Karlsruhe Institute of Technology

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

Master's Thesis Proposal of

Moritz Gstür

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

| | |
|---|---|
| First examiner: | Dr.-Ing. Ghada Elbez |
| Second examiner: | Prof. TBD |
| First advisor: | Dr. Mohammed Ramadan |

15. April 2024 – 05. August 2024

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, 05. August 2024**

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

(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 (CASC-SAS). Our approach consists of two core concepts referred to as CASA and SABAAC. The certificateless attribute-based server-aided authentication (CASA) provides asymmetric cryptographic protocols and schemes that serve as a foundation for cybersecurity approaches in a SAS. The server-aided attribute-based authorization and access control (SABAAC) prevents unauthorized access to devices of a SAS based on ABAC policies. By employing server-aided protocols and speedup techniques, our approach take the strict time and resource constraints of the SAS domain into account. To demonstrate our approach, we propose realizing it as a fully functional test bed that mimics the behavior of a real interconnected SAS. Moreover, we propose a theoretical and experimental evaluation of the security, performance, and economic aspects of our approach.

# Zusammenfassung

The German abstract will be included in the final thesis.

# 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 [64]. 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. [64], 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 infrastructures prioritize system availability and reliability.

Within the scope of this thesis, we focus on the occurring communication in a specific type of OT system. The field of application of the approach proposed by this thesis is known as Substation Automation System (SAS). A SAS represents the entirety of communication and control equipment of a substation [51]. A substation is a facility of a high-voltage electricity grid connecting power transmission and distribution lines that use different voltage levels [50]. 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.

The IEC 61850 series provides standards for the communication networks of digital energy systems [32]. The goal of the IEC 61850 series is seamless communication and interoperability of systems in a smart energy grid. Although standards for the communication in a SAS are provided by the IEC 61850 standards, 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 series provides standardized security means for communication compliant to IEC 61850 [33]. Moreover, Part 8 of the IEC 62351 series provides a role-based access control concept for power system [34].

## 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 [14]. The Communications Security Establishment Canada documents 28 OT-related cybersecurity incidents between 2010 and 2020. 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 [22], the Shamoon malware used against Saudi Aramco in 2012 [23], the Blackenergy malware used to attack Ukrainian power distribution systems in 2015 [17], the Industroyer/CrashOverride malware used to shut down remote terminal units of a Ukrainian power transmission facility in 2016 [69, 16], the Triton/Trisis malware used to attack Triconex Safety Instrumented System (SIS) controllers in 2017 [37], and other incidents in 2013, 2014, 2015, 2017, 2018 as well as 2020 [68, 18, 10, 31, 15, 19, 36, 66].

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 [35], 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, Keller, and Hagenmeyer [21], the strict time constraints of the low latency communication in substations are key factors for the information security. Accordingly, asymmetric cryptography formerly specified by the IEC 62351 standards is not appropriate 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 asymmetric cryptography 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 asymmetric cryptography in time-critical OT systems. Therefor, the following research questions are going to be answered in the course of this thesis:

**RQ1** How can a secure certificateless server-aided public key cryptography approach be designed and implemented, that is able to serve as a foundation for a malleable and extendable cryptosystem in the time-critical SAS environment?

**RQ2** How can expressive and flexible but yet computationally expensive access control approaches such as ABAC be employed to enable prevention of unauthorized access in a time-critical SAS environment?

**RQ3** How can authentication, authorization, and access control be integrated into a certificateless attribute-based server-aided 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 based on asymmetric cryptography. The provided protocols, algorithms, and schemes aim to satisfy SAS security requirements such as integrity, authenticity, access control, and non-repudiation. The approach aims to not only enhance the security of SAS communication by satisfying security objectives, but also takes the specific characteristics, risks, and priorities of OT, ICS, and SAS into account. Our proposed cryptosystem is based on a dual-path four-layered architecture. Furthermore, the CASC-SAS approach is divided into two core concepts or components. These two concepts are referred to as CASA and SABAAC.

### Certificateless Attribute-Based Server-Aided Authentication (CASA)

CASA is an attribute-based public key approach. Instead of relying on trusted Certificate Authorities (CA), the approach employs certificateless public key cryptography. CASA provides cryptographic protocols, algorithms, and schemes that serve as a foundation for cybersecurity approaches in a SAS. To take the time criticality of SAS communication into account, the approach features server-aided algorithms and online/offline cryptography.

### Server-Aided Attribute-Based Authorization & Access Control (SABAAC)

SABAAC is a server-aided Attribute-Based Access Control (ABAC) approach. The approach uses the provided cryptographic protocols, algorithms, and schemes of CASA as a foundation. It enables the prevention of unauthorized access to devices of a SAS. Our approach aims to demonstrate the applicability of expressive and flexible but yet computationally expensive access control policies in an environment with strict time and resource constraints. For this purpose, SABAAC not only relies on server-aided algorithms, but also provides speedup techniques such as access decision caching and policy evaluation precomputation.

## 1.3 Structure

The following section outlines the planned structure of the proposed thesis. The proposed thesis will be divided into seven chapters. The preliminary work, comprising the initial four chapters of the proposed thesis, is presented in this thesis proposal.

Chapter 1 of the thesis serves to introduce the topic. The introduction includes the objective, research questions, and contribution of this thesis. Chapter 2 presents the fundamental concepts that form the basis for this thesis. The fundamentals introduce the concepts of OT, ICS, information security, system safety, access control, and public key cryptography. Subsequently, chapter 3 presents a review of the existing literature on this topic. Chapter 4 defines the proposed cryptosystem and its system model. Furthermore, this chapter describes the components, algorithms, schemes, and protocols of the proposed cryptosystem. The fifth chapter of the thesis will address the realization of the proposed cryptosystem. The proposed realization is discussed in section 4.5. The sixth chapter will provide a security, performance, and economic evaluation of the proposed approach. The proposed evaluation is discussed in section 4.6. Finally, the seventh chapter will provide a conclusion and identify the limitations as well as possible future work of the thesis.

# 2 Fundamentals

The purpose of this chapter is to introduce, define, and describe the fundamental terms and concepts of this thesis proposal. 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 proposed 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 symmetric and asymmetric 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 [20]. 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 [20], an IT system is a dynamic technical system which is able to process and store information. An IT system is a 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 interactions of information storage and processing procedures in a physical environment. According to Stouffer et al. [64] 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 [64]. 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 proposed thesis especially 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 [63]. In the scope of the thesis proposal, 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. [63] 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 [24] 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 [64].

#### 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 [5], 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 [64]. 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 [5]. 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. [64] further described four basic communication topologies for modern SCADA networks that were initially introduced by the American Gas Association [2]. 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 but 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 [64]. 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 [24], a DCS is a process-driven system rather than an event-driven system like SCADA. Moreover, 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 [64]. 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 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 I/O, processing or communication abilities. According to Galloway and Hancke [24], modularity of PLCs eases maintenance and grants more flexibility for the installation. A modular PLC generally consists of a power supply, processing modules, I/O 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 [64]. 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 [64]. 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 [24], 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 [2]. As stated by Stouffer et al. [64], 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. [64], 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 [20] 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 [64]. 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. [64], 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 [65]. 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 [48], 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 [20], 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. [64], 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 [20]. 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 [39], 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 [20]. 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 [20]. 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 [20]. 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 [20]. 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 [20]. 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 [20], 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 [62]. 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. [62], 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 [65]. In other words, each object and subject within the system is labeled according to its security levels 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 [12].

### 2.3.4 Policy

According to Anderson, Stajano, and Lee [4], 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 [65]. Moreover, a security policy can be seen as a set of rules for system entity behavior. 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 [20]. 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 [64]. 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 [47], 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 proposal, the term access control is going to be used to describe logical access control for IT and OT systems.

In other words, as stated by Hu et al. [30], 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. [30] 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 [20]. 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 [65] 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 [20]. 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 [65] defines MAC as an access control policy uniformly enforced over all subjects and objects within a system. Moreover, 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 [30]. In other words, as stated by the CNSS Glossary Working Group [12], IBAC represents an access control assigning access authorizations to objects based on the user identity.

An example of an IBAC mechanism capturing the identities of subjects and their access privileges is an access control list (ACL) [30]. 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 [20]. 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. [30], 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 [65], 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 [65]. In other words, as stated by Hu et al. [30], 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. As a consequence, ABAC is also referred to as aspect-based access control [3] 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 [30]. 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 [30]. 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. [30], 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 [63] and OT systems [64]. 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 [7, 6, 12]. 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 [6]. 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 [46]. Moreover, a cryptosystem comprises sets of valid inputs and outputs as well as required cryptographic keys [20]. 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 [9]. 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 [7, 9]. 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 [38]. According to Shannon, it shall be assumed that the adversary knows the system being used [60].

### 2.6.1 Symmetric Cryptography (Secret-Key Cryptography)

Symmetric cryptography relies on algorithms which use the same key for a cryptographic operation and its inverse operation [6, 20]. 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 [9]. 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, so-called Message Authentication Codes (MAC) are used in symmetric cryptography. Therefor, the sender computes a so-called 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 [6]. 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 Asymmetric Cryptography (Public-Key Cryptography)

Asymmetric cryptography, also referred to as Public-Key Cryptography (PKC), relies on algorithms which use a pair of two related keys for a cryptographic operation and its inverse operation [6, 12, 20]. 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 [9]. 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 symmetric cryptography [6, 20]: Firstly, PKC does not require a secure channel or secure protocol 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 [59]. Identity-Based Public-Key Cryptography (ID-PKC) approaches allow the derivation of public keys from subject identities [59, 9]. 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 [54, 9]. 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 [54]. 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 [13, 9]. 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) [54]. Certificate-based PKC approaches such as PKI use a TTP called Certificate Authority (CA) to issue, store, and revoke digital certificates [20]. 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 [54]. 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 [54]. After the generation, the public key may be shared with other entities directly or via public directories. Similar to 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 [57, 27, 29]. 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. Similar to the concept of ABE, Attribute-Based Signatures (ABS) enable the integration of attributes into signing and verification algorithms [41, 45].

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 [27, 8, 29]. 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 [8]. 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 proposed thesis is presented. The introduced related work serves as a foundation for the proposed approach presented in chapter 4. Moreover, the similarities, differences, and applications of the related work within the scope of the proposed approach are going to be highlighted.

## 3.1 Secure Communication in Substations

An authenticated communication approach for network packets between IEDs and merging units is presented by Ishchenko and Nuqui [35]. 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 packets incoming packets, blocks compromised packets after a certain threshold, and tags all outgoing packets. Moreover, the security filter alarms the IED about the 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 [32] using a HMAC and GMAC algorithm even on commodity of-the-shelf ARM hardware.

This thesis proposal 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. Moreover, the proposed 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, Keller, and Hagenmeyer [21]. As stated by the authors, ensuring integrity and authenticity of substation communication is critical. Similar to the approach presented by Ishchenko and Nuqui, 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 schemes for GOOSE messages. Firstly, the authors present the digital signature authentication approach specified by IEC 62351 [33]. This approach is based on asymmetric cryptography using the RSA Probabilistic Signature Scheme with Appendix (RSASSA-PSS) algorithm. Secondly, the authors present a keyed Hash Message Authentication Code (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, Keller, and Hagenmeyer 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, Keller, and Hagenmeyer, 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 of GOOSE messages.

