Secure Management of Certificates for Industrial Control Systems

Sebastian Obermeier, Ragnar Schierholz, Hadeli Hadeli,
Robert R. Enderlein, Ana Hristova, and Thomas Locher
[first name].[last name]@ch.abb.com
ABB Corporate Research, Industrial Software Systems, Switzerland

Abstract—*In order to secure the communication of industrial automation and control systems (IACS), recent cyber security standards demand the use of certificates, but do not answer the problem of certificate management. In the IACS domain, the use cases and the corresponding implementation differ from traditional IT systems.*

This paper points out the use cases and challenges that an efficient certificate handling technique has to address, and proposes a comprehensive certificate management approach based on existing cryptographic solutions. The paper covers all phases of an embedded device's life cycle up to the operation of the devices. Finally, the paper shows how the proposed solution is able to withstand several attack vectors.

Keywords: security; industrial control systems; certificate management

1. Introduction

Industrial applications are often controlled and supervised by industrial automation and control systems (IACS),[1] which are highly distributed systems used to control dispersed assets, often scattered over a large geographical area. The role of an IACS is to acquire data of an industry process and allow an operator to operate and issue control commands to the associated assets if needed.

In the past, IACS were isolated systems and field devices were connected to the control system via dedicated lines. This traditional approach relied much on the special purpose components and isolation of systems to achieve security. Thus, traditional IACS technology itself had little to no protection against attacks.

Currently, a lot of open standards and technologies are in use to replace old technologies in the IACS due to maintainability issues (e.g., product discontinuations) or connectivity issues (e.g, a demand to connect Enterprise resource planning systems to the IACS).

As nowadays technologies that follow open standards are used in industrial control systems, they inherit all the vulnerabilities of those technologies as well. Consequently, this implies new challenges for the security of those systems and the associated communications. Recently, new security standards (for instance OPC UA) emphasize the need for information security, but yet they do not solve all the issues. OPC UA, and also other standards, demand X.509 certificates without specifying how the certificates are to be managed.

Securing an IACS environment using standard security protocols is challenging mainly because an industrial control system has different security requirements than enterprise information systems. In particular, in contrast to enterprise information systems, IACS follow a different prioritization regarding the relevance of security objectives, cf. [1], [2]. For instance, availability, authenticity, and integrity, are of paramount importance for IACS, while the priority of confidentiality is usually lower.

In office environments, *certificates* are used for achieving authenticity. Certificates bind a public key to identity information, and provide a convenient way to achieve mutual authentication. When certificates should be used for embedded device configuration, however, the devices' life cycle imposes several problems regarding certificate management. For example, a solution to manage the initial authentication between the device and the issuer of the certificate—which entails installing the root certificate, as well as issuing the client certificate—is required.

In addition, the requirement for constant availability, in combination with the fact that embedded devices in general have little computational power, makes complex certificate validations difficult. As in many scenarios, a breach of security can lead to a severe safety breakdown, the security properties "integrity" (making sure commands arrive as intended), "strong authentication and authorization" (making sure only those entities send commands that are entitled to do so), as well as "auditability" (being able to audit in case of an issue or to audit whether a given configuration corresponds to the policy), have the highest priority that any certificate management solution must obey.

Our **contributions** are the following. In this paper, we

- identify the life cycle of a typical embedded IACS device,
- point out use cases for embedded device certificate management including initial deployment, system integration, and certificate operation,
- specify the critical attack vectors, and
- propose and discuss solutions that are able to withstand these attacks for each use case.

[1] These systems are also known as SCADA systems.

2. Related Work

In a public key infrastructure (PKI) scheme, the signatures on a certificate are attestations by the certificate signer that the identity information and the public key belong together. [3] discusses a classification of various certificates.

In the embedded device scenario, however, there is no single CA involved but multiple CAs, including manufacturer and operator CAs as in the lifecycle of an industrial embedded device, different organizations provide the function of the CA. However, generally none of the organizations is available throughout the entire lifecycle. Thus, certificate management for industrial embedded devices has to consider the specific lifecycle requirements of the devices.

