Practical Security Recommendations for building OPC UA Applications

Version 3 // June 2018

Whitepaper

Security Working Group

Editorial

The increasing networking and digitization of industrial systems entails new security challenges that need to be addressed systematically. A trustworthy, secure handling of sensitive data such as product and production knowledge is just as necessary as the protection against attacks on the networked systems. In order to counteract potential damage, Information Technology (IT) security should be ensured throughout the development process of a system and its software, from the requirements phase all the way to the decommissioning of the system.

The OPC Foundation established a user group to build secure, connected products. The objective of this group is to enhance the use of IT security mechanisms in the context of Operational Technology (OT) through practical examples.

The group members develop best practices and guidelines for typical use cases under the management of the Fraunhofer IEM and the Hochschule Offenburg. The security expertise is based on the extensive industry knowledge and the latest research in this area. The group took into account the requirements of device and machine builders as well as those of industrial operators. The implementation of the presented solutions within self-run projects and customer projects underlined the need for security. The steady increase in attacks on critical infrastructure and industrial automation solutions, the economic and social threats, and the lack of understanding in security principles make it necessary to build a community to share requirements, use cases, and best practices. An open mindset and a thorough examination of the present situation, including defining the threats and risks, is a good starting point for improving protection of assets.

The group focuses on the communication standard OPC UA. In this proceeding second version of the document “Practical Security Recommendations for building OPC UA Applications”, the group adds recommendations for storing private keys and the use of pull and push certificate management. Furthermore, the group describes two real-world use cases that are in operation and use X.509 certificates for signing and encrypting OPC UA messages. Overall, the guideline gives an overview of the OPC UA security concept and how to use it.

As the chairmen of the group, we thank all participants for sharing their knowledge and their contributions to the user group. Finally, we would like to invite you to read this brochure and to contact us for participation and further information.

Uwe Pohlmann, Fraunhofer IEM

Prof. Dr.-Ing. Axel Sikora, Hochschule Offenburg

The members of the group are:

- → Ascolab
- → Beckhoff Automation
- → CERN, European Organization for Nuclear Research
- → DS Interoperability
- → exceet Secure Solutions
- → Fraunhofer IEM
- → Hochschule Offenburg
- → Microsoft Corporation
- → Software AG
- → Sparhawk Software Inc
- → TE Connectivity

Introduction

Today's devices and machines produce high-value data. For example, a production machine logs at which time it is used. However, the available data only becomes viable if it can be processed and used to improve a product, to offer a service, or to reduce the costs. For example, knowing the utilization of production systems can be used for offering overcapacity of the production system to other parties. Currently, the value of the available data is lost as the data is locked within its machine. Communication enables remote access to and processing of the data. Internet-based smart services enable new business cases, like production as a service, which mine the value of the available data. A prerequisite for smart services is that devices, machines, and smart services exchange data in a secure way. Otherwise, data, machines, and devices might be compromised or the value of the data might be monetarized by external parties. Figure 1 shows a typical use case for a connected factory. OPC UA is the best solution that realizes the use case in a secure way. Device and machine builders must ensure the data integrity and the data confidentiality. Furthermore, they must guarantee that the sovereignty of the data remains with the data owner. Currently, many devices and machine builders are struggling with these security challenges. Thereby, they give away the ability to use the data securely to improve or extend their own products and services or to reduce their operational costs in a secure way.

This is why members of the OPC Foundation have joined together their expertise, founding a security user group.

This document seeks to give a condensed overview of the recommended security measures, which are used in "best practice" installations.

This description

→ Gives an overview on the possible countermeasures laid out in the OPC UA specification

→ Reports on typical installations. Having said this, this paper does not claim to be complete.

→ Gives a snapshot of the situation at the time of compilation of this white paper (spring 2018). It is clear that security solutions need a regular update over time.

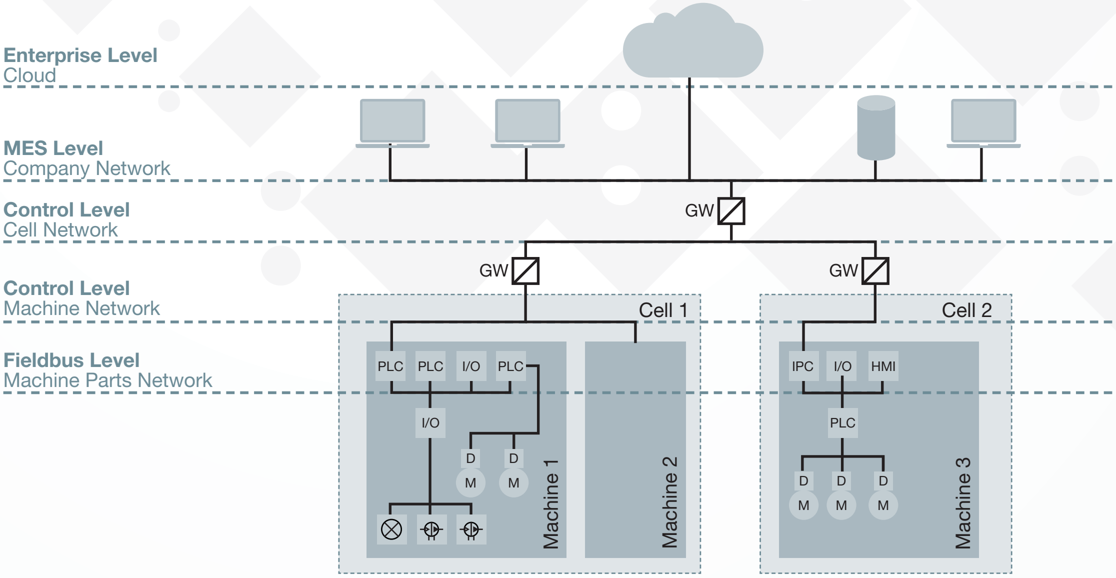
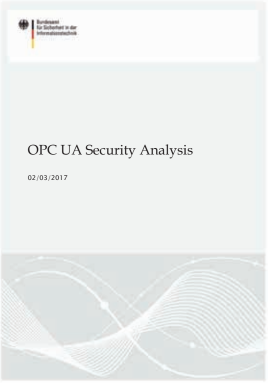