An authentication and encryption approach for substation communication using the protocols GOOSE and SV is presented by Rodriguez et al. [55]. 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 [33]. 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 [32].

Besides securing the intra-substation communication based on the GOOSE and SV protocol, the thesis proposal 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, the proposed approach is software-based rather than hardware-based. Furthermore, the proposed approach does not rely on a symmetric-key algorithms, but rather 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. [28], 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 proposal is inspired by the usage of domain-specific knowledge to detect and block attacks. The proposed approach uses available domain-specific knowledge to design and implement a substation-specific cryptosystem. Moreover, the incremental framework of the proposed 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 [56]. 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 presented 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. On the one hand, within the scope of substations, devices connected to the outer station bus include Human Machine Interfaces (HMI), station computers, and WAN gateways. On the other hand, the inner station bus connects IEDs and enables low-latency GOOSE or 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 [49]. 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 proposal 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, the proposed approach controls access requests to any device within the substation that requires access control. For this purpose, not a single but rather distributed ABAC firewall is used. As a consequence, the firewall does not represent a communication bottleneck or single point of failure of an ICS in the proposed approach.

A real-time capable ABAC approach is presented by Burmester, Magkos, and Chrissikopoulos [11]. 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 other words, resources may not be available in time leading to events threatening the system state not being addressed within strict time limits. 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 real-time 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.

An IEC 61850 and IEC 62351 compliant RBAC approach for substations is presented by Lee et al. [40]. 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 [34] using XACML [49]. 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 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 [34]. 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 [32].

Instead of exclusively relying on roles, the approach presented in the thesis proposal 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 [32].

A distributed RBAC approach for subscription-based remote network services is presented by Ma and Woodhead [43, 44]. 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 leading 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 proposal is inspired by the concept of delegation presented by Ma and Woodhead. The authors illustrate the concept of delegation within the context of a subscription-based remote network service environment. The proposed approach entails the utilization of authentication and authorization delegation in substations. Moreover, the proposed approach elevates the degree of abstraction of the presented delegation concept by decoupling it from the concrete access control model used. The proposed 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, Lopez, and Wolthusen [1]. According to the authors, the presented approach integrates into a smart grid system with supernode networking architecture. As stated by Samuel, Zhuang, and Preiss [58], 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, Lopez, and Wolthusen 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, the Policy Enforcement Point (PEP) of the subject infrastructure provides the authentication token and the desired type of action on the target object to a Policy Decision Point (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 proposal relies on a system model with an architecture similar to the approach presented by Alcaraz, Lopez, and Wolthusen. 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, Lopez, and Wolthusen depends on the subject-role associations of RBAC for decision-making as specified by IEC 62351-8 [34]. 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. [42]. 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 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.

# 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 introduction and discussion of the proposed approach is organized as follows. 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 and defining its requirements. Based on the presented system model and requirements, we introduce the CASC-SAS approach in section 4.2. The two main CASC-SAS concepts, its cryptographic scheme and server-aided access control, are introduced in section 4.3 and section 4.4. In section 4.5 we present the planned realization of the CASC-SAS approach. Subsequently, we present the proposed evaluation strategies and metrics of the approach in section 4.6.

## 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 Substation Automation System (SAS) [51]. Although the proposed approach is tailored to the power system domain and substation environment, its main concepts may also be applied to other ICS with similar requirements and constraints.

### 4.1.1 Architecture

The architecture of the presented system model is based on the IEC 61850 standard [32]. The presented system model architecture consists of four layers called network level, station level, bay level, and process level. The process, bay, and station level represent the internal layers of a SAS architecture. The SAS architecture containing the process, bay, and station level as well as the station and process bus further discussed in subsection 4.1.2 is shown in Figure 4.1. 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 measurement signals 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 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 [32], 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.

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.

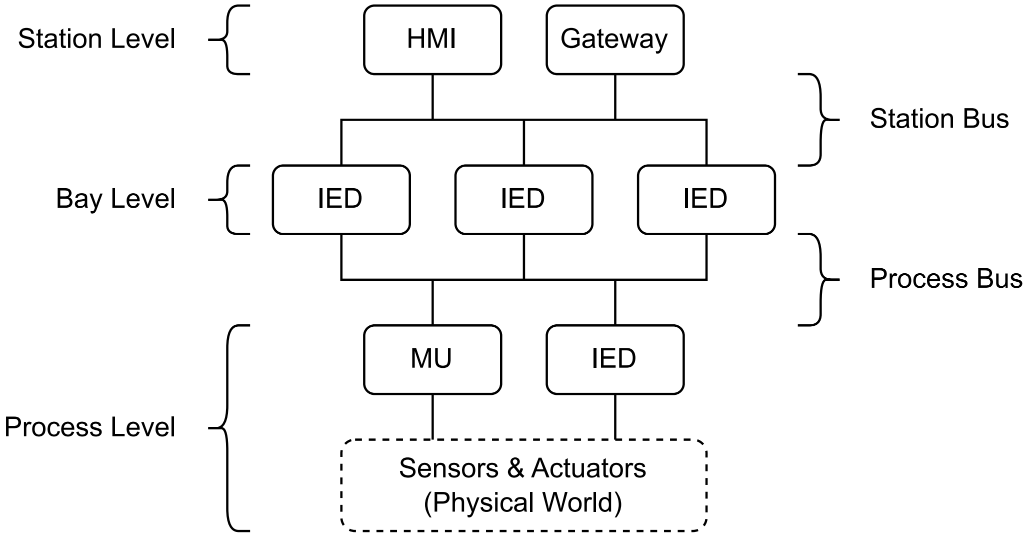


**Figure 4.1:** The internal SAS architecture consisting of three layers called system level, bay level, and process level which are connected via station bus and process bus.

#### 4.1.2.1 Classification Characteristics

The communication relationships between devices can be classified using different communication characteristics. 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 (subsystem) internal communication, whereas communication relationships including an external device are classified as (subsystem) external communication. Furthermore, a communication relationship is not limited to a single receiver (unicast) but rather a group of devices (multicast) or all devices (broadcast) may receive a sender's message.

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 might 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.

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 times and transmission times required to exchange information between involved devices. The transmission time is the time required to transmit a message over a network link with a specific throughput. The processing time represents the time required for a device to send, forward, or receive a message or packet. 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 or packet the processing time consists of enqueue and dequeue delays, cryptographic overhead, and message coding. As a consequence, the communication latency represents the time required for a message from being put into the sending buffer at the sender to the point when the message is taken from the receiving buffer at the receiver.

#### 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 standard [32]. The defined message type 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 [32] message types 1A and 4. The low latency messages are used for SAS-internal exchange of sampled values and state values. In IEC 61850 compliant substations the sampled values are exchanged using the Sampled Values (SV) protocol between MUs and IEDs (vertical) or between MUs (horizontal). Moreover, state values and state changes are exchanged horizontally between IEDs using the Generic Object Oriented Substation Events (GOOSE) protocol.

The medium latency message type corresponds to the IEC 61850 message types 1B and 2. The medium latency messages are used for internal and external as well as horizontal and vertical session-based client-server communication. In IEC 61850 substations IEDs use the Manufacturing Message Specification (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 61850 substations MMS as well as SCADA protocols are used for high latency communication.

**Table 4.1:** Message types of the presented system model classified with regard to their topology, continuity, and latency constraints of the communication relationships.

| 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 |

#### 4.1.2.3 Communication Buses

The presented system model uses a bus-based approach for SAS-internal message exchange between the system architecture layers. The implementation of SAS-internal buses is typically based on Ethernet and open or proprietary fieldbus technology. The bus-based approach as well as the two concrete buses introduced in the following are based on the IEC 61850 standard [32].

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. GOOSE and SV are the IEC 61850 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.

### 4.1.3 Requirements

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

#### 4.1.3.1 Security

The system model has to satisfy six requirements related to information security. The security requirements of the system model are defined as follows:

**RQ.SEC.1** Integrity
The system detects unauthorized manipulation of stored and exchanged data [20].

**RQ.SEC.1A** Device Integrity
The system detects unauthorized manipulation of data that is stored on system devices.

**RQ.SEC.1B** Message Integrity
The system detects unauthorized manipulation of data that is exchanged between system devices.

**RQ.SEC.2** Authenticity
The system can proof the authenticity and trustworthiness of subjects and data objects present in the system [20].

**RQ.SEC.3** Access Control
The system prohibits unauthorized access to sensitive information stored on devices.

**RQ.SEC.4** Non-Repudiation
The system ensures that a subject cannot dispute its authorship of data and requests [20].

**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 [65].

**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 [65].

#### 4.1.3.2 Safety

The system model has to satisfy two safety-related requirements. The safety requirements of the system model are defined as follows:

**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 [61]. In other words, the system never transitions into an unsafe state.

#### 4.1.3.3 Availability

The system model has to satisfy two availability-related requirements. The availability requirements of the system model are defined as follows:

**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 [61].

#### 4.1.3.4 Performance

The system model has to satisfy three requirements related to system performance. The performance requirements of the system model are defined as follows:

**RQ.PER.1** Communication Latency
The system must respect the latency constraints for network communication defined in subsubsection 4.1.2.2. Consequently, the maximum communication latency for low latency message exchange on the data path must not exceed three milliseconds.

**RQ.PER.2** Computational Complexity
The system must respect the limited performance of resource-constrained devices in the SAS. Consequently, computationally complex algorithms have to be executed by performance-oriented servers.

**RQ.PER.3** Energy & Power Saving
The system must respect the energy and power constraints of resource-constrained devices in the SAS.

#### 4.1.3.5 Compatibility

The system model has to satisfy two requirements related to system compatibility. The compatibility requirements of the system model are defined as follows:

**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 [32].

**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 [32].

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

To satisfy the requirements of the above-mentioned system model, 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 approach is a cryptosystem that provides security policies, cybersecurity mechanisms, and cryptographic algorithms and schemes. 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 Certificateless Attribute-Based Server-Aided Authentication (CASA). This concept represents the foundation of the CASC-SAS approach. The concept provides cryptographic algorithms and schemes for authentication including a public key cryptography 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, non-repudiation, and privacy preservation. This core concept is further discussed in section 4.3.

The second core concept of the approach is the Server-Aided Attribute-Based Authorization and Access Control (SABAAC). This concept provides mechanisms to enable attribute-based authorization and ABAC for time-critical SAS communication. The concept represents a more abstract cybersecurity means to provide access control, PoLP, and SoD. For this purpose, the concept relies on authentication services provided by CASA. This concept is further discussed in section 4.4.

### 4.2.1 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 subsubsection 4.2.1.1. Moreover, the two paths of the architecture are further discussed in subsubsection 4.2.1.2.

#### 4.2.1.1 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 rather bypass lower layers. The four layers of the CASC-SAS architecture and their provided services are defined in the following sections.