A restriction of the time span during which a certificate and the associated private key can be used is important for the following reasons, cf. [4]). A limited certificate lifetime

1) limits the amount of information protected by a given key that is available for cryptanalysis,
2) limits the amount of exposure if a single key is compromised,
3) limits the use of a particular algorithm to its estimated effective lifetime,
4) limits the time available for attempts to penetrate physical, procedural, and logical access mechanisms that protect a key from unauthorized disclosure.
5) limits the period within which information may be compromised by inadvertent disclosure of keying material to unauthorized entities, and
6) limits the time available for computationally intensive cryptanalytic attacks (in applications where long-term key protection is not required).

3. Model

3.1 Roles

Since the various roles in an IACS context are different from the roles in an office IT environment, we proceed by quickly describing all roles that are relevant for the management of certificates.

Manufacturers of the devices physically assemble the devices and install firmware and/or software on them.

System Integrators customize devices, integrate them into an entire system (potentially devices from multiple manufacturers), and perform commissioning. This may be the manufacturer, the asset owner, or an external company.

Operators monitor the system during their normal operation and respond to alarms. Typically done by the asset owner or an external company.

Certificate Authorities are chosen by the system integrator potentially based on operator's requirements. A certificate authority manages certificates and handles revocation during the lifetime of the plant.

Service Units are responsible for maintaining and repairing devices. The role can be performed by the manufacturer, the asset owner, or an external company.

3.2 Life Cycle of an Embedded Device

The life cycle with respect to certificate management of embedded industrial devices, e.g., industrial embedded controllers, is illustrated in Figure 1. After the device has been manufactured, it is integrated into the system within the commissioning phase. Within this phase, all devices forming the system are engineered, customized, and tested according to the operational requirements of the asset owner. Usually two tests are conducted: the "Factory Acceptance Test" (FAT) and the "Site Acceptance Test" (SAT). The goal of the FAT is to ensure that the system engineered for the customer itself works, while the SAT actually ensures that this system works in its intended environment. The asset owner usually chooses the manufacturer, a third party, or even himself to be responsible for this phase. After this phase, the operational phase begins. Again, the asset owner chooses the responsible party for the operation and maintenance of the plant. Service units repair or replace devices if the necessity occurs during operation. Finally, the devices are decommissioned at the end of their lifetime.

During the device's life cycle indicated in Figure 1, five requirements for certificate management can be identified:

Installation of the manufacturer's default root certificate: The manufacturer installs an initial certificate onto the device. The goal is to establish a trust relationship between the devices and the manufacturer's root Certificate Authority (CA) by installing the CA's root certificate on the device inside of a trusted device production environment. Since the ultimate destination of the devices is not known at this point, the device certificates are temporary and will have to be replaced before operation.

Installation of system integrator root certificate: As devices will probably operate in a multi-vendor environment, the operator will likely wish to set up (or ask a third party, e.g., the system integrator, to set up) a Certificate Authority distinct from the manufacturer CA probably even specific to a plant or at least the operator's organization. The system integrator's new root certificate will have to be installed on the devices in a secure manner in order to establish a trust relationship between the system integrator and the devices. The crux of this phase is actually to establish a trust relationship between the manufacturer's and the system integrator's respective CAs. This step can be performed any time before the Factory Acceptance Test (FAT).

Operation-time certificate installation: As the default certificate that the device has received at production time was not customized for operation-time and was issued by the manufacturer's CA, the new CA will issue a new "operation-time" certificate to each device. This step is functionally very

