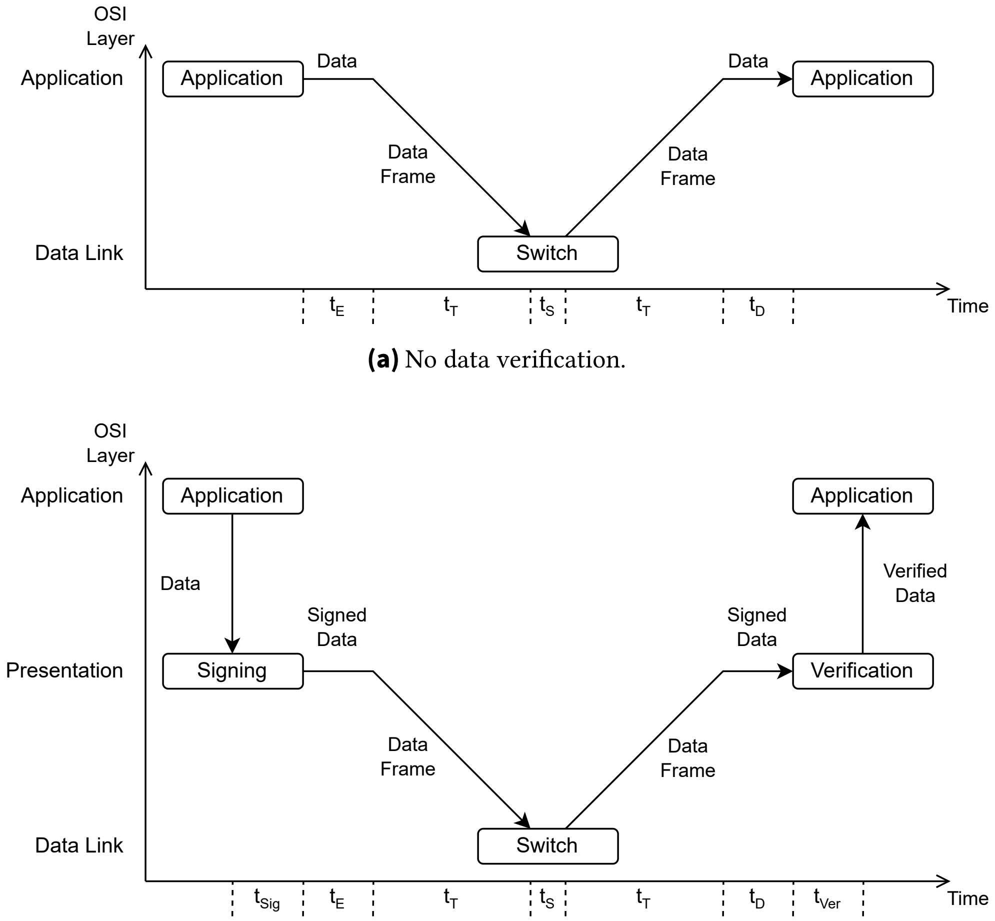
Topology-Based Classification Topological communication characteristics can be used to classify the device relationships based on their relative or absolute location within the system model. Accordingly, communication can either occur between devices on the same layer or different layers of the system model. Communication on the same layer of the system model is referred to as horizontal communication, whereas communication between devices on different layers is referred to as vertical communication. Moreover, communication can occur between devices of the same or different subsystems. Communication between devices of the same subsystem is classified as internal communication, whereas communication relationships including an external device are classified as external communication. Furthermore, a communication relationship is not limited to a single receiver using unicast, but rather a group of devices via multicast or all devices via broadcast may receive a sender's message.

Continuity-Based Classification Besides the topology-based classification, communication relationships can be classified based on their continuity. Continuous, session-oriented, or stateful communication requires an initial session establishment between the involved devices. While the first message exchange requires additional initialization overhead, subsequent latencies may benefit from the established communication session. Discontinuous, message-oriented, or stateless communication enables communication without initial overhead for the involved devices. Consequently, discontinuous communication does not lead to latency emerging from session initialization and management.

Latency-Based Classification Since communication in ICS and SAS is time-critical, communication relationships can be classified based on their communication latency constraints. Within the scope of the proposed approach, we define communication latency as sum of processing time and transmission time required to exchange information between involved devices. As a consequence, the communication latency represents the time an individual message requires to be delivered from the sending buffer of a host to the receiving buffer of another host.

Transmission time is the time required to transmit a message over a network link with a specific throughput, whereas processing time represents the time required for a device to send, forward, or receive a message. For intermediate network devices like routers and switches the processing time depends on queuing delay and forwarding delay. For the sender and receiver of a message the processing time consists of enqueue and dequeue delays, cryptographic overhead, and message coding.

To support Ethernet-based SAS protocols such as GOOSE and SV, communication latency constraints for each individual data frame of the data link layer must be taken into account. The composition of the end-to-end latency of Ethernet-based communication is visualized in Figure 4.2. In Figure 4.2a, the communication latency is composed of time for encoding (t_E), time for transmission to another host or intermediate system (t_T), time for switching in an intermediate system (t_S), and time for decoding (t_D). In Figure 4.2b, device-local data verification is employed, which leads to additional processing time required for signing (t_{Sig}) and verification (t_{Ver}).



(a) No data verification.

(b) Device-local data verification.

Figure 4.2: Composition of the end-to-end latency using Ethernet-based communication.

4.1.2.2 Message Types

The defined message types of the presented system model are based on the classification characteristics defined above. Furthermore, the defined message types have been adapted from the message types and performance classes of the IEC 61850 standards [3]. The defined message types as well as their typical communication topology, continuity, and latency constraints are shown in Table 4.1.

The low latency message type corresponds to the IEC 61850 [3] message types 1A and 4. These messages are used for SAS-internal exchange of sampled values and state values. In IEC-compliant substations, the sampled values are exchanged via multicast using the SV protocol between MUs and IEDs (vertical) or between MUs (horizontal). Moreover, state values and state changes are exchanged between IEDs (horizontal) using the GOOSE protocol.

Table 4.1: Message types of the presented system model.

Message Type	Topology			Continuity	Latency
	Externality	Verticality	Receiver		Constraint
Low Latency	Internal	Horiz./Vert.	Multicast	Message-Based	3 ms
Medium Latency	Int./Ext.	Horiz./Vert.	Unicast	Session-Based	20-100 ms
High Latency	Int./Ext.	Horiz./Vert.	Unicast	Session-Based	500 ms

The medium latency message type corresponds to the IEC 61850 message types 1B and 2. These messages are used for SAS-internal and SAS-external, as well as horizontal and vertical session-based client-server communication. In IEC-compliant substations, IEDs use the MMS protocol to communicate with other IEDs and higher-level devices.

The high latency message type corresponds to the IEC 61850 message types 3 and 5. This message type is used for HMI interactions as well as non-time-critical operations like file transfers. In IEC-compliant substations, MMS as well as SCADA protocols are used for high latency communication.

4.1.2.3 Communication Buses

The presented system model uses a bus-based approach for message exchange within and between the system architecture layers. The realization of SAS-internal buses is typically based on Ethernet, and on open or proprietary fieldbus technology. The bus-based approach as well as the two specific buses introduced in the following are based on the IEC 61850 standards [3].

The first bus for SAS-internal message exchange is referred to as process bus. The process bus is located between the bay level and the process level. The process bus is used for time-critical message-based publisher-subscriber communication, i.e., multicast or broadcast communication. GOOSE and SV are the protocols used for process bus communication.

The second bus for SAS-internal message exchange is referred to as station bus. The station bus is located between the station level and the bay level. The station bus connects IEDs at the bay level with each other as well as with gateways and interfaces at the station level. The communication at the station bus is typically session-based unicast communication with less strict time requirements compared to the process bus.

SAS-external message exchange between devices on the station level and network level use WAN telecommunication technologies including Internet, satellite, cellular, and radio technology. Secure tunneling approaches like Virtual Private Networks (VPN) can be used to enhance the security of SAS message exchange over an unsecure communication medium.

4.2 Requirements

In the following, we introduce the requirements of the presented approach. Based on the identified requirements, functional and non-functional characteristics of the proposed approach are derived and evaluated. Each requirement is associated with a requirement category. We define five requirement categories for the introduced system requirements. The requirement categories consist of security (RQ.SEC), safety (RQ.SAF), availability (RQ.AVA), performance (RQ.PER), and compatibility (RQ.COM).

4.2.1 Security

RQ.SEC.1 Data Frame Payload Integrity
A SAS device detects unauthorized manipulation of data frames that are exchanged between itself and another device.

RQ.SEC.2 Data Frame Sender Authenticity
Each SAS device can prove the authenticity and trustworthiness of a sender of a data frame.

RQ.SEC.3 Data Frame Authorship Non-Repudiation
A SAS device cannot dispute its authorship of a data frame sent.

RQ.SEC.4 Access Control
The system prohibits unauthorized access to sensitive information stored on devices.

RQ.SEC.5 Principle of Least Privilege (PoLP)
The system ensures that each subject has the least number of privileges necessary to perform its function [22].

RQ.SEC.6 Separation of Duties (SoD)
The system ensures that no subject has enough privileges to be able to misuse the system without collusion [22].

4.2.2 Safety

RQ.SAF.1 Safe Operation
Under possible operating conditions, the system must not pose a threat to itself and its environment.

RQ.SAF.2 Fail-Safe
In case of failure, the system terminates without causing harm to the system or system environment [58]. In other words, the system never transitions into an unsafe state.

4.2.3 Availability

RQ.AVA.1 Continuing Operation
Under possible operating conditions, the system must continue its operation as stated by the system requirements.

RQ.AVA.2 Fail-Operational
In case of failure, the system aims to continue its operation by selectively terminating failing system functions. The selective termination of non-essential system functions in case of a failure is also referred to as fail-soft [58].

4.2.4 Performance

RQ.PER.1 Constrained Data Frame Delivery Time
The latency constraints for network communication, as defined in subsubsection 4.1.2.2, must be satisfied. To support Ethernet-based SAS protocols, the approach must be able to satisfy time constraints for each individual data frame of the data link layer.

RQ.PER.2 Constrained Computational Performance
The limited performance of resource-constrained devices of an SAS must be taken into account. Consequently, computationally complex algorithms must be executed by performance-oriented TTPs.

RQ.PER.3 Constrained Energy & Power
The limited energy and power of resource-constrained devices of an SAS must be taken into account. Consequently, energy-intensive tasks, such as long-running computations, or power-intensive tasks, such as tasks leading to high CPU loads, must be executed by performance-oriented TTPs.

4.2.5 Compatibility

RQ.COM.1 Interoperability
The system components are capable of exchanging information and providing services, irrespective of whether they originate from a single vendor or multiple vendors [3].

RQ.COM.2 Interchangeability
The system's behavior and functionality may not be influenced by an exchange of devices with an equal range of functions from a single vendor or multiple vendors [3].

4.3 Adversarial Attacks

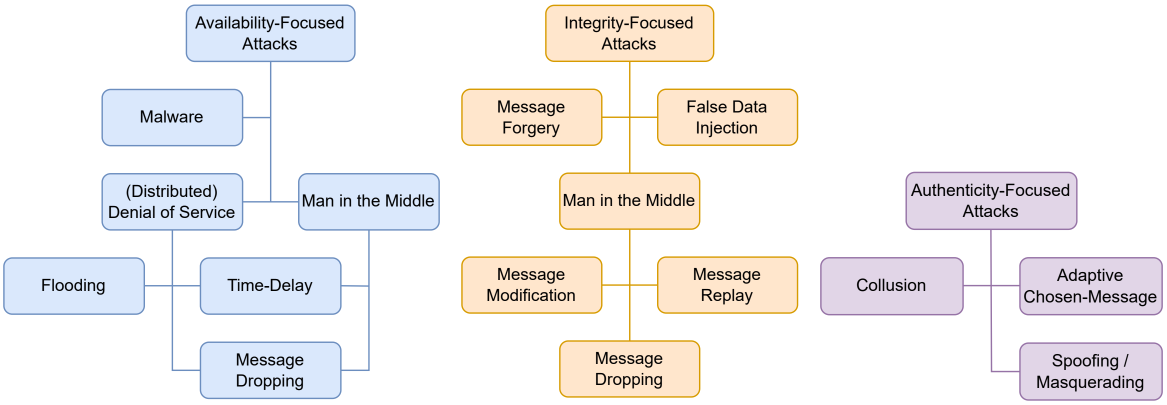
In addition to the aforementioned requirements, this section introduces an adversary model and provides an enumeration and classification of cyberattacks. While not being exclusively relevant to SAS, the presented adversaries and attacks pose a threat to the state and operation of systems that correspond to the system model defined in section 4.1. A plethora of different threats, adversaries, and cyberattack classifications applicable to SCADA, SAS, ICS, or smart grid systems are discussed in the literature [59, 60, 61, 62, 63, 64]. In the following paragraphs, these concepts are aggregated and transferred to the area of application of the CASC-SAS approach.

For the purpose of design, realization, and evaluation of the CASC-SAS approach, we assume an adversary corresponding to the Dolev-Yao model [65]. The Dolev-Yao adversary is adapted to the SAS-specific network characteristics, including the usage of two separated buses for the exchange of messages. Based on the adversary classification presented by Hof [66] and Ponikwar et al. [67], the defined Dolev-Yao-like CASC-SAS adversary is a malicious, global, cooperative, dynamic, active, insider adversary. Accordingly, a possible CASC-SAS adversary is characterized by five assumptions:

1. The adversary has physical or remote access to at least one of the internal SAS networks, i.e., the process bus or the station bus.
2. The adversary is able to initiate arbitrary message exchanges with any device on the networks.
3. The adversary is able to receive messages from any device on the networks.
4. The adversary is able to capture, alter, and drop messages exchanged on the networks.
5. The adversary is unable to bypass or break cryptographic procedures without first obtaining the necessary key material.

The types of cyberattacks that a CASC-SAS adversary can carry out are visualized in Figure 4.3. Each cyberattack is classified based on the security objective affected. The objective of availability-focused attacks, as illustrated in Figure 4.3a, is to disrupt system services in a manner that renders the continuation of operations impossible. In contrast, integrity-focused attacks, as illustrated in Figure 4.3b, aim to disturb the system's integrity by transitioning the system into a state that is either invalid or beneficial for an adversary. The authenticity-focused attacks shown in Figure 4.3c enable an adversary to impersonate a legitimate subject of the system, thereby abusing the subject's granted privileges.

In addition to the classification based on the affected security objective, we propose a classification of potential cyberattacks based on the adversary's objective. These objectives are modeled as attack trees, as the objective of an adversary is typically unachievable by a single operation but rather requires a sequence of attacks to be achieved. This modeling approach allows for a more comprehensive understanding of the adversary's strategy and the potential avenues for defense. With regard to the CASC-SAS approach, two primary objectives of an adversary were identified. The first objective is an attack against the SAS



(a) Availability-focused attacks. **(b)** Integrity-focused attacks. **(c)** Authenticity-focused attacks.

Figure 4.3: Classification of adversarial attacks based on security objectives.

network communication, i.e., against the SAS protocols. The second objective is an attack against SAS devices. Accordingly, two independent attack trees are provided in Figure 4.4 and Figure 4.5. The colors utilized for the visualization of attacks within the attack trees correspond to the colors utilized in Figure 4.3 and, thus, represent the affected security objective.

The first attack tree, as illustrated in Figure 4.4, represents an adversary with the objective of compromising the system by disrupting communication between two or more legitimate system subjects. In order to achieve this objective by compromising the communication integrity, the adversary may either replay messages that have been captured on the network, or modify them. While replaying only requires the adversary to be able to eavesdrop on communication, message modification additionally requires the adversary to masquerade as a legitimate subject. This can be achieved by either breaking authenticity and integrity protection mechanisms, such as digital signatures, or colluding with other adversaries, such as infiltrated system devices. Furthermore, an adversary may disrupt communication via (distributed) denial of service (DoS) attacks. Examples of DoS attacks in a SAS include the intentional delay of time-critical messages, also known as time-delay attacks [68], and dropping of messages.

The second attack tree, as illustrated in Figure 4.5, represents an adversary with the objective of compromising the system by directly attacking the system devices. An adversary may seek to either disrupt the device's availability or compromise its integrity. An adversary may achieve disruption of the device's availability via the deployment of malware or (distributed) DoS attacks, such as flooding attacks. A device's integrity can be compromised by modifying its state using accessible service interfaces. Consequently, an adversary may either create a legitimate request, in case of an unsecured service, or forge a request. As discussed above with regard to message modification, request forgery might require an adversary to masquerade as a legitimate subject to circumvent authenticity and integrity protection mechanisms.

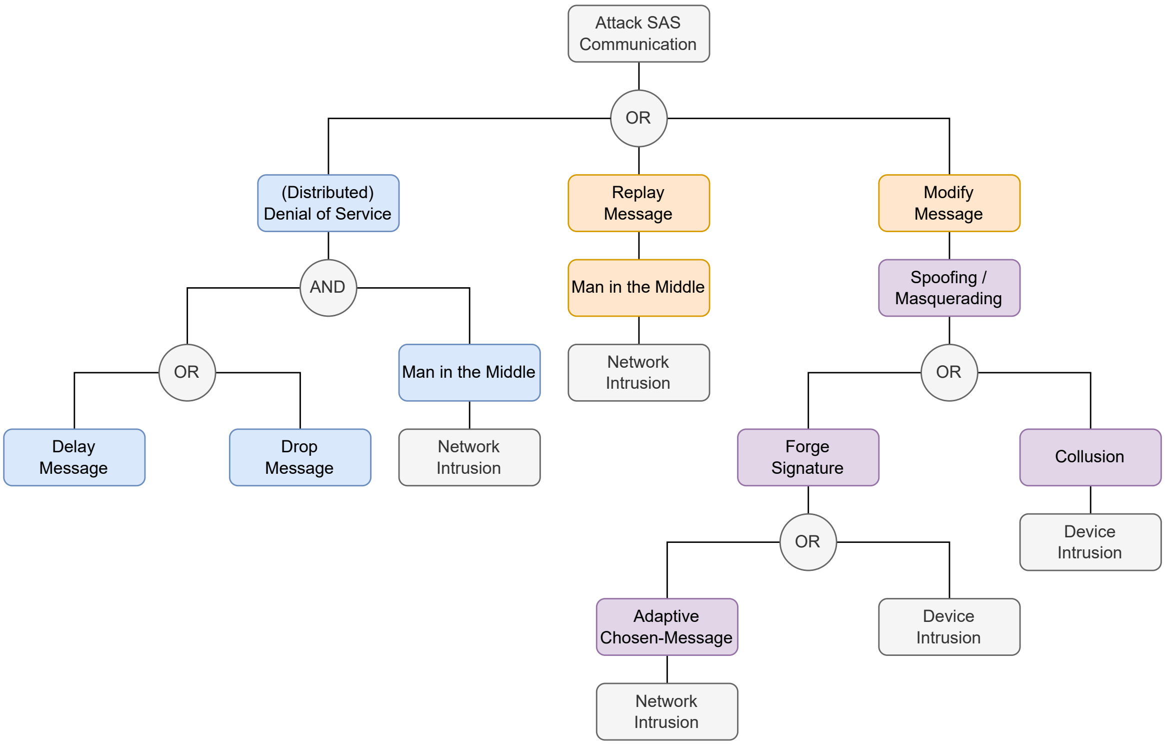


Figure 4.4: Attack tree comprising cyberattacks that endanger SAS message exchange protocols.

4.4 Security Policies

The CASC-SAS approach enforces a set of security policies with the purpose of satisfying the aforementioned requirements and defending a SAS against cyberattacks. The objective of these policies is not merely to safeguard the equipment of the CASC-SAS approach, but rather to ensure the continuous operation of SAS devices, including IEDs and MUs. The security policies represent mandatory rules for the operation of a SAS secured by the CASC-SAS approach. While the attack trees discussed in section 4.3 provide insight into potential adversarial strategies, the proposed security policies represent mitigation strategies for potential cyberattacks. The cyberattacks that can be mitigated by enforcing the security policies are shown in Table 4.2. The classification of the cyberattacks is based on the classifications illustrated in Figure 4.3, Figure 4.4, and Figure 4.5. Moreover, the security requirements, as introduced in section 4.2, that can be satisfied by enforcing the security policies are shown in Table 4.3.