**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.

**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.

**Layer 1: Cryptography** The cryptography layer provides cryptographic algorithms and schemes to higher layers of the architecture. It relies on reliable and unreliable control message exchange. 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, non-repudiation, and privacy preservation.

**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 network stack.

**Example: Domain-Specific Communication** An exemplary domain-specific communication between a sending entity and receiving entity is shown in Figure 4.2. 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 exchange resulting from an invocation may occur bidirectional. The presented domain-specific communication is initiated by entity A. Therefor, entity A creates a single or multiple domain-specific messages and delivers them to the cybersecurity layer. The yet unsigned and non-authorized messages are then authorized by SABAAC and forwarded to CASA at the cryptography layer for signing. Subsequently, the signed and authorized messages are forwarded to the receiver using the reliable and unreliable network transport services provided by the message exchange layer. Upon arrival at the receiver, the message exchange layer delivers the signed and authorized messages 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 access control policies by verifying the validity of the message authorization. Moreover, the message is forwarded to the cryptography layer for cryptographic verification.

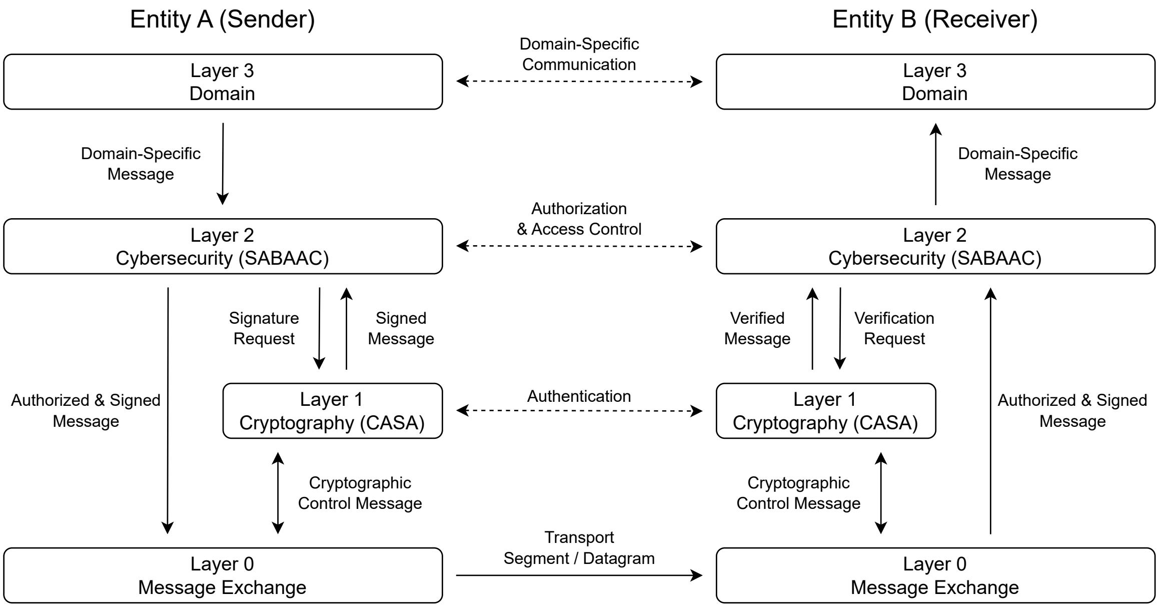


**Figure 4.2:** An exemplary domain-specific communication between a sending and receiving entity including the involved CASC-SAS layers and messages exchanged between the architectural layers.

#### 4.2.1.2 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 used for the forwarding of domain-specific payload from a sending entity to a receiving entity. Besides the concrete domain-specific messages, communication 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 domain-specific payload. 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.

## 4.3 Certificateless Attribute-Based Server-Aided Authentication (CASA)

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. The goal of CASA is to provide cryptographic algorithms and schemes for key generation, key distribution, key revocation, 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.

Since CASA is a CL-PKC approach, neither certificates nor key escrow is required [54]. Moreover, the CASA approach proposes a key generation that is not only based on subject identities but rather enables public keys and private keys based on arbitrary attributes of subjects or even groups of subjects. The key generation of the CASA approach is inspired by the alternative CL-PKC key generation technique proposed by Al-Riyami and Paterson [54]. The defining characteristics of the alternative key generation is the derivation of partial private keys from public keys and identities. As a consequence, an entity has to generate its public key before it can request a partial private key from the KGC. This alternative key generation enables sending of partial private keys over unsecure channels and reduces the required trust in the KGC. Furthermore, this technique allows only one public key to be created for a specific private key.

### 4.3.1 Server-Aided Cryptography

As PKC mechanisms may consist of computationally complex algorithms and operations such as bilinear pairing, we propose a server-aided PKC approach. Therefor, we propose an extension of the CL-PKC concept and schemes to make time-critical steps server-aided. To make CASA server-aided, an Untrusted Cryptography Server (UCS) supports devices by handling computationally expensive algorithms instead of executing them locally on resource-constrained devices. To minimize the required trust, the UCS may only handle certain computations, i.e., partially sign or verify a request of a device. This server-aided approach enables resource-constrained devices to apply secure algorithms and schemes of CASA 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. [67], 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.

### 4.3.2 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 to reduce the required time for cryptographic algorithms.

### 4.3.3 Signature Scheme $\mathcal{S}_{CASA}$

The CASA signature scheme $\mathcal{S}_{CASA} = (I, G_{VAL}, G_{PK}, G_{PPK}, G_{SK}, S, V_{ENT}, V_{SAV}, V_{FIN})$ is a nine-tuple of algorithms. The algorithms comprise an initialization algorithm $I$, a secret value generation algorithm $G_{VAL}$, a public key generation algorithm $G_{PK}$, a partial private key generation algorithm $G_{PPK}$, a private key generation algorithm $G_{SK}$, a signing algorithm $S$, a partial entity verification algorithm $V_{ENT}$, a partial server verification algorithm $V_{SAV}$, and a final entity verification algorithm $V_{FIN}$. In the following sections, the specific algorithms are further discussed.

The definition of the CASA signature scheme is based on the definition of digital signature schemes provided by Boneh and Shoup [9]. Moreover, since CASA is a CL-PKC approach, the signature scheme has been adapted from the schemes and concepts presented by Al-Riyami and Paterson [54] and Ramadan, Elbez, and Hagenmeyer [52]. The proposed server-aided verification concept and algorithms are inspired by schemes proposed by Ramadan et al. [53], Girault and Lefranc [25] and Wu et al. [67].

#### 4.3.3.1 Initialization Algorithm $I$

The initialization algorithm $(\rho, s) \leftarrow I(\lambda)$ takes the security parameter $\lambda$ as input and outputs the public system parameters $\rho$ and the private master key $s$. The initialization algorithm is executed by the KGC. After the execution, $\rho$ is publicly available to all entities, whereas $s$ is only known by the KGC.

#### 4.3.3.2 Secret Value Generation Algorithm $G_{VAL}$

The secret value generation algorithm $\chi_A \leftarrow G_{VAL}(\rho)$ takes the public system parameters $\rho$ as input and outputs the secret value $\chi_A$ of an entity $A$. The secret value generation algorithm is executed by an entity. The secret value $\chi_A$ is never shared with other entities and may only be known to entity $A$.

#### 4.3.3.3 Public Key Generation Algorithm $G_{PK}$

The public key generation algorithm $pk_A \leftarrow G_{PK}(\rho, \chi_A, ATT_A)$ takes the public system parameters $\rho$, the secret value of an entity $\chi_A$, and the defining attributes of entity $A$ $ATT_A$ as input. The algorithm outputs the public verification key $pk_A$ of entity $A$. The algorithm is executed by an entity.

#### 4.3.3.4 Partial Private Key Generation Algorithm $G_{PPK}$

The partial private key generation algorithm $ppk_A \leftarrow G_{PPK}(\rho, s, ATT_A, pk_A)$ takes the public system parameters $\rho$, the private master key of the KGC $s$, the defining attributes of entity $A$ $ATT_A$, and the $A$'s public verification key $pk_A$ as input. The algorithm outputs the partial private key $ppk_A$ of entity $A$. The partial private key generation is executed by the KGC on request of an entity. After the execution, the KGC provides the partial private key to the corresponding entity.

#### 4.3.3.5 Private Key Generation Algorithm $G_{SK}$

The private key generation algorithm $sk_A \leftarrow G_{SK}(\rho, \chi_A, ppk_A)$ takes the public system parameters $\rho$, the secret value of an entity $\chi_A$, and the partial private key $ppk_A$ of entity $A$ as input. The algorithm outputs the private signing key $sk_A$ of entity $A$.

#### 4.3.3.6 Signing Algorithm $S$

The signing algorithm $\sigma \leftarrow S(sk_A, m)$ takes the private signing key $sk_A$ and message $m$ as input, and outputs the signature $\sigma$. In other words, the signing algorithm $S$ is used by the sender $A$ of a message $m$ to generate a digital signature $\sigma$. The generated digital signature $\sigma$ is associated with the message $m$ and the sender's private signing key $sk_A$.

#### 4.3.3.7 Partial Entity Verification Algorithm $V_{ENT}$

The partial entity verification algorithm $\sigma_{ENT} \leftarrow V_{ENT}(\rho, pk_A, m, \sigma)$ represents the first step of the server-aided verification process. The algorithm takes the public system parameters $\rho$, the public verification key $pk_A$ of entity $A$, the message $m$, and the signature $\sigma$ as input. The algorithm outputs the partially verified signature $\sigma_{ENT}$. The receiver of a message executes the partial entity verification algorithm $V_{ENT}$ and sends the partially verified signature $\sigma_{ENT}$ to the UCS for server-aided verification.

#### 4.3.3.8 Partial Server Verification Algorithm $V_{SAV}$

The partial server verification algorithm $\sigma_{SAV} \leftarrow V_{SAV}(\sigma_{ENT})$ represents the second step of the server-aided verification process. The algorithm takes the partially verified signature $\sigma_{ENT}$ as input and outputs the partially verified signature $\sigma_{SAV}$. The algorithm is executed by the UCS on request of a message receiving entity. After the execution, the UCS provides the partially verified signature $\sigma_{SAV}$ to the requestor.

#### 4.3.3.9 Final Entity Verification Algorithm $V_{FIN}$

The final entity verification algorithm $\delta \in \{accept, reject\} \leftarrow V_{FIN}(\rho, pk_A, m, \sigma, \sigma_{ENT}, \sigma_{SAV})$ represents the third and last step of the server-aided verification process. The algorithm takes the public system parameters $\rho$, the public verification key $pk_A$ of entity $A$, the message $m$, the signature $\sigma$, and the partially verified signatures $\sigma_{ENT}$ and $\sigma_{SAV}$ as input. The algorithm outputs the verification decision $\delta$ which is either $accept$ or $reject$. In other words, the verification algorithm $V_{FIN}$ is used by the receiver to verify a received message $m$ sent by entity $A$ based on an appended signature $\sigma$. As $\sigma$ is associated with the message $m$ and the sender's private signing key $sk_A$, it allows the receiver to verify the integrity and authenticity of the received message $m$ using the sender's public verification key $pk_A$.