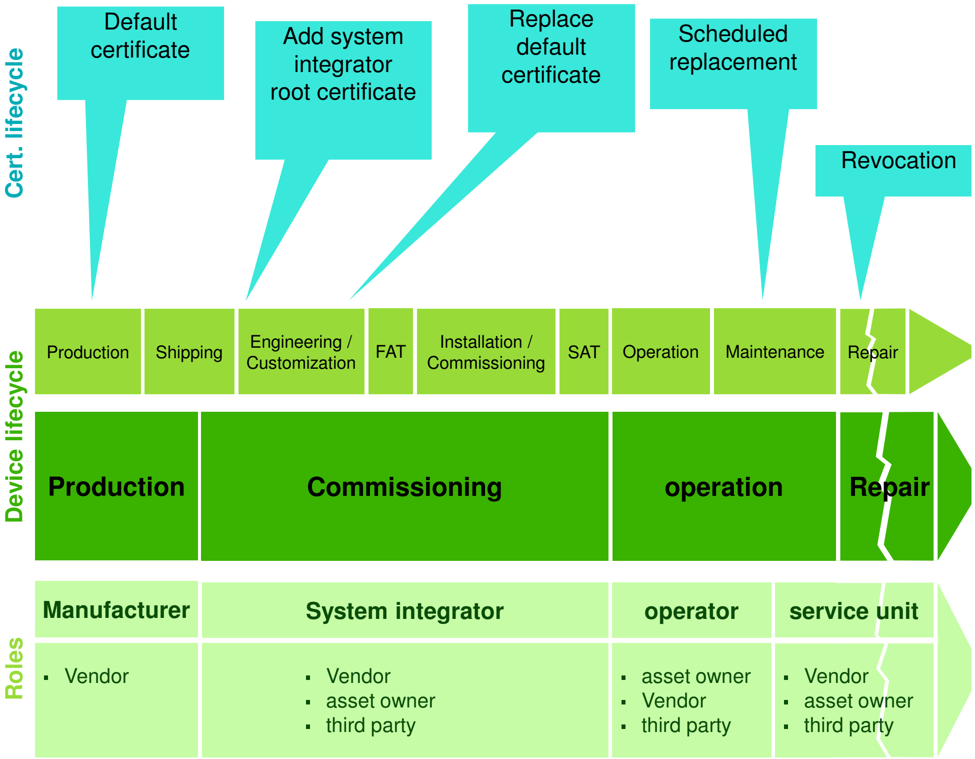

Fig. 1: Embedded Device Lifecycle

similar to the next phase, but is performed after the root certificate has been installed and before the SAT.

Renewal of operation-time certificate: The operation-time certificate of each device will have to be replaced periodically (because all certificates eventually expire). The operator's CA will have to keep track of which certificates need replacement and then issue the new certificates in a secure manner.

Revocation: A properly set-up revocation framework allows the CA to revoke a certificate that needs to be prematurely invalidated.

Within this article, necessary requirements up to a plant's operation are discussed. Certificate renewal and certificate revocation, however, are omitted due to size limitations.

Table 1 shows the mapping between a device's lifecycle phases and the phases defined in RFC4210 [5]. While RFC4210 mandates out-of-band authentication for some of these phase, the exact nature of these out-of-band channels is considered out of scope and not discussed in this paper.

Figure 2 illustrates an overview of the certificate management roles and responsibilities. In total, four CAs are assumed: the manufacturer root CA, the (device) factory CA, the system integrator root CA, and the plant CA.

3.3 Attack Vectors

As mentioned before, certificates protect the integrity of commands and the authenticity of all devices and involved parties. In this paper, we make the common assumption that it is not feasible to break the authentication scheme itself, i.e., we assume that the standard cryptographic algorithms are secure. Therefore, it is only possible for an attacker to launch a successful attack by somehow getting its own key onto the device. During the life cycle of a typical device in an IACS, there are several stages where an attacker may try to tamper with the device. The various possibilities for an attacker to corrupt the device are summarized in the following attack vectors:

1) During the production process, an attacker installs its own root key on the device. The goal is to later send commands signed by the attacker to the device in order

Table 1: Mapping between RFC 4210 and Embedded Device Life Cycle

RFC 4210	Mapping to Device Life Cycle
CA establishment	Set up of manufacturer CA
End-entity initialization	Installation of manufacturer root certificate
Initial registration/certification	Installation of device default certificate
Cross-certificate request	Installation of system integrator root certificate (loosely)
Certificate update (+ key pair update)	Replacement of the default / operation-time certificate, certificate publication
Revocation request, CRL publication	Revocation

to instruct the device to perform malicious actions.

2) During the device's shipment, an attacker installs its own root key.
3) During operation, an attacker manages to install its own root key.
4) During operation, an attacker resets the device, installs its own root key, and restores the initial configuration including the plant's root key.