Policy I: Data Frame Signing and Verification

The authenticity, integrity, and non-repudiation of exchanged messages on SAS networks are safeguarded by employing cryptographic mechanisms for signing and verification, i.e., digital signatures if PKC is used or MAC if SKC is used. This security policy is enforced for each individual data frame of the data link layer to support Ethernet-based SAS protocols such as GOOSE and SV. The integration of signing and verification into the OSI protocol stack of two generic Ethernet-based SAS applications

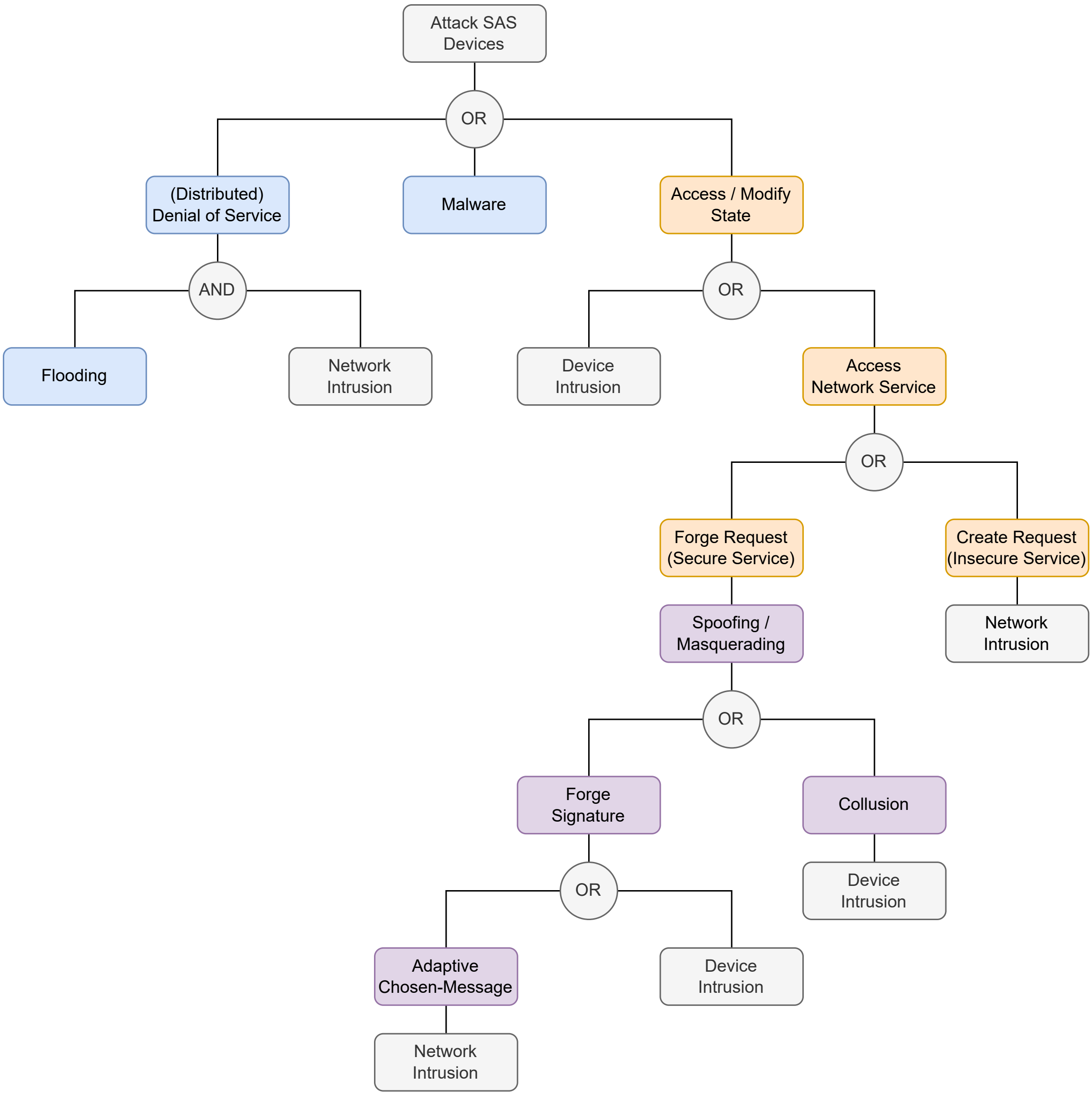


Figure 4.5: Attack tree comprising cyberattacks that endanger the functionality of SAS devices.

within a switched LAN is visualized in Figure 4.6. In the shown protocol stack, signing and verification reside within the presentation layer directly below an Ethernet-based application, e.g., a GOOSE or SV application.

Policy II: Data Frame Access Control
Unauthorized access to service interfaces provided by SAS devices is prevented by employing access control. As discussed above, this security policy is enforced for each individual data frame of the data link layer to support Ethernet-based SAS protocols. In other words, CASC-SAS checks if a data frame is authorized before delivering it to a SAS application.

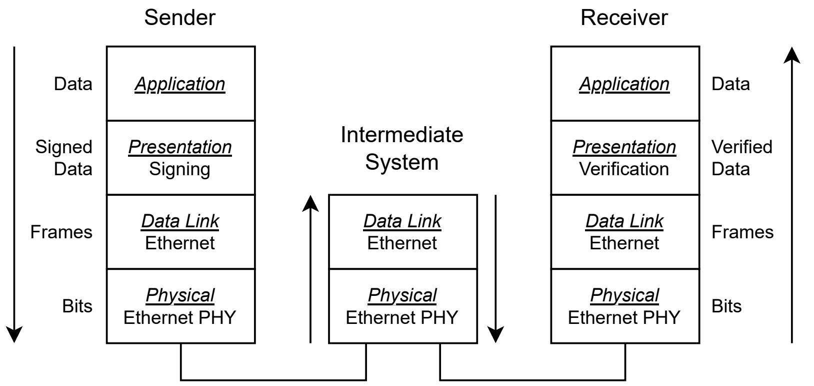


Figure 4.6: Protocol stack of Ethernet-based SAS applications using data verification.

Policy III: Flow-Based Time-Dependent Authorization

Authorization mechanisms are deployed to specify the privileges of communicating entities in a SAS and, thus, enable access control. Since a SAS environment is non-static with regard to its state, authorizations have to be time-dependent as well. Moreover, authorizations have to be fine-grained and flexible to satisfy the PoLP and SoD security requirements. Nevertheless, authorizations have to be network-traffic-flow-based, i.e., applicable to any sequence of data frames with common properties passing through the SAS network, to reduce evaluation overhead and management overhead.

Policy IV: Data Frame Sequencing

Sequencing of exchanged messages is used to safeguard the system's integrity by mitigating intentional or accidental re-ordering, replaying, and delaying of messages. For this purpose, a timestamp is appended to each individual data frame sent. A receiver may utilize the timestamp for two distinct purposes. Firstly, the receiver interprets the timestamp as sequence number to reject re-ordered and replayed data frames of a network traffic flow. Secondly, the receiver may use the timestamp to calculate the average communication latency and its standard deviation to detect infrequent delaying of data frames. For the latter case, no time synchronization between sender and receiver is required, as only the deviation from the average is relevant for the classification of delayed data frames.

Policy V: Flow-Based Ingress Buffer Management

Congestions of exchanged messages in network ingress buffers of SAS devices have to be avoided to mitigate intentional or accidental DoS due to message flooding. CASC-SAS employs flow-based buffer management to avoid message congestions. The flow-based buffer management limits the number of processed messages of a network traffic flow within a specific time interval. If the processing limit is reached, a receiver may either reject received data frames belonging to the network traffic flow or replace buffered data frames with received ones.

Table 4.2: Adversarial attacks mitigated by CASC-SAS security policies.

Adversarial Attack	Classification		Policy				
	Security Objective	Adversarial Objective	I	II	III	IV	V
Malware	Availability	Device		X	X		
Flooding	Availability	Device					X
Time-Delay	Availability	Communication				X	
False Data Injection	Integrity	Dev. / Comm.	X	X	X		
Message Forgery	Integrity	Device	X				
Message Modification	Integrity	Communication	X				
Message Replay	Integrity	Communication				X	
Spoofing / Masquerading	Authenticity	Dev. / Comm.	X				
Adaptive Chosen-Message	Authenticity	Dev. / Comm.	X				
Collusion	Authenticity	Dev. / Comm.	X				

Table 4.3: Security requirements satisfied by CASC-SAS security policies.

Requirement		Policy				
		I	II	III	IV	V
RQ.SEC.1:	Payload Integrity	X			X[1]	X[2]
RQ.SEC.2:	Sender Authenticity	X				
RQ.SEC.3:	Authorship Non-Repudiation	X				
RQ.SEC.4:	Access Control		X			
RQ.SEC.5:	Principle of Least Privilege			X		
RQ.SEC.6:	Separation of Duties			X		

[1] Prevents re-ordering, replaying, and delaying of valid messages.

[2] Prevents loss of valid messages due to message flooding.

4.5 Security Architecture

In the following, we present the security architecture of the proposed approach. The CASC-SAS approach is based on a dual-path four-layered architecture. The four layers of the architecture are presented in subsection 4.5.1. Moreover, the two paths of the architecture are further discussed in subsection 4.5.2.

4.5.1 Four-Layered Architecture

The CASC-SAS architecture is non-strictly layered and consists of four open layers. The goal of the layered architecture is the separation of different domains and levels of abstraction within the CASC-SAS approach. An upper layer may use services provided by a lower layer but not vice versa. Moreover, since the layering is non-strict, an upper layer is not restricted to the services provided by its direct predecessor, but may bypass lower layers. The four layers of the CASC-SAS architecture and their provided services are defined in the following sections.

4.5.1.1 Layer 3: Domain

The domain layer is the uppermost layer of the architecture. The domain layer represents the domain-specific applications and the exchange of domain-specific messages. We assume that the domain layer does not provide means for secure message exchange between entities. As a consequence, the domain layer relies on the secure message exchange provided by lower layers.

4.5.1.2 Layer 2: Cybersecurity

The cybersecurity layer encompasses algorithms and protocols used to satisfy the security requirements. Additionally, security workflows and mechanisms for the enforcement of security policies are located at this layer. Consequently, the cybersecurity layer provides secure message exchange services to the domain layer. The SABAAC concept of CASC-SAS is part of this architectural layer. SABAAC provides authorization and access control to satisfy the security requirements access control, PoLP, and SoD.

4.5.1.3 Layer 1: Cryptography

The cryptography layer provides cryptographic algorithms and schemes to higher layers of the architecture. The exchange of cryptographic control messages enables cryptographic workflows such as key generation, key distribution, key revocation, and server-aided cryptography. The CASA core concept of CASC-SAS is located at the cryptography layer. CASA provides authentication means via digital signatures to higher levels of the architecture. Accordingly, CASA provides services that satisfy the security requirements integrity, authenticity, and non-repudiation.

4.5.1.4 Layer 0: Message Exchange

The lowermost layer of the CASC-SAS architecture is referred to as message exchange layer. The message exchange layer provides reliable and unreliable message exchange between devices in a network to higher layers. The message exchange layer represents an abstraction of the physical layer, data link layer, network layer, and transport layer of a conventional OSI network stack.

4.5.1.5 Example: Domain-Specific Communication

An exemplary domain-specific communication between a sending entity and receiving entity is shown in Figure 4.7. The figure shows the four layers of the CASC-SAS architecture at the sender and receiver. Moreover, the different messages exchanged between the layers are shown. While the invocation of services is restricted to predecessor layers, message

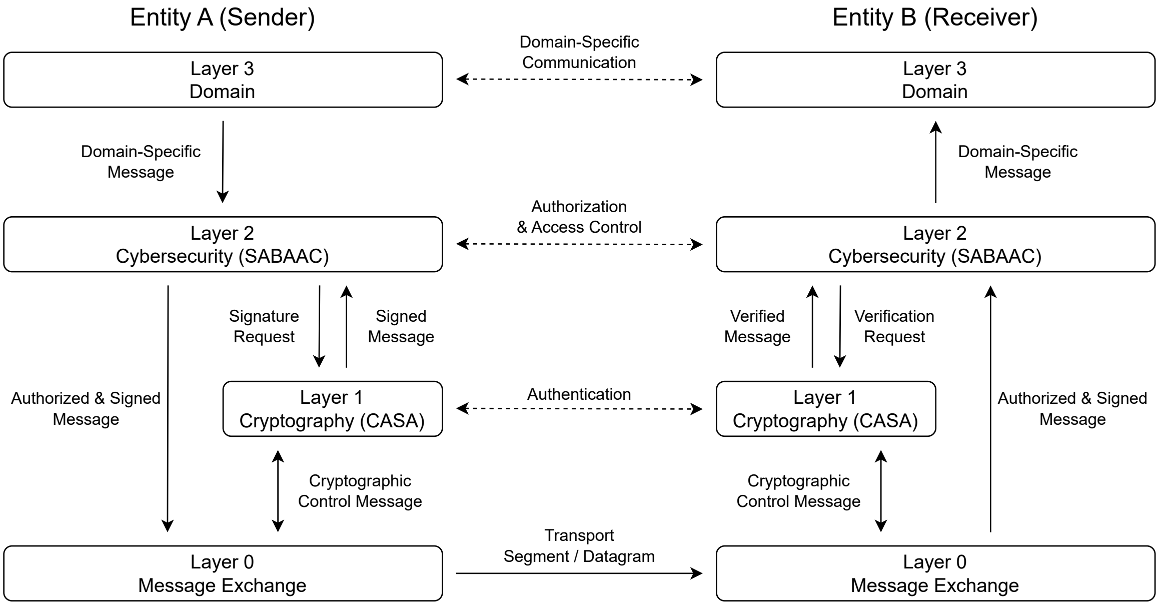


Figure 4.7: Exemplary message exchange in four-layered CASC-SAS architecture.

exchange resulting from an invocation may occur bidirectionally. The presented domain-specific communication is initiated by entity A. Therefor, an application at the domain layer of entity A creates a domain-specific message and delivers it to the cybersecurity layer. The yet unsigned and non-authorized message is then authorized by SABAAC and forwarded to CASA at the cryptography layer for signing. Subsequently, the signed and authorized message is forwarded to the receiver using the either reliable or unreliable network transport services provided by the message exchange layer. Upon arrival at the receiver, the message exchange layer delivers the signed and authorized message to the cybersecurity layer. The messages are then verified by the cybersecurity layer before forwarding them to the domain layer and application. For the purpose of message verification, the cybersecurity layer enforces the CASC-SAS security policies. The access control policy is enforced by verifying the message authorization. Moreover, the message is forwarded to the cryptography layer for digital signature verification.

4.5.2 Dual-Path Architecture

In addition to the separation into different layers, the occurring message exchanges within the CASC-SAS architecture are logically divided into two communication paths. The two paths are referred to as data path and control path.

The messages on the data path are directly related to the forwarding of domain-specific payload from a sending entity to a receiving entity. Besides the domain-specific messages, control messages required for the message forwarding are transported on the data path.

This message-related communication includes server-aided signing and verification requests as well as access control. As a consequence, the data path is used for traffic-intensive and time-critical message exchange.

The messages on the control path are used for the exchange of management information and do not carry payload that is directly related to domain-specific messages. The components of the CASC-SAS approach use control path messages for layer-internal communication between different devices. The cryptography layer uses control messages for key generation, distribution, and revocation. The cybersecurity layer uses control messages for tasks such as policy management. As a result, the communication occurring on the control path is less traffic-intensive and less time-critical.

4.6 Certificateless Attribute-Based Server-Aided Authentication

In the following section, we present the **C**ertificateless **A**ttribute-Based **S**erver-Aided **A**uthentication (CASA) concept. CASA is a CL-PKC approach. Consequently, neither certificates nor key escrow are required [38]. The goal of CASA is to provide cryptographic protocols, algorithms, and schemes for key generation, distribution, and revocation as well as signing and verification. Moreover, the goal of CASA is to enable and support more abstract cybersecurity mechanisms like authorization and access control of the CASC-SAS approach. Therefore, CASA represents the foundation of the employed CASC-SAS cybersecurity mechanisms.

4.6.1 Administration & Processing Platform

The CASA Administration and Processing Platform (CAPP) represents the central component of the CASA approach. The main objective of the CAPP is the provisioning of services required for the realization of cryptographic protocols. While CASA also provides its own signature scheme $\mathcal{S}_{CASA}$, which is further discussed in subsection 4.6.3, the CAPP and its protocols are algorithm-agnostic. The use of an algorithm-agnostic central component in the CASA approach is based on the idea, that different cryptographic algorithms and schemes might be optimal solutions for different problems. This idea is inspired by the cipher suites and cipher transitioning of the TLS protocol [46]. The support of different algorithms and schemes enables devices to choose the cryptographic approach, that fits their security requirements and performance constraints best.

4.6.1.1 Server-Aided Cryptography

As PKC mechanisms may consist of computationally complex algorithms and operations, a core task of the CAPP is to enable and support server-aided PKC. By supporting server-aided PKC, CASA encourages the utilization of server-aided CL-PKC schemes for time-critical applications. To make CASA server-aided, the CAPP supports devices by handling computationally expensive algorithms instead of executing them locally on resource-constrained devices. To minimize the required trust, the CAPP may only handle certain computations, e.g., partially sign or verify a request of a device. This server-aided approach enables resource-constrained devices to apply secure PKC algorithms and schemes in a time-critical OT environment.

In the following, we employ the concept of server-aided PKC for the verification process. As stated by Wu et al. [69], a server-aided verification process has to satisfy the property of being computation-saving. A server-aided verification process V_{Aided} is computation-saving if the computational costs for the verifier are strictly less than the costs of non-server-aided verification $V_{Conventional}$. In other words, V_{Aided} is computation-saving if the equation $Cost(V_{Aided}) < Cost(V_{Conventional})$ holds.

4.6.1.2 Online & Offline Cryptography

Since CASA is tailored for time-critical communication, the approach aims to reduce the required time for cryptographic algorithms. In addition to server-aided cryptography, this time reduction is achieved by precomputation. For this purpose, each step of an algorithm is classified as either online or offline. Online steps depend on the sender's public key, the digital signature, or the message. Consequently, online steps cannot be precomputed. Nevertheless, specific online steps can be accelerated via server-aided cryptography. Offline steps depend on information that is available before any message exchange occurs. Therefore, offline steps can be precomputed either at the CAPP or at client-side to reduce the required time for cryptographic operations.

4.6.2 Algorithm-Agnostic Public-Key Exchange Protocol

The Algorithm-agnostic PKC EXchange (APEX) protocol represents the primary protocol of CASA. The main objective of the protocol is the exchange of information between the CAPP and other devices within a SAS. An APEX message exchange is always initiated by sending a request to the CAPP. Consequently, the protocol can be classified as a request-response client-server protocol. Furthermore, the protocol is based on a stateless communication pattern that establishes no sessions at the CAPP. In other words, a message exchange is completed as soon as the APEX reply is sent to the requestor.

The APEX protocol comprises five so-called transactions. A transaction represents a request-response pair of the protocol. The transactions of the protocol are defined as follows:

Transaction I: Registration

A device within a SAS sends a registration request to establish its identity and its initial public key at the CAPP. For this purpose, the device encapsulates its public key and its identifying attributes in the registration request.

Transaction II: Re-Registration
The process of linking a new public key to existing attributes, or modifying an already registered public key is referred to as re-registration. By issuing a re-registration, devices may register multiple public keys, e.g., for different cryptographic algorithms. A re-registration request has to be verifiable with one of the registered public keys of a device. If the re-registration request cannot be verified, it is rejected by the CAPP.

Transaction III: Revocation
The revocation transaction allows devices to revoke a registered public key by removing it from the CAPP. The revocation is referred to as deregistration, if it is used to revoke the last registered public key of a device.

Transaction IV: Query
The query transaction allows devices to retrieve a specific public key from the CAPP. To identify the requested public key, the sender of the request has to encapsulate the identifying attributes of the key owner.

Transaction V: Computation
A device may issue a computation request in order to make use of server-aided operations provided by the CAPP. Besides the operation-dependent parameters, the request must contain an identification of the requested operation.

4.6.3 Signature Scheme $\mathcal{S}_{CASA}$

The CASA signature scheme $\mathcal{S}_{CASA} = (I, G_{PPK}, G_{SK}, S_i, S_{AGG}, V_{SAV}, V_{ENT})$ is a seven-tuple of algorithms. The algorithms comprise an initialization algorithm I, a partial private key generation algorithm G_{PPK}, a private key generation algorithm G_{SK}, a signing algorithm S_i, a signature aggregation algorithm S_{AGG}, a partial server verification algorithm V_{SAV}, and an entity verification algorithm V_{ENT}. In the following sections, the specific algorithms are further discussed.