### 4.3.4 Security Model

The proposed signature scheme $\mathcal{S}_{CASA}$ is a secure signature scheme if it is existentially unforgeable under an adaptive chosen-message attack (EUF-CMA) [9, 26]. To create an existential forgery, i.e., output a valid pair of message and signature for a new message, an adversary carrying out a CMA can request valid signatures from an entity for any message of his choice. While non-adaptive CMA restricts the adversary to a fixed set of messages chosen prior to the attack, adaptive CMA allows the adversary to request signatures of messages depending on previously obtained signatures.

## 4.4 Server-Aided Attribute-Based Authorization & Access Control (SABAAC)

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 but 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 cybersecurity concept that is located on the cybersecurity layer of the CASC-SAS architecture. Since the approach is located on the cybersecurity layer, 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 4.4.1. Furthermore, the approach is divided into two central tasks or protocols. 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, and distribution. This process partially takes place prior to occurring access requests and corresponding access decisions. The delegated attribute-based authorization protocol is further discussed in subsection 4.4.3. The second central task is referred to as delegated ABAC. The delegated ABAC is responsible for the policy decision and policy enforcement. This process takes place when an entity initiates the communication with another entity. The delegated ABAC protocol is further discussed in subsection 4.4.4. An overview of the SABAAC architecture, components, and protocols as well as the integration of CASA components and services into the SABAAC approach are shown in Figure 4.3.

### 4.4.1 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. [30]. Each functional unit is represented by a component that offers function-oriented services. The components of the architecture are TTPs since the semantic validity of their provided services is not verifiable by the service consumers. We specify the components as semi-trusted for the access control problem due to the restriction to access control tasks and available mitigation approaches such as multiple instances of a single component. The four components of the architecture are discussed in the following:

**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. Therefor, 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.

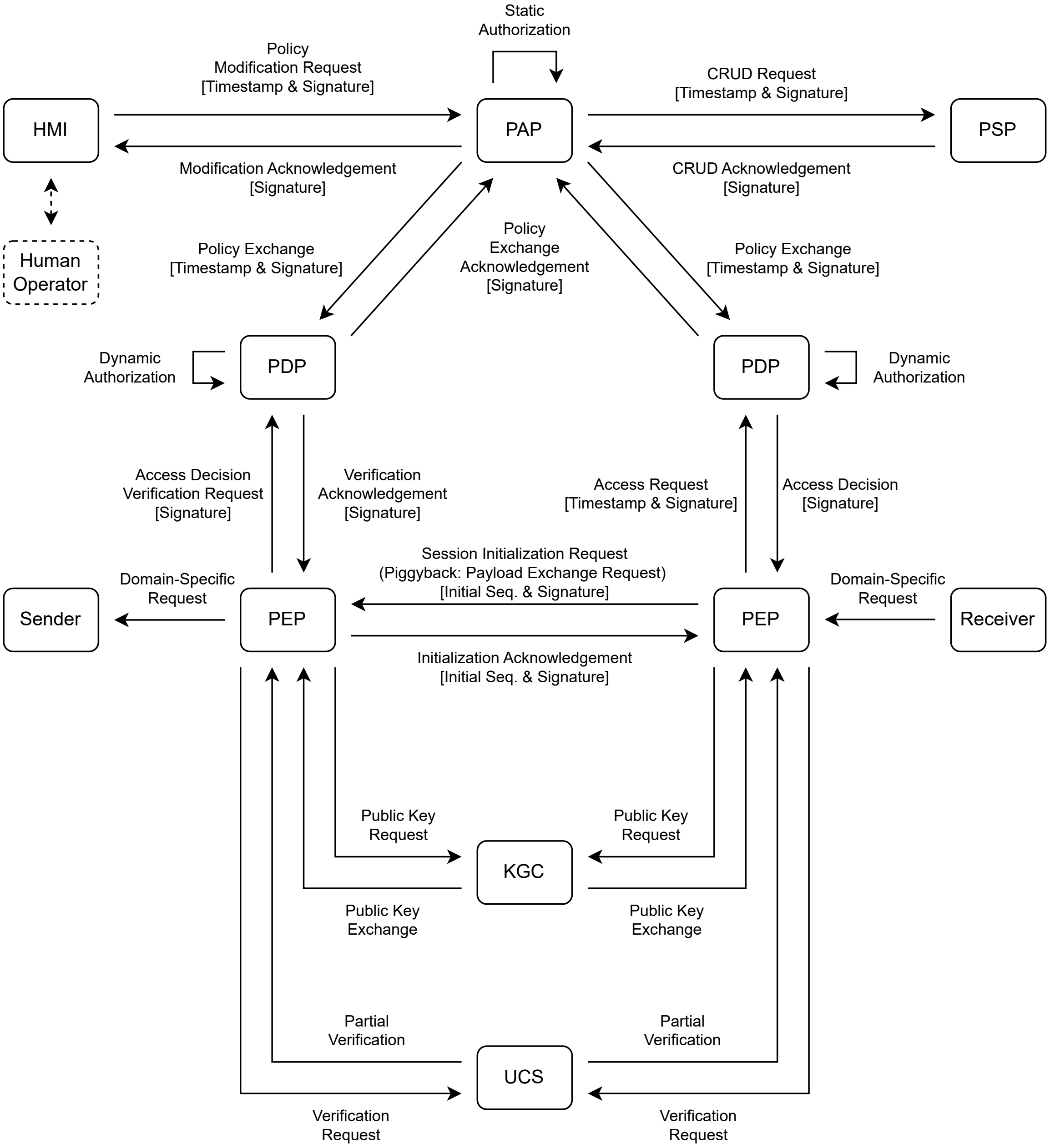


**Figure 4.3:** Component-based architecture and communication protocols of the SABAAC approach, including the four SABAAC components, their interrelationships, and the integration of CASA components and services into the authorization and access control workflow.

**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. [30]. Therefore, the PDP not only takes access control decision on request but is rather 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) introduced by Hu et al. [30]. Accordingly, the SABAAC PEP provides attributes related to its protected objects to the PDP.

### 4.4.2 Access Control Policy

According to Hu et al. [30], 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 benefits: 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. Moreover, as stated by Hu et al. [30], the use of ABAC can avoid explicit authorizations prior to a request. 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, Magkos, and Chrissikopoulos [11], 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 $\lambda_a$ of a real-time attribute $a$ is defined by $E_{ATT}(a, t) = \lambda_a$. To handle policies based on their degree of time-variability, the SABAAC approach classifies policies as follows:

**Dynamic Policy** A dynamic policy $\rho$ is an ABAC policy whose evaluation relies on at least one time-variable subject, object, environment, or action attribute. A policy $\rho$ is dynamic iff $\exists a \in \rho : \forall t_i \neq t_j : E(a, t_i) \neq E(a, t_j)$. 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 $\rho$ is an ABAC policy whose evaluation does not rely on time-variable subject, object, environment, or action attributes. A policy $\rho$ is static iff $\forall a \in \rho : \forall t_i, t_j : E(a, t_i) = E(a, t_j)$. 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.

### 4.4.3 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 [20]. According to Eckert [20], 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, management, storage, and distribution. For this purpose, the authorization protocol provides policy management services for human operators at the PAP. Moreover, the authorization protocol provides services for the exchange of policies between the PAP, PSP, and PDP.

The delegated attribute-based authorization protocol is a message-based protocol. It uses digital signatures and verification provided by CASA to verify the integrity and authenticity of messages. Moreover, the services provided by the authorization protocol use reliable and acknowledged transport layer protocols for the message exchange. Since the authorization protocol is delegated, authorizations are not managed by the protected devices, but management is rather delegated to a TTP. In the SABAAC approach this TTP is represented by the PAP, PSP, and PDP. Accordingly, the protected objects or devices delegate the authorization process to these SABAAC components. Therefore, the authorization can also be referred to as server-aided authorization.

Since no cybersecurity layer sessions are established and consequently neither nonces nor sequence numbers can be used, message exchange of the authorization protocol is vulnerable to replay attacks. Therefore, we propose using timestamp-based replay protection for authorization messages. At the sender, timestamp-based replay protection can be achieved by adding a timestamp to the message and digitally signing the message and timestamp together. The message is then sent over the network to the receiver. The receiver only processes the message payload if its timestamp has a reasonable deviation from the current time. Moreover, if the receiver keeps record of the last timestamp received from each sender, the receiver can acknowledge duplicated and older packets without processing them.

Authorizations in the SABAAC approach are realized using ABAC policies. As discussed in subsection 4.4.2, access control policies are classified as either static or dynamic. Consequently, the authorization protocol has to take the time-variability into account. For this purpose, the authorization protocol consists of three parts or sub-protocols:

**Static Authorization** The static authorization process is responsible for the creation, modification, and deletion of policies. The static authorization is initiated by a human operator. The human operator sends a policy creation, modification, or deletion request to a PAP. The PAP then translates the request into a valid SABAAC policy or policy modification and sends it to the PSP in the form of a CRUD request. Finally,

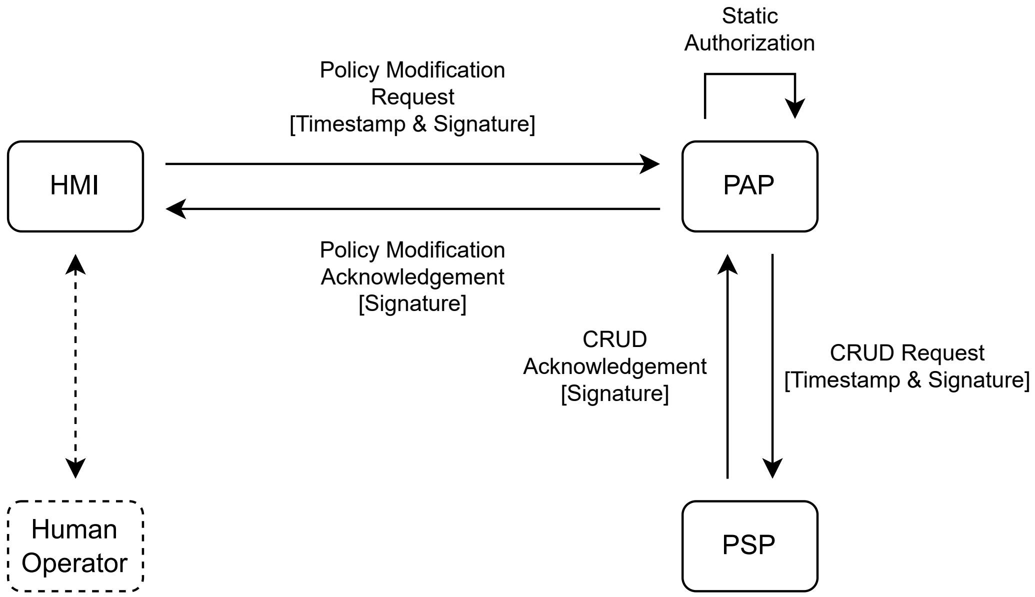

Static Authorization
Policy Modification Request [Timestamp & Signature]
HMI
PAP
Policy Modification Acknowledgement [Signature]
CRUD Acknowledgement [Signature]
CRUD Request [Timestamp & Signature]
Human Operator
PSP

