From Standard to Practice: Towards ISA/IEC 62443-conform Public Key Infrastructures

Michael P. Heinl[1,2], Maximilian Pursche[1,2], Nikolai Puch[2], Sebastian N. Peters[2], and Alexander Giehl[1,2]

[1] Technical University of Munich, Germany; TUM School of Computation, Information and Technology; Department of Computer Engineering
[2] Fraunhofer AISEC, Germany; Department Product Protection and Industrial Security

Abstract Public key infrastructures (PKIs) are a cornerstone for the security of modern information systems. They also offer a wide range of security mechanisms to industrial automation and control systems (IACS) and can represent an important building block for concepts like zero trust architectures and defense in depth. Hence, the ISA/IEC 62443 series of standards addresses the PKI paradigm, but there is little practical guidance on how to actually apply it to an IACS. This paper analyzes ISA/IEC 62443 for explicit and implicit requirements regarding PKI deployment to provide a guideline for developing and operating a standard-conform PKI. For this purpose, the analyzed requirements and IACS-specific constraints are combined with current research and best practices. To assess its viability, a tangible PKI use case is implemented in a test environment. The evaluation of this use case shows that common IACS components are capable of supporting PKI, but that important features are missing. For instance, the handling of PKI turns out to be time-consuming and involves many manual operations, a potential factor to render large-scale operations impractical at this point in time.

Keywords: PKI · ISA/IEC 62443 · IACS · Security Engineering · Zero Trust

1 Introduction

The Industrial Internet of Things (IIoT) increasingly connects operational technology (OT) involved in production processes to the business IT network and even the internet. This enables new types of value creation, ranging from more efficient processes to individual made-to-order production. However, it also leads to an increased risk of cyber attacks. The ISA/IEC 62443 series of standards provides guidance on the question *what* has to be done to reduce such risks to an acceptable level considering the special characteristics of Industrial Automation and Control Systems (IACS), e.g., the long life cycles and rigorous availability requirements. One measure is to apply the paradigm of public key infrastructures (PKIs) to IACS. This way, security services, such as encryption and authentication, can be utilized by OT devices as well, establishing a basis for security models like defense-in-depth and zero trust architectures. The goal

of this paper is to specify *how* the different aspects related to PKI should be applied by providing a guideline with recommendations for a PKI concept that can be practically used to secure IACS in accordance with ISA/IEC 62443. For this, the paper synthesizes current research, best practices learned from the WebPKI, and requirements from ISA/IEC 62443. The viability is then evaluated by deriving a PKI concept for a tangible use case within a physical IACS testbed. The corresponding research questions answered in this paper are:

RQ 1: Which ISA/IEC 62443 requirements are relevant for PKIs?
RQ 2: Which aspects have to be considered when designing a PKI for IACS?
RQ 3: To what extent is PKI currently supported in industrial environments?

2 Related Work

Scientific papers dealing specifically with PKI in industrial environments are scarce. Hanke [11] analyzed PKI use cases within IACS in 2007, but ISA/IEC 62443 or its predecessor ISA99 were not taken into account. A more recent paper by Yunakovsky et al. [41] from 2021 provides recommendations for PKI in production environments with regards to post-quantum security. While some general recommendations may overlap with the suggestions given during the course of this paper, their focus is solely on post-quantum attacks that PKI systems might face and which cryptographic algorithms are needed to correspondingly protect IACS environments. Apart from these publications, other relevant papers focus on ISA/IEC 62443 threat analysis [9] and its application in engineering projects [22,21] without specifically addressing PKIs.

3 Requirements Analysis

In order to develop a guideline for an ISA/IEC 62443-conform PKI, it is necessary to extract the requirements the standard places on a PKI as covered in Table 1. It is important to note that ISA/IEC 62443 does neither consider a PKI as a System under Consideration (SuC) nor a component of an IACS. Rather, a PKI is considered a security measure. Therefore, requirements are rarely imposed on the PKI itself but on how the IACS components and subsystems shall interact with it. Nevertheless, directions for the design of a PKI can be directly or indirectly derived from these requirements. Furthermore, it is paramount that the architecture of the PKI does not hinder any security- or safety-related operation and that it integrates well into the environment of the IACS.

As a basis for the actual requirements, ISA/IEC 62443 defines three common control system security constraints generalizing the high availability and integrity requirements of IACS [17]. The first constraint, *support of essential functions*, is crucial when designing a PKI for IACS as it dictates that security measures shall not negatively impact health, safety, environment (HSE), or the availability of the system. As an example, ISA/IEC 62443 stipulates that failure of a PKI service shall not interrupt or significantly delay essential IACS

Table 1. PKI Architecture Requirements.