4.6.3.1 Initialization Algorithm I

The initialization algorithm $(\rho, s) \leftarrow I(\lambda)$ takes the security parameter λ as input and outputs the public system parameters ρ and the master secret s. The initialization algorithm is executed by the CAPP. After the execution, ρ is publicly available to all entities, whereas s is only known to the CAPP. The initialization algorithm consists of the following steps:

1. Define the bilinear groups G_1, G_2, and G_T of prime order q, generators $g_1 \in G_1$ and $g_2 \in G_2$, and a bilinear pairing e:

   $$e : G_1 \times G_2 \longrightarrow G_T$$

   A map e is a bilinear pairing if it fulfills the following properties:

   a) Bilinearity: $e(P^a, Q^b) = e(P, Q)^{ab}$ for all $P \in G_1, Q \in G_2$ and $a, b \in \mathbb{Z}_q^*$.

   b) Non-Degeneracy: $e(P, Q) \neq 1$ for $P \in G_1, Q \in G_2$

   c) Computability: There is an efficient algorithm to compute $e(P, Q)$ for any $P \in G_1, Q \in G_2$

2. Define the cryptographic hash functions H_1, H_2, and H_3:

$$H_1 : \{0,1\}^* \longrightarrow G_1, \quad H_2 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*, \quad H_3 : \mathbb{Z}_q^* \longrightarrow G_2.$$

3. Generate the master secret s and compute the master public key pk_m:

$$s \in \mathbb{Z}_q^*, \quad pk_m = g_2^s.$$

4. Publish the public system parameters ρ:

$$\rho = (G_1, G_2, G_T, q, e, g_1, g_2, pk_m, H_1, H_2, H_3).$$

4.6.3.2 Partial Private Key Generation Algorithm G_{PPK}

The partial private key generation algorithm $ppk_i \longleftarrow G_{PPK}(\rho, s, ID_i, ATT_i)$ takes the public system parameters ρ, the master secret of the CAPP s, the identifier ID_i of entity A_i, and the attributes ATT_i of entity A_i as input. The algorithm outputs the partial private key ppk_i of entity A_i. The partial private key generation is executed by the CAPP on request of entity A_i. After the execution, the CAPP provides the partial private key to the corresponding entity. The partial private key generation algorithm consists of the following steps:

1. Compute the partial private key ppk_i on request of entity A_i:

$$ppk_i = H_1(ID_i || ATT_i)^s.$$

2. Provide the partial private key ppk_i to entity A_i.

4.6.3.3 Private Key Generation Algorithm G_{SK}

The private key generation algorithm $sk_i \longleftarrow G_{SK}(\rho, ppk_i)$ takes the public system parameters ρ, and the partial private key ppk_i of entity A_i as input. The algorithm outputs the private signing key sk_i of entity A_i. The private key generation algorithm consists of the following steps:

1. Generate a random secret value χ_i:

$$\chi_i \in \mathbb{Z}_q^*.$$

2. Set the private signing key sk_i:

$$sk_i = (ppk_i, \chi_i).$$

4.6.3.4 Signing Algorithm S_i

The signing algorithm $\sigma_i \leftarrow S_i(\rho, sk_i, m, T)$ takes the public system parameters ρ, the private signing key sk_i of entity A_i, a message m, and an access policy T as input, and outputs the signature σ_i. In other words, the signing algorithm S_i is used by the sender A_i of a message m to generate a digital signature σ_i. The generated digital signature σ_i is associated with the message m, the sender's private signing key sk_i, and an access policy T. The signing algorithm consists of the following steps:

1. Check if the attributes ATT_i of entity A_i satisfy the policy T, and abort the process otherwise.
2. Compute the hash h of the message m:

$$h = H_2(m||T).$$

3. Compute the signature σ_i using the private signing key sk_i:

$$\sigma_i = ppk_i \cdot H_3(h)^{x_i}.$$

4.6.3.5 Signature Aggregation Algorithm S_{AGG}

The signature aggregation algorithm $\sigma_{agg} \leftarrow S_{AGG}(\sigma_{1\ldots n})$ takes n signatures $\{\sigma_i | 1 \leq i \leq n\}$ as input. The algorithm outputs an aggregated signature σ_{agg}:

$$\sigma_{agg} = \prod_{i=1}^{n} \sigma_i.$$

4.6.3.6 Partial Server Verification Algorithm V_{SAV}

The partial server verification algorithm $P_{CAPP} \leftarrow V_{SAV}(\rho, s, \sigma_{agg}, ID_{1\ldots n}, ATT_{1\ldots n})$ takes the public system parameters ρ, the master secret of the CAPP s, an aggregated signature σ_{agg}, the entity identifiers $\{ID_i | 1 \leq i \leq n\}$, and the entity attributes $\{ATT_i | 1 \leq i \leq n\}$ as input. The algorithm outputs the partial verification P_{CAPP}. The partial server verification algorithm consists of the following steps:

1. Compute the aggregated public key pk_{agg}:

$$pk_{agg} = \prod_{i=1}^{n} H_1(ID_i||ATT_i)^s.$$

2. Compute the partial verification P_{CAPP}:

$$P_{CAPP} = e(pk_{agg} \cdot \frac{\sigma_{agg}}{pk_{agg}}, g_2).$$

4.6.3.7 Entity Verification Algorithm V_{ENT}

The entity verification algorithm $\delta \in \{accept, reject\} \longleftarrow V_{ENT}(\rho, \sigma_{agg}, P_{CAPP})$ represents the final step of the verification process. The algorithm takes the public system parameters ρ, the aggregated signature σ_{agg}, and the partial verification P_{CAPP} as input. The algorithm outputs the verification decision δ which is either *accept* or *reject*. In other words, the verification algorithm V_{ENT} is used by a message receiver to verify a message m sent by entity A_i based on an appended signature σ_{agg}. As σ_{agg} is associated with the message m and the sender's private signing key sk_i, it allows the receiver to verify the integrity and authenticity of the received message m. The entity verification algorithm consists of the following steps:

1. Compute the bilinear pairing P_{Entity}:

$$P_{Entity} = e(\sigma_{agg}, g_2).$$

2. Accept m if the following equation holds and reject otherwise:

$$P_{Entity} \overset{!}{=} P_{CAPP}$$

4.7 Server-Aided Attribute-Based Authorization & Access Control

The second core concept of the CASC-SAS approach is the **S**erver-Aided **A**ttribute-**B**ased **A**uthorization and **A**ccess **C**ontrol (SABAAC). The SABAAC approach enables the employment of attribute-based authorization and access control for time-critical SAS communication. Therefore, the approach prevents unauthorized access and extraction of information. The approach enables CASC-SAS to satisfy the access control, PoLP, and SoD security requirements. Moreover, the expressive and flexible yet computationally expensive ABAC policies are handled in a server-aided manner to satisfy the strict time constraints of the SAS domain.

Our authorization and access control approach represents a security concept that is located on the cybersecurity layer of the CASC-SAS architecture. Thus, it relies on secure authentication services provided by CASA. As a consequence, the approach assumes that efficient and secure signing and verification algorithms are available. In other words, CASA provides secure cryptographic algorithms and schemes that enable SABAAC to realize secure authorization and access control.

The proposed authorization and access control approach is based on a function-oriented component-based architecture. The architecture and components are further discussed in subsection 4.7.1. Furthermore, the approach is divided into two central tasks. The first task is referred to as delegated attribute-based authorization. The delegated attribute-based authorization is responsible for the access control policy creation, management, storage,

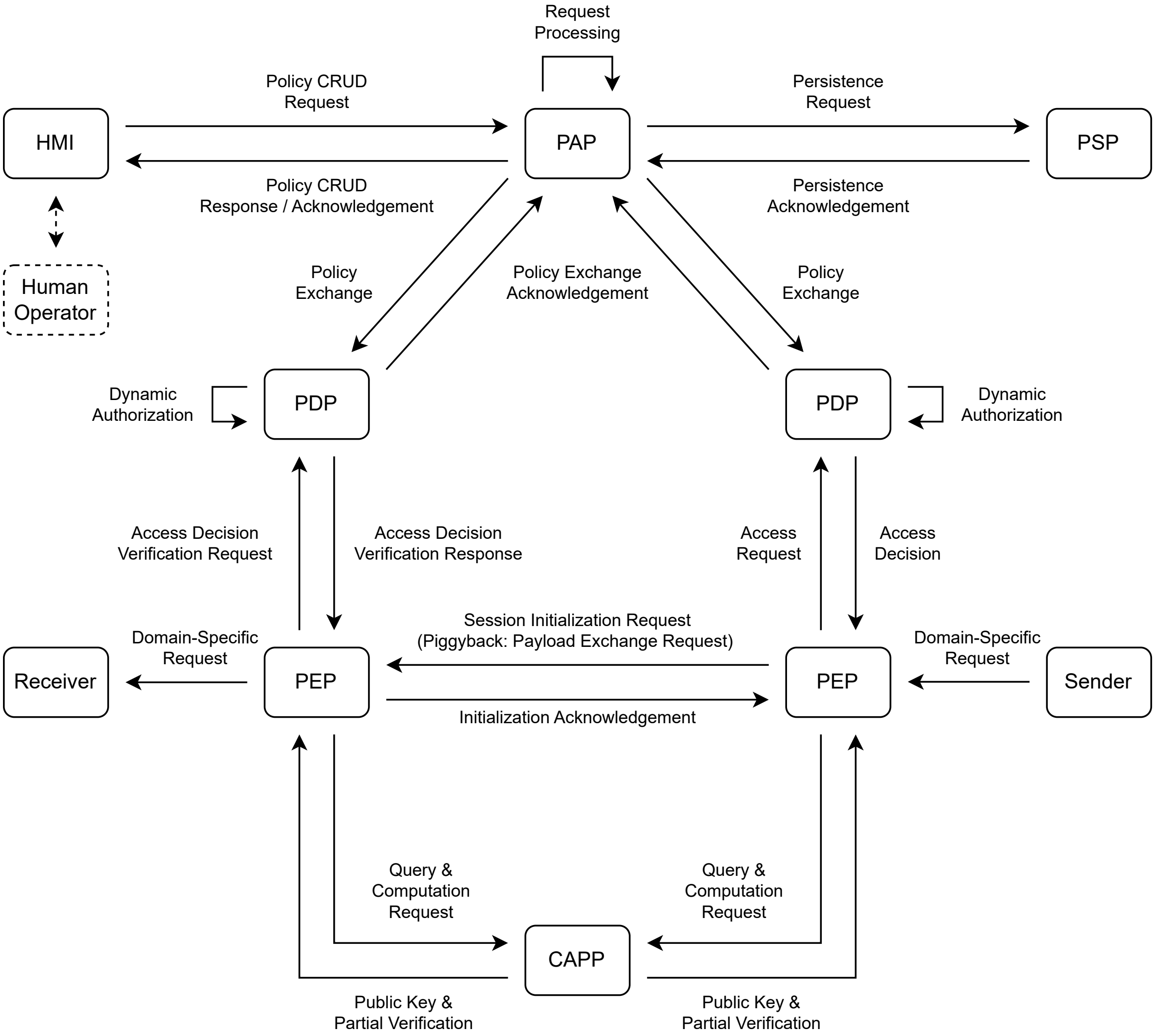


Figure 4.8: Function-oriented component-based architecture of the SABAAC approach.

and distribution. This task partially takes place prior to access requests and corresponding access decisions. The delegated attribute-based authorization protocol is further discussed in subsection 4.7.3. The second central task is referred to as delegated ABAC. The delegated ABAC is responsible for the policy decision exchange and policy enforcement. This task takes place when an entity initiates the communication with another entity. The delegated ABAC protocol is further discussed in subsection 4.7.4. An overview of the SABAAC architecture, components, and protocols is shown in Figure 4.8.

4.7.1 Authorization & Access Control Architecture

The component-based architecture of our authorization and access control approach consists of four functional units. These functional units have been adapted from the access control mechanism functional points presented by Hu et al. [29]. Each functional unit is represented by a component that offers a set of services. The components of the architecture are TTPs since the semantic validity of their provided services is not verifiable by the service consumers. Strategies to reduce the required trust in specific components, including the deployment of multiple instances of a single component, are discussed in the following sections. The four components of the architecture are defined below:

Policy Administration Point (PAP) The PAP offers services for the policy creation, management, and distribution. The PAP is a component of the delegated attribute-based authorization process and executes the corresponding authorization protocols. Moreover, it provides interfaces for policy management services to human operators. The PAP accesses the PSP to persist policies and policy changes.

Policy Storage Point (PSP) The PSP acts as a repository to make created policies and changes to policies persistent. For this purpose, the PSP offers Create, Read, Update, and Delete (CRUD) services to PAP instances. The physical PSP instance may be integrated with the PAP component to avoid network communication overhead.

Policy Decision Point (PDP) The PDP takes access control decisions by evaluating policies. The PDP takes decision on request of a PEP and provides the access control decision to the requesting PEP. As a consequence, the PDP is part of the delegated ABAC task of SABAAC. Furthermore, in the SABAAC architecture the PDP incorporates the services provided by the context handler, which was introduced by Hu et al. [29]. Therefore, the PDP not only takes access control decisions on request but is responsible for the policy and attribute evaluation workflow. This workflow includes the retrieval of required attributes and speedup techniques such as access decision caching and policy evaluation precomputation.

Policy Enforcement Point (PEP) The PEP enforces access control decisions by controlling access to protected objects. As a consequence, the PEP is part of the delegated ABAC task of SABAAC. The services provided by the PEP rely on access control decisions taken by the PDP. Moreover, in the SABAAC architecture the PEP incorporates the services provided by the Policy Information Point (PIP), which was introduced by Hu et al. [29]. Accordingly, the SABAAC PEP provides attributes related to its protected objects to the PDP.

4.7.2 Access Control Policy

According to Hu et al. [29], ABAC is an access control model that enables access decisions based on attributes associated with subjects, objects, actions, and the environment of a system. An ABAC policy represents a set of rules that describe under which environmental conditions a certain subject is granted to perform certain actions on a specific protected object. The SABAAC approach relies on the concept of attribute-based policies and access control due to the following benefit: ABAC enables multifactor policy expression, while RBAC and IBAC limit the policy expressiveness by only relying on either roles or identities. Consequently, the multifactor policy expression enables fine-grained and flexible access control.

4.7.2.1 Policy Specification

The SABAAC approach features a novel strategy for the representation and specification of access control policies. In this section we introduce the structure of the policies and define the different types of attributes that are supported. As defined by the second and third security policy of CASC-SAS in section 4.4, the policies are enforced for data frame flows, i.e., sets of data frames with common properties passing through the SAS network. Accordingly, the objective of an access control policy is twofold. On the one hand, a policy has to contain information that specifies if the policy is applicable to a certain data frame flow. On the other hand, a policy must specify which actions should be taken for a matching data frame flow.

Access Control Policy: $\rho = (action_\rho, flow_\rho, auxiliary_\rho)$.
A SABAAC policy ρ is represented by a three-tuple, which contains the $action$ to be taken, the $flow$-specifying pattern, and a set of $auxiliary$ attributes specifying additional non-flow-related system characteristics.

Action: $action \in \{GRANT, DENY\}$.
The action of a policy specifies whether a matching data frame flow should be granted or denied. A data frame from a granted flow may be delivered to a SAS device, whereas a data frame from a denied flow is dropped by a PEP. The default action of SABAAC is $DENY$ which leads to dropping of all data frames of non-explicitly granted flows.

Flow Pattern: $flow = \{p_1, \ldots, p_n\}$, $p_i(frame) \in \{TRUE, FALSE\}$.
The flow pattern of a policy specifies whether the policy is applicable to a specific data frame or not. For this purpose, the flow pattern consists of a set of frame predicates $p_1, \ldots, p_n$. A frame predicate p_i is a function that assigns a boolean value to an arbitrary data frame. A predicate p_i matches a frame f, if $p_i(f) = TRUE$. A policy is applicable to a specific data frame if all frame predicates match the frame. In other words, a policy $\rho = (action_\rho, flow_\rho, auxiliary_\rho)$ is applicable to a frame f if the following equation holds:

$$\forall p_i \in flow_\rho : p_i(f) = TRUE$$

Auxiliary Attributes: $auxiliary = \{a_1, \ldots, a_n\}$, $a_i(system) \in \{TRUE, FALSE\}$.
The auxiliary attributes of a policy specify non-flow-related system attributes that have to be taken into account in order to apply an access control policy. The auxiliary attributes are specified in the form of so-called system predicates $a_1, \ldots, a_n$. A system predicate a_i is a function that assigns a boolean value to the current system state. A policy $\rho = (action_\rho, flow_\rho, auxiliary_\rho)$ is applicable in a system s if all system predicates match the current system state, i.e., a policy is applicable if the following equation holds:

$$\forall a_i \in auxiliary_\rho : a_i(s) = TRUE$$

4.7.2.2 Policy Classification

The utilization of ABAC can avoid explicit authorizations prior to a request [29]. In other words, an ABAC policy can be dynamically evaluated at the time of a request. This dynamic evaluation allows the use of attributes from a time-variable environment. As stated by Burmester et al. [51], a real-time attribute represents an attribute whose value is time-dependent. Given an attribute evaluation function E_{ATT} and a point of time t, the value λ_a of a real-time attribute a is defined by $E_{ATT}(a, t) = \lambda_a$.

Each SABAAC policy ρ is related to a set of attributes $ATT_\rho = ATT_{flow_\rho} \cup ATT_{auxiliary_\rho}$. The flow attributes ATT_{flow_ρ} consist of all frame-related attributes required for the evaluation of the frame predicates $p_1, \ldots, p_n$. The auxiliary attributes $ATT_{auxiliary_\rho}$ consist of all non-flow-related system attributes required for the evaluation of the system predicates $a_1, \ldots, a_n$. To handle policies based on their degree of time-variability, the SABAAC approach classifies policies as follows:

Dynamic Policy A dynamic policy ρ is an ABAC policy whose evaluation relies on at least one time-variable subject, object, environment, or action attribute. A policy ρ is dynamic iff $\exists a \in ATT_\rho : \exists t_i \neq t_j : E_{ATT}(a, t_i) \neq E_{ATT}(a, t_j)$. Due to the time-variable evaluation of dynamic policies, access decisions must have a limited time of validity that corresponds to the change rate of the underlying attribute values. As a result, caching of access decisions that are based on dynamic policies should be avoided. A dynamic policy is also referred to as real-time policy.

Static Policy A static policy ρ is an ABAC policy whose evaluation does not rely on time-variable subject, object, environment, or action attributes. A policy ρ is static iff $\forall a \in ATT_\rho : \forall t_i, t_j : E_{ATT}(a, t_i) = E_{ATT}(a, t_j)$. Since static policies do not rely on time-variable attributes, access decisions can be cached. Moreover, due to the non-frequent attribute retrieval and evaluation as well as access decision caching, static policies are a viable solution for low latency message exchange. A static policy is also referred to as non-real-time policy.

4.7.2.3 Policy Evaluation

The policy evaluation is the process of deriving an access control decision from an access control policy and enforcing the decision in the SAS. While the access control policy may depend upon the current system state, the corresponding access control decision is static during a specified period of validity. As a consequence, the process of evaluating a SABAAC policy can be divided into two related processes: The PDP-driven dynamic authorization derives an access control decision from an access control policy for the current system state. The PEP-driven decision enforcement utilizes the result of the dynamic authorization to enforce the SAS policies in a timely manner.

Access Control Decision: $decision_\rho = \{flow_\rho, action_\rho, nexthop, validity\}$.
An access control decision or access decision consists of a $flow$-specifying pattern, an $action$ to be taken, a $nexthop$ set specifying the PEPs a matching data frame should be forwarded to, and a period of $validity$. Accordingly, an access decision is valid for a specific data frame flow in a specific system during a specified period of time.

Dynamic Authorization: $E_{POL} : Policy \times System \longrightarrow Decision$.
In the SABAAC approach, the policy evaluation is server-aided and takes partially place at the PDP. To derive an access decision from an access control policy at the PDP, the dynamic authorization function is used. The dynamic authorization function of SABAAC is defined as follows:

$$E_{POL}(\rho, s_t) = \begin{cases} (flow_\rho, action_\rho, nexthop, validity) & \text{, if } \forall a_i \in auxiliary_\rho : a_i(s) = TRUE, \\ (flow_\rho, DENY, \emptyset, validity) & \text{, otherwise.} \end{cases}$$

Thus, if the system predicates $a_1, \dots, a_n$ of the policy match the current system state s_t, the policy is applicable to data frame flows in the system s. To evaluate the system predicates, the PDP fetches the auxiliary attributes $ATT_{auxiliary_\rho} \subseteq s_t$ prior to the evaluation of $a_1, \dots, a_n$. The validity of the resulting access decision at the point of time t equals the minimal validity of the attribute values in $\{\lambda_k = E_{ATT}(k, t) | k \in ATT_{auxiliary_\rho} \subseteq s_t\}$. Consequently, the validity of an access decision of a dynamic policy is determined by the attribute value that expires first. The validity of an access decision of a static policy may be limited by specifying a non-attribute-related maximum time of validity, to avoid the utilization of invalid or outdated access decisions.

Decision Enforcement: $E_{DEC} : Decision \times Frame \longrightarrow (Action, Nexthop)$.
A PEP uses an access control decision d_ρ taken by a PDP to enforce a policy ρ in the SAS. For this purpose, each data frame f traversing a PEP is matched using $flow_\rho$ to identify the corresponding access decision. The process of enforcing a policy based on a priorly taken access decision is referred to as decision enforcement. At the PEPs the decision enforcement for a decision $d_\rho = \{flow_\rho, action_\rho, nexthop, validity\}$ and a frame f is performed using the decision enforcement function E_{DEC}:

$$E_{DEC}(d_\rho, f) = \begin{cases} (action_\rho, nexthop) & \text{, if } \forall p_i \in flow_\rho : p_i(f) = TRUE \\ & \quad \wedge validity.contains(time.now), \\ (DENY, \emptyset) & \text{, otherwise.} \end{cases}$$

Thus, if the flow predicates match a data frame, and the decision is still valid, the decision enforcement function returns the action to be taken and the PEPs the frame has to be forwarded to. Two cases have to be distinguished, if $E_{DEC}(d_\rho, f) = (action_\rho, nexthop)$:

- **Outgoing Data Frame:** An outgoing data frame f is forwarded to each $PEP \in nexthop$, if $action_\rho = GRANT$.
- **Incoming Data Frame:** An incoming data frame f is accepted by a receiving PEP, if $action_\rho = GRANT$ and $PEP \in nexthop$.

If multiple decisions match a data frame f identified by flow attributes ATT_f, a PEP may choose the decision corresponding to the most specifically matching flow predicates. When two flow patterns $flow_\gamma$ and $flow_\delta$ of the access control policies γ and δ match a data frame $f \supseteq ATT_f$, $flow_\gamma$ is said to match f more specifically than $flow_\delta$ if $ATT_f \supseteq ATT_{flow_\gamma} \supsetneq ATT_{flow_\delta}$. In case of $(ATT_{flow_\gamma} \not\supseteq ATT_{flow_\delta}) \wedge (ATT_{flow_\gamma} \not\subseteq ATT_{flow_\delta})$, the matching flow patterns and their corresponding access control policies and decisions are said to be conflicting. A PEP may use a so-called composite decision d_C to resolve two or more conflicting access control decisions $d_1, \ldots, d_n$:

$$d_C = \begin{cases} (\bigcup_{i=1}^{n} flow_i, GRANT, \bigcup_{i=1}^{n} nexthop_i, \min_{i=1}^{n} validity_i) & \text{, if } \forall d_i \in \{d_1, \ldots, d_n\} : \\ & \quad action_i = GRANT, \\ (\bigcup_{i=1}^{n} flow_i, DENY, \emptyset, \min_{i=1}^{n} validity_i) & \text{, otherwise.} \end{cases}$$

Evaluation Strategies: A Priori, A Posteriori, & Predicted Evaluation
To speed up the policy evaluation and increase the data frame throughput, PEP and PDP instances may rely upon different policy evaluation strategies. The SABAAC approach distinguishes between three evaluation strategies shown in Figure 4.9 and defined below:

- **A Priori:** The a priori or precomputed evaluation aims to maximize the policy evaluation speed. A finite-state machine representing the a priori evaluation strategy at a PDP and PEP is shown in Figure 4.9a. At the PDP, the strategy is applied via precomputation of the dynamic authorization function E_{POL} for all available access control policies of the SAS. Accordingly, a PDP refreshes and caches each access decision periodically, and provides it to PEPs if requested. At the PEP, a priori evaluation can be achieved by requesting access decisions before data frames are matched and forwarded, i.e., before E_{DEC} is evaluated. For this purpose, a PEP requests all relevant access decisions at startup and re-request them from a PDP before the period of validity ends. The precomputation strategy results in an increased policy evaluation speed, increased data frame throughput, and decreased network jitter sensitivity. The a priori strategy is particularly suitable for periodic message exchanges, such as messages exchanges of the SV protocol. However, precomputation and caching of access decisions leads to increased memory utilization. Furthermore, periodic refreshing of unused or rarely used access decisions results in non-optimal power efficiency.