**Figure 4.4:** Components and exchanged messages of the static authorization process.

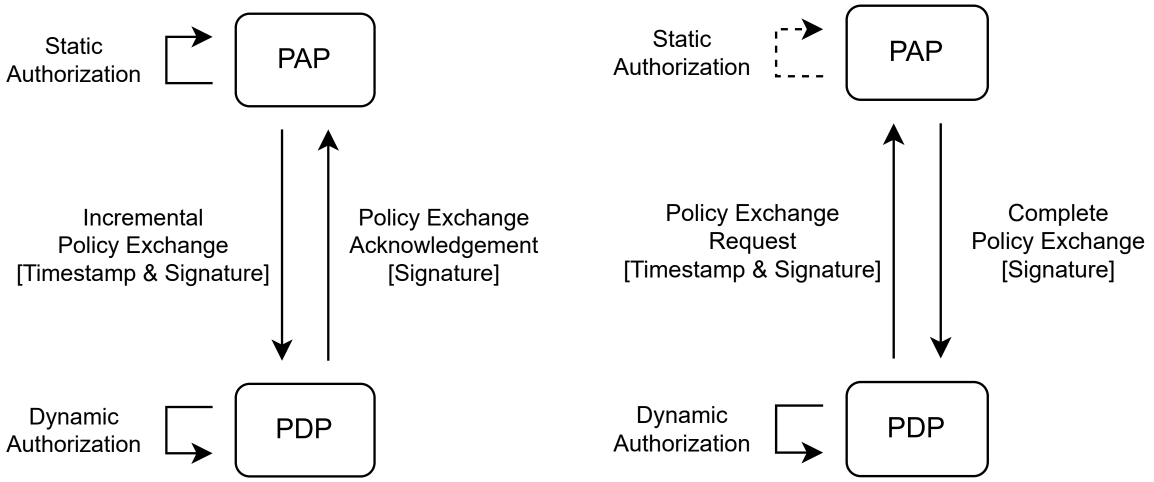
the PSP persists the created or modified policy and sends a response to the PAP. The static authorization process, the involved components, and the occurring message exchanges are visualized in Figure 4.4.

**Policy Exchange** After a policy is created or modified via static authorization, it is shared with the PDPs via policy exchange. The policy exchange is an interaction between a PAP and a PDP either initiated by a static authorization or on request of the involved PDP. The static authorization is a prerequisite for both policy exchange procedures. The policy exchange is a prerequisite for dynamic authorization at a PDP. A repeatedly executed dynamic authorization process may be initially triggered by a policy exchange.

In the event that a static authorization triggers the policy exchange, the PAP sends a policy exchange message to a PDP, which contains the newly created or modified policy. This type of policy exchange is an incremental process. Consequently, only newly created or modified policies are exchanged. The incremental policy exchange is shown in Figure 4.5a.

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 4.5b.

**Dynamic Authorization** The dynamic authorization process is responsible for the evaluation of SABAAC policies. The dynamic authorization process is executed by the PDP. The goal of the process is to compute whether a certain subject is authorized for a specific request. Therefor, the PDP retrieves the attributes related to a policy and derives an access decision. The dynamic authorization is either executed ad-hoc to respond to an access control request or executed prior to a request. The former execution strategy


Static Authorization
PAP
Incremental Policy Exchange [Timestamp & Signature]
Policy Exchange Acknowledgement [Signature]
Dynamic Authorization
PDP
Static Authorization
PAP
Policy Exchange Request [Timestamp & Signature]
Complete Policy Exchange [Signature]
Dynamic Authorization
PDP

**(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 4.5:** Components and exchanged messages of the incremental and complete policy exchange procedures.

has less management overhead, since it requires no caching of policy evaluations. On the contrary, ad-hoc policy evaluation increases the response time for access control requests. The latter execution strategy evaluates policies prior to a request and caches the corresponding access decision. Consequently, this precomputation strategy enables low latency communication by reducing the response time for access control requests.

### 4.4.4 Delegated Attribute-Based Access Control Protocol

In the following, we discuss the delegated attribute-based access control protocol of the SABAAC approach. The goal of the access control protocol is the exchange and enforcement of access control decisions. The access control decisions are derived from the dynamic authorization process discussed in subsection 4.4.3. Since the PDP instances execute the dynamic authorization process, the provisioning of access decisions is delegated to the PDP instances as well. In other words, the PDP instances compute access decision using dynamic authorization and provide these decisions to devices using the access control protocol. Moreover, devices including requesting subjects and protected objects delegate the enforcement of access decisions to trusted PEP instances.

The access control protocol of the SABAAC approach uses digitally signed and partially acknowledged message exchange. Similar to the authorization protocol, it uses digital signatures and verification provided by CASA to verify the integrity and authenticity of messages. To exchange messages on the control path, the access control protocol relies on reliable transport services provided by the message exchange layer. Data path message exchange of the access control protocol relies on either reliable or unreliable transport services depending on requirements such as message integrity and congestion tolerance. Moreover, the protocol is based on discontinuous message-based communication as well as continuous session-based communication at the cybersecurity layer. The access control sessions are initialized at the sending and receiving entities prior to the exchange of domain-specific messages.

The security of the access control protocol is vulnerable to three distinct threats. Due to its communication architecture the protocol is vulnerable to replay, Denial-of-Service (DoS), and collusion attacks. Replay of messages represents a threat for session-based and message-based communication. Session-based communication of the protocol uses session identifiers and sequence numbers to provide replay protection. Message-based communication employs timestamp-based replay protection, similar to the discontinuous communication used by the authorization protocol. DoS attacks represent a threat for session-based communication due its stateful concept including session initialization and management. To mitigate malicious DoS attacks and DoS due to configuration and system errors, the access control uses a soft-state session-based message exchange. Accordingly, the session states used by the communicating entities have to be refreshed periodically and unused or invalid states are deleted. As a consequence, devices may not make assumptions about the current state of other devices. Moreover, a device has to handle session state exceptions via reinitialization of access control sessions. The third type of security threat is the collusion of malicious PDP and PEP instances. A collusion attack occurs if a PDP forges an access decision that is used by a PEP to access a protected object in an unauthorized manner. To mitigate collusion and reduce the trust in a PDP, a PEP can use server-aided access decision verification. Therefor, a PEP can request the verification of a PDP access decision from another PDP.

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 the access request phase is to exchange an access decision between a PDP and PEP. The access request phase is initiated by a PEP instance on behalf of a domain subject. Therefor, the PEP sends an access request including a digital signature and timestamp to a PDP. The PDP verifies the request based on the signature and timestamp. The PDP then derives an access decision from the access request and the dynamic authorization process. The access decision can either grant or deny the requested access. Finally, the signed access decision is returned to the requesting PEP. A PEP instance can either initiate the access request ad-hoc as soon as a domain request arrives or execute the access request prior to an arriving domain request. The latter option reduces the time before domain-specific payload can be transmitted to the receiving devices.

To take the time-variability of access control policies into account, the exchanged access decision has to contain a period of validity. As long as an access decision is valid, it can be cached and reused by the PEPs. Moreover, the access decision has to contain parameters to uniquely identify the access request. These request parameters are used at the sending and receiving PEPs to identify the access control session. Since arbitrary header-specific and payload-specific information can be used for session identification, the SABAAC approach defines the mandatory identification parameters. Domain-specific message exchange that is based on data link layer protocols such as Ethernet is at least identified by a triple of parameters. The identifying triple consists of source address, destination address, and communication protocol. Domain-specific message exchange that is based on transport layer protocols such as the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP) is at least identified by a five-tuple of parameters. The identifying five-tuple consists of source address, source port, destination address, destination port, and communication protocol.

**Session Initialization** The session initialization is executed by a PEP instance that is granted access via access request phase. To initialize an access control session, a PEP sends a session initialization request to another PEP. The digitally signed request has to contain a PDP-signed granted access decision and an initial sequence number. Moreover, the PEP may send the initialization request along with a domain-specific request 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 verifies the signature of the request, the signature of the access decision, and its period of validity. The PEP may optionally use server-aided access decision verification that is further discussed in the following section. If the received request is valid, the PEP initializes a session state, sends a positive initialization acknowledgement to the requestor, and starts processing piggybacked domain-specific requests. If the request is invalid, the PEP sends a negative initialization acknowledgement to the requestor and discards the piggybacked domain-specific request.

A successful session initialization between two PEPs is shown in Figure 4.6. Besides the session initialization procedure, the figure visualizes a preceding access request. The access request procedure relies on an interaction between a PDP and a PEP and is a prerequisite for the session initialization. As illustrated in the figure, the session initialization relies on an interaction between two PEP instances. Moreover, a server-aided access decision verification at the receiving PEP is shown. The receiver's PEP uses a server-aided access decision verification provided by a PDP to reduce the trust in another PDP. Additionally, the shown SABAAC components, especially resource-constrained PEP instances, rely on the services provided by a CASA UCS for SAV.

**Access Decision Verification** The optional server-aided access decision verification enables a PEP to verify received access decisions. Therefor, the PEP sends a verification request containing the PDP-signed access decision to another PDP. The PDP verifies the access decision and sends a positive or negative verification acknowledgement back to the requestor.