Thus, the goal is to ensure that the certificate management scheme mitigates exactly these security risks, assuming that the manufacturer and system integrator are not acting maliciously. Our approach is discussed in the subsequent section.

4. Default Certificate Installation

4.1 Problem Description

At this point in time, the final IP address, or other identifiers (DNS name, IEC 61850 Logical Device Name, or any other application specific naming scheme), of a device is not known. The solution must ensure that a pre-installed certificate can be securely replaced later on.

An intuitive approach would be to ship a device without any certificate. This, however, leads to the obvious problem that an attacker could secretly install a root certificate onto a device and ship it to the system integrator. Assuming the system integrator does not notice the certificate, the attacker can always access the device later on. Therefore, the chain of trust must be established right after the production process of each device.

4.2 Solution

The proposed solution requires the manufacturer to generate a private key for each device and create a matching default certificate (using the manufacturer CA to sign it) during device production. The private key, default device certificate, and root certificate of the manufacturer are then installed on each device while in a trusted environment. Devices freshly out of the factory need this default certificate to be able to authenticate themselves. At this stage, the device can only hold one root certificate. The problem of storing the key on a tamper proof (or at least tamper-evident), secure storage is an additional problem that is not discussed in this paper.

At this point, the final device identifiers (IP address or other distinguished names) are not known, thus the default certificates cannot be used for plant operation. However, a unique hardware identifier, such as the serial number or the MAC address of the network interface, can be used as an identifier for the devices. The devices are configured in such way that all certificates issued by the manufacturer have no privileges other than validating an "add root certificate" command and authenticating a key replacement (this authorization is not part of the certificate).

Since the contents of the subject DN (distinguished name) in X.509 certificates are not well-defined in the various standards, Table 2 shows a way to fill in the various parts of the subject in the device certificate.

Table 2: Proposed contents for X.509 certificate RelativeDN record

RelativeDN Type	RelativeDistinguishedName
C (Country)	The country the device was manufactured in.
O (Organization)	The name of the manufacturer.
OU (Organizational Unit)	The name of the factory the device was produced in. This field could also be used for the "security domain", for instance by appending a number to the factory name.
CN (Common Name)	MAC address or serial number of the device; must be unique for all certificates issued by that manufacturer / that CA.

The validity period of these default certificates must be long enough to accommodate the time span between production and the first certificate replacement. While a long key cannot be used for the certificates used in time-critical operations because the devices may have limited computational capabilities, a long key (e.g., RSA key $\geq$ 2048 bits) may be used for the default certificates, which are not used for time-critical operations. Moreover, a long key ensures that the certificate will not expire until the device is deployed.

This procedure mitigates the first attack vector as described in Section 3.3: A device that leaves the manufacturing process can only contain a single certificate root. If an attacker has installed its own root key during the production process, it is not possible to install an additional root key signed by the factory CA. In such case, the attacker would have to sign the command to install additional root keys. However, the out-of-band telephone call using SAS ensures the authenticity of the manufacturer. Thus, a malicious root key can be identified during the device's commissioning.

5. Installation of New Root Certificates

5.1 Problem Description

Once the system integrator has set up his CA infrastructure[2], the plant CA's root certificate must be installed on the devices in a secure manner. Unfortunately, since at that point there is no trust relationship between the plant CA and the rest of the world, an out-of-band method is required.

[2] The same mechanisms can be used to add an additional trusted root CA without the cooperation of the system integrator's CA (provided the manufacturer root CA was not deleted), for instance because the CA chosen by the system integrator has to be replaced.