- **A Posteriori:** The a posteriori or ad-hoc evaluation aims to minimize the memory utilization. A finite-state machine representing the a posteriori evaluation strategy at a PDP and PEP is shown in Figure 4.9b. In contrast to the a priori strategy, neither the PDP-driven dynamic authorization nor the PEP-driven access decision requests are precomputed or cached. As a consequence, for each traversing data frame a PEP requests an access decision from a PDP. The PDP evaluates E_{POL} on request and provides the result to the PEP. As no access decisions are cached at the PDP and PEP, the a posteriori strategy minimizes the memory utilization. The strategy is particularly suitable for non-recurring and non-periodic message exchanges without message fragmentation. However, for periodic message exchanges and frequently matching flow patterns the strategy results in non-optimal power efficiency and decreased data frame throughput, as access policies are repeatedly evaluated even if the corresponding access decisions are still valid.

- **Predicted:** The predicted, conditional, or hybrid evaluation aims to optimize data frame throughput, memory utilization, and power efficiency. A finite-state machine representing the predicted evaluation strategy at a PDP and PEP is shown in Figure 4.9c. This strategy is inspired by the branch prediction used in microprocessor architectures. Instead of relying exclusively upon a priori or a posteriori evaluation, the hybrid evaluation uses a predictor to decide whether an access decision should be refreshed and cached. By predicting whether an access decision is required in the next prediction period, PDPs and PEPs are able to avoid unnecessary computation, requesting, and caching of non-required access decisions. As for branch predictors in microprocessor architectures, the primary objective of an access decision predictor is the avoidance of cache misses, while optimizing the memory utilization and power efficiency.

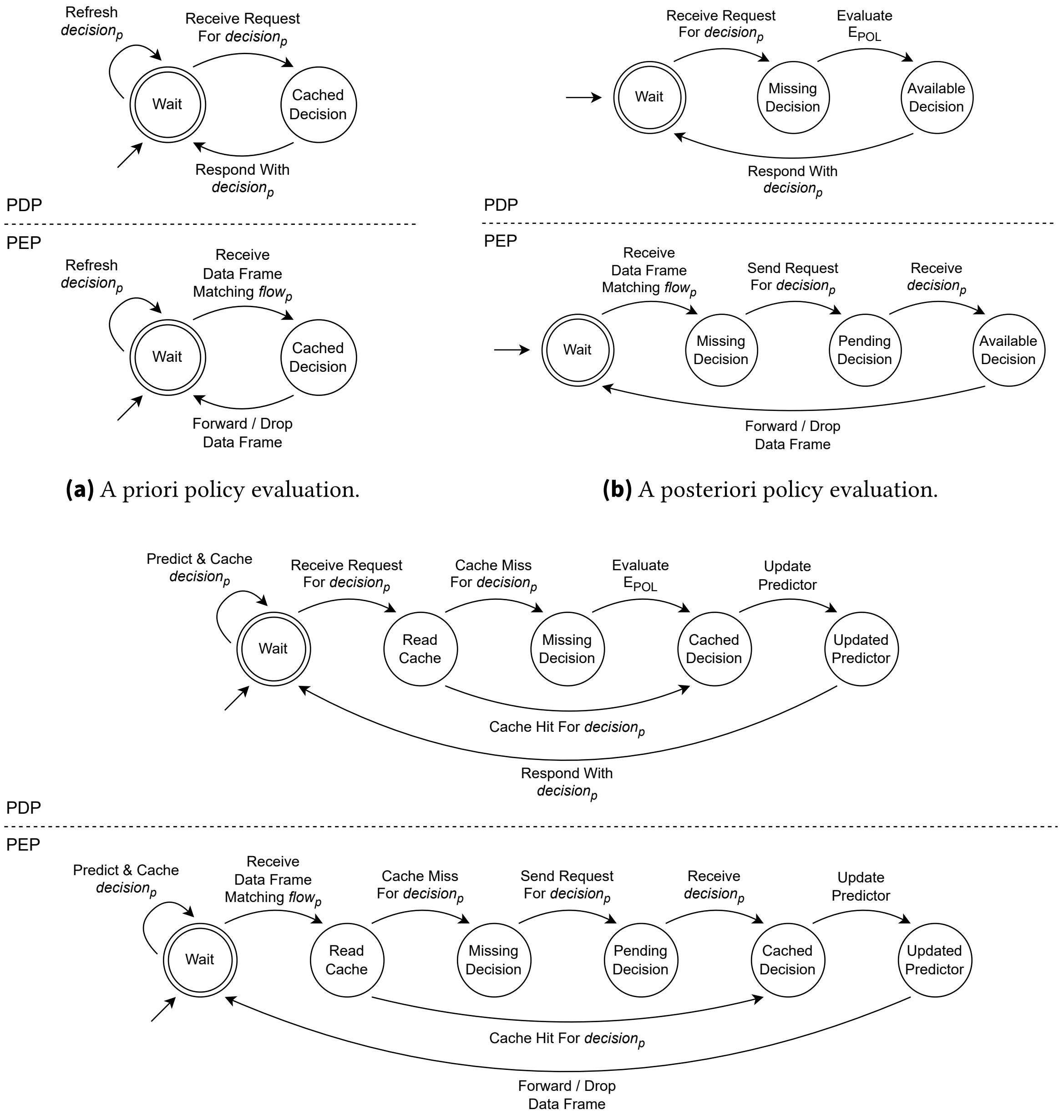
4.7.3 Delegated Attribute-Based Authorization Protocol

In the following, we discuss the delegated attribute-based authorization protocol of the SABAAC approach. Authorization is the process of assigning access privileges for protected objects to subjects [17]. A subject is said to be authorized for a specific request if it has the required access privileges for the request. We propose an authorization protocol that is responsible for the policy creation, modification, storage, and distribution. For this purpose, the authorization protocol provides policy management services at the PAP. Moreover, the authorization protocol provides services for the exchange of policies between the PAP, PSP, and PDP. The authorization protocol is part of the control path of CASC-SAS, as neither policy management nor policy exchange are directly related to domain-specific messages.

The delegated attribute-based authorization protocol offers reliable services to entities involved in the policy management process, including human operators or intrusion detection and prevention systems. It uses signing and verification services provided by CASA to safeguard the integrity and authenticity of messages. The exchanged messages of the protocol are sequenced, as defined by security policy IV in section 4.4, to protect the protocol against intentional or accidental re-ordering, replaying, and delaying of messages. To guarantee the delivery of messages, the protocol relies on an Automatic Repeat Query (ARQ) approach. Accordingly, positive acknowledgements and timeout-based retransmissions are used to achieve reliable message transmission over an unreliable communication channel.

As discussed in subsection 4.7.2, access control policies are classified as either static or dynamic. Consequently, the authorization protocol has to take the time-variability of access control policies into account. For this purpose, the authorization protocol consists of three parts or sub-protocols:

Static Authorization The static authorization process is responsible for handling CRUD requests for access control policies. The process, the involved components, and the occurring message exchanges are visualized in Figure 4.10. The static authorization is



(a) A priori policy evaluation.

(b) A posteriori policy evaluation.

(c) Predicted policy evaluation.

Figure 4.9: Finite-state machines of the SABAAC policy evaluation strategies at the PDP and PEP.

initiated by a human operator or an external system responsible for the management of policies. When a policy CRUD request arrives at a PAP, it is processed by the PAP according to the requested action. In case of a read request, the PAP fetches the requested access control policy from its PSP and returns it to the requestor. Create, update, and delete requests are processed by computing a set of changes to be made to the persistent set of access control policies. These changes are sent to the PSP via

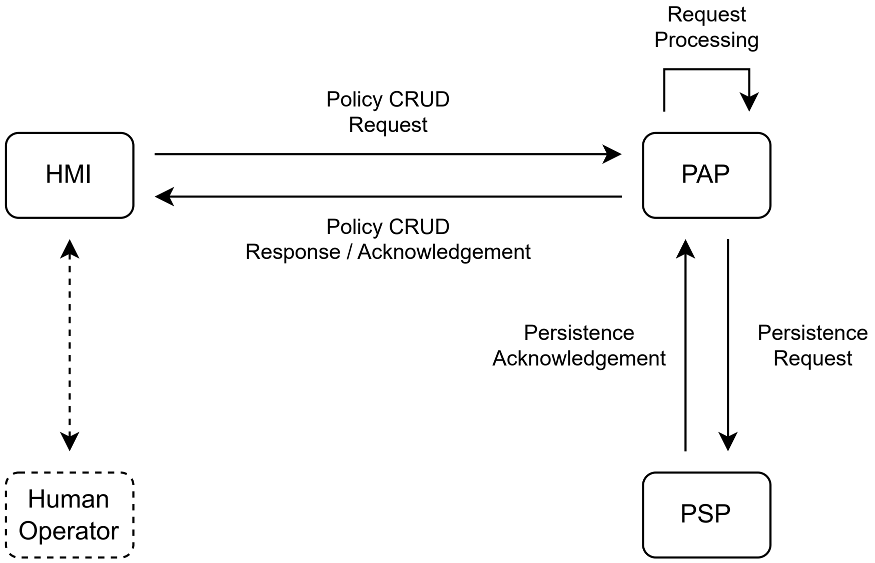


Figure 4.10: Exchanged messages of the static authorization process.

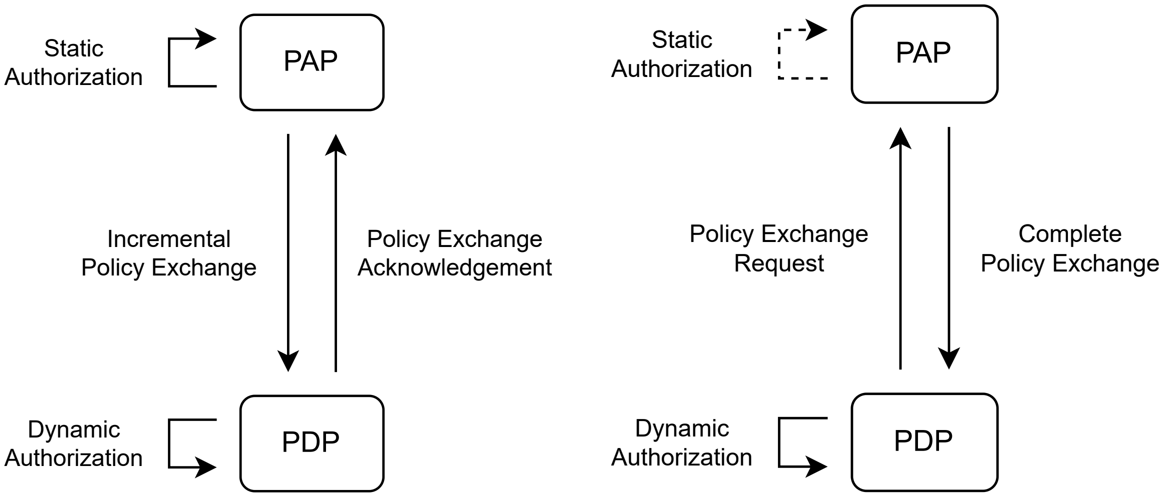
persistence requests. As soon as all changes are made to the access control policies, the PSP sends a persistence acknowledgement to the PAP and the PAP acknowledges the initial policy CRUD request.

Policy Exchange After a policy is created, modified, or deleted during static authorization, it is shared with the PDPs via policy exchange. The policy exchange is an interaction between a PAP and a PDP. The interaction is either initiated by a PAP as a result of a static authorization, or on request of a PDP.

If a static authorization triggers the policy exchange, the PAP sends a policy exchange message to a PDP. This type of policy exchange is referred to as incremental policy exchange., i.e., only newly created, modified, or removed policies are exchanged. The incremental policy exchange is shown in Figure 4.11a.

A policy exchange containing all relevant policies can be initiated by a PDP by sending a policy exchange request to a PAP. This type of policy exchange is referred to as complete policy exchange. The complete policy exchange is shown in Figure 4.11b.

Dynamic Authorization The dynamic authorization process is responsible for deriving an access decision from an access control policy at the PDP. The process, the involved components, and the occurring message exchanges are visualized in Figure 4.12. Algorithm 1 shows the steps of the dynamic authorization process executed by a PDP, i.e., the process of deriving an access control decision in a given system for each access control policy available to the PDP. Depending on the evaluation strategy used, the dynamic authorization might either be triggered automatically prior to an access request of a PEP, or be triggered by an access request. As discussed in subsection 4.7.2, the dynamic authorization process is based on the dynamic authorization function $E_{POL} : Policy \times System \longrightarrow Decision$. Prior to the evaluation of the dynamic authorization function for a policy, the PDP fetches the auxiliary attributes from other system



(a) Incremental policy exchange procedure initiated by the static authorization of a PAP.

(b) Complete policy exchange procedure initiated by a policy exchange request of a PDP.

Figure 4.11: Exchanged messages of the policy exchange procedures.

entities via attribute request messages. These entities respond to the attribute requests with attribute resolution messages. As soon as the auxiliary attributes of a policy are available at the PDP, the dynamic authorization function E_{POL} is evaluated.

Algorithm 1 Dynamic authorization process of a PDP deriving an access control decision for each access control policy in a given system.

```
function DynamicAuthorization($Policies$, $System$)
    $Decisions \leftarrow \{\,\}$ // Initialize an empty set of access decisions
    $ATT_{System} \leftarrow \{\,\}$ // Initialize an empty system state

    for all $\rho = (action_\rho, flow_\rho, auxiliary_\rho) \in Policies$ do
        $ATT_{auxiliary_\rho} \leftarrow \{\,\}$ // Initialize the auxiliary attributes of policy $\rho$
        for all $a_i \in auxiliary_\rho$ do
            // Request the attribute values for $a_i \in auxiliary_\rho$ from the system
            $ATT_{a_i} \leftarrow$ ResolveAttributes($a_i$, $System$)
            $ATT_{auxiliary_\rho} \leftarrow ATT_{auxiliary_\rho} \cup ATT_{a_i}$
        end for

        // Add the auxiliary attributes of policy $\rho$ to the current system state $ATT_{System}$
        $ATT_{System} \leftarrow ATT_{System} \cup ATT_{auxiliary_\rho}$

        // Derive an access decision from policy $\rho$ & the current system state $ATT_{System}$
        $Decisions \leftarrow Decisions \cup E_{POL}(\rho, ATT_{System})$
    end for
    return $Decisions$
end function
```

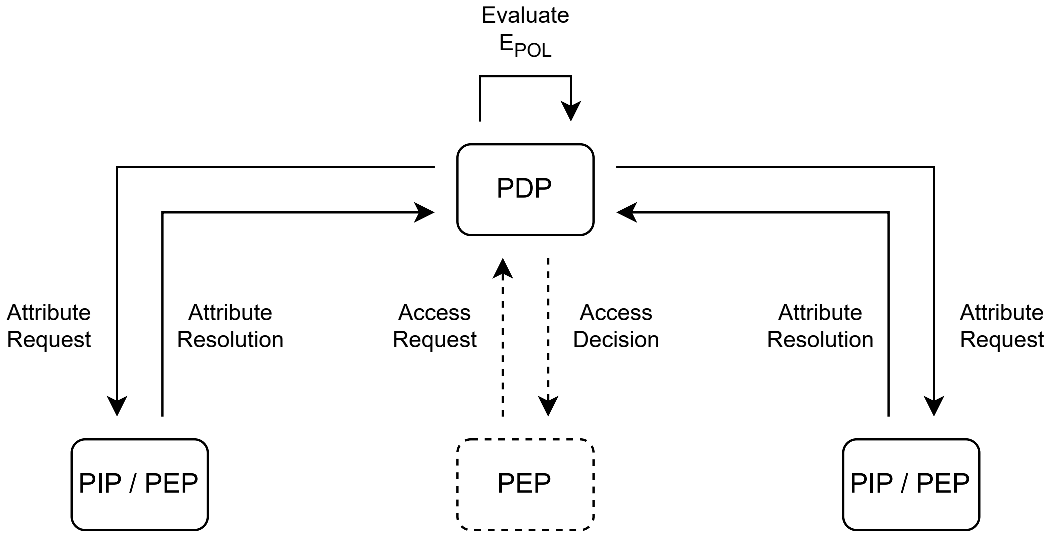


Figure 4.12: Exchanged messages of the dynamic authorization process.

4.7.4 Delegated Attribute-Based Access Control Protocol

In the following section, we discuss the delegated attribute-based access control protocol of the SABAAC approach. The goal of the access control protocol is the request, exchange, and enforcement of access control decisions. The devices of a SAS, including IEDs and MUs, neither request nor enforce the access control decisions but delegate these tasks to trusted PEP instances. The access control decisions result from the dynamic authorization process discussed in subsection 4.7.3. Since the PDP instances execute the dynamic authorization, the provisioning of access control decisions is delegated to the PDP instances as well.

The delegated attribute-based access control protocol comprises reliable and unreliable services. These services are provided by PDPs and used by PEPs. Since the services provided by the protocol are directly related to domain-specific messages, the protocol is part of the CASC-SAS data path. Accordingly, the occurring message exchanges are time-critical and traffic-intensive. As with the delegated attribute-based authorization protocol, the access control protocol uses signing and verification services provided by CASA to safeguard the integrity and authenticity of messages. Furthermore, messages of the protocol are sequenced, as defined by security policy IV in section 4.4, to protect the protocol against intentional or accidental re-ordering, replaying, and delaying of messages. The parts of the protocol that guarantee the delivery of messages, use an ARQ approach with positive acknowledgements and timeout-based retransmissions to achieve reliable message exchange. As the message exchanges of the protocol are time-critical, negative acknowledgements are used in addition to timeout-based retransmissions to indicate errors and trigger retransmissions.

The workflow of the access control protocol is divided into three mandatory phases and an optional verification phase. The phases of the access control protocol are defined as follows:

Access Request The access request represents the initial phase of the access control procedure. The goal of this phase is the exchange of an access decision between a PDP and PEP. The access request phase is initiated by a PEP on behalf of a domain subject. Depending on the evaluation strategy used, the PEP either sends an access request to a PDP before or after a domain-specific request arrives at the PEP. On receipt of an access request, the PDP verifies the request, fetches the requested access decision from its cache or derives it via dynamic authorization, and returns the access decision to the requesting PEP.

To identify the requested access decision and the corresponding access control policy at the PDP, the PEP appends a set of flow attributes $ATT_f \subseteq f$ of a data frame flow f to the access request. The PDP uses these flow attributes to evaluate the flow patterns of the available access control policies. Consequently, the flow attributes are used by the PDP to identify all applicable policies. The PDP returns the access decision of the applicable access control policy with the most specific flow pattern match. As discussed in subsection 4.7.2, if multiple conflicting access control policies are applicable, the composite decision of the corresponding access decisions can be used to resolve the conflict. If no access control policy is applicable, the PDP returns a default access decision $d_{Default}$ that matches the flow attributes ATT_f exactly. Since the default action of SABAAC is $DENY$, the default access decision is $d_{SABAAC} = \{flow_{ATT_f}, DENY, \emptyset, validity_{Default}\}$.

Session Initialization The session initialization is executed by a PEP that is granted access during the access request phase. To initialize an access control session, a PEP sends a session initialization request to another PEP. The request has to contain a PDP-signed granted access decision. The PEP may send the initialization request along with domain-specific data by piggybacking a payload exchange request. A more detailed examination of the piggybacking approach is provided below. On receipt of an initialization request, a PEP may optionally use server-aided access decision verification that is further discussed in the following section. If a received initialization request is valid, the PEP initializes a session state by adding the encapsulated access decision to an internal set of access decisions for incoming messages. Furthermore, the PEP sends a positive initialization acknowledgement and starts processing piggybacked domain-specific requests. If the request is invalid, the PEP sends a negative initialization acknowledgement and discards the piggybacked domain-specific request.

A successful session initialization between two PEPs is shown in Figure 4.13. Besides the session initialization procedure, the figure shows a preceding access request and a server-aided access decision verification at the receiving PEP. Additionally, the shown SABAAC components rely on computation requests for server-aided cryptography provided by the APEX protocol of CASA.

Access Decision Verification The optional server-aided access decision verification enables a PEP to verify received access decisions. This optional step is used to reduce the trust in a single PDP instance. To initiate the verification process, a PEP appends a

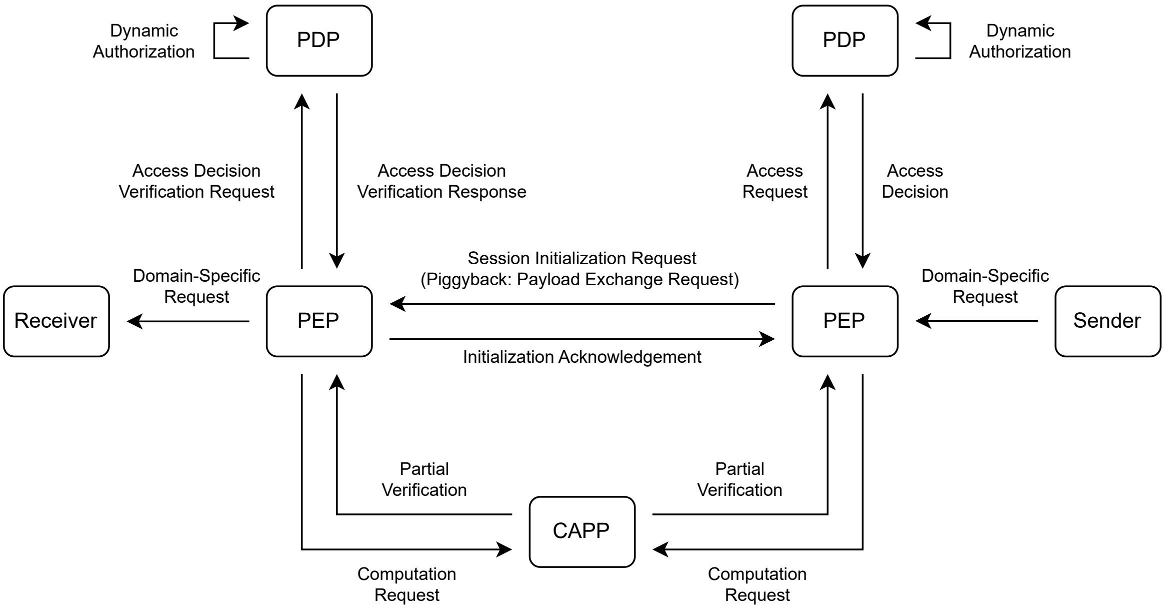


Figure 4.13: Exchanged messages of an access request with unidirectional session initialization and access decision verification.