**Payload Exchange** The payload exchange phase is the final phase of the access control protocol. This phase can be repeatedly executed until the period of validity of an underlying access control session ends. The goal of the payload exchange phase is to securely exchange domain-specific requests between a sender and a receiver

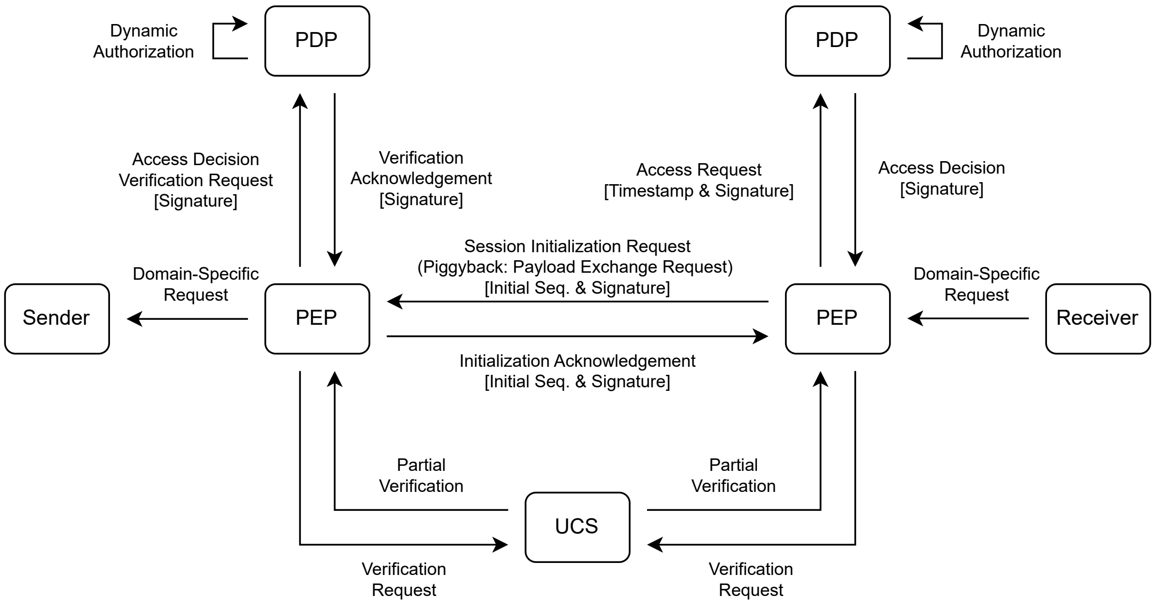

Dynamic Authorization
PDP
PDP
Dynamic Authorization
Access Decision Verification Request [Signature]
Verification Acknowledgement [Signature]
Access Request [Timestamp & Signature]
Access Decision [Signature]
Session Initialization Request (Piggyback: Payload Exchange Request) [Initial Seq. & Signature]
Domain-Specific Request
Domain-Specific Request
Sender
PEP
PEP
Receiver
Initialization Acknowledgement [Initial Seq. & Signature]
Partial Verification
Partial Verification
UCS
Verification Request
Verification Request

**Figure 4.6:** Components and exchanged messages of a successful access request, unidirectional session initialization, and access decision verification.

by employing authentication, authorization, and access control mechanisms. The payload exchange is initiated by a domain entity via delivery of a domain-specific request to its PEP. On receipt of a domain-specific request, the sender's PEP appends a valid session-specific sequence number to the request. This extended request is then digitally signed and sent to the receiver's PEP. On receipt of the extended request, the receiving PEP derives the session identifying parameters from the request. Based on these parameters, the PEP verifies that a valid access control session for the received domain-specific request exists. Moreover, the PEP verifies the sequence number and signature of the extended message. If the received extended request is valid, the receiving PEP extracts the domain-specific request and delivers it to the corresponding domain entity.

A successful unidirectional payload exchange between a sender and receiver is shown in Figure 4.7. The shown procedure relies solely on the interaction of PEP instances, with the exception of a single server-aided CASA verification on message receipt.

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. Moreover, the payload exchange phase relies on a Negative Acknowledgement (NACK) concept A NACK is sent in case of verification or authorization exceptions. The NACK concept avoids acknowledgement implosions in multicast and broadcast communication scenarios. A received NACK may trigger a session reinitialization workflow at a PEP.

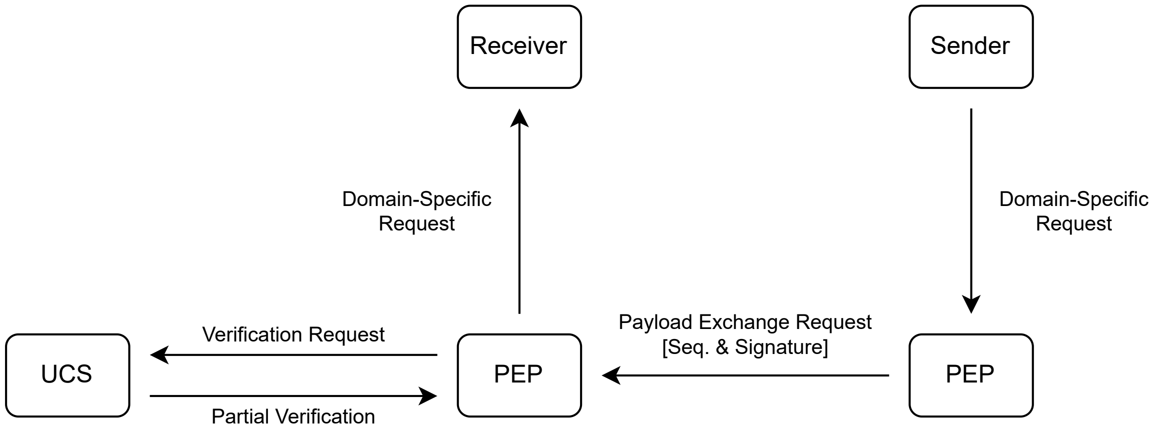
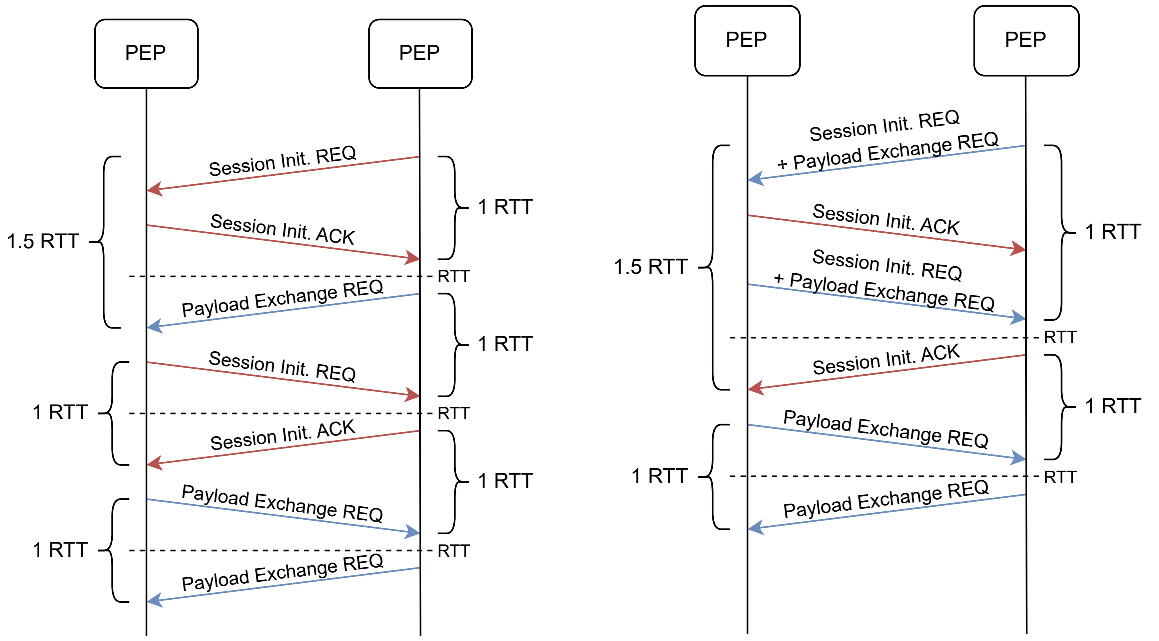


**Figure 4.7:** Components and exchanged messages of a successful unidirectional payload exchange procedure between two domain entities.

To reduce the overhead of session initialization handshaking, the requesting PEP may send the session initialization request along with a domain-specific request by piggybacking a payload exchange request. The processing of piggybacked payload exchange requests starts as soon as the received initialization request is successfully verified. If an initialization request is invalid, the PEP discards the piggybacked payload exchange request. The usage of piggybacking decreases the required time until a domain-specific request arrives at a PEP if no session was initiated prior to the request. Since session initialization requires at least one RTT for handshaking, a non-piggybacked domain-specific request is delayed by at least one RTT. Moreover, due to the reason that session initialization is handled unidirectionally, handshaking leads to at least two RTT delay for bidirectional domain-specific communication.

A simplified session initialization procedure between two PEP instances is visualized in Figure 4.8. For RTT comparison purposes, the semantically identical session initialization handshakes and message exchanges are shown without and with piggybacking of payload exchange requests. A session initialization procedure without piggybacking is shown in Figure 4.8a, whereas Figure 4.8b visualizes the semantically identical procedure using piggybacked requests. As shown in Figure 4.8a, 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 two RTTs to a half RTT 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 for both scenarios.



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

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

**Figure 4.8:** Protocol sequence diagrams showing a piggybacking and non-piggybacking initialization of bidirectional message exchange sessions between two PEP instances.

## 4.5 Realization

In the following section, we discuss the proposed realization of the CASC-SAS approach and its two core concepts CASA and SABAAC. The approach and its two concepts introduce components that are defined and discussed in section 4.3 and section 4.4. The adaptation of the layered SAS architecture to the CASC-SAS approach entails a modification of the architecture to accommodate these additional components. This integration of CASC-SAS components into the SAS architecture is visualized in Figure 4.9. The components depicted in blue represent elements of the SAS architecture, whereas the components depicted in red are introduced by the CASC-SAS approach. The components with a color gradient represent elements of the SAS architecture that have been adapted to support CASC-SAS concepts. The PEP, PDP, UCS, and KGC components have to be present locally in every adapted SAS. This is necessary due to the strict time constraints of SAS-internal low latency message exchange. The PAP and PSP instances may be centrally deployed, since static authorization is part of the non-time-critical control path communication. Any non-intermediate SAS component that participates in a communication relationship must either support the CASC-SAS protocols or use the services provided by a PEP to secure occurring message exchanges.

To examine the feasibility and perform the above-mentioned integration, we propose a hardware and software realization of our approach. The aim of the realization is to provide implementations of CASA and SABAAC with a full range of functions. The software is going