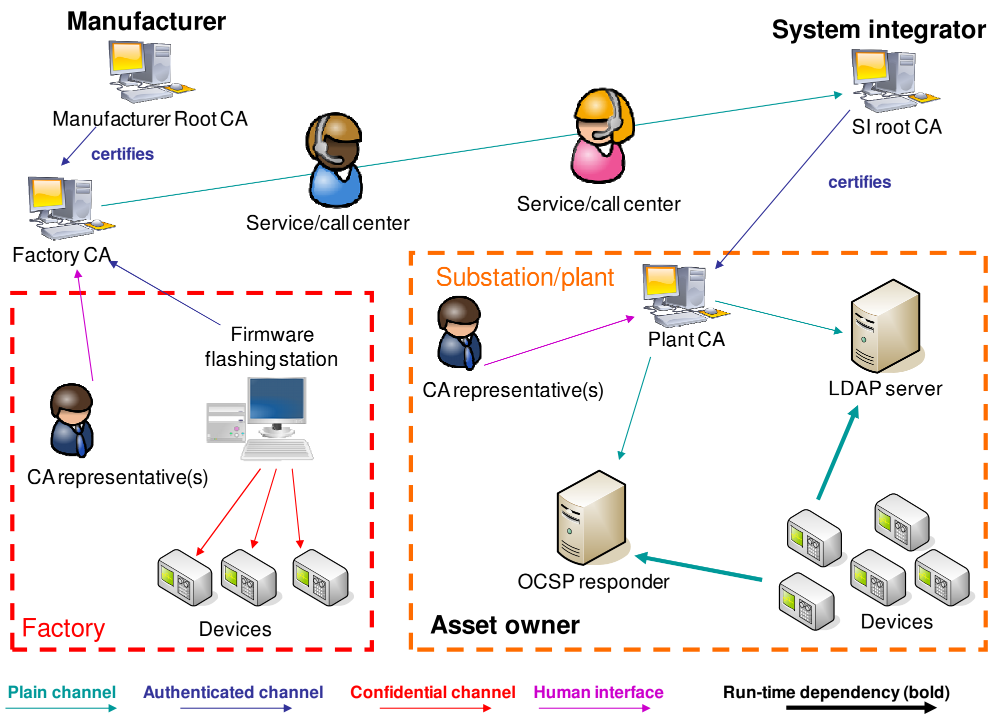

Fig. 2: Certificate Management Overview

5.2 Solution

A straightforward solution is to have cross-certified CAs between the manufacturer and the system integrator to extend the trust between both CAs [6]. The system integrator would be fully trusted and could immediately setup the devices. However, we cannot assume that such cross certified CAs exist for all system integrators and manufacturers. Thus, we present two options that allow the dynamic establishment of a trust relationship between manufacturer, system integrator, and plant.

The first option is for the manufacturer to "introduce" the system integrator's CA to them. For this method, an authenticated channel from the system integrator to the manufacturer is required. The second option is for the manufacturer to issue and sign a certificate for the system integrator. The latter can now install his root CA's certificate on the devices by using this certificate. This method requires a confidential channel[3] from the manufacturer to the system integrator in order to enable a secret transmission of the corresponding private key. There are several possibilities for an out-of-band authentication:

Face-to-face meeting: A representative of the system integrator hands over the root certificate to a representative of the manufacturer. If the CA is already up, the best possible time for this transfer would be during contract signing.

Paper mail: The system integrator sends printout of the cryptographic hash ($\geq$ 160 bits) of the root certificate on paper. The actual root certificate can now be sent over any channel (for instance on a USB key send with the letter, or by a separate e-mail).

Telephone (hash): The system integrator calls the manufacturer and transmits a cryptographic hash ($\geq$160 bits) of the root certificate over the authenticated voice channel. The actual root certificate can now be sent over any channel (for instance by a separate e-mail).

Telephone: The system integrator calls the manufacturer, and both parties perform a Short Authenticated Strings (SAS) protocol [7]. The SAS protocol minimizes the amount of authenticated data that has to be transmitted over the voice channel (20 bits $\approx$ 6 digits that must be transmitted). The non-authenticated part of the SAS protocol will take place on a plain data channel (e.g., TCP). To make the hash transfer user-friendly and less error prone, an agreed-upon list of words can be used instead of digits, for example the PGP word list [8], [9], which was developed for transmitting cryptographic hashes over the phone.