PDP-signed access decision to an access decision verification request, and sends the request to another PDP. The PDP verifies the access decision and sends a positive or negative access decision verification response back to the PEP.

Payload Exchange The payload exchange phase is the final phase of the access control protocol. The goal of the payload exchange phase is to securely exchange domain-specific requests or data between a sending and a receiving domain entity, i.e., between two SAS devices, such as IEDs or MUs. As the payload exchange is time-critical and traffic-intensive, delivery of messages cannot be guaranteed via positive acknowledgements and timeout-based retransmissions. Consequently, the service provided by the payload exchange phase is unreliable with regard to dropping of messages. However, the payload exchange phase relies on a Negative Acknowledgement (NACK) concept. A NACK is sent in case of an exception, thus acknowledgement implosions in multicast and broadcast communication scenarios are avoided. A received NACK may trigger a session re-initialization workflow at a PEP.

The payload exchange is initiated by a domain entity via delivery of a domain-specific data frame to its PEP. On receipt of a domain-specific data frame f, the sender's PEP identifies all matching access control decisions $d_1, \ldots, d_n$ by evaluating their flow patterns for data frame f. As in the access request phase and as discussed in subsection 4.7.2, the PEP uses the access decision of the applicable access control policy with the most specific flow pattern match, or a composite decision in case of a conflict. If no matching access decision is available, the PEP initiates an access request and session initialization procedure. As soon as the access decision is available, the PEP executes the decision enforcement function $E_{DEC} : Decision \times Frame \rightarrow (Action, Nexthop)$. If the access decision is granting, the PEP encapsulates the data

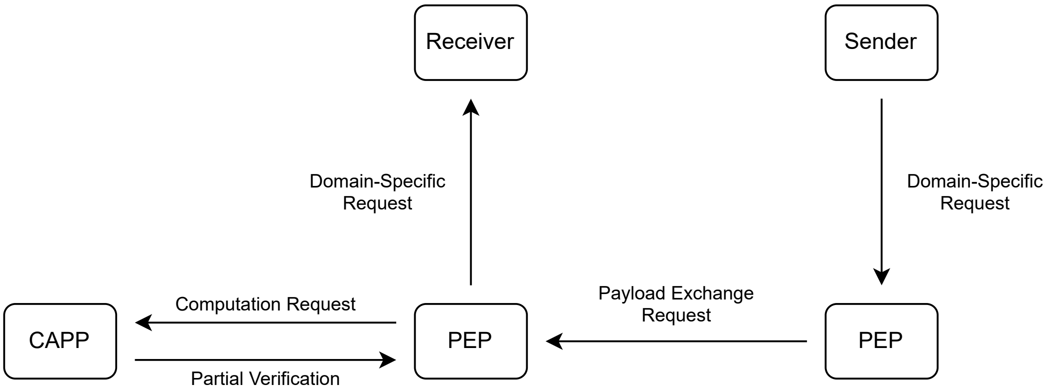
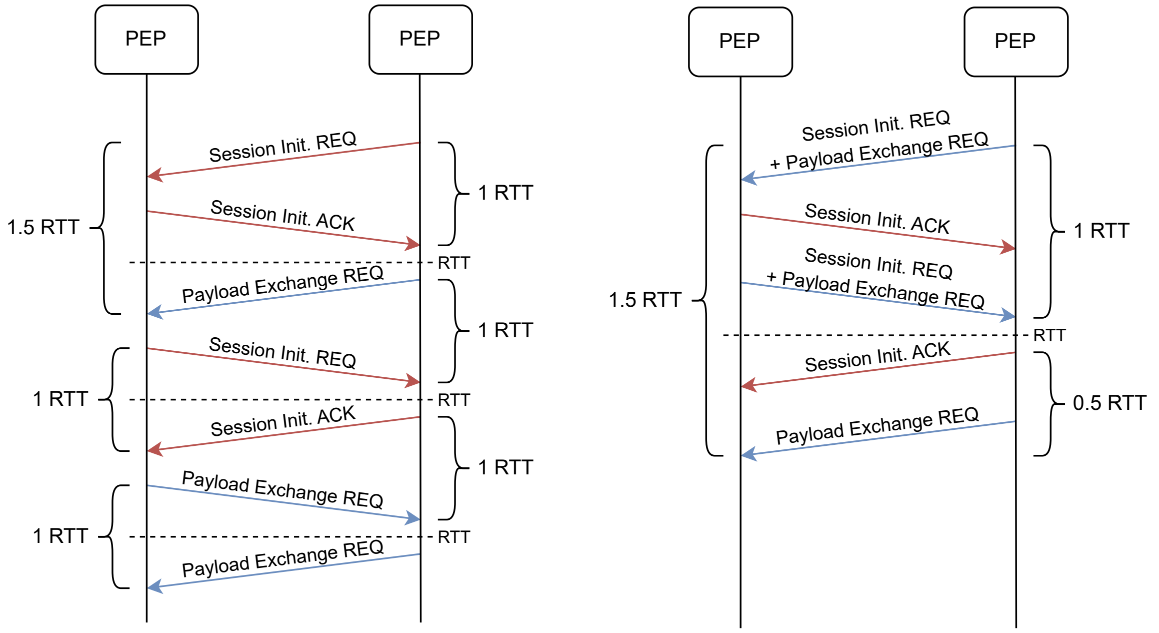


Figure 4.14: Exchanged messages of a unidirectional payload exchange procedure.

frame in a payload exchange request, and forwards the request to each $PEP \in nexthop$. On receipt of the payload exchange request, the receiving PEP fetches the most specific access decision and evaluates E_{DEC}. If the access decision is granting and the receiving PEP is part of the $nexthop$ set, the encapsulated data frame is forwarded to the receiving domain entity. Otherwise, the payload exchange request is discarded. A successful unidirectional payload exchange between a sender and receiver is shown in Figure 4.14. The shown procedure relies solely on the interaction of PEP instances, with the exception of a computation request for server-aided cryptography on message receipt.

Piggybacked Payload Exchange Domain-specific communication is unidirectionally handled by the access control protocol of SABAAC. Consequently, a response to a domain-specific request is handled as independent message exchange by the PEPs. To reduce the overhead of session initialization handshaking, the requesting PEP may send the session initialization request along with a domain-specific data frame by piggybacking a payload exchange request. The processing of piggybacked payload exchange requests starts as soon as the corresponding initialization request is processed. The usage of piggybacking decreases the required time until a domain-specific request arrives at a PEP. Since session initialization requires at least one RTT for handshaking, a non-piggybacked domain-specific request is delayed by at least one RTT. As session initialization is handled unidirectionally, the handshaking leads to a delay of at least two RTTs for bidirectional domain-specific communication.

A simplified session initialization procedure between two PEPs is shown in Figure 4.15. A session initialization procedure without piggybacking is shown in Figure 4.15a, whereas Figure 4.15b shows the same payload exchanges using piggybacked requests. Three RTTs are the minimum time until a response to a domain-specific request can be received if no piggybacking is used. This three RTT delay consists of one RTT for forward session initialization, a half RTT for the forward payload exchange request, one RTT for backward session initialization, and a half RTT for the backward payload exchange request. The use of piggybacking reduces the minimum time required to deliver the first payload exchange request from one and a half RTTs to a half RTT



(a) Bidirectional session initialization and payload exchange without piggybacking.

(b) Bidirectional session initialization and payload exchange with piggybacking.

Figure 4.15: Protocol sequence diagrams of bidirectional session initialization and message exchange.

under the assumption of symmetric transmission times. The minimum time until a response is delivered for the initial payload exchange request is reduced from three RTTs to a single RTT. After the bidirectional initialization of sessions, the minimum time required for a bidirectional payload exchange equals one RTT in both scenarios.

4.8 Realization

In the following section, we discuss the realization of the CASC-SAS approach and its two core concepts CASA and SABAAC. The approach and its two core concepts introduce components that are defined and discussed in section 4.6 and section 4.7. To employ the CASC-SAS approach in a SAS, these components have to be integrated into the system architecture of a newly constructed or retrofitted SAS. The three-layered architecture of a SAS, as introduced in the IEC 61850 standards [70], is shown in Figure 4.1. Our adaptation of the layered SAS architecture integrating the introduced components of the CASC-SAS approach is shown in Figure 4.16. Any non-intermediate SAS devices that participate in a communication relationship must either support the CASC-SAS protocols or use the services provided by a PEP to secure occurring message exchanges according to the CASC-SAS security policies introduced in section 4.4. The components depicted in blue represent devices of a typical SAS, whereas the components depicted in red are introduced by the CASC-SAS approach. The components with a color gradient represent SAS devices that have been adapted to support CASC-SAS concepts.

The employed components of CASA and SABAAC provide different types of services, and have to be deployed differently to correspond to the presented protocols. Since the delegated attribute-based authorization protocol is part of the time-critical and traffic-intensive data path of CASC-SAS, the PEPs and PDPs have to be present locally in every adapted SAS. Furthermore, at least one CAPP instance has to be present locally in a SAS to support query and computation transactions in a timely manner, i.e., to take the strict time constraints of SAS-internal low latency message exchanges into account. The PAP and PSP may be centrally deployed, as the delegated attribute-based authorization protocol of SABAAC is part of the non-time-critical control path communication. Consequently, while some components may be deployed centrally, other components have to be distributed to individual substations. This leads to differing hardware requirements for the deployed components. With the exception of PEP instances, the components provide their services by using a client-server pattern. The PEP instances partially use a client-server pattern and partially provide their services in the form of a Bump-In-The-Wire (BITW) solution. The services of the delegated attribute-based access control protocol at the PEPs are provided via BITW pattern. Therefore, these services are invisible to the corresponding service consumers, i.e., to the SAS devices secured by the PEPs. The BITW pattern is inspired by the security filter approach presented by Ishchenko and Nuqui [15]. Taking the differing provision patterns and deployment structures into account, we propose the usage of performance-oriented server hardware for the PAP, PSP, PDP, and CAPP to avoid bottlenecks and mitigate the risk of accidental or malicious DoS. For the highly distributed PEP instances, we propose the usage of inexpensive off-the-shelf hardware. To reduce the trust in single component instances and achieve a fault-tolerant and scalable system, components can be deployed redundantly, as discussed for access decision verification in subsection 4.7.4.

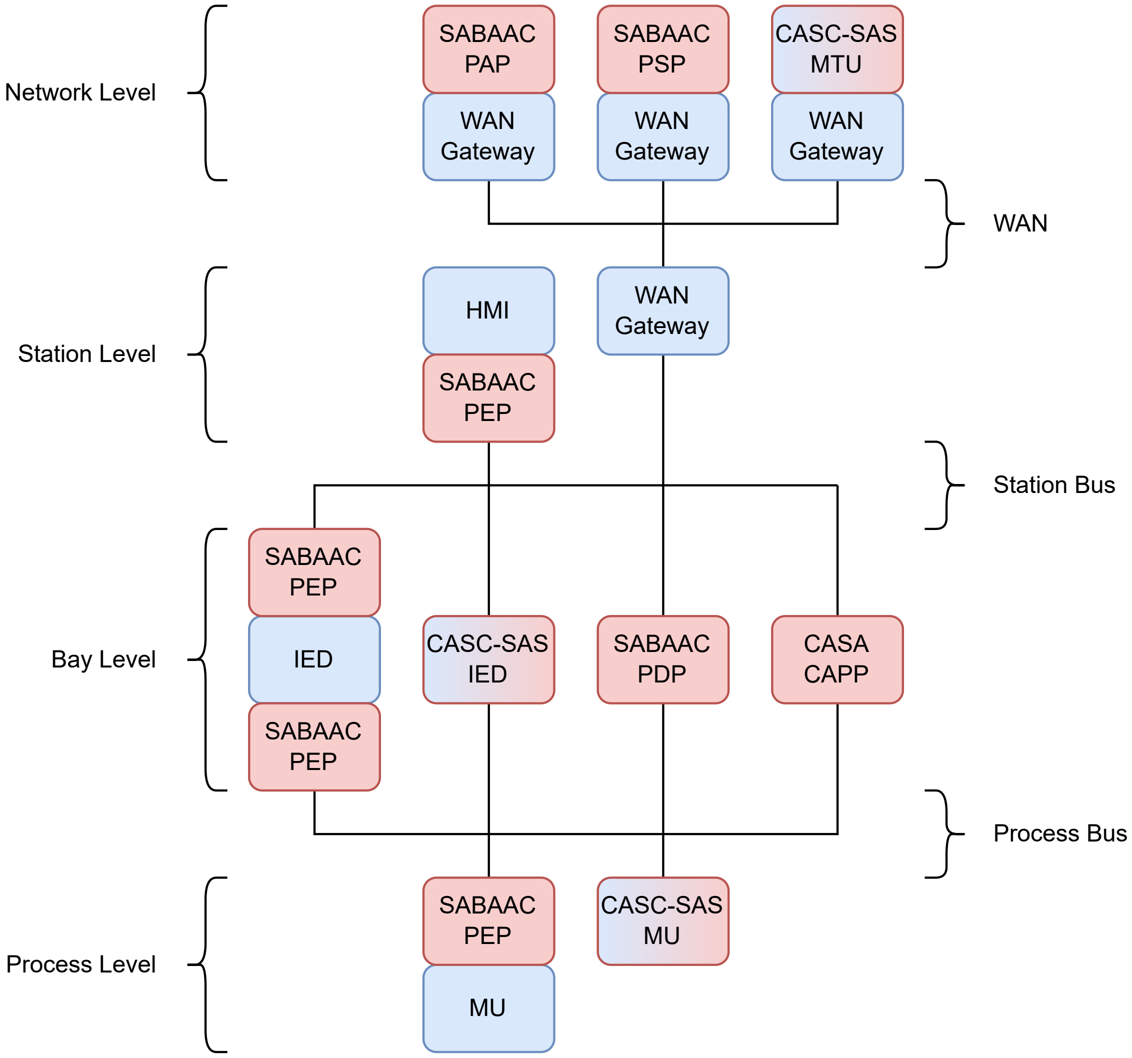


Figure 4.16: Adaptation of the layered SAS architecture to the CASC-SAS approach.

5 Evaluation

In this section, we analyze and evaluate the CASC-SAS approach and discuss the findings of the evaluation. The goal of the evaluation is to derive quantitative and qualitative characteristics of the approach. These characteristics are used to verify the applicability of the approach in the presented field of application, i.e., the employment of the approach in newly constructed or retrofitted substations. Furthermore, the characteristics are used to identify limitations and future directions of the approach.

5.1 Method

The evaluation is performed theoretically as well as experimentally. For the theoretical parts of the evaluation, we employ proofs to demonstrate and guarantee certain characteristics of our approach. The experimentally performed parts of the evaluation are based on a testbed implementation of our approach and the concepts discussed in section 4.8. The areas and metrics covered by the different parts of the evaluation are discussed in the following section.

5.1.1 Evaluation Areas & Metrics

The evaluation of our approach is based on the goal-question-metric (GQM) approach [71, 72]. The GQM approach aims to analyze whether an overall goal was achieved by answering a set of questions that represent the different areas of interest of the evaluation. These questions are answered by deriving and evaluating quantitative and qualitative metrics. The evaluation of the CASC-SAS approach covers three areas of interest. The three areas of interest, i.e., questions to be answered, and their corresponding metrics are defined below:

Goal: Protect the time-constrained and traffic-intensive communication of a newly constructed or retrofitted SAS against domain-typical adversaries and attacks.

- **Question: Security** Does CASC-SAS provide security against typical SAS adversaries and attacks?
  - **Metric:** Which security, safety, and availability requirements can be satisfied by deploying the approach in a SAS?
  - **Metric:** Which adversary and system characteristics are assumed?

- **Metric:** Which attacks can be mitigated, and how can these attacks be mitigated with regard to their corresponding mitigation strategy?
- **Metric:** How does the attack surface of the SAS change?

Question: Performance Is CASC-SAS capable of securing time-constrained and traffic-intensive communication of an SAS in an efficient and scalable manner?

- **Metric:** Which performance requirements can be satisfied by deploying the approach in a SAS?
- **Metric:** Which communication characteristics are assumed?
- **Metric:** Which message types are supported?

Question: Compatibility Is CASC-SAS a viable solution to enhance the security of newly constructed or retrofitted substations?

- **Metric:** Which compatibility requirements can be satisfied by deploying the approach in a SAS?
- **Metric:** Which device requirements are assumed?
- **Metric:** What are the additional costs for SAS construction and retrofitting?
- **Metric:** Is the approach feasible with regard to SAS retrofitting?

5.1.2 Testbed

To analyze and evaluate the integration of our approach into the SAS architecture, as discussed in section 4.8, we implemented the approach in hardware and software as a testbed. The software is implemented component-wise using object-oriented high-level programming languages. The components are primarily implemented using the programming languages Java and Kotlin. The software implementation of our approach is published open source on GitHub [73] under the European Union Public Licence (EUPL) [74]. The implementation is divided into three main packages. These packages, their sub-packages, and the package interrelationships are shown in Figure 5.1. The common package contains functionalities that are required by all other parts of the implementation. Among others, the common package contains classes and interfaces related to message ingress and egress, message serialization, concurrency, and cryptography. The second package and its sub-packages are dedicated to the CASA components and protocols. The third package and its sub-packages are dedicated to the SABAAC components and protocols. To avoid circular dependencies between the packages and achieve loose coupling of components, we employ the dependency inversion principle by using interfaces.

To be able to conduct experiments while taking the behavior of physical network communication into account, we transformed the software implementation into a physical system by deploying the SABAAC and CASA components to hardware. In contrast to deterministic

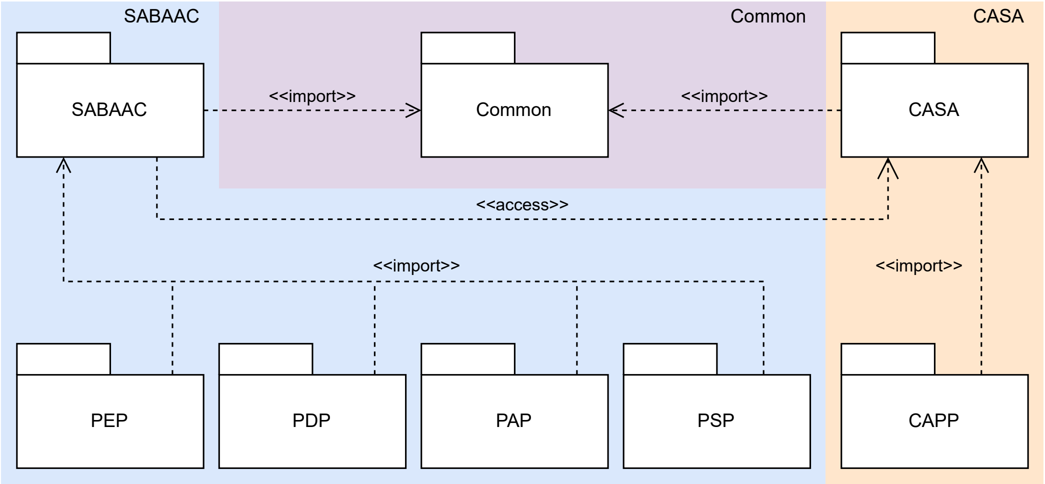


Figure 5.1: Structural package diagram of the CASC-SAS testbed implementation.

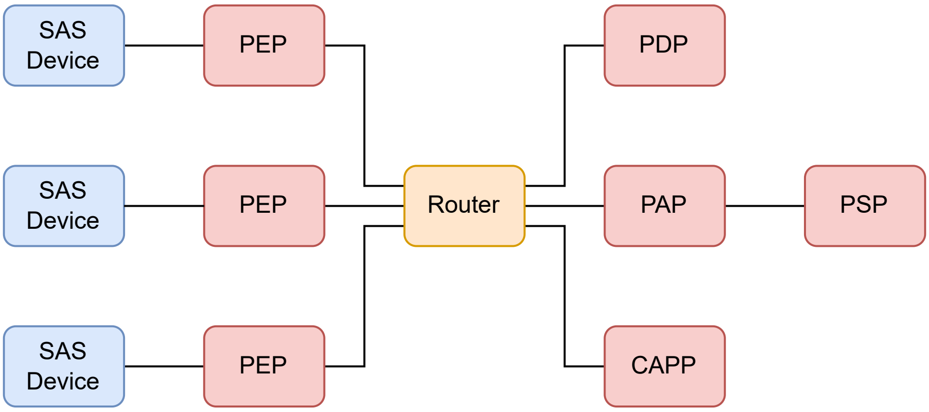


Figure 5.2: Conceptual network topology of the CASC-SAS testbed.

software-based analysis, the testbed evaluation results are practice-oriented and transferable to the SAS domain. The conceptual topology of our testbed network is visualized in Figure 5.2. The components depicted in blue represent domain-specific devices of an SAS, including IEDs and MUs. The components depicted in yellow represent intermediate network devices for frame and datagram forwarding on the data link layer and network layer. The components depicted in red are part of the CASC-SAS approach. The specific hardware used for the experiments is discussed for each experiment individually in the following sections.