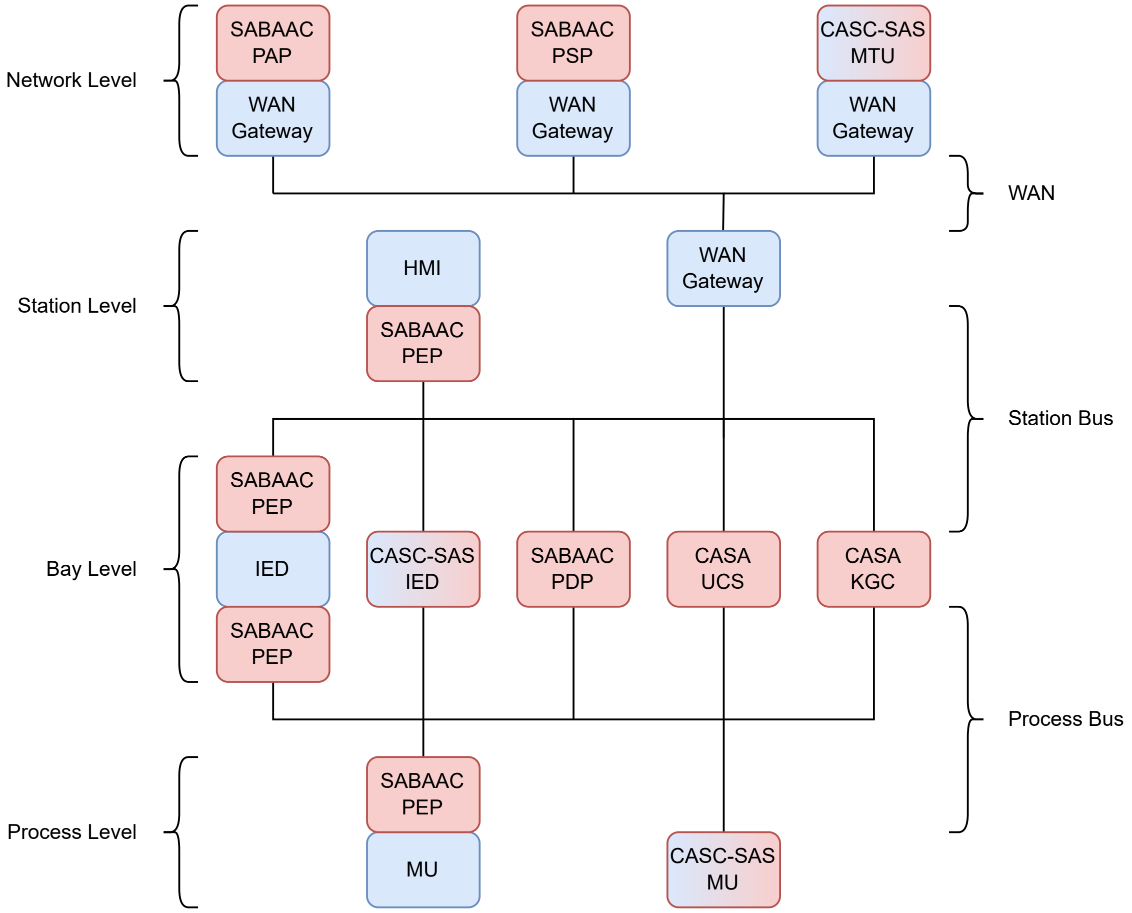


**Figure 4.9:** Adaptation of the layered SAS architecture to the CASC-SAS approach.

to be implemented component-wise using high-level programming languages. Depending on the complexity and time constraints of a specific component, different programming languages might be used for the implementation. Due to internal dependencies, the software implementation is split into two parts. The first part is dedicated to the CASA approach, as it represents the foundation of all employed protocols. The second part is dedicated to the SABAAC components and protocols.

The employed components of CASA and SABAAC provide different services. Furthermore, the components have to be deployed differently to correspond to the proposed protocols. While PAP and PSP may be deployed centrally, components such as PEP, PDP, UCS, and KGC are distributed to individual SAS instances. This leads to differing hardware requirements for the implemented components. Moreover, 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 provided by PEP instances via BITW pattern are automatically applied to captured packets. Therefore, these services are invisible to the corresponding service consumers. This BITW pattern is inspired by the security filter approach presented by Ishchenko and Nuqui [35]. Taking the differing provision patterns and deployment structures into account, we propose the usage of performance-oriented server hardware for the PAP, PSP, PDP, UCS, and KGC to avoid bottlenecks and mitigate the risk of accidental or malicious DoS. Moreover, we propose the usage of inexpensive off-the-shelf hardware for the highly distributed PEP instances.

## 4.6 Evaluation

In this section, the evaluation of the CASC-SAS approach is discussed. 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 proposed approach in the presented field of application. Moreover, the characteristics are used to identify limitations and future work of the proposed approach.

### 4.6.1 Strategy

The evaluation is performed theoretically as well as experimentally. For the theoretical parts of evaluation, formal and informal methods are used to proof certain characteristics of the proposed approach. The experimentally performed part of the evaluation is based on the realization presented in section 4.5. The component implementations are used to construct a network simulation and network test bed. The proposed network test bed is visualized in Figure 4.10. The components depicted in blue represent computers that mimic the behavior of domain-specific senders and receivers of an SAS. The components depicted in red are part of the CASC-SAS approach. The components depicted in yellow represent message-forwarding intermediate network devices.

These test environments mimic the behavior of a real interconnected SAS for occurring message exchanges. The simulation and test bed strategy have differing advantages and disadvantages. On the one hand, the simulation strategy has the advantage of repeatability and reproducibility due to deterministic behavior, whereas the behavior of the test bed is non-deterministic. On the other hand, the test bed results are practice-oriented and transferable to the physical SAS domain, whereas the behavior of a real SAS cannot be compared to the deterministic behavior of the network simulation.

### 4.6.2 Evaluation Areas & Metrics

The evaluation aims to derive quantitative and qualitative metrics for different areas of interest. We propose three areas of interest for the evaluation of the CASC-SAS approach. The three areas of interest and their corresponding metrics are defined as follows:

**Security Evaluation** Does CASC-SAS provide security against typical SAS adversaries and attacks?

1. Satisfied security, safety, and availability requirements

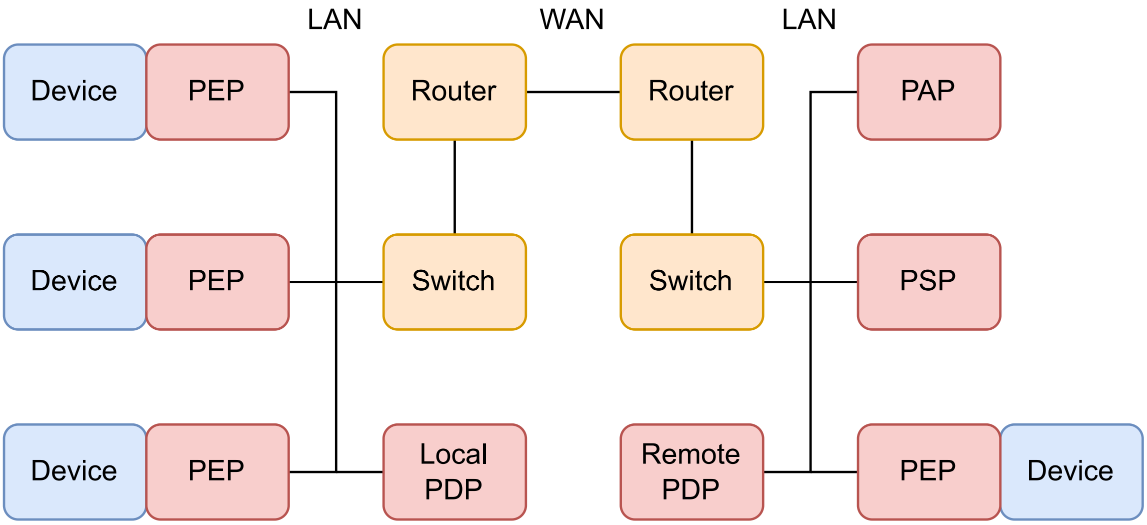


**Figure 4.10:** Architecture of the network test bed used for the experimental evaluation of the CASC-SAS approach.

2. Assumed system characteristics
3. Assumed adversary characteristics
4. Mitigated attacks
5. Introduced attack surface

**Performance Evaluation** Is CASC-SAS capable of securing time-constrained communication of an SAS?

1. Satisfied performance requirements
2. Assumed communication characteristics
3. Supported message types
4. Resistance against network exceptions including congestion, delay, jitter, duplicated packets, lost packets, and out-of-order packet delivery

**Economic Evaluation** Is CASC-SAS an economically feasible solution for the construction or retrofitting of an SAS?

1. Satisfied compatibility requirements
2. Assumed device requirements
3. Additional costs for SAS construction and retrofitting
4. Feasibility with regard to SAS retrofitting
5. Cost-benefit efficiency compared to alternative approaches

# 5 Project Plan

In the following section, the project plan of the proposed master's thesis is discussed. The project plan consists of a work plan, time schedule, and risk assessment. The work plan defines the project objectives, milestones, tasks, and deliverables and is presented in section 5.1. The time schedule represents the mapping of the proposed work plan to calendar weeks and is discussed in section 5.2. The risk assessment evaluates technical and non-technical risks for the proposed project plan and is discussed in section 5.3.

## 5.1 Work Plan

The work plan structures the proposed thesis into six milestones. Each milestone represents a major phase of the proposed master's thesis. The milestones serve as intermediate project goals which enable monitoring and reporting of project progress. Each milestone is defined by its duration in Calendar Weeks (CW), deliverables, and tasks. Each milestone consists of at least one task. A task of a milestone is referred to as increment. The duration of a milestone depends on the duration of its increments and increment interdependencies. The projected timescales for the completion of milestones and increments have been calculated based on empirical values. The milestones of the proposed master's thesis are defined as follows:

**Milestone I:** Preliminary Work

- **Duration:** 14 Weeks (15. April 2024 (CW 16) – 22. July 2024 (CW 30))
- **Deliverables:** Master's thesis proposal

**Milestone II:** Realization

- **Duration:** 9 Weeks (22. July 2024 (CW 30) – 23. September 2024 (CW 39))
- **Deliverables:**
  a) Software Design, Implementation, Tests (Unit, Integration, and System), & Test Coverage Report of CASC-SAS
  b) Hardware Deployment Scripts of CASC-SAS
  c) Thesis Chapter: Realization
- **Increments:**
  1) Design, Implementation, Tests, & Deployment of CASA (8 Weeks)
  2) Design, Implementation, Tests, & Deployment of SABAAC (8 Weeks)

3) Architecture & Code Review (Optional, Single Meeting)

4) Writing of Documentation (5 Weeks)

**Milestone III:** Evaluation

**Duration:** 9 Weeks (23. September 2024 (CW 39) – 25. November 2024 (CW 48))

**Deliverables:**
a) Evaluation Results of CASC-SAS
b) Thesis Chapter: Evaluation

**Increments:**
1) Security Evaluation (9 Weeks)
2) Performance Evaluation (9 Weeks)
3) Economic Evaluation (5 Weeks)
4) Writing of Documentation (4 Weeks)

**Milestone IV:** Conclusion

**Duration:** 2 Weeks (25. November 2024 (CW 48) – 09. December 2024 (CW 50))

**Deliverables:**
a) Thesis Chapter: Conclusion
b) Thesis Chapter: Limitations
c) Thesis Chapter: Future Work
d) Thesis Chapter: Abstract

**Increment:** Writing of Documentation: Conduct a review of results and derive a conclusion, limitations, and future work with regard to the research questions (2 Weeks)

**Milestone V:** Review

**Duration:** 5 Weeks (09. December 2024 (CW 50) – 13. January 2025 (CW 03))

**Deliverables:** Reviewed & Proofread Master's Thesis

**Increments:**
1) Internal Review: Proofreading & Correction by the Authors (2 Weeks)
2) External Review: Proofreading & Correction by External Readers (4 Weeks)