	Requirement	Name	PKI-related Content
PKI Architecture Requirements	SR/CR 1.8	Public key infrastructure certificates	A PKI has to follow best practices and the company's Certificate Policy (CP). A Certificate Policy (CP), for example, defines network locations of PKI entities or trust store configurations. RFC 3647 [30] is explicitly mentioned for guidance. Secure processes for the operation of the PKI need to be in place and should not negatively affect the system's performance.
	SR/CR 1.9	Strength of public key authentication	The IACS and its components must, e.g., be able to validate signatures and the chain of trust up until a trusted (CA) certificate and check the certificate's revocation status. Components must also ensure that the used key and signature algorithm follows cryptographic guidelines. The number of roots of trust (RoT) has to be minimized and the secrecy of private keys ensured. There is an RE requiring *hardware security for public key authentication*, e.g., TPMs.
	SR/CR 2.8	Auditable events	IACS and components must be able to produce audit logs for security-related events, incl. PKI operations. An RE requires the IACS to have the capability to maintain a *centrally managed, system-wide audit trail* and export in standardized formats.
	SR/CR 4.2	Information persistence	It must be ensured that decommissioned systems and components do not leak confidential information. This implies the presence of certificate/key life cycle processes including proper sanitization.
	SR/CR 4.3	Use of cryptography	Cryptography and key management shall follow international standards and best practices, e.g., by U.S. National Institute of Standards and Technology [24,26]. The strength of keys and algorithms should be chosen appropriate to the information it protects.
	SR/CR 5.1	Network segmentation	Network segmentation must be possible to support the zone model and to break connections between segments during an incident without essential functions failing. This means that zones, components, and PKI entities like CA or RA should withstand being cut off from each other. Two REs require *independence from non-control system networks* and *logical and physical isolation of critical networks* from non-critical networks.
	SR 5.2	Zone boundary protection	Traffic between zones must be monitored and only allowed if necessary. REs require communication between zones being preventable in case of an incident (*island mode*) or operational failure (*fail close*).
	SR/CR 7.3	Control system backup	IACS, components, and PKI entities must be able to perform backups without endangering confidential information like private keys. This implies either the exclusion of keys or encryption of backups.
	SR/CR 7.4	IACS recovery and reconstitution	Recovery to a secure state after disruption must be possible, incl. configuration loaded from backup. A PKI must ensure the valid state of time-sensitive data after reconstitution, e.g., renew expired certificates before resuming operation.
PKI End Entity/Relying Party Requirements	SR/CR 1.5	Authenticator management	IACS and components shall be able to initialize, change/refresh, and protect authenticators (including certificates and corresponding private keys) from disclosure in transit and storage. Components that are unable to meet this CR cannot use PKI security functionalities.
	SR/CR 2.11	Timestamps	Timestamps are required for audit logs. Two REs require IACS-*internal time synchronization* with a central time source and the *protection of time source integrity*, which is also necessary to check expiration of time sensitive PKI information like certificates or CRLs.
	SR/CR 3.3	Security functionality verification	Testing of security functions must be supported, e.g., authentication and proper handling of revoked certificates as well as test cases for further PKI-related functions.
	SR/CR 3.7	Error handling	Error handling shall support remediation without revealing sensitive information to adversaries.
	SR/CR 4.1	Information confidentiality	The confidentiality of sensitive information at rest and in transit must be protected. This extends SR/CR 1.5 by including PKI process information and configuration to impede reconnaissance.
	SR/CR 7.1	Denial of service protection	IACS and components shall be able to maintain essential functions in case of Denial of Service events and *manage communication load from component* flooding according to an RE. Another RE requires IACS to *limit DoS effects to other systems or networks*. Hence, PKI-related communication must not cause DoS events.
	SR/CR 7.2	Resource Mgmt.	Security functions have to be managed in a resource-efficient way to prevent overload and delay.
	HDR/EDR/NDR 3.12	Provisioning product supplier RoT	Host, embedded, and network components must be capable of being provisioned with supplier's RoT and protecting their integrity, authenticity, and confidentiality.
	HDR/EDR/NDR 3.13	Provisioning asset owner roots of trust	These types of components must also be capable of being provisioned with and protecting the owner's RoT without reliance external to their own security zone.

communication. *Compensating countermeasures* means that security requirements the system or component should fulfill can also be fulfilled by an external component if an appropriate interface is given. In case of an PKI, this could be, e.g., an Online Certificate Status Protocol (OCSP) responder providing Validation Authority (VA) services. Eventually, *least privilege* specifies that permissions concerning resources and information of the IACS must only be mapped to a specific role if they are necessary to fulfill the role's intended purpose. Based on this, ISA/IEC 62443 defines seven Foundational Requirements (FRs) [20,17]. Each FR is detailed by a set of System Requirements (SRs) [17] and Component Requirements (CRs) [18]. SRs/CRs consist of a baseline requirement and possible Requirement Enhancements (REs). Moreover, some CRs have specific variations depending on the type of the respective component, i.e., software application (SAR), embedded device (EDR), host device (HDR), or network device (NDR). Each SR/CR is associated with Security Levels (SLs). These SLs range from SL 1 to SL 4. The higher the SL, the better the corresponding protection. To accomplish a higher SL, a system/component often needs to meet REs in addition to the baseline requirement. A more in-depth explanation of the different types of SLs can be found in Annex A of ISA/IEC 62443 3-3 [17].