5.2 Security Analysis

In this section, we conduct an analysis of the security of the CASC-SAS approach and its core concepts CASA and SABAAC. The primary objective of the security analysis is to demonstrate and guarantee certain security-related characteristics of the approach. As the security analysis is performed theoretically, proofs will be provided demonstrating that the approach is able to mitigate adversarial attacks that endanger these security-related characteristics.

Definition. Unforgeability.
Unforgeability ensures that no adversary can create a valid signature for a message under a policy unless their set of attributes satisfies the policy. Unforgeability is defined by using a game between a challenger and an adversary $\mathcal{A}$:

- Setup: The challenger runs the `Setup` algorithm to generate the public parameters PK and the master secret key MSK. The public parameters are given to $\mathcal{A}$, while the MSK is kept secret.
- KeyGen: $\mathcal{A}$ can query the `KeyGen` oracle to obtain private keys for sets of attributes of its choice. The challenger responds with the corresponding private keys.
- SignQueries: $\mathcal{A}$ can request signatures for messages and policies from the `Sign` oracle. The oracle returns valid signatures if the queried attributes satisfy the signing policy.
- Forgery: $\mathcal{A}$ outputs a forged signature (M^*, T^*, σ^*) for a message M^* and policy T^*. $\mathcal{A}$ wins if the following conditions hold:
  - $\mathcal{A}$ did not request a signature on (M^*, T^*) from the `Sign` oracle.
  - $\mathcal{A}$ does not possess a private key whose attributes satisfy the policy T^*.
  - The verification algorithm accepts σ^* as a valid signature under T^*.

Definition. Existential Unforgeability under Chosen-Message Attacks (EU-CMA).
An adversary $\mathcal{A}$ is given access to public parameters, hash oracles, and a signing oracle. A scheme is secure if $\mathcal{A}$ cannot forge a valid signature σ^* for a new message M^* without knowing the signer's full private key. In other words, to create an existential forgery, i.e., create a valid pair of message and signature for a new message, an adversary carrying out a CMA can request valid signatures for any message of his choice [75]. The adversary's advantage in this game is its probability of generating a valid forgery. We say the ABS scheme is *existentially unforgeable* if the adversary's advantage is negligible.

Theorem. $\mathcal{S}_{CASA}$ is EU-CMA secure under the Computational Diffie-Hellman (CDH) assumption in the random oracle model.
Proof. After querying the signing oracle, assume $\mathcal{A}$ forges a signature σ^* for M^*. The challenger interacts with $\mathcal{A}$ as follows:

- *Setup*: The challenger generates the public system parameters ρ and the hash oracles H_1, H_2, H_3 for $\mathcal{A}$. The challenger also programs the random oracles H_1, H_2, H_3 to embed a CDH instance g_1^a and g_1^b, where a, b are random exponents.
- *HashQueries*: The challenger responds to H_1, H_2, H_3 queries with random values, ensuring consistency.

- *SignQueries*: When $\mathcal{A}$ requests a signature on a message M under an access policy T, the challenger computes the signature as:

$$\sigma_i = ppk_i \cdot H_3(h)^{\chi_i},$$

  where $h = H_2(M||T)$.

- *Forgery*: Eventually, the adversary outputs a forged signature σ^* on a message M^* under an access policy T^*. The challenger extracts the solution to the CDH problem g_1^{ab} from the forged signature by exploiting the structure of the random oracle responses.

$\mathcal{A}$'s success in forging a signature implies the ability to solve the CDH problem. Since the CDH problem is assumed to be hard, $\mathcal{A}$'s advantage in breaking the scheme is negligible. Thus, the $\mathcal{S}_{CASA}$ scheme is secure under the EU-CMA model and under the assumption of the hardness of the CDH problem in the random oracle model.

Definition. Collusion Attack.
An adversary $\mathcal{A}$ colludes with a TTP and corrupted signers to derive private keys or forge valid signatures. The scheme is secure if such a collusion does not compromise honest signers and does not allow forgery.

Theorem. $\mathcal{S}_{CASA}$ resists collusion attacks under the Discrete Logarithm Problem (DLP).
Proof. Suppose $\mathcal{A}$ colludes with the CAPP to derive $sk_i = (ppk_i, \chi_i)$, then the following steps are performed:

- The CAPP knows $ppk_i = H_1(ID_i||ATT_i)^s$.
- The signer independently chooses χ_i and the DLP ensures $\mathcal{A}$ cannot derive χ_i.
- Without χ_i, $\mathcal{A}$ cannot compute:

$$\sigma_i = ppk_i \cdot H_3(h)^{\chi_i}.$$

Thus, collusion cannot compromise security. The $\mathcal{S}_{CASA}$ scheme is secure against collusion attacks.

Definition. Message Replay.
To perform a message replay, an adversary captures and repeats the messages exchanged between two or more network devices. The adversary aims to inject false data into the system, or disrupt the operation of the network devices.

Theorem. CASC-SAS protects SAS devices against message replay attacks.
Proof. Suppose two SAS devices, *Alice* and *Bob*, exchange messages over a network. We assume that an adversary $\mathcal{A}$, as introduced in section 4.3, is able to eavesdrop and replay the messages sent from *Alice* to *Bob*, and vice versa. For the exchange of a message m between *Alice* and *Bob*, as shown in Figure 5.3, the following steps are performed:

- *Alice* sends the message m to *Bob* via PEP_{Alice}.

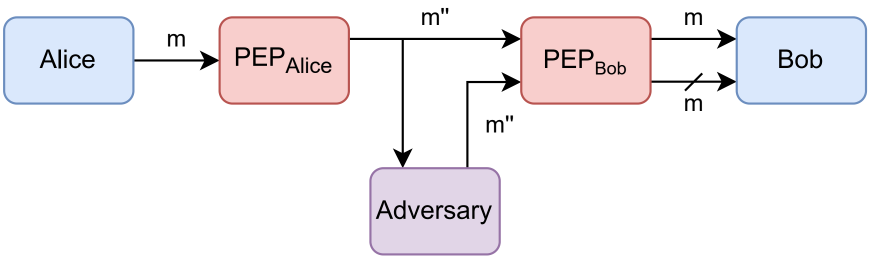


Figure 5.3: Malicious replay of a message exchanged between two PEP-protected SAS devices.

- To satisfy the security policies I and IV of CASC-SAS, PEP_{Alice} encapsulates m and a sequence number seq_m in a packet $m' = (Alice, Bob, m, seq_m)$ and signs m' to get $m'' = (m', \sigma_{m'})$.
- PEP_{Alice} sends m'' to PEP_{Bob} using the network. PEP_{Bob} receives m'' from PEP_{Alice}, verifies the signature $\sigma_{m'}$, sets the last sequence number received from PEP_{Alice} to seq_m, and delivers m to Bob.
- $\mathcal{A}$ eavesdrops the message exchange, receives m'' at the same time as PEP_{Bob}, and replays m'' to PEP_{Bob}.
- PEP_{Bob} receives m'' from $\mathcal{A}$. As PEP_{Bob} already processed a packet from PEP_{Alice} with sequence number seq_m, m'' is discarded.

$\mathcal{A}$ gains no advantage by replaying m'', as neither false data is injected into $Alice$ or Bob, nor is the operation of $Alice$ or Bob disrupted. Thus, SAS devices are protected against message replay attacks.

Definition. Message Forgery.
To perform a message forgery, an adversary masquerades as a legitimate device to send malicious messages to other devices. By using message forgery, the adversary injects false data into the system, or disrupts the operation of devices.

Theorem. CASC-SAS protects SAS devices against message forgery attacks.
Proof. Suppose two SAS devices, $Alice$ and Bob, exchange messages over a network. We assume that only the two devices have the necessary key material to sign and verify exchanged messages. As defined in section 4.3, we assume that an adversary $\mathcal{A}$ is able to initiate arbitrary message exchanges but is unable to bypass or break cryptographic procedures. To send a malicious message $m*$ from $\mathcal{A}$ to Bob, as shown in Figure 5.4, $\mathcal{A}$ has to perform the following steps:

- $\mathcal{A}$ creates a message f of its choice.
- $\mathcal{A}$ encapsulates f in a packet $m^* = ((Alice, Bob, f, seq_{m^*}), \sigma_{m^*})$, to masquerade as PEP_{Alice}.
- $\mathcal{A}$ sends m^* to PEP_{Bob}.
- PEP_{Bob} discards m^* and does not deliver f to Bob, as the signature σ_{m^*} is not created by PEP_{Alice}.

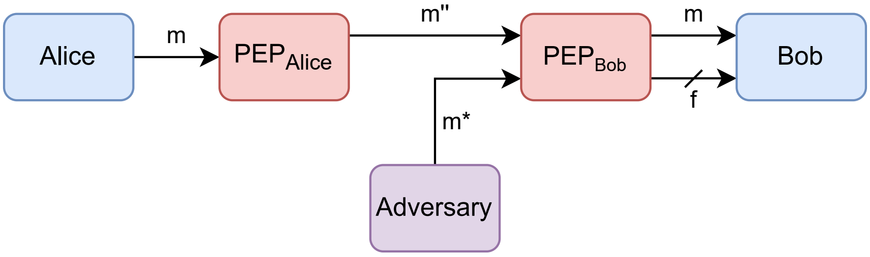


Figure 5.4: Forgery of a message by masquerading as a PEP-protected SAS device.

Since only PEP_{Alice} has the necessary key material to sign messages from *Alice*, $\mathcal{A}$ is unable to masquerade as *Alice*. Thus, SAS devices are protected against message forgery attacks.

Definition. Message Modification.
To perform a message modification, an adversary captures and alters messages exchanged between two or more network devices. Accordingly, message modification is a special type of message forgery that derives a malicious message from a captured message.

Theorem. CASC-SAS protects SAS devices against message modification attacks.
Proof. Suppose two SAS devices, *Alice* and *Bob*, exchange messages over a network. As with message forgery, we assume that only *Alice* and *Bob* have the necessary key material to sign and verify exchanged messages, and that an adversary $\mathcal{A}$ is unable to bypass or break cryptographic procedures. We assume that $\mathcal{A}$ performs the attack using a man in the middle approach, i.e., *Alice* and *Bob* are not directly connected and the exchanged messages traverse $\mathcal{A}$. The modification of a message m from *Alice* to *Bob* using a man in the middle approach is shown in Figure 5.5. To carry out a message modification attack, $\mathcal{A}$ has to perform the following steps:

- *Alice* sends the message m to PEP_{Alice}.
- To satisfy the security policies I and IV of CASC-SAS, PEP_{Alice} encapsulates m and a sequence number seq_m in a packet $m' = (Alice, Bob, m, seq_m)$ and signs m' to get $m'' = (m', \sigma_{m'})$.
- PEP_{Alice} sends m'' unintentionally to $\mathcal{A}$ using the network.
- $\mathcal{A}$ modifies m and encapsulates the modified message f in a packet $m^* = ((Alice, Bob, f, seq_m), \sigma_m)$.
- $\mathcal{A}$ sends m^* to PEP_{Bob}.
- PEP_{Bob} discards m^* and does not deliver f to *Bob*, as the signature $\sigma_{m'}$ does not match the content of m^*.

Since only PEP_{Alice} has the necessary key material to sign messages from *Alice*, $\mathcal{A}$ is unable to renew the signature after modifying the encapsulated message. Thus, SAS devices are protected against message modification attacks.

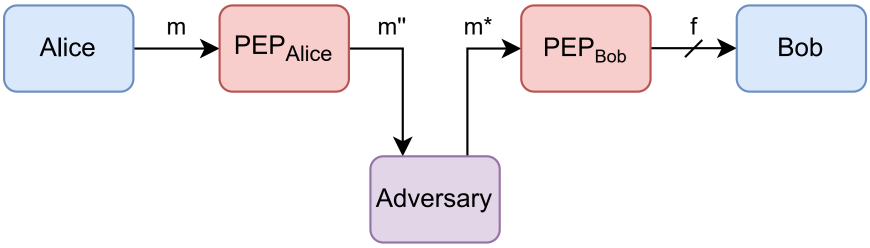


Figure 5.5: Malicious modification of a message exchanged between two PEP-protected SAS devices.

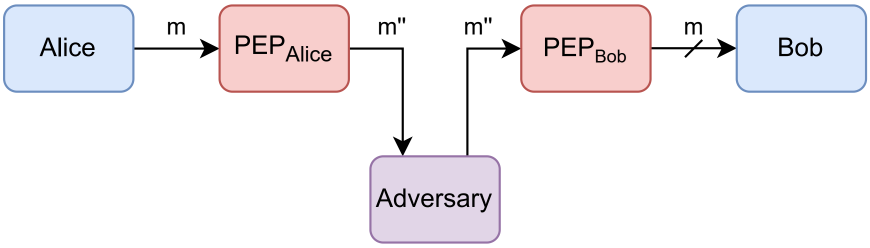


Figure 5.6: Malicious delaying of a message exchanged between two PEP-protected SAS devices.

Definition. Time-Delay Attack.

A time-delay attack is the intentional delaying of time-critical messages in a network. To perform a time-delay attack, a man in the middle adversary captures a message sent by a network device, and waits a certain time before forwarding it to the message receiver. By maliciously delaying exchanged messages, the adversary may either inject outdated data into the system or disrupt the operation of devices.

Theorem. CASC-SAS protects SAS devices against time-delay attacks.

Proof. Suppose two SAS devices, *Alice* and *Bob*, exchange messages over a network in the presence of an adversary $\mathcal{A}$, which performs a time-delay attack using a man in the middle approach. The performed time-delay attack is shown in Figure 5.6. To carry out a message modification attack, $\mathcal{A}$ has to perform the following steps:

- *Alice* sends the message m to PEP_{Alice}.
- To satisfy the security policies I and IV of CASC-SAS, PEP_{Alice} encapsulates m with a timestamp-based sequence number seq_m in a packet $m' = (Alice, Bob, m, seq_m)$ and signs m' to get $m'' = (m', \sigma_{m'})$.
- PEP_{Alice} sends m'' unintentionally to $\mathcal{A}$ using the network.
- $\mathcal{A}$ receives m'' and waits a certain time before sending it to PEP_{Bob}.
- PEP_{Bob} discards m'' and does not deliver m to *Bob*, as the sequence number seq_m indicates that the packet was maliciously or accidentally delayed.

The delay of m'' would only be unnoticeable if $\mathcal{A}$ was able to update the sequence number. However, as only PEP_{Alice} has the necessary key material to sign messages from *Alice*, $\mathcal{A}$ is unable to update the sequence number. Thus, SAS devices are protected against time-delay attacks.

Table 5.1: Hardware used for the performance analysis testbed.

Manufacturer	Device	Task	Amount
TP-Link	Omada ER605	Gigabit Router	1
Raspberry Pi Ltd	Raspberry Pi 5 8GB	PEP & Domain Entity	4
Lenovo	ThinkPad T480	PDP/PAP/PSP/CAPP	1
Bechtle	ARTICONA Adapter	USB-A to RJ45 Adapter	2

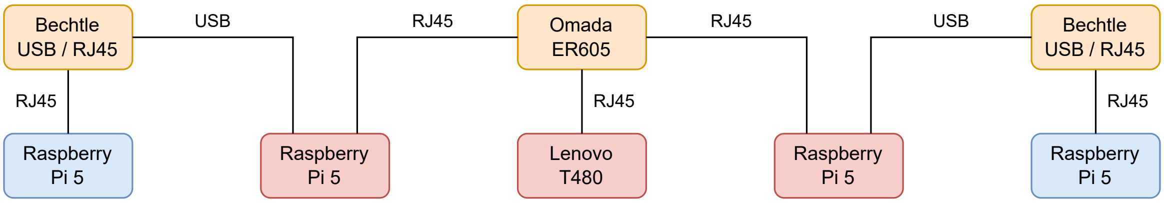


Figure 5.7: Network topology of the performance analysis testbed.

5.3 Performance Analysis

In this section, we conduct an analysis of the performance aspects of our approaches CASA and SABAAC. The objective of the performance analysis is to demonstrate that our approaches are viable solutions to secure message exchanges, taking the strict time and resource constraints of a SAS into account. For this purpose, we conducted an experimental estimation of message exchange latencies using our testbed implementation and off-the-shelf hardware. In subsection 5.3.1 the setup of the experiment is discussed in detail. In subsection 5.3.2 we describe the procedure and results of the experiment.

5.3.1 Experimental Setup

The testbed of the performance analysis consists of eight devices. The hardware devices used for the experiment are listed in Table 5.1. Two Raspberry Pi 5 were deployed to mimic domain entities that communicate with each other. Another two Raspberry Pi 5 were used as PEPs protecting the domain entities. A ThinkPad T480 provided the services of the PAP, PSP, PDP, and CAPP.

The network topology of the devices used for the performance analysis is shown in Figure 5.7. The PAP, PSP, PDP, CAPP, and PEPs were connected to an industrial grade TP-Link Omada router using Ethernet over twisted-pair. Each domain entity was connected to its corresponding PEP using Ethernet over twisted-pair, i.e., by using the on-board RJ45 Ethernet connector of the Raspberry Pi 5 domain entity. However, since the Raspberry Pi 5 only possesses a single on-board RJ45 Ethernet connector, an additional USB-A to RJ45 Ethernet adapter had to be used for both PEPs in order to connect them to the router and to the domain entities.

5.3.2 Procedure & Results

To demonstrate the viability of our approach with regard to securing time critical message exchanges, we aimed to evaluate the extent to which our approach was capable of handling different message types. For this purpose, we conducted an experiment to estimate the end-to-end communication latency of a message exchange between two domain entities. The three message types that were considered for the evaluation are discussed in subsubsection 4.1.2.2 and listed in Table 4.1.

We realized the message exchange latency estimation by implementing a benchmark program and deploying it to the domain entities. The benchmark program was implemented in Python. The program is published open source on GitHub [73] alongside the implementation of our approach. The program estimated the end-to-end latency between the two domain entities based on the RTT of a bidirectional message exchange. We chose UDP as a message exchange protocol for the experiment in order to avoid external latency influences, such as the flow control and congestion control of the Transmission Control Protocol (TCP). To measure the RTT, the so-called active benchmark entity sent a timestamp to the passive benchmark entity. The passive entity replied to the message with the received timestamp. Thus, after receiving the response from the passive entity, the active entity was able to calculate the RTT by subtracting the received timestamp from the current timestamp. As a consequence, no time synchronization was required between the domain entities. Furthermore, under the assumption of symmetric transmission times, the accuracy of the RTT measurements only depended on the accuracy of the active entity's system clock. To avoid RTT fluctuations or an offset caused by the router's buffering and forwarding strategy, messages were sent sequentially, i.e., the active entity waited for the arrival of a response before sending another timestamp message.

The procedure of the experimental latency estimation consisted of ten key events. The sequence of events and the corresponding messages exchanged between the devices are shown in Figure 5.8. To improve the readability of the shown message exchanges, the USB-A to RJ45 Ethernet adapters were omitted from the figure. The steps of the experiment procedure are defined in the following:

Step 1: Send Request

At the initial state of the experiment, no domain-specific messages were exchanged between the devices. The necessary key material for signing and verification was already exchanged prior to the experiment. The PDP used the precomputed evaluation strategy for access policies, i.e., the access decisions were periodically refreshed and cached. The PEPs used a hybrid access decision evaluation strategy, i.e., access decisions were requested and cached as soon as they were needed. To initiate the end-to-end latency estimation, the active domain entity sent a UDP packet with its current system clock timestamp to the passive domain entity.

Step 2: Request Access
As the active entity was protected by a PEP, the outgoing UDP packet was captured by the active entity's PEP. The PEP checked if an applicable and valid access decision for the packet-related data frames was available in its cache. If no applicable or valid access decision was available, the PEP sent an access request to the PDP.

Step 3: Exchange Request Payload
As soon as an applicable and valid access decision was available, the active entity's PEP processed the UDP packet as discussed in subsection 4.7.4 and sent a payload exchange request to the passive entity's PEP.

Step 4: Forward Request
On receipt of a payload exchange request, the passive entity's PEP verified the request. After the request verification, the contained UDP packet was forwarded to the passive domain entity.

Step 5: Send Response
On receipt of the UDP packet, the benchmark program of the passive domain entity extracted the timestamp, created a new UDP packet containing the same timestamp, and sent the new UDP packet to the active domain entity.

Step 6: Request Access
As the passive entity was protected by a PEP, the outgoing UDP packet was captured by the passive entity's PEP. The PEP requested and enforced the access decision for the packet as discussed in the second step.

Step 7: Exchange Response Payload
As soon as an applicable and valid access decision was available, the passive entity's PEP processed the UDP response packet and sent a payload exchange request to the active entity's PEP.

Step 8: Forward Response
On receipt of a payload exchange request, the active entity's PEP verified the request and forwarded the contained UDP response packet to the active entity.

Step 9: Estimate RTT
The benchmark program of the active domain entity extracted the timestamp from the response packet and calculated the RTT of the packet by subtracting the received timestamp from the current system clock timestamp. To compensate for fluctuations in the RTT measurements and to increase the confidence in the RTT estimation, the active entity repeated the RTT measurement procedure.