The manufacturer then signs an "Add root certificate" command (containing the operator's CA root certificate, authorized for the relevant security domain/list of devices), and transmits it to the system integrator, which can then relay this message to the devices. The devices then consider the operator's CA root certificate to be a trusted root. Finally, the system integrator can issue new certificates to the devices,

[3] If the system integrator generates the private key for this kind of certificate, the same situation applies as discussed the preceding paragraph, just with one level of indirection more (and more points of vulnerability).

containing all the required parameters and permissions. The system integrator may also delete the manufacturer's root certificate from the trusted store.

5.2.1 System Integrator Based Root Certificate Installation

An alternative is the transmission of a user certificate that is signed by the manufacturer to reconfigure the list of trusted certificates on the devices. The private key that goes along with that certificate must be transmitted to the system integrator in a confidential manner. The key must remain confidential until the root certificate of the manufacturer has been removed from the device, otherwise an adversary who finds the key can take control over the devices, for instance, by changing the root certificate, and demanding a ransom to reveal the corresponding private key.

Possible ways to establish a confidential channel:

Face-to-face meeting: Direct handover of the private key corresponding to the user certificate.

Paper mail: Since mail can be intercepted and read by a third party, it must be ensured that someone who intercepts the letter cannot access the private key. One way to proceed would be the following: first, the manufacturer sends a passphrase (at least 128 bits of entropy is recommended) in a tamper-evident envelope (as banks use to transmit the PIN code for credit cards). If that letter is lost or has been tampered with, the manufacturer can retry by sending a new letter with a different passphrase until the system integrator received the unopened letter (to be fully secure, an authenticated return channel for the system integrator is required to transmit an acknowledgement back to the manufacturer). Otherwise, if an adversary intercepts the letter, he could impersonate the system integrator telling the manufacturer that the passphrase was received properly), the manufacturer can send him the private key encrypted with the passphrase, along with the user certificate over any channel (e.g., by e-mail). All of this can happen in parallel to the device shipment, therefore latency is not a critical issue.

Having established a confidential channel, the system integrator can now add its own root certificate to the device and issue new certificates to the devices, containing all the required parameters and permissions. The system integrator should also remove the manufacturer's root certificate from the devices as otherwise anyone who finds out the private key corresponding to the user certificate could compromise the device. The devices must not allow the removal of the last root certificate with "root certificate management" privileges.

5.2.2 Discussion

The proposed solutions are analyzed regarding the following criteria:

Strengths Attributes that enhance the security of the approach.

Weaknesses Weaknesses that may harm the security of the approach.

Threats External conditions that may harm the security of the approach.

Cross-certified CAs

Strengths No further interaction between manufacturer and system integrator is required.

Weaknesses Substantial overhead beforehand.

Threats n/a

Manufacturer Based – Face-to-face Meeting

Strengths Authentication and authorization trivial.

Weaknesses Requires the CA of the system integrator to be set up before contract signing.

Threats The contract signers might not be familiar with security issues.

Manufacturer Based – Paper Mail

Strengths Low Cost.

Weaknesses High latency. Difficult to authenticate sender and to determine whether the letter is delivered to the authorized recipient.

Threats Recipient might not bother checking the cryptographic hash. Liability of manufacturer if the letter was a forgery.

Manufacturer Based – Telephone (Hash and SAS)

Strengths Low Cost.

Weaknesses Difficult to determine if caller is authorized by the asset owner. Inconvenience of transmitting relatively complex data over the phone. This process can be facilitated by using PGP wordlists [8], [9].

Threats Recipient might not bother checking the cryptographic hash. SAS is a relatively new cryptographic protocol.

System Integrator Based – Face-to-face

Strengths Authentication/authorization is trivial.

Weaknesses If the private key of the user certificate is compromised, devices can be rendered unusable.

Threats The contract signers might not be familiar with security issues.

System Integrator Based – Paper Mail

Strengths Low Cost.

Weaknesses If the private key of the user certificate is compromised, devices can be rendered unusable. "Return channel" depends on mail delivery company.

Threats Mail delivery service must be trusted for giving the letter to the right person.

Cross-certified CAs between manufacturer and system integrator renders the problem at hand trivial, but requires additional overhead beforehand. It can be considered the ideal choice if both manufacturer and system integrator already have a CA, i.e., the effort is already spent, or both plan to interact more frequently in the future.