4 PKI Guideline

This section discusses the major structural and procedural aspects of a PKI. A PKI primarily targeting Machine-to-Machine (M2M) authentication faces different challenges than a PKI targeting Human-to-Machine (H2M) authentication. While a lot of architectural requirements are similar, an H2M PKI requires additional processes during operation, e.g., addressing identification and fluctuation of employees. Although employment of a PKI targeting H2M is a valid use case for IACS, it is beyond the scope of this paper.

ISA/IEC 62443's two central requirements directed towards PKI are SR/CR 1.8 describing how to handle PKI operation and 1.9 detailing how systems and components should interact with certificates. The most important point in SR/CR 1.8 is the requirement to follow best practices and a Certificate Policy (CP). It points towards RFC 3647 [30] that assists in writing a CP and a Certificate Practice Statement (CPS). A CP is a document that defines roles, duties, and requirements for entities within a PKI, for example, Certificate Authority (CA), Registration Authority (RA), and End Entity (EE). It also provides legal and liability statements. A CPS, on the other hand, is more practical and provides details on how a PKI meets the set requirements. To employ a PKI in an IACS to the best possible extent, one would have to first define a CP based on security standards, best practices, and laws applicable to the industry. Subsequently, a CPS details how to fulfill each requirement set by the CP depending on the technical and organizational environment of the PKI. This paper discusses key points and gives recommendations to meet the requirements set by ISA/IEC 62443, enriched by best practices [5,4,7,8].

4.1 PKI Structure

A fundamental consideration that needs to be evaluated is whether to integrate the own PKI into a public PKI or to utilize a private PKI. SR/CR 1.8 mentions both possibilities. A public PKI has the advantage that a lot of the security recommendations are ideally already met, revocation procedures are in place, and certificates issued by most commercial CAs are publicly trusted, meaning their root CA certificate is present in most trust stores. Component and system configuration with such certificates is usually easier. This is especially advantageous, if EEs are to provide a public service or communicate with third-party entities. It means, however, to invest a certain amount of trust in the used CA. A way to qualitatively evaluate and compare the trustworthiness of CAs is to analyze their CPs and CPSs. Complementarily, Heinl et al. [12] demonstrated a method of assessing trustworthiness of CAs quantitatively. Utilizing a private PKI provides more sovereignty in terms of architecture, procedures, and operation. It is more transparent and thus trustworthy to the operator at the cost of securing such PKI is the operator's own liability and will take more effort.

PKI Hierarchy A PKI is inherently hierarchical with trust delegated from the root CA down to EE certificates. The recommended hierarchy structure

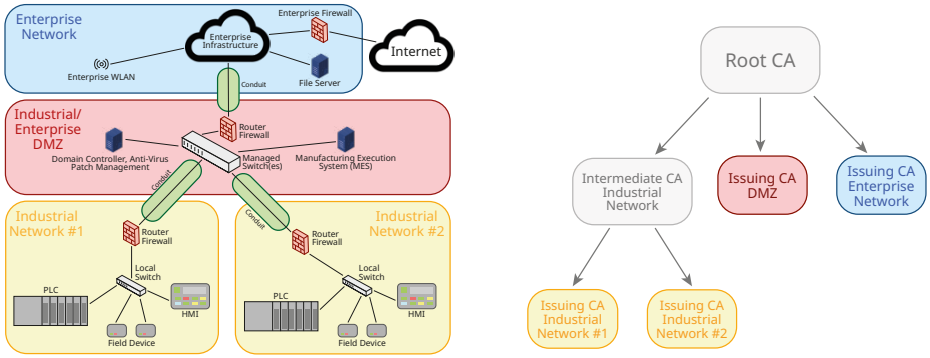

Figure 1. Simplified CA hierarchy mapped to zone model based on ISA/IEC 62443 [17].

includes at least three levels: a root CA, issuing CA(s), and EE(s). In a more complex PKI, there are often additional intermediate CAs. The root CA is the central trust source and should not be used to sign EE certificates [5]. Instead, it issues SubCA certificates that can be used as intermediate or issuing CAs. Intermediate CAs can, for example, be used for different company branches. Their main advantage is the division of responsibility in a complex technical or organizational structure. Issuing CAs are responsible for signing and issuing EE certificates. In the context of ISA/IEC 62443, this hierarchy is advantageous, since the IACS environment is divided into zones according to use case and security requirements. If applicable, each zone should be provided with its own issuing CA, exemplified in Figure 1. This allows revocation of an entire zone without affecting other zones in case of a security incident [41] as implicitly required by SR/CR 5.1 [17,18].

Roles and Responsibilities When designing a PKI, trusted roles as well as their responsibilities and necessary permissions need to be identified on every level of the PKI hierarchy. While it may not be feasible that every role is covered by a different person, separation of duty and the principle of least privilege are crucial to prevent misuse of power and to provide non-repudiation among privileged roles [4,8]. For example, a role only entrusted with certificate issuance should only be able to use but not to read/export the issuing CA's private key. If relevant employees leave or change roles within the company, administration keys and passwords have to be changed [4]. Since PKI centralizes trust, CAs (especially the root CA) represent a potential single point of failure (SPoF) and special protection must be in place to secure them.