The results of the latency estimation experiment are shown in Table 5.2 and Table 5.3. The results are published open source on GitHub [73] alongside the implementation of our approach. Since CASA is an algorithm-agnostic approach, we conducted the message exchange latency estimation procedure for six different authentication algorithms. For each of the authentication algorithms, the benchmark program sent 1000 sequential packets to estimate the RTT. Based on the measurements, we calculated the arithmetic mean, median,

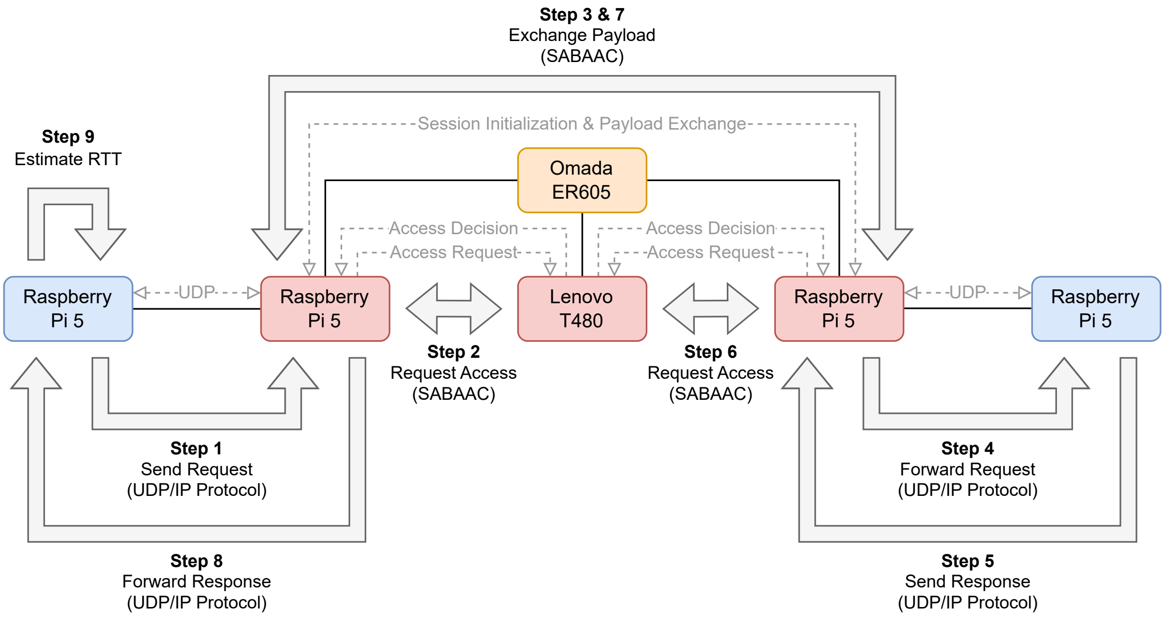


Figure 5.8: Sequence of events of the experimental message exchange latency estimation.

standard deviation, and extrema-related values for each experiment. Furthermore, we calculated the throughput of the PEPs in Packets Per Second (PPS), and the cumulative share of the 1000 packets in the three message types listed in Table 4.1.

To measure the latency offset caused by packet capturing and forwarding, we performed the initial latency estimation without authorizations, i.e., neither access control nor authentication were used. From this data, we can see that a bidirectional message exchange between two domain entities requires 1.809 ms on average, which leads to a PEP throughput of 528.2 sequential PPS. Without authentication, authorization, and access control in place, the measured RTTs are consistent, and each bidirectional message exchange was finished in less than 6 ms RTT, which is required to support the low latency message type.

To measure the influence of the authorization and access control, we performed a latency estimation with authorization and access control but without authentication. For this purpose, we implemented a so-called no-operation authenticator, which processed the packets without signing or verifying them. The measurements of the RTTs with the no-operation authenticator show that the authorization and access control workflow of SABAAC leads to an RTT increase of 1.1 ms on average. Moreover, the increased range of minimum and maximum time indicates that SABAAC leads to an increase in RTT fluctuation.

In order to evaluate the performance of our approach in combination with symmetric cryptography, we performed a latency estimation for HMAC authentication based on SHA-512. The results show that 99.8 % of the message exchanges were finished in less than 6 ms RTT. The remaining two packets or 0.2 % of the 1000 packets satisfied the 20 ms time constraint of the medium latency message type. By employing HMAC authentication, CASA and SABAAC achieved a throughput of 289.1 PPS at the PEPs.

Table 5.2: Results of the RTT estimation based on 1000 measurements per authentication algorithm.

Authentication	Mean	Median	Deviation	Extrema			
	$\bar{x}$	$\widetilde{x}$	σ	Min	Max	Range	Mid-Range
Unauthorized	1.809	1.799	0.067	1.704	2.400	0.696	2.052
Unauthenticated	2.937	2.924	0.146	2.808	4.213	1.405	3.511
HMAC	3.342	3.279	0.507	2.779	6.695	3.915	4.736
Ed25519	11.096	11.224	1.591	9.537	31.971	22.434	20.754
RSA-2048	12.703	12.034	1.179	11.741	16.804	5.063	14.273
S_{CASA}	119.851	113.127	28.534	111.385	509.392	398.007	310.389

Table 5.3: Throughput and cumulative message type share of the analyzed authentication algorithms.

Authentication	Throughput	Cumulative Share in Message Types		
		Low Latency ≤ 6 *ms*	Medium Latency ≤ 20 *ms*	High Latency ≤ 500 *ms*
Unauthorized	528.2 PPS	1000 (100 %)	1000 (100 %)	1000 (100 %)
Unauthenticated	328.2 PPS	1000 (100 %)	1000 (100 %)	1000 (100 %)
HMAC	289.1 PPS	998 (99.8 %)	1000 (100 %)	1000 (100 %)
Ed25519	88.9 PPS	0 (0 %)	998 (99.8 %)	1000 (100 %)
RSA-2048	77.4 PPS	0 (0 %)	1000 (100 %)	1000 (100 %)
S_{CASA}	8.2 PPS	0 (0 %)	0 (0 %)	998 (99.8 %)

To evaluate the performance of our approach in combination with PKC, we conducted latency estimations for three PKC algorithms. For the latency estimations we chose the Ed25519 algorithm, which is an elliptic-curve digital signature algorithm, RSA-2048, and our S_{CASA} signature scheme. All three PKC approaches did not satisfy the time constraint of the low latency message type. Ed25519 and RSA satisfied the time constraint of the medium latency message type. S_{CASA} was able to finish 99.8 % of the message exchanges within the time constraints of the high latency message type, i.e., in less than 500 ms. The data indicates that the RTTs of the PKC approaches are subject to fluctuations with higher magnitude compared to the fluctuations caused by SABAAC and symmetric cryptography. Furthermore, the throughput of the PEPs was reduced by more than 70 % compared to the HMAC authentication.

5.4 Compatibility Analysis

In this section, we conduct an analysis of the compatibility aspects of our approaches CASA and SABAAC. The objective of the compatibility analysis is to demonstrate that our approach is a viable solution to enhance the communication security in a newly constructed or retrofitted substation. Accordingly, the compatibility analysis serves the purpose of demonstrating that the SAS behavior and functionality is not influenced by our approach,

Table 5.4: Hardware used for the laboratory-based experimental demonstration of applicability.

Manufacturer	Device	Task	SAS	CASC-SAS
General Electric	Reason MU320	Process Bus Merging Unit	X	
ABB	REL670	Intelligent Electronic Device	X	
Siemens	SIPROTEC 5 6MD84	Input/Output Box	X	
Hirschmann	MACH	Managed Ethernet Switch	X	
Hirschmann	RSP35	Managed Ethernet Switch	X	
OMICRON	CMC 356	Universal Relay Test Set	X	
/	Circuit Breaker	Electrical Grid Switch	X	
Raspberry Pi Ltd	Raspberry Pi 5 8GB	PEP		X
Lenovo	ThinkPad T480	PDP/PAP/PSP/CAPP		X
Bechtle	ARTICONA Adapter	USB-A to RJ45 Adapter		X

and that SAS devices protected by CASA and SABAAC are able to provide their services and exchange information. For this purpose, we conducted a laboratory-based experimental demonstration of applicability with industrial SAS devices. In subsection 5.4.1 the setup of the experiment is discussed. In subsection 5.4.2 we describe the procedure and results of the experiment.

5.4.1 Experimental Setup

In total twelve devices were used to set up the experiment. The hardware devices used for the experiment are listed in Table 5.4. Four of these devices were industrial SAS devices, including a MU from General Electric, an IED from ABB, an I/O box from Siemens, and a circuit breaker. Additionally, a relay test device from Omicron was used to generate three-phase electric power, which was measured by the MU to generate SV frames. Besides these five SAS-related devices, two Raspberry Pi 5 were used as PEPs. A ThinkPad T480 provided the services of the PAP, PSP, PDP, and CAPP.

The network topology of the devices used for the experiment is shown in Figure 5.9. In accordance with the layered SAS architecture shown in Figure 4.1, we introduced a layering of devices for the setup of the experiment. The PAP, PSP, PDP, and CAPP were located at the bay level together with the PEP-protected IED. As the IED only supported fiber optic network connections, we employed a Hirschmann RSP35 switch as a protocol converter. With the protocol converter in place, the IED was connected to its PEP using Ethernet over twisted-pair. The PEP was then connected to a Hirschmann MACH switch representing the process bus switch. Since the Raspberry Pi 5 only possesses a single on-board RJ45 Ethernet connector, a USB-A to RJ45 Ethernet adapter had to be used for both PEPs.

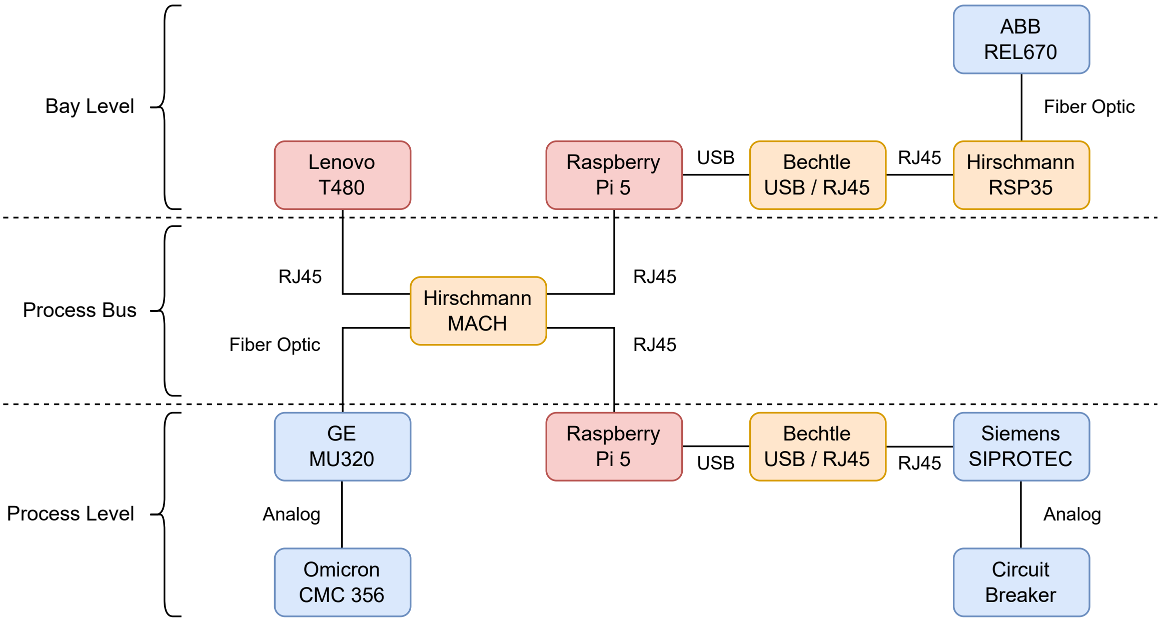


Figure 5.9: Network topology of the laboratory-based experimental demonstration of applicability.

At the process level of the experiment's network topology the MU was located. The MU was connected to the relay test device using an analog connection, and was connected to the process bus switch using a fiber optic connection. Consequently, the MU was not PEP-protected. In addition to the MU, the PEP-protected I/O box was located at the process level, and was connected to the circuit breaker via an analog connection.

5.4.2 Procedure & Results

The procedure of the experiment comprised four key events. The sequence of events and the corresponding messages exchanged between the devices are shown in Figure 5.10 and are discussed in the following:

Step 1: Generate Overcurrent

At the initial state of the experiment, the voltages and currents of all three phases generated by the Omicron CMC 356 were within certain boundaries to be detected as normal grid situation. Accordingly, the circuit breaker connected to the IED was closed and allowed power to flow through the grid. To start the experiment, we manually adjusted the generated three-phase electric power. The current was set to a higher level to simulate an overcurrent situation in the grid. This situation was communicated to the MU via a direct analog connection, i.e., the MU was measuring the voltages and currents of the three phases.

Step 2: Send Sampled Values

The MU sampled the voltage and current values provided by the relay test device. The sampled values were sent to the IED using the SV protocol. As the MU was not

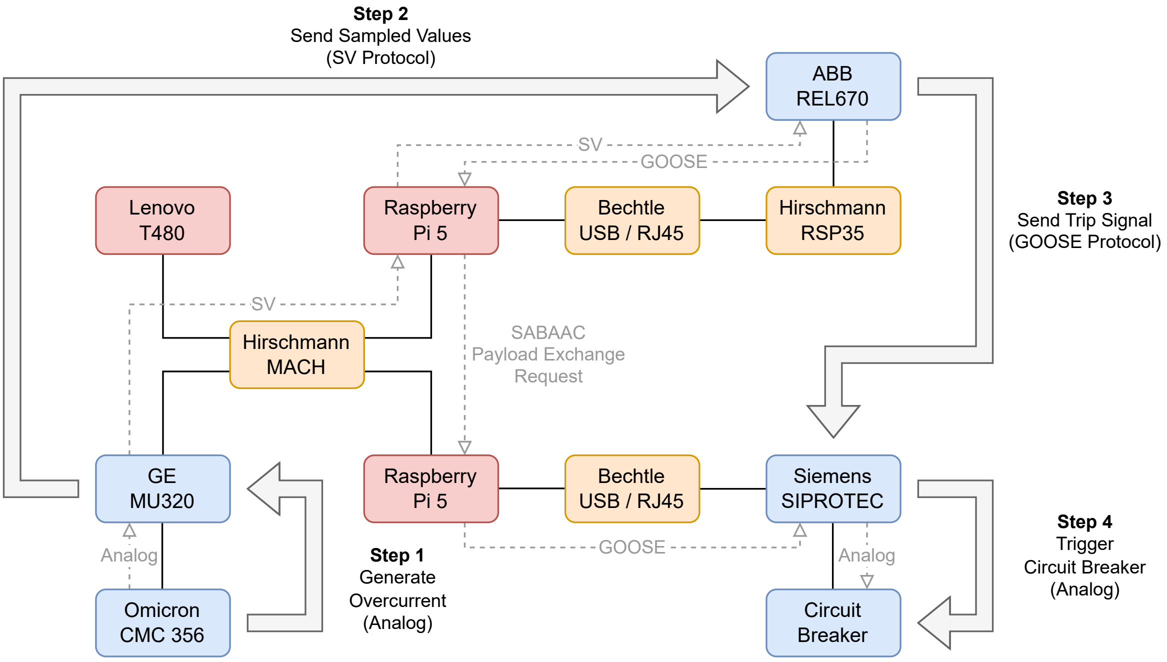


Figure 5.10: Sequence of events of the laboratory-based experimental demonstration of applicability.

protected by a PEP, we bypassed the SV frames at the IED's PEP. For this purpose, we programmed a static bypass rule into the IED's PEP that analyzed Ethernet frames to detect and forward SV frames sent by the MU.

Step 3: Send Trip Signal
The IED received and processed the SV frames, and detected the overcurrent situation. To resolve the overcurrent situation, the IED sent a GOOSE frame to the I/O box to open the circuit breaker. As the IED was protected by a PEP, the outgoing GOOSE frame was captured by the PEP and processed as discussed in subsection 4.7.4. The authenticated and authorized payload exchange message, which contained the GOOSE frame, was then forwarded to the PEP of the I/O box. The PEP verified the incoming payload exchange message, extracted the encapsulated GOOSE frame, and forwarded the GOOSE frame to the I/O box.

Step 4: Trigger Circuit Breaker
The I/O box received the original GOOSE frame, which was sent by the IED. As the GOOSE frame signalled to the I/O box to open the circuit breaker, the I/O box used an analog signal to open the circuit breaker. At the end of the experiment, we were able to visually verify that the circuit breaker had opened successfully.

The simulated exception in the grid was successfully propagated through the SAS network during the experiment. Consequently, the experiment demonstrated that the operation of the MU, IED, and I/O box was not disrupted by the employment of CASA and SABAAC. Since we deployed the security-related components to inexpensive off-the-shelf hardware, we were able to demonstrate that our approach is a feasible solution for SAS environments not only security-wise and performance-wise but also cost-wise. Due to the BITW concept of our approach, no adaptations had to be made to the SAS devices. This indicates that our approach is a viable solution for the retrofitting of existing substations. Furthermore, the usage of a static bypass rule in one of the PEPs suggests that incompatible or legacy devices could continue their operation in a retrofitted SAS. Thus, the interoperability and interchangeability requirements of an IEC 61850 substation remain satisfied, while the communication security is increased by the enforcement of the CASC-SAS security policies.

5.5 Discussion & Comparison

In this section we provide a summary of the findings of the evaluation, we contextualize our approach within the existing literature, and we describe the limitations and constraints inherent to the evaluation and our approach. The conducted evaluation covered security, performance, and compatibility aspects of CASC-SAS, CASA, and SABAAC. By performing a theoretical analysis of the security aspects of CASA and SABAAC, we demonstrated that our approaches are able to mitigate a set of adversarial attacks. We proved that our signature scheme S_{CASA} is existentially unforgeable under chosen-message attacks and that it resists collusion attacks under the DLP. Furthermore, we provided definitions for message replay, forgery, modification, and time-delay attacks, and demonstrated how CASA and SABAAC interact to mitigate these attacks. Our findings indicate that SAS devices benefit from being PEP-protected, as domain-typical DoS and false data injection attacks can be mitigated by the employment of our approach. However, by employing CASA and SABAAC the attack surface of an SAS changes. While IEDs, MUs, and other SAS devices benefit from being protected by using our BITW approach, the increased total number of devices in the SAS and their communication relations lead to an increased risk of DoS attacks. In particular, the dependence on certain centralized components, such as the PDP, PAP, PSP, and CAPP, introduces new attack vectors for SAS adversaries. Furthermore, as we conducted the security analysis theoretically, our findings might not be transferable to real SAS environments, and the behavior of CASA and SABAAC components under attack may be different from what we discussed theoretically.

In contrast to the security analysis, the performance and compatibility evaluations of our approach were conducted experimentally. The results of both experimental evaluations suggest that CASA and SABAAC are able to secure time-critical communication in a SAS. As we conducted the experiments using inexpensive off-the-shelf-hardware to deploy the CASC-SAS components, we demonstrated that our approach is lightweight and efficient with regard to its hardware requirements and corresponding deployment costs. Nevertheless, the results of the performance evaluation indicate that computational performance is the primary challenge for the deployment of information security approaches in a SAS. The findings demonstrate that lightweight yet inflexible symmetric cryptography approaches seem to be the only viable solution to secure low latency message exchanges. However, the performance evaluation also revealed that the attribute-based authorization and access control workflows of SABAAC have only a limited impact on the overall message exchange latency. Accordingly, the results emphasize the appropriateness of expressive and flexible yet computationally expensive access control approaches, such as ABAC, even in time and resource constrained environments.

Furthermore, the findings of the security, performance, and compatibility analysis indicate that authentication, authorization, and access control benefit from the consolidation of competencies. Related approaches that we discussed in chapter 3, focus on either secure communication or access control in substations. While existing approaches for the security of SAS communication, including the BITW security filter by Ishchenko and Nuqui [15], and the FPGA-based GOOSE and SV hardware encryption by Rodriguez et al. [47], are optimal solutions under certain circumstances, we emphasize the advantages of malleable security systems. By integrating CASA and SABAAC into a cybersecurity and cryptography architecture for SAS, our approach enables adaption to future requirements, while it maintains a well-defined set of provided services. In particular, our algorithm-agnostic CASA approach demonstrated the advantages of a malleable cryptography platform by supporting the employment of different algorithms for different use cases within a single SAS.

6 Conclusion

This concluding chapter presents a summary and potential future directions of our approach. The potential future directions are discussed in detail in section 6.1. In section 6.2 we provide a summary of the contributions and findings of the thesis.

6.1 Future Directions

Our findings indicate that the CASC-SAS approach, its cryptographic approach CASA, and its authorization and access control approach SABAAC serve to enhance the communication security of a SAS. Further research could be conducted to determine, whether this enhancement of security is also achievable by employing a purely cryptographic approach. To answer this question, we propose the design and realization of an AB-PKC approach that satisfies the requirements of the SAS domain. In contrast to cryptography-dependent but scheme-agnostic ABAC, as we proposed it in CASC-SAS, the AB-PKC approach could allow the accomplishment of additional security objectives, including privacy and anonymity.

In addition to the changes in the approach paradigm, further research might investigate whether the CASA approach could be expanded to encompass encryption and decryption in conjunction with its signing and verification operations. While confidentiality for power systems via encryption is explicitly non-recommended for time-critical communication in the IEC standards [4], further research might elucidate how confidentiality can be achieved even in such time-critical systems.

With regard to the cryptographic services provided by CASA, further studies could be carried out to evaluate the advantages and disadvantages of hardware-based cryptography acceleration. It is anticipated that the required computation time will decrease, leading to an increase in message throughput through the utilization of hardware accelerators for cryptographic algorithms. However, factors such as algorithm compatibility, costs per acceleration unit, and computation time consistency may result in a less beneficial influence on the system than currently expected.

The evaluation demonstrated that CASC-SAS is capable of securing application protocols of a SAS, as well as multipurpose transport protocols. However, network time protocols, such as the Network Time Protocol (NTP) and the Precision Time Protocol (PTP), were bypassed by the PEP entities, as the operation of these protocols is susceptible to temporal inconsistencies resulting from authentication and access control. Further research could investigate how network time protocols could benefit from being processed by a PEP and what additional requirements and constraints have to be satisfied with regard to computation performance and time consistency. Moreover, lower-layer network management protocols, such as the Address Resolution Protocol (ARP), were bypassed since these protocols provide services not only to SAS devices but also to auxiliary intermediate devices, including network switches and routers. Future studies could evaluate the feasibility of CASC-SAS to process these network management protocols and mitigate attacks related to them.