**Milestone VI:** Finalization

**Duration:** 6 Weeks (09. December 2024 (CW 50) – 20. January 2025 (CW 04))

**Deliverables:**
a) Printed & Bound Master's Thesis
b) Master's Thesis Presentation Slides

**Increments:**
1) Thesis Presentation Preparation (5 Weeks)
2) Printing & Binding of Thesis (1 Weeks)

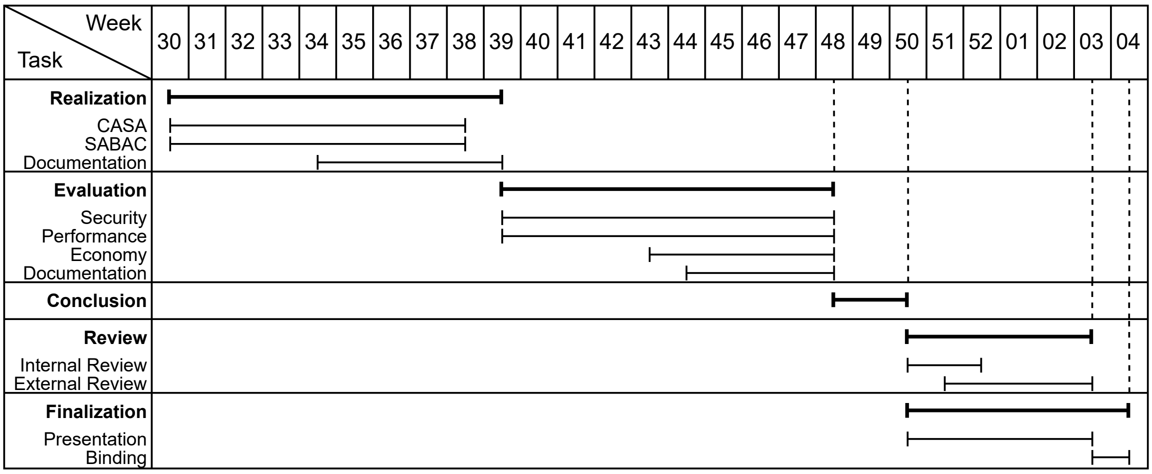


**Figure 5.1:** Time schedule of the proposed master's thesis.

## 5.2 Time Schedule

The time schedule maps the milestones and increments as defined in section 5.1 to specific calendar weeks. The mapping of the proposed master's thesis is shown in Figure 5.1. The proposed time schedule covers the period from the 30th calendar week of 2024 to the 4th calendar week of 2025. In other words, the time schedule spans from July 22nd, 2024 to January 20th, 2025. The work plan stipulates that the approach presented in this thesis proposal has to be realized and evaluated within the 26-week period. Additionally, the remaining chapters must be completed and the written elaboration of the thesis must be finalized within this timeframe.

The visualized time schedule shown in Figure 5.1 illustrates the milestones II to VI. The initial milestone represents the preliminary work and has therefore been excluded from the visual representation. Multiple increments mapped to the same calendar weeks may be performed in parallel, as there are either no interdependencies or existing dependencies can be resolved beforehand. Accordingly, the indicated duration of an increment takes the occurrence of parallelization into account. In regard to the realization of CASA and SABAAC, the dependency inversion principle and service interfaces are employed to resolve existing dependencies. Other increments such as the different areas of the evaluation do not have interdependencies. The realization, evaluation, and conclusion milestones depend on the results of their predecessors. Consequently, the preceding milestones must be completed before the next milestone can commence. The finalization milestone is only partially dependent on the review milestone due to its binding increment. Accordingly, independent increments of these milestones can be performed simultaneously.

## 5.3 Risk Assessment

The risk assessment identifies potential risks and assesses their impact on the proposed project plan. Additionally, it presents mitigation strategies for the identified risks. By considering the occurrence of undesirable events, the risk assessment enables the meeting of deadlines. The identified and assessed risks are classified as either technical or non-technical risks. The technical risks are discussed in detail in subsection 5.3.1. The non-technical risks consist of organizational and project management risks and are discussed in detail in subsection 5.3.2.

### 5.3.1 Technical Risks

The following technical risks represent deficiencies in the design and implementation of the approach, which could result in non-compliance with stated requirements. To ensure the compliance with stated requirements, it is recommended that software testing and system evaluation are conducted in an automated manner.

#### 5.3.1.1 Software Implementation Flaws

Software implementation flaws have an impact on the performance, availability, security, and safety of the approach. Implementation flaws such as bugs should be avoided by using automated software testing. The automated software testing should consist of automated unit, integration, system, and acceptance tests. The unit, integration, and system tests assure the correct and failure-free operation of the software under valid and invalid system conditions. The acceptance tests check the compliance to stated system requirements. The employed software testing methods have to provide high source code coverage. Besides automated software testing, code reviews after the implementation of the CASC-SAS software can increase the source code quality and mitigate the risk of implementation flaws.

#### 5.3.1.2 Software Design Flaws

Software design flaws have an impact on the performance, availability, security, and safety of the approach. To mitigate the risk of software design flaws and therefore avoid re-implementation, architecture reviews should be conducted after the design of the CASC-SAS software. Furthermore, automated acceptance tests should check the compliance of already implemented software to stated system requirements to identify software design flaws.

#### 5.3.1.3 Transient Hardware Faults

Transient faults of system hardware have an impact on the performance and availability of the approach. Transient faults have to be taken into account during the design and implementation of the system. This can be achieved by employing failure-avoidance strategies such as redundancy and automated system monitoring.

#### 5.3.1.4 Persistent Hardware Faults

Persistent faults of system hardware have an impact on the performance and availability of the approach. Persistent faults have to identified via automated system monitoring and resolved by replacing corresponding hardware components.

#### 5.3.1.5 Unsuitable Hardware

Unsuitable hardware has an impact on the performance and economic aspects of the approach. As a consequence, unsuitable hardware has to be replaced by suitable hardware in order to satisfy the system requirements. To make the approach performant and economically feasible, the system has to use components that provide neither too much nor too less performance for their designated tasks.

### 5.3.2 Organizational & Project Management Risks

The following risks represent deficiencies in the project organization and management. Undesirable events caused by these risks lead to deviations from the proposed work plan and time schedule. To mitigate the following risks, a preliminary milestone was created with the objective of reducing the number and complexity of tasks to be completed within the limited thesis period.

#### 5.3.2.1 Inaccurate Estimation of Milestone & Increment Duration

The projected durations of milestones and increments have been calculated based on empirical values. However, this calculation approach may result in inaccurate estimations, which could lead to a deviation from the proposed time schedule. To mitigate the risk of inaccurate estimations, an additional buffer time has been incorporated into each duration associated with an increment or milestone.

#### 5.3.2.2 Illness-Related Delay

The projected durations of milestones and increments have been calculated based on empirical values. Furthermore, additional buffer times have been incorporated into the projected durations. Nevertheless, illness may result in a deviation from the proposed time schedule. Small deviations, in the order of weeks, can be compensated by the additional buffer times. Large deviations, in the order of months, require either an extension of the limited thesis period or a prioritization of increments. The possibility of extending the thesis period by up to three months is governed by the examination regulations.

# Bibliography

[1] 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.

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

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

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography*. 2023. URL: https://toc.cryptobook.us/book.pdf (visited on 06/22/2024).

[10] Bundesamt für Sicherheit in der Informationstechnik (BSI). *Die Lage der IT-Sicherheit in Deutschland 2014*. 2014. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2014.pdf (visited on 08/03/2024).

[11] 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.

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

[13] 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).

[14] 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 08/03/2024).

[15] Joseph Cox. *GCHQ Says Hackers Have Likely Compromised UK Energy Sector Targets*. 2017. URL: https://www.vice.com/en/article/9kwg4a/gchq-says-hackers-have-likely-compromised-uk-energy-sector-targets (visited on 08/03/2024).

[16] Cybersecurity & Infrastructure Security Agency (CISA). *CrashOverride Malware*. 2021. URL: https://www.cisa.gov/news-events/alerts/2017/06/12/crashoverride-malware (visited on 08/03/2024).

[17] 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 08/03/2024).

[18] Cybersecurity & Infrastructure Security Agency (CISA). *ICS Focused Malware (Update A)*. 2018. URL: https://www.cisa.gov/news-events/ics-alerts/ics-alert-14-176-02a (visited on 08/03/2024).

[19] Cybersecurity & Infrastructure Security Agency (CISA). *Russian Government Cyber Activity Targeting Energy and Other Critical Infrastructure Sectors*. 2018. URL: https://www.cisa.gov/news-events/alerts/2018/03/15/russian-government-cyber-activity-targeting-energy-and-other-critical-infrastructure-sectors (visited on 08/03/2024).

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

[21] 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.

[22] Jonathan Fildes. *Stuxnet worm targeted high-value Iranian assets*. 2010. URL: https://www.bbc.com/news/technology-11388018 (visited on 08/03/2024).

[23] Jim Finkle. *Exclusive: 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 08/03/2024).

[24] 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.

[25] Marc Girault and David Lefranc. “Server-Aided Verification: Theory and Practice”. In: *Advances in Cryptology - ASIACRYPT 2005*. Springer Berlin Heidelberg, 2005, pp. 605–623. ISBN: 9783540322672. DOI: 10.1007/11593447_33.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] Hürriyet Daily News. *Major cyber-attack on Turkish Energy Ministry claimed*. 2017. URL: https://www.hurriyetdailynews.com/major-cyber-attack-on-turkish-energy-ministry-claimed-107981 (visited on 08/03/2024).

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

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

[34] 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).

[35] 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.

[36] The Times of Israel. *Cyber attacks again hit Israel’s water system, shutting agricultural pumps*. 2020. URL: https://www.timesofisrael.com/cyber-attacks-again-hit-israels-water-system-shutting-agricultural-pumps (visited on 08/03/2024).

[37] 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 08/03/2024).

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

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

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

[47] 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).

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

[49] 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 05/06/2024).

[50] Occupational Safety and Health Administration (OSHA). *Illustrated Glossary - Substations*. URL: https://www.osha.gov/etools/electric-power/illustrated-glossary/sub-station (visited on 08/02/2024).

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

[52] Mohammed Ramadan, Ghada Elbez, and Veit Hagenmeyer. “Verifiable Certificateless Signcryption Scheme for Smart Grids”. In: *2023 7th International Conference on System Reliability and Safety (ICSRS)*. IEEE, Nov. 2023. DOI: 10.1109/icsrs59833.2023.10381069.

[53] Mohammed Ramadan et al. “Identity-Based Signature With Server-Aided Verification Scheme for 5G Mobile Systems”. In: *IEEE Access* 8 (2020), pp. 51810–51820. ISSN: 2169-3536. DOI: 10.1109/access.2020.2980213.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] Robert W. Shirey. *Internet Security Glossary, Version 2*. RFC 4949. Aug. 2007. DOI: 10.17487/RFC4949. URL: https://www.rfc-editor.org/info/rfc4949.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] Joby Warrick and Ellen Nakashima. *Officials: Israel linked to a disruptive cyberattack on Iranian port facility*. 2020. URL: https://www.washingtonpost.com/national-security/officials-israel-linked-to-a-disruptive-cyberattack-on-iranian-port-facility/2020/05/18/9d1da866-9942-11ea-89fd-28fb313d1886_story.html (visited on 08/03/2024).

[67] 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`.

[68] Danny Yadron. *Iranian Hackers Infiltrated New York Dam in 2013*. 2015. URL: `https://www.wsj.com/articles/iranian-hackers-infiltrated-new-york-dam-in-2013-1450662559` (visited on 08/03/2024).

[69] 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 08/03/2024).