Certificate Profiles Certificates should only be issued for a certain purpose defined in a certificate profile encompassing the X.509v3 certificate fields and extensions [31]. After an ISA/IEC 62443 risk assessment [16] comes to the conclusion which PKI services have to be employed in which zone, corresponding

certificate profiles must be defined, depending on PKI hierarchy level as well as zone. There are three important extensions restricting what certificates can be used for. The first one is the *Basic Constraints* extension. It includes the `CA` field, indicating if the key pair can be used to sign other certificates. This must never be true for EE certificates, but needs to be set for CA certificates [5,31] along with the `KeyCertSign` and `CRLSign` bits of the second extension, *Key Usage*. If OCSP is utilized by this CA, `DigitalSignature` must be set, too. Other purposes should not be set for CA certificates to ensure that they are used for signing certificates and revocation procedures only [5]. For EE certificates, this extension should be adjusted to the intended application. For example, a typical TLS certificate will have `DigitalSignature` and `KeyEncipherment` set. Another use case leveraging `DigitalSignature` in combination with `NonRepudiation` would be signing audit logs sent to a central log server (cf. SR 2.8 RE 1). The final extension, *Extended Key Usage*, contains use case restrictions like server or client authentication within the TLS protocol and is not limited to the options specified in RFC 5280 [31]. While certificates that are used within a public context must adhere to this specifications, some extensions can be repurposed in a private IACS environment, e.g., to authenticate license keys. However, if an RFC 5280-compliant certificate parser cannot process an extension with the `critical` flag set to `true`, it will reject such certificate [31].

Network Segmentation When a network utilizes PKI for a critical service, e.g., secure communication between critical components, it should not cause loss of availability when communication between network segments is broken or the zone goes into island mode. E.g., a network that should fulfill SR 5.1 RE 2 will need to have its issuing CA and revocation method within its own segment. To reduce the exposure of the PKI when CAs are deployed to every zone, it is useful to restrict the certificates a SubCA can issue. The field `PathLenConstraint` in the *Basic Constraints* extension enforces a maximum number of CA certificates that may follow in the certificate chain [31]. This should be `0` for all issuing CAs, since they shall only issue EE but not CA certificates. The extension *NameConstraints* restricts the name space for which a CA can issue certificates. To utilize it in an IACS, its name spaces should reflect the zone partitions.

Computer Security Computers hosting PKI services must be hardened, e.g., running only tested and trustworthy software, changing or disabling default accounts/credentials [4,7], and require personalization and multi-factor authentication for all privileged roles [8]. Patch management shall be established [19] and security patches be implemented no later than six months after they became available, unless they conflict with other functionality or dependencies [4].

Physical Security Zones or networks with high security requirements, e.g., an offline root CA, should be physically protected. E.g., only authorized personnel should have access, every entry and exit be logged, and portable media containing sensitive information not be brought out without authorization [8,4].

Monitoring and Logging Monitoring must cover all PKI entities, incl. network reachability, CPU utilization, disk capacity, and logging processes [4,7]. In case of failure, responsible personnel should be alerted. Audit logs are a common way to detect security incidents and to provide accountability [7]. PKI-related events, e.g., certificate issuance, revocation, as well as general security events should be logged [5] and centrally aggregated (cf. SR 2.10 RE 1). This allows the detection and timely response to events. The retention period should be sufficiently long, e.g., at least two years [5], to enable post-incident forensics.

Cryptographic Recommendations A PKI is built upon cryptographic primitives with a lot of research and development going on. In general, commonly accepted recommendations, e.g., by NIST [26,25,24] or the BSI [1,2,3], shall be followed to stay on top of these developments. However, there are environment-specific aspects, e.g., regarding key pair generation or signature algorithm, which have to be considered in the design phase of the PKI due to potential trade-offs between security and factors like performance, latency, and cost.

RSA has the advantage that most networked IACS components support it out of the box. Drawbacks are relatively long public keys and computationally expensive key operations like signature generation and decryption [40,23]. This stands in contrast to real-time constraints of IACS. The most prevalent alternative to RSA is Elliptic Curve Cryptography (ECC) needing a significantly smaller key size to achieve a similar effective key length (*security strength*), e.g., 256 bit (ECC) compared to 3072 bit (RSA) for a security strength of 128 bit [1,3,26]. ECC is often faster (with exceptions, e.g., signature validation), more energy-efficient, and the shorter key lengths make key handling easier for components [40,23]. Hence, depending on the application, RSA can be recommended for heterogeneous environments and a focus on time-critical signature validation whereas ECC can be recommended if there are constraints regarding storage, bandwidth, power consumption, and little to no legacy devices. It is recommended to use key lengths with at least 128 bit of security strength for both RSA and ECDSA as well as a SHA-2 or SHA-3 hash function with the same security strength for digital signatures [26,3]. For environments requiring post-quantum (PQ) security, entirely different algorithms have to be used [41,1]. Methods which can provide a smooth transition to PQ cryptography include hybrid certificates [28] and mixed certificate chains [29].