Furthermore, future studies could investigate the impact of redundancy protocols in time-critical networks on the operation of our approach. To this end, CASC-SAS could be deployed and assessed in systems utilizing the Parallel Redundancy Protocol (PRP) or Media Redundancy Protocol (MRP).

To simplify the architectural complexity of CASC-SAS, reduce the overall costs of deployment, and enable processing of the above-mentioned network protocols, we propose the integration of CASC-SAS into network switches as an alternative realization approach. For this purpose, further research could investigate the potential benefits of realizing CASC-SAS through the use of Software-Defined Networking (SDN) solutions. This SDN-based CASC-SAS could aggregate the tasks of multiple PEPs by deploying a virtual PEP for each port of a network switch. Furthermore, distributed SDN controllers might provide the PAP, PSP, PDP, and CAPP services.

While the proposed PAP entities provide policy management services for human operators, future research could investigate how CASC-SAS might benefit from the utilization of artificial intelligence (AI). The integration of AI-based intrusion detection could facilitate the creation and modification of security policies that are enforced within a SAS, thereby enabling our approach to mitigate a wider range of cyberattacks in a timelier manner.

In addition to the deployment in a SAS, further research is required to evaluate the applicability of CASC-SAS for other time-critical systems. Therefore, we propose the evaluation of our approach in time-critical systems that have similar requirements as a SAS. Systems that might potentially benefit from the enhanced communication security provided by our approach include industry 4.0, robotics, avionics, and medical systems.

6.2 Summary

To address the increasing relevance of cybersecurity for smart grid systems and to overcome the limitations of existing standards like IEC 61850 and IEC 62351, we presented CASC-SAS, a novel cryptography and cybersecurity approach for the enhancement of SAS security. The two attribute-based and server-aided approaches CASA and SABAAC represent the central parts of the four-layered dual-path CASC-SAS security architecture.

CASA provides algorithm-agnostic cryptographic protocols and services that serve as a foundation for the employment of other cryptography and cybersecurity mechanisms in a SAS. The main objective of CASA is to enable and support secure authentication procedures that safeguard the integrity, authenticity, and non-repudiation of SAS communication. As a central component of the CASA approach we presented the CAPP, an algorithm-agnostic administration and processing platform that provides key generation, key distribution, key revocation, and server-aided computation services via our APEX protocol to resource-constrained devices of a SAS. Furthermore, in order to take the time-criticality of communication in a SAS into account, we discussed the importance and advantages of precomputation and server-aided computation for cryptographic procedures. We demonstrated the viability of these computation techniques by presenting our server-aided AB-PKC signature scheme S_{CASA}.

SABAAC enables the administration and enforcement of ABAC policies for time-critical and time-variable environments. We introduced the concept of time-dependency for attributes and ABAC policies, and discussed methods to manage, distribute, and enforce such expressive and flexible yet computationally expensive access control policies. With regard to the management and evaluation of access control policies, we proposed a delegated attribute-based authorization protocol responsible for the policy creation, management, storage, and distribution. With regard to the enforcement of policies, we introduced a delegated attribute-based access control protocol. To facilitate the enforcement of policies in time-critical systems, we presented novel policy enforcement strategies, which combine server-side precomputation and client-side caching of access decisions.

We implemented CASA and SABAAC in software and deployed it to a hardware testbed. The software is implemented component-wise using primarily the object-oriented high-level programming languages Java and Kotlin. The software implementation of our approach is published open source on GitHub [73] under the European Union Public Licence (EUPL) [74]. To assess the applicability of our approach for SAS environments, we conducted a theoretical and experimental evaluation. The theorem-based theoretical security analysis has shown that CASC-SAS is capable of enhancing the communication security of a SAS by mitigating domain-typical adversarial attacks performed by a Dolev-Yao-like adversary. We demonstrated that our approach mitigates, among others, message forgery, modification, replay, and time-delay attacks. While SAS-typical cyberattacks can be mitigated by employing our approach, we also discussed the change of the attack surface, leading to an increased risk of DoS as a result of the additional components and protocols deployed in a SAS. Based on the testbed implementation, the performance analysis demonstrated the ability of the approach to secure time-critical message exchanges. Furthermore, the performance analysis identified the advantages and disadvantages of different authentication schemes with regard to satisfied time constraints. In accordance with the related literature, we identified the strict time constraints of low latency communication in a SAS as a key challenge for information security. The compatibility analysis demonstrated that our approach is a feasible solution for SAS environments not only security-wise and performance-wise but also cost-wise and due to its highly-compatible BITW concept, which allows retrofitting of existing systems. As we conducted a laboratory-based demonstration of applicability, our evaluation demonstrated the ability of our approach to secure time-critical message exchanges of the GOOSE and SV protocol between an IED made by ABB, a MU made by General Electric, and an I/O box made by Siemens. Accordingly, the results of the evaluation of our approach indicate that CASA and SABAAC are viable solutions to enhance the communication security in a newly constructed or retrofitted substation.

Bibliography

[1] Keith Stouffer et al. *Guide to Operational Technology (OT) Security.* Tech. rep. NIST Special Publication 800-82,Rev.3. National Institute of Standards and Technology, 2023. DOI: `10.6028/nist.sp.800-82r3`.

[2] Xi Fang et al. “Smart Grid — The New and Improved Power Grid: A Survey”. In: *IEEE Communications Surveys & Tutorials* 14.4 (2012), pp. 944–980. ISSN: 1553-877X. DOI: `10.1109/surv.2011.101911.00087`.

[3] International Electrotechnical Commission. “Part 5: Communication requirements for functions and device models”. In: *Communication networks and systems for power utility automation (IEC 61850)* (2014).

[4] International Electrotechnical Commission. “Part 6: Security for IEC 61850”. In: *Power systems management and associated information exchange - Data and communications security (IEC 62351)* (2020).

[5] International Electrotechnical Commission. “Part 8: Role-based access control for power system management”. In: *Power systems management and associated information exchange - Data and communications security (IEC 62351)* (2020).

[6] Evelio Padilla. *Substation Automation Systems: Design and Implementation.* Wiley, Oct. 2015. ISBN: 9781118987216. DOI: `10.1002/9781118987216`.

[7] Occupational Safety and Health Administration (OSHA). *Illustrated Glossary - Substations.* URL: `https://www.osha.gov/etools/electric-power/illustrated-glossary/sub-station` (visited on 01/30/2025).

[8] Communications Security Establishment Canada. *Cyber threat bulletin: Cyber threat to operational technology.* Dec. 2021. URL: `https://open.canada.ca/data/dataset/98bad300-28f1-49b9-9b34-2d46de4c9a58` (visited on 01/30/2025).

[9] Jonathan Fildes. *Stuxnet worm targeted high-value Iranian assets.* 2010. URL: `https://www.bbc.com/news/technology-11388018` (visited on 01/30/2025).

[10] Jim Finkle. *Insiders suspected in Saudi cyber attack.* 2012. URL: `https://www.reuters.com/article/net-us-saudi-aramco-hack/exclusive-insiders-suspected-in-saudi-cyber-attack-idUSBRE8860CR20120907` (visited on 01/30/2025).

[11] Cybersecurity & Infrastructure Security Agency (CISA). *Cyber-Attack Against Ukrainian Critical Infrastructure.* 2021. URL: `https://www.cisa.gov/news-events/ics-alerts/ir-alert-h-16-056-01` (visited on 01/30/2025).

[12] Natalia Zinets. *Ukraine hit by 6,500 hack attacks, sees Russian cyberwar.* 2016. URL: `https://www.reuters.com/article/us-ukraine-crisis-cyber-idUSKBN14I1QC` (visited on 01/30/2025).

[13] Cybersecurity & Infrastructure Security Agency (CISA). *CrashOverride Malware.* 2021. URL: `https://www.cisa.gov/news-events/alerts/2017/06/12/crashoverride-malware` (visited on 01/30/2025).

[14] Blake Johnson et al. *Attackers Deploy New ICS Attack Framework "TRITON" and Cause Operational Disruption to Critical Infrastructure.* 2017. URL: `https://cloud.google.com/blog/topics/threat-intelligence/attackers-deploy-new-ics-attack-framework-triton` (visited on 01/30/2025).

[15] Dmitry Ishchenko and Reynaldo Nuqui. “Secure Communication of Intelligent Electronic Devices in Digital Substations”. In: *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D).* IEEE, Apr. 2018. DOI: `10.1109/tdc.2018.8440438`.

[16] Ghada Elbez, Hubert B. Keller, and Veit Hagenmeyer. “Authentication of GOOSE Messages under Timing Constraints in IEC 61850 Substations”. In: *Electronic Workshops in Computing.* BCS Learning & Development, Sept. 2019. DOI: `10.14236/ewic/icscsr19.17`.

[17] Claudia Eckert. *IT-Sicherheit: Konzepte – Verfahren – Protokolle.* De Gruyter, Jan. 2023. ISBN: 9783110985115. DOI: `10.1515/9783110985115`.

[18] Keith Stouffer et al. *Guide to Industrial Control Systems (ICS) Security.* Tech. rep. NIST Special Publication 800-82,Rev.2. National Institute of Standards and Technology, 2015. DOI: `10.6028/NIST.SP.800-82r2`.

[19] Brendan Galloway and Gerhard P. Hancke. “Introduction to Industrial Control Networks”. In: *IEEE Communications Surveys & Tutorials* 15.2 (2013), pp. 860–880. ISSN: 1553-877X. DOI: `10.1109/surv.2012.071812.00124`.

[20] David Bailey and Edwin Wright. *Practical SCADA for industry.* Elsevier, 2003. ISBN: 9780750658058. DOI: `10.1016/B978-0-7506-5805-8.X5000-4`.

[21] American Gas Association. “Cryptographic Protection of SCADA Communications”. In: *AGA Report* 12 (2006).

[22] Joint Task Force Interagency Working Group. *Security and Privacy Controls for Information Systems and Organizations.* Tech. rep. NIST Special Publication 800-53,Rev.5. National Institute of Standards and Technology, Sept. 2020. DOI: `10.6028/nist.sp.800-53r5`.

[23] National Security Agency. “Introduction and general model”. In: *Common Criteria for Information Technology Security Evaluation* (2009).

[24] Butler W. Lampson. “A note on the confinement problem”. In: *Communications of the ACM* 16.10 (Oct. 1973), pp. 613–615. ISSN: 1557-7317. DOI: `10.1145/362375.362389`.

[25] Kevin Stine et al. *Guide for Mapping Types of Information and Information Systems to Security Categories.* Tech. rep. NIST Special Publication 800-60,Vol.1,Rev.1. National Institute of Standards and Technology, 2008. DOI: `10.6028/NIST.SP.800-60v1r1`.

[26] CNSS Glossary Working Group. “CNSS Glossary”. In: *Committee on National Security Systems Instruction (CNSSI)* 4009 (2022).

[27] Ross Anderson, Frank Stajano, and Jong-Hyeon Lee. “Security policies”. In: *Advances in Computers*. Elsevier, 2002, pp. 185–235. DOI: 10.1016/s0065-2458(01)80030-9.

[28] National Institute of Standards and Technology. “Personal Identity Verification (PIV) of Federal Employees and Contractors”. In: *Federal Information Processing Standards Publication (FIPS PUB)* 201-3 (2022).

[29] Vincent C. Hu et al. *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*. Tech. rep. NIST Special Publication 800-162. National Institute of Standards and Technology, Jan. 2014. DOI: 10.6028/nist.sp.800-162.

[30] Ross Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, Dec. 2020. ISBN: 9781119644682. DOI: 10.1002/9781119644682.

[31] Elaine Barker and William Barker. *Guideline for Using Cryptographic Standards in the Federal Government: Directives, Mandates and Policies*. Tech. rep. NIST Special Publication 800-175A. National Institute of Standards and Technology, 2016. DOI: 10.6028/NIST.SP.800-175A.

[32] Elaine Barker. *Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms*. Tech. rep. NIST Special Publication 800-175B,Rev.1. National Institute of Standards and Technology, 2020. DOI: 10.6028/NIST.SP.800-175Br1.

[33] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. DOI: 10.1201/9780429466335.

[34] Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography*. 2023. URL: https://toc.cryptobook.us/book.pdf (visited on 01/30/2025).

[35] Auguste Kerckhoffs. “La cryptographie militaire”. In: *Journal des sciences militaires* IX (1883).

[36] C. E. Shannon. “Communication Theory of Secrecy Systems”. In: *Bell System Technical Journal* 28.4 (Oct. 1949), pp. 656–715. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1949.tb00928.x.

[37] Adi Shamir. “Identity-Based Cryptosystems and Signature Schemes”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1985, pp. 47–53. ISBN: 9783540156581. DOI: 10.1007/3-540-39568-7_5.

[38] Sattam S. Al-Riyami and Kenneth G. Paterson. “Certificateless Public Key Cryptography”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2003, pp. 452–473. ISBN: 9783540400615. DOI: 10.1007/978-3-540-40061-5_29.

[39] Committee on National Security Systems (CNSS). “Instruction for Secret National Security Systems Public Key Infrastructure X.509 Certificate Policy”. In: *Committee on National Security Systems Instruction (CNSSI)* 1300 (2021).

[40] Amit Sahai and Brent Waters. “Fuzzy Identity-Based Encryption”. In: *Advances in Cryptology – EUROCRYPT 2005*. Springer Berlin Heidelberg, 2005, pp. 457–473. ISBN: 9783540320555. DOI: 10.1007/11426639_27.

[41] Vipul Goyal et al. "Attribute-based encryption for fine-grained access control of encrypted data". In: *Proceedings of the 13th ACM conference on Computer and communications security*. CCS06. ACM, Oct. 2006. DOI: 10.1145/1180405.1180418.

[42] Vincent C Hu. *Overview and considerations of access control based on attribute encryption*. Tech. rep. NIST Internal Report 8450-upd1. National Institute of Standards and Technology (U.S.), 2023. DOI: 10.6028/nist.ir.8450-upd1.

[43] Jin Li et al. "Attribute-based signature and its applications". In: *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. ASIA CCS '10. ACM, Apr. 2010. DOI: 10.1145/1755688.1755697.

[44] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. "Attribute-Based Signatures". In: *Topics in Cryptology – CT-RSA 2011*. Springer Berlin Heidelberg, 2011, pp. 376–392. ISBN: 9783642190742. DOI: 10.1007/978-3-642-19074-2_24.

[45] John Bethencourt, Amit Sahai, and Brent Waters. "Ciphertext-Policy Attribute-Based Encryption". In: *2007 IEEE Symposium on Security and Privacy (SP '07)*. IEEE, May 2007. DOI: 10.1109/sp.2007.11.

[46] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Aug. 2018. DOI: 10.17487/RFC8446.

[47] Mikel Rodriguez et al. "A Fixed-Latency Architecture to Secure GOOSE and Sampled Value Messages in Substation Systems". In: *IEEE Access* 9 (2021), pp. 51646–51658. ISSN: 2169-3536. DOI: 10.1109/access.2021.3069088.

[48] Junho Hong et al. "Cyber Attack Resilient Distance Protection and Circuit Breaker Control for Digital Substations". In: *IEEE Transactions on Industrial Informatics* 15.7 (July 2019), pp. 4332–4341. ISSN: 1941-0050. DOI: 10.1109/tii.2018.2884728.

[49] Christoph Ruland and Jochen Sassmannshausen. "Firewall for Attribute-Based Access Control in Smart Grids". In: *2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*. IEEE, Aug. 2018. DOI: 10.1109/sege.2018.8499306.

[50] OASIS Open. *eXtensible Access Control Markup Language (XACML) Version 3.0*. 2013. URL: https://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf (visited on 01/30/2025).

[51] Mike Burmester, Emmanouil Magkos, and Vassilis Chrissikopoulos. "T-ABAC: An attribute-based access control model for real-time availability in highly dynamic systems". In: *2013 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, July 2013. DOI: 10.1109/iscc.2013.6754936.

[52] Byunghun Lee et al. "Role-based access control for substation automation systems using XACML". In: *Information Systems* 53 (Oct. 2015), pp. 237–249. ISSN: 0306-4379. DOI: 10.1016/j.is.2015.01.007.

[53] Mingchao Ma and Steve Woodhead. "Constraint-Enabled Distributed RBAC for Subscription-Based Remote Network Services". In: *The Sixth IEEE International Conference on Computer and Information Technology (CIT'06)*. IEEE, 2006. DOI: 10.1109/cit.2006.63.

[54] Mingchao Ma and Steve Woodhead. "Authentication delegation for subscription-based remote network services". In: *Computers & Security* 25.5 (July 2006), pp. 371–378. ISSN: 0167-4048. DOI: 10.1016/j.cose.2006.03.006.

[55] Cristina Alcaraz, Javier Lopez, and Stephen Wolthusen. "Policy enforcement system for secure interoperable control in distributed Smart Grid systems". In: *Journal of Network and Computer Applications* 59 (Jan. 2016), pp. 301–314. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2015.05.023.

[56] H. Samuel, W. Zhuang, and B. Preiss. "Routing over Interconnected Heterogeneous Wireless Networks with Intermittent Connections". In: *2008 IEEE International Conference on Communications*. IEEE, 2008. DOI: 10.1109/icc.2008.435.

[57] Nian Liu et al. "Study on PMI based access control of substation automation system". In: *2006 IEEE Power Engineering Society General Meeting*. IEEE, 2006. DOI: 10.1109/pes.2006.1709324.

[58] Robert W. Shirey. *Internet Security Glossary, Version 2*. RFC 4949. Aug. 2007. DOI: 10.17487/RFC4949.

[59] A. Nicholson et al. "SCADA security in the light of Cyber-Warfare". In: *Computers & Security* 31.4 (June 2012), pp. 418–436. ISSN: 0167-4048. DOI: 10.1016/j.cose.2012.02.009.

[60] Fadi Aloul et al. "Smart Grid Security: Threats, Vulnerabilities and Solutions". In: *International Journal of Smart Grid and Clean Energy* (2012), pp. 1–6. ISSN: 2315-4462. DOI: 10.12720/sgce.1.1.1-6.

[61] Tiago Antonio Rizzetti et al. "Cyber security and communications network on SCADA systems in the context of Smart Grids". In: *2015 50th International Universities Power Engineering Conference (UPEC)*. IEEE, Sept. 2015, pp. 1–6. DOI: 10.1109/upec.2015.7339762.

[62] Tarek A. Youssef et al. "IEC 61850: Technology standards and cyber-threats". In: *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*. IEEE, June 2016. DOI: 10.1109/eeeic.2016.7555647.

[63] Jiran Cai, Yongkang Zheng, and Zhenyu Zhou. "Review of cyber-security challenges and measures in smart substation". In: *2016 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*. IEEE, Oct. 2016, pp. 65–69. DOI: 10.1109/icsgce.2016.7876027.

[64] Shahbaz Hussain et al. "Vulnerabilities and countermeasures in electrical substations". In: *International Journal of Critical Infrastructure Protection* 33 (June 2021), p. 100406. ISSN: 1874-5482. DOI: 10.1016/j.ijcip.2020.100406.

[65] D. Dolev and A. Yao. "On the security of public key protocols". In: *IEEE Transactions on Information Theory* 29.2 (Mar. 1983), pp. 198–208. ISSN: 0018-9448. DOI: 10.1109/tit.1983.1056650.

[66] Hans-Joachim Hof. "Sichere Dienste-Suche in Sensornetzen". PhD thesis. Karlsruhe Institute of Technology, 2007. DOI: 10.5445/IR/1000007852.

[67] Christoph Ponikwar et al. “Beyond the Dolev-Yao Model: Realistic Application-Specific Attacker Models for Applications Using Vehicular Communication”. In: *CoRR* abs/1607.08277 (2016). DOI: `10.48550/arXiv.1607.08277`.

[68] Gianluca Bianchin and Fabio Pasqualetti. “Time-Delay Attacks in Network Systems”. In: *Cyber-Physical Systems Security*. Springer International Publishing, 2018, pp. 157–174. ISBN: 9783319989358. DOI: `10.1007/978-3-319-98935-8_8`.

[69] Wei Wu et al. “Server-Aided Verification Signatures: Definitions and New Constructions”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, pp. 141–155. ISBN: 9783540887331. DOI: `10.1007/978-3-540-88733-1_10`.

[70] International Electrotechnical Commission. “Part 81: Specific communication service mapping (SCSM) - Mappings to MMS (ISO 95061 and ISO 95062) and to ISO/IEC 88023”. In: *Communication networks and systems for power utility automation (IEC 61850)* (2022).

[71] Victor R. Basili and David M. Weiss. “A Methodology for Collecting Valid Software Engineering Data”. In: *IEEE Trans. Software Eng.* 10.6 (1984).

[72] Victor R. Basili. *Software Modeling and Measurement: The Goal/Question/Metric Paradigm*. Tech. rep. CS-TR-2956, UMIACS-TR-92-96. University of Maryland, 1992. URL: `http://hdl.handle.net/1903/7538` (visited on 01/30/2025).

[73] Moritz Gstuer. *Certificateless Attribute-Based Server-Aided Cryptosystem for Substation Automation Systems (CASC-SAS)*. 2025. URL: `https://github.com/gstuer/CASC-SAS` (visited on 01/30/2025).

[74] European Union. *European Union Public Licence (EUPL) Version 1.2*. 2017. URL: `https://joinup.ec.europa.eu/collection/eupl` (visited on 01/30/2025).

[75] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. “A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks”. In: *SIAM Journal on Computing* 17.2 (Apr. 1988), pp. 281–308. ISSN: 1095-7111. DOI: `10.1137/0217017`.