If this is not the case, the use of the SAS mechanism in combination with PGP wordlists can be considered the most promising approach for a practically feasible out-of-band authentication. This allows the system integrator to call the manufacturer and transmit only a few words to the operator.

Establishing trust between manufacturers and system integrators and ensuring the verification of the certificates as described above mitigates the remaining attack vectors:

2. An attacker cannot install an arbitrary root key after the device has left the manufacturing process as the root key can only be installed along with a "install root key" command signed by the manufacturer's factory CA. In order to install its own root key, an attacker would have to call the manufacturer's call center and pretend to be the regular system integrator or operator. This, however, is noticed by the manufacturer during the commissioning phase when the regular customer calls the manufacturer's call center to install his root key as well. The second call will raise an alarm and instruct the customer to verify installed root keys carefully.

3. An attacker cannot install its own root key during operation: each new root key must be signed by the plant CA.

4. In order to reset a device, an attacker would have to get physical access. In addition, the failure of the device would immediately raise an alarm within the system software and instruct operators to inspect the device. In order to install the malicious root key, an attacker would have to perform the telephone authentication and successfully receive a signed device command and restore the initial configuration. As logs disappear from the device due to the factory reset, the system software detects the factory reset and issues a high priority alarm when the device is re-integrated into the system. If such an attack should be mitigated, a complete removal of the manufacturer's root key from the device can prevent the manufacturer from adding new root certificates to the device.

6. Summary and Conclusion

Certificate management for embedded devices faces many challenges, of which, based on the device's life cycle, the use cases "default certificate installation", "system integration", and "certificate replacement" have been discussed. Different alternative approaches have been proposed that meet the requirements for each use case with respect to industrial automation control systems. A discussion of the feasibility of different solution approaches has been presented and a solution has been recommended. Finally, an evaluation of the proposed solution has shown that it withstands critical attack vectors.

In the future, the authors plan to discuss the problem of certificate replacement and revocation to support certificates throughout the whole embedded device lifecycle.

References

[1] D. Dzung, M. Naedele, T. von Hoff, and M. Crevatin, "Security for industrial communication systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1152–1177, 2005.

[2] M. Naedele, "Addressing IT security for critical control systems," in *HICSS*. IEEE Computer Society, 2007, p. 115.

[3] J. Lopez, R. Oppliger, and G. Pernul, "Classifying public key certificates," in *EuroPKI*, ser. Lecture Notes in Computer Science, D. W. Chadwick and G. Zhao, Eds., vol. 3545. Springer, 2005, pp. 135–143.

[4] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "NIST SP800-57: Recommendation for Key Management – Part 1: General(Revised)," Tech. Rep., March 2007.

[5] C. Adams, S. Farrell, T. Kause, and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)," RFC 4210 (Proposed Standard), Internet Engineering Task Force, Sept. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4210.txt

[6] V. Casola, A. Mazzeo, N. Mazzocca, and M. Rak, "An innovative policy-based cross certification methodology for public key infrastructures," in *EuroPKI*, ser. Lecture Notes in Computer Science, D. W. Chadwick and G. Zhao, Eds., vol. 3545. Springer, 2005, pp. 100–117.

[7] S. Vaudenay, "Secure communications over insecure channels based on short authenticated strings," in *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, vol. 3621. Springer, November 2005, pp. 309–326.

[8] P. Juola and P. Zimmermann, "Whole-word phonetic distances and the pgpfone alphabet," in *Proceedings of the 4th International Conference on Spoken Language Processing, Philadelphia, PA, USA*, 1996, pp. 98–101.

[9] P. Juola, "Isolated word confusion metrics and the pgpfone alphabet," in *Proceedings of the Second International Conference on New Methods in Language Processing , Ankara, Turkey*, 1996.

[10] D. W. Chadwick and G. Zhao, Eds., *Public Key Infrastructure, Second European PKI Workshop: Research and Applications, EuroPKI 2005, Canterbury, UK, June 30 - July 1, 2005, Revised Selected Papers*, ser. Lecture Notes in Computer Science, vol. 3545. Springer, 2005.