Long-term keys should be stored in a trusted platform module (TPM) or a hardware security module (HSM). They provide hardware-based protection and often functionality like binding a key pair to a device (TPM) or a multi-user authorization scheme (HSM) for very sensitive keys, e.g., of the root CA.

4.2 PKI Processes

Besides the structure, procedural aspects must also be evaluated.

Deploying EE Certificates After defining a certificate profile, the key pair and a corresponding certificate signing request has to be generated. *Key generation*

can either be done by the EE or by the CA in case the EE itself is not able to due to a missing cryptographically secure random number generator. The latter allows for key recovery, however, it also impairs non-repudiation. Once the certificate is signed, the EE certificate, the certificate chain, and the keys must then be transported to the EE via a secure channel [8]. While it seems desirable to deploy certificates to as many devices and utilize them as often as possible, their employment must be carefully considered. ISA/IEC 62443 classifies keys as authenticators and sensitive data which results in additional operational requirements. If a device does not handle sensitive data in terms of confidentiality or integrity, the operational effort by handling certificates may outweigh optional security functionality.

Long *validity periods* may be acceptable in IACS. If an ISA/IEC 62443 risk assessment [16] comes to a different conclusion for specific zones, then regular renewal of certificates may be necessary. In this case, automation protocols, like SCEP [35] or EST [34], which allow EEs to automatically obtain a certificate from a CA, should be taken into consideration. If components support such automatism, it can reduce operational effort and minimize human error while enabling short certificate life times. However, it must be ensured that the employed mechanisms meet the set requirements, especially regarding availability. Otherwise, manual renewal of certificates can mean serious operational overhead and even downtime. SR/CR 1.9 does not explicitly cover checks of a certificates' validity period because for some applications with very high availability requirements, communication with an expired certificate can be more acceptable than unsecured or no communication at all. In these cases, expired certificates may be accepted as a fallback under exceptional circumstances [27] as long as they are not revoked and their keys provide adequate protection.

Revocation Revocation mechanisms like Certificate Revocation Lists (CRLs) [31] and OCSP [33] fulfill the purpose of indicating lost or otherwise compromised key material, that should not be accepted or used by any entity within the PKI [5,8]. While validity period checks may not be necessary within IACS, revocation status must be evaluated in every step of the certificate path (cf. SR/CR 1.9). CRLs are the basic form of revocation and are relatively independent from CA uptime or other PKI services by deploying the lists to every EE. Their main disadvantage is the update and maintenance process. A CRL can grow quite large and must be redeployed to every component once another certificate is added. Delta CRLs only containing certificates revoked since the last base CRL [31] can be an alternative to reduce overhead. In static environments, where communication only happens within a security zone, the disadvantages of CRLs might not matter as much, since they will be short and updates are unlikely. The most prevalent alternative to CRLs is OCSP, centralizing revocation checking in an OCSP reponder. EEs query the responder for the status of certificates they process, decreasing storage and computation effort of EEs but increasing network traffic and representing a potential SPoF. OCSP stapling [32] can solve some of the potential problems of OCSP by offering a signed and

timestamped revocation status to certificate holders who can then present it to relying parties during authentication. This way, the revocation status can be verified without direct communication with the OCSP responder decreasing the communication load and availability constraints. Currently, OCSP stapling is only standardized as a TLS extension, but the principle could be used in other protocols as well.

Backup and Recovery PKI entities need to be included in backup procedures following commonly accepted standards [15,4]. Confidential information has to be excluded from a component backup or encrypted [8,18,17]. Recovery procedures should be regularly tested, to ensure that IACS can resume operation [5] even if certificates are invalid at the time of restore. In such a case as well as for component backups excluding private keys, a certificate issuing process should be included into the restore procedure. There might be PKI entities depending on each other's configuration, for example an OCSP responder and the corresponding CA, that have to be backed up and restored together. Such dependencies should be analyzed and documented [7].

End of Life Procedures When devices reach their end of service, processes must be in place to ensure that sensitive information, e.g., private keys, is purged (cf. SR/CR 4.2), including from backups and potential redundant systems [5,8,3]. If it cannot be ensured that all copies of a private key are destroyed, the corresponding certificate has to be revoked in addition.

Compliance and Auditing Risk assessments, zone definitions, and usage evaluations should not only be done before initial deployment, but regularly [16]. Guidelines and requirements for PKIs will change and should be reviewed on a regular basis in order to incorporate them into the security program.

5 Implementation

This section describes a practical PKI implementation with the goal to test the feasibility of PKI usage in a representative IACS testbed. It builds on previous research [10], which identified possible attacks on the present IACS. The implementation focuses on preventing one of the identified attacks by implementing TLS on top of an existing communication protocol and outlines possible challenges that need to be considered when employing PKI in an IACS.

5.1 IACS Environment

The testbed represents a small production facility consisting of three production isles. Each isle is made up of a base and an application module. Modules are composed of components, with the base usually consisting of a main Programmable Logic Controller (PLC) *SIMATIC ET 200SP*, a Human-Machine

Interface (HMI) *SIMATIC HMI TP700 Comfort*, and a conveyor belt. The application modules have components specific to their task and may contain additional PLCs. The main PLCs are connected to a router via Ethernet, which connects them to the Manufacturing Execution System (MES) which is the central control unit of the IACS. The MES runs the *MES4* software by Festo for managing manufacturing orders as well as the *TIA* portal (version 15.1) and *CODESYS* (version 3.5.14) to program the PLCs of the associated manufacturer. The implementation focuses on the station shown in Figure 2, housing a storage application and marking the starting point of the manufacturing process. A smaller version of the main PLC from Siemens (*SIMATIC S7-1200*) controls the storage unit. The MES and the main PLC communicate via TCP/IP. The demonstrated attack [10] targets this communication channel by initially obtaining a MitM position using ARP spoofing and then altering data sent on the TCP layer. This results in full control of the application, including picking wrong starting materials from the storage without the MES noticing. To prevent this kind of attack, TLS is implemented.

5.2 Existing PKI Interfaces

Before implementing TLS, existing interfaces for certificate deployment are identified to provide an overview of the extent to which PKI can be deployed and which use cases are already implemented. The MES is Windows 10-based allowing to import any X.509v3 certificate to authenticate users, e.g., via smart card [13], using the Windows certificate utility. The MES software (MES SW) does neither have any documented certificate interfaces nor does it utilize the built-in Windows certificate store. The SIMATIC ET 200SP has a certificate store that can be configured via the TIA portal's central certificate manager [37]. It has two modes: the centrally managed, project-wide certificate store and an independent mode. When using the independent mode, certificates cannot be imported and only self-signed certificates can be created. The centrally managed certificate store comes with an own root CA and allows the issuance of device certificates signed by this very root CA or import of other certificates. A PLC's possible certificate usage depends on the built-in CPU. The used SIMATIC ET 200SP allows for four certificate use cases: TLS communication either as server or client, OPC UA authentication, securing PLC to HMI communication, and employment of the HTTPS protocol for the PLC-hosted web server [39,38]. The smaller SIMATIC S7-1200 has a S7-1200

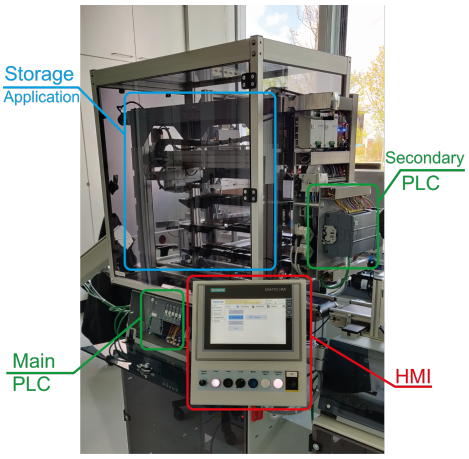

Figure 2. Testbed setup.

CPU built in and does not have any certificate store that could be configured. The only documented certificate utilization is a self-signed certificate used for HTTPS access to the web server running on the device [36]. This certificate could not be configured, regardless of whether the global certificate manager is employed or not. The Festo PLC CECC-LK can be configured via CODESYS with the option to secure the production PLC code with an X.509 certificate, encrypting or signing the project by utilizing the Windows certificate manager. Similar to the TIA portal, the used CODESYS version 3.5.14 also supports certificates for OPC UA but no other use cases. In 2020, version 3.5.16 was released enabling TLS with configurable certificates [6].

5.3 Selecting a PKI Tool

In order to make an informed tool selection, the landscape of open-source PKI software solutions is analyzed regarding *security*, *usability*, as well as *scalability and integration*. The decision matrix provided in Table 2 shows the results of the four most promising candidates, namely Dogtag, EJBCA, OpenXPKI, and Step-CA, indicating that none of them can be seen as a clear favorite. Eventually, EJBCA is chosen for the implementation, especially due to its modular architecture.

Table 2. Open-source PKI tool decision matrix.

			Dogtag		**EJBCA**		**Step-CA**		**OpenXPKI**	
Groupings	Selection Criteria	Weights	Rating	Weighted	Rating	Weighted	Rating	Weighted	Rating	Weighted
Security	Confidentially (in transit and DB)	25,00 %	5	1,25	5	1,25	5	1,25	5	1,25
	Integrity in Database	25,00 %	5	1,25	5	1,25	5	1,25	5	1,25
	Access and Rights Management	25,00 %	5	1	5	1,25	2	0,5	5	1,25
	Release and Patch Cycle	25,00 %	4	1	5	1,25	5	1,25	4	1
Usability	GUI Usability	20,00 %	4	0,8	4	0,8	0	0	4	0,8
	CLI Functionality	20,00 %	4	0,8	4	0,8	5	1	5	1
	Documentation	35,00 %	4	1,4	4	1,4	4	1,4	3	1,05
	Vendor Support	25,00 %	1	0,25	5	1,25	5	1,25	5	1,25
Scalability and Integration	Variety of Supported Components	15,00 %	4	0,6	3	0,45	4	0,6	5	0,75
	Automation Possibilities	25,00 %	4	1	4	1	5	1,25	5	1,25
	Availability Concepts	30,00 %	3	0,9	4	1,2	2	0,6	3	0,9
	Multi Instance Operation	30,00 %	5	1,5	4	1,2	3	0,9	4	1,2
Total Score				11,75		13,1		11,25		12,95

5.4 PKI Installation and Configuration

EJBCA is installed including CA, RA, and VA functionality. Subsequently, the PKI is configured including the generation of certificate profiles as well as root CA and SubCA certificates. Eventually, two PKCS #12-formatted EE certificates are issued in the RA web GUI and the CA certificate chain is exported. Due to a lack of support for management protocols, e.g., SCEP or EST, the certificates have to be transferred to the TIA portal to assign them to the SIMATIC PLCs. For this purpose, the previously exported CA certificate chain as well as the PKCS#12 file containing the certificate and private key for the PLC are imported, requiring a restart of the device. The TIA portal limits the usage of certificates to RSA with SHA-1 or SHA-2. The certificates issued by the TIA portal's own CA use SHA-1 and 2048 bit RSA keys by default. The max. length for keys generated by the TIA portal is 3072 bit but it can handle larger keys if certificates are

imported. Attempts to import ECDSA keys/certificates were rejected. Neither CRLs nor any other revocation method are supported [39].

5.5 Communication Configuration

The communication between PLC (client) and MES SW (server) utilizes two TCP/IP sessions. One session is used for order inquiries (query session) while the other transmits status information to the MES (state session). Once the query session is established, it is only used when a sled reaches the storage application. The PLC then queries the MES SW for instructions regarding the next operation. The state session communicates runtime information to the MES including production mode and error codes. Mutually authenticated TLS is implemented on both sessions to secure communication between MES SW and PLC. Since the existing proprietary MES SW does not provide TLS capabilities, the TLS wrapper *stunnel* is used on the MES to tunnel the unencrypted communication through a secure channel as illustrated by Figure 3. The SIMATIC PLC is programmed via the TIA portal which utilizes STEP 7, an IEC 61131-compliant [14] software for programming PLCs. In STEP 7 V14, the datatype `TCON_IP_V4_SEC` was added to support TLS 1.2. The unsecured MES communication took place on TCP ports 2000 (query session) and 2001 (state session). These ports are now used internally via loopback interface, while TCP ports 2005/2006 are used for the external TLS connection via *stunnel*. Since revocation checking is not supported by the SIMATIC ET 200SP, it is done server-sided. *stunnel* is therefore configured with a CRL that is checked when a certificate is verified.

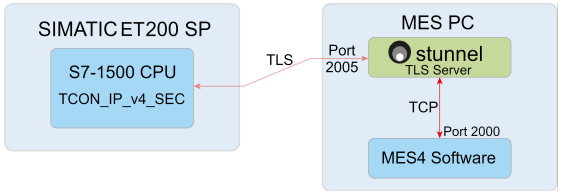

Figure 3. TLS communication with the help of *stunnel*.

6 Evaluation

The probably most serious limitation for a PKI implementation within the present IACS, is the inability of the PLC to check certificate revocation status. With none of the standardized revocation methods supported, the only mechanism available to limit the damage compromised keys can cause, are short certificate lifetimes. Maintaining short lifetimes, in turn, is not a trivial task due to the fact that certificates can only be imported manually into the PLCs. Moreover, the PLCs need to reboot on reconfiguration, halting production. This makes short certificate lifetimes operationally unmanageable in a complex IACS comprising a high number of devices with similar constraints. Another missing feature is the use of EC key material. The shorter keys and faster computations could lower the strain on components' resources. Interestingly, the PLC uses

ECDHE for the session key exchange during the TLS handshake which suggests that at least some ECC operations are implemented in the PLC's firmware. Another security-relevant aspect is the missing crypto agility and weak default parameters of the used version of the TIA portal.

7 Conclusion

This paper collected direct and indirect requirements set by ISA/IEC 62443 and contextualized them with more tangible recommendations and best practices. With its hierarchical structure, PKI fits well into the ISA/IEC 62443 zone concept. However, there is also a discrepancy between the security requirements for the WebPKI and a PKI for IACS environments. This is mainly due to the different prioritization of the security goals resulting in some interesting differences, e.g., regarding certificate validity periods. The implementation shows that it is possible to implement a PKI use case with common IACS components. However, it also reveals that PKI support is rudimentary and lacks important features, e.g., certificate revocation. Overall, it confirmed the impression that IACS are only slowly evolving due to their proprietary devices and long life cycles. However, the rapidly increasing importance of ISA/IEC 62443 suggests that stakeholders are aware of these circumstances, which in turn might also lead to more capable components enabling fully compliant PKI deployments. Finally, it must be taken into account that recommendations and security requirements are not static. Once ISA/IEC 62443 or best practices change, the present guideline also has to be revised.

References

1. BSI: Kryptographische Verfahren: Empfehlungen und Schlüssellangen (2022)
2. BSI: Kryptographische Verfahren: Empfehlungen und Schlüssellangen Teil 2 – Verwendung von Transport Layer Security (TLS) (2022)
3. BSI: Kryptographische Vorgaben für Projekte der Bundesregierung Teil 4: Kommunikationsverfahren in Anwendungen (2022)
4. CA/Browser Forum: Network & Certificate System Security Requirements (2021)
5. CA/Browser Forum: Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates (2022)
6. CODESYS GmbH: Features and Improvements CODESYS V3.5 SP16 (2020)
7. ETSI EN 319 401 V2.3.1: Electronic Signatures and Infrastructures; General Policy Requirements for Trust Service Providers (2021)
8. ETSI EN 319 411-1 V1.3.1: Electronic Signatures and Infrastructures; Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements (2021)
9. Fockel, M., et al.: Designing and Integrating IEC 62443 Compliant Threat Analysis. In: EuroSPI 2019 (2019)
10. Hagen, B.: Security analysis of an interconnected industrial automation testbed (production line). Master's thesis, Hochschule Augsburg (2022)
11. Hanke, M.: Embedded PKI in Industrial Facilities. In: ISSE/SECURE 2007 (2007)

12. Heinl, M.P., et al.: MERCAT: A Metric for the Evaluation and Reconsideration of Certificate Authority Trustworthiness. In: CCSW'19 (2019)
13. Hughes, L.E.: Issue and manage windows logon certificates. In: Pro Active Directory Certificate Services: Creating and Managing Digital Certificates for Use in Microsoft Networks, pp. 405–436. Apress, Berkeley, CA (2022)
14. IEC 61131-3:2013: Programming languages (2013)
15. IEC 62443-2-1:2010: Establishing an IACS security program (2010)
16. IEC 62443-3-2:2020: Security risk assessment for system design (2020)
17. IEC 62443-3-3:2013: System security requirements and security levels (2013)
18. IEC 62443-4-2:2019: Technical security requirements for IACS components (2019)
19. IEC TR 62443-2-3:2015: Patch management in the IACS environment (2015)
20. IEC TS 62443-1-1:2009: Terminology, concepts and models (2009)
21. Leander, B., et al.: Applicability of the IEC 62443 standard in Industry 4.0/IIoT. In: ARES'19 (2019)
22. Maidl, M., et al.: A Comprehensive Framework for Security in Engineering Projects - Based on IEC 62443. In: IEEE ISSREW 2018 (2018)
23. Maletsky, K.: RSA vs. ECC Comparison for Embedded Systems (Microchip) (2020)
24. NIST: FIPS 140-3: Security Requirements for Cryptographic Modules (2019)
25. NIST: SP 800-57 Part 2 Rev. 1 - Recom. for Key Management: Part 2 – Best Practices for Key Management Organizations (2019)
26. NIST: SP 800-57 Part 1 Rev. 5 - Recom. for Key Management: Part 1 – General (2020)
27. OPC UA Foundation: Practical Security Recommendations for building OPC UA Applications. Whitepaper Security Working Group (2018)
28. Paul, S., et al.: Towards Post-Quantum Security for Cyber-Physical Systems: Integrating PQC into Industrial M2M Communication. In: ESORICS 2020 (2020)
29. Paul, S., et al.: Mixed Certificate Chains for the Transition to Post-Quantum Authentication in TLS 1.3. In: ASIA CCS '22 (2022)
30. RFC 3647: Internet X.509 PKI Certificate Policy & Certification Pract. Framew. (2003)
31. RFC 5280: Internet X.509 PKI Certificate and CRL Profile (2008)
32. RFC 6066: Transport Layer Security (TLS) Extensions: Extension Definitions (2011)
33. RFC 6960: X.509 Internet PKI Online Certificate Status Protocol (2013)
34. RFC 7030: Enrollment over Secure Transport (2013)
35. RFC 8894: Simple Certificate Enrolment Protocol (2020)
36. Siemens AG: SIMATIC S7-1200 Programmable controller (2015), https://cache.industry.siemens.com/dl/files/121/109478121/att_851433/v1/s71200_system_manual_en-US_en-US.pdf
37. Siemens AG: Using Certificates with TIA Portal (2019), https://support.industry.siemens.com/cs/attachments/109769068/109769068_CertificateHandlingTIAPortal_V1_0_en.pdf
38. Siemens AG: Config. of TLS-based PG/HMI Com. and the Protection of Confidential PLC Config. Data (2021), https://support.industry.siemens.com/cs/attachments/109772940/s71200_system_manual_en-US_en-US.pdf
39. Siemens AG: SIMATIC S7-1500, ET 200MP, ET 200SP, ET 200AL, ET 200pro Communication (2021), https://cache.industry.siemens.com/dl/files/942/84133942/att_1098064/v1/et200sp_manual_collection_en-US.pdf
40. Vahdati, Z., et al.: Comparison of ecc and rsa algorithms in iot devices. JATIT (2019)
41. Yunakovsky, S.E., et al.: Towards security recommendations for public-key infrastructures for production environments in the post-quantum era. EPJ Quantum Technology **8**(1) (2021)