Figure 1: Connected Factory

Secure By Design

In 2015, and under the consortium leadership of the TÜV SÜD Rail, the Federal Office for Information Security (BSI) has performed a security analysis of OPC UA. The BSI is the first and foremost the central IT security service provider and national cyber security authority for the federal government in Germany. It shapes information security in digitization through prevention, detection and reaction for government, business and society. The OPC UA communication was analyzed systematically with regard to the Secure-Channel, Session and Discovery services according to the specification. The specification analysis has revealed no systematic errors, and has thus shown that OPC UA, in contrast to many other industrial protocols, provides a high level of security. On the basis of the analysis results, the OPC Foundation has improved the OPC UA specification and provided an annotated edition of the OPC UA security analysis. [1]



Bundesamt für Sicherheit in der Informationstechnik

OPC UA Security Analysis

02/03/2017

https://opcfoundation.org/security/

Figure 2: Building of the German Federal Office for Information Security in Bonn, Germany

Source: Bundesamt für Sicherheit in der Informationstechnik

Scope of the Security Model

The OPC UA security architecture comprises the following concepts [2]:

Trusted Information (CIA triad)

- → Confidentiality, by encrypting messages on the transport layer
- → Integrity and authenticity, by signing messages on the transport layer
- → Availability, by restricting the message size and returning no security related codes

Access Control (AAA Framework)

- → Authentication by username and password or X.509 certificate on the application layer
- → Authorization to read, write values of a node or to browse the information model based on the access rights of the information model, access rights of the user or of the user's role
- → Accountability, by generating audit events for security related operations

The following concepts are outside the scope of the OPC UA security architecture [2]:

- → Organizational Issues, like security training of personnel, the security lifecycles and policies or how to handle physical access. OPC UA does not replace the information security management sys-tem (ISM) that the ISO 27001 defines. OPC UA security aspects should be used to im-ple-ment defense/security in depth.

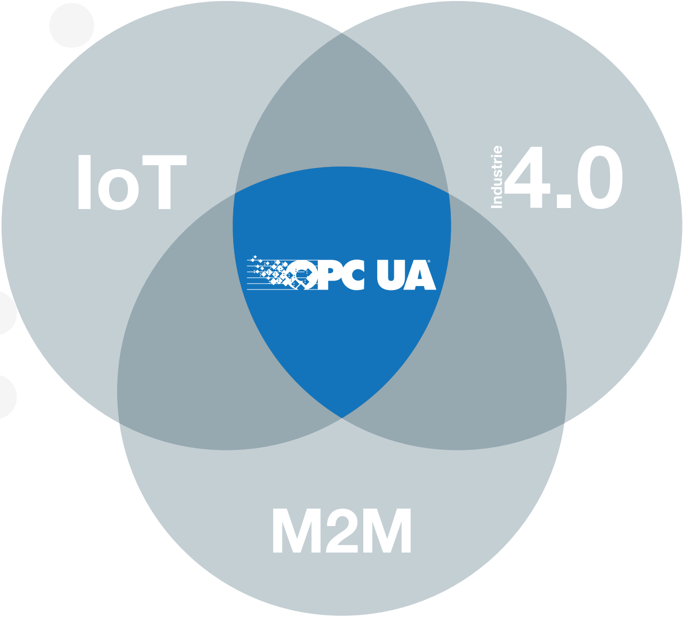


Figure 3: OPC UA serves as the common data connectivity and collaboration standard for local and remote device access in IoT, M2M, and Industrie4.0 settings.

Security Model

Security is a fundamental requirement for OPC UA and it is therefore tightly integrated into the architecture. UA security mechanisms are based on a detailed analysis of security threats. UA security deals with authentication of users and UA applications, integrity and confidentiality of the exchanged messages and the validation of function profiles.

UA Security complements the preexisting security infrastructure within a company. Figure 4 shows the scalable UA security concept. It consists of three levels: user security, application security, and transport security.

The mechanisms of UA user level security grant access to a specific user and its role while setting up a new session.

UA application level security is also part of the communication session and includes the exchange of digitally signed **X.509 certificates**. Application instance certificates that are exchanged during secure channel establishment are used to authenticate an application. The supported UA security profile that can be certified by the OPC Foundation defines which security mechanisms a UA application supports.

Transport-level security can be used to sign and encrypt each message during a communication session. Signing ensures the message's integrity and authenticity, while encryption prevents eavesdropping.

The UA security mechanisms are implemented in the UA stack, i.e., they are included in the software package distributed by UA stack vendors, so UA applications just have to make use of it. It is however the responsibility of the UA application developer (i.e. the machine builder, etc.) to configure the UA server, according to the requirements that he has to adhere to. Refer to [3] for further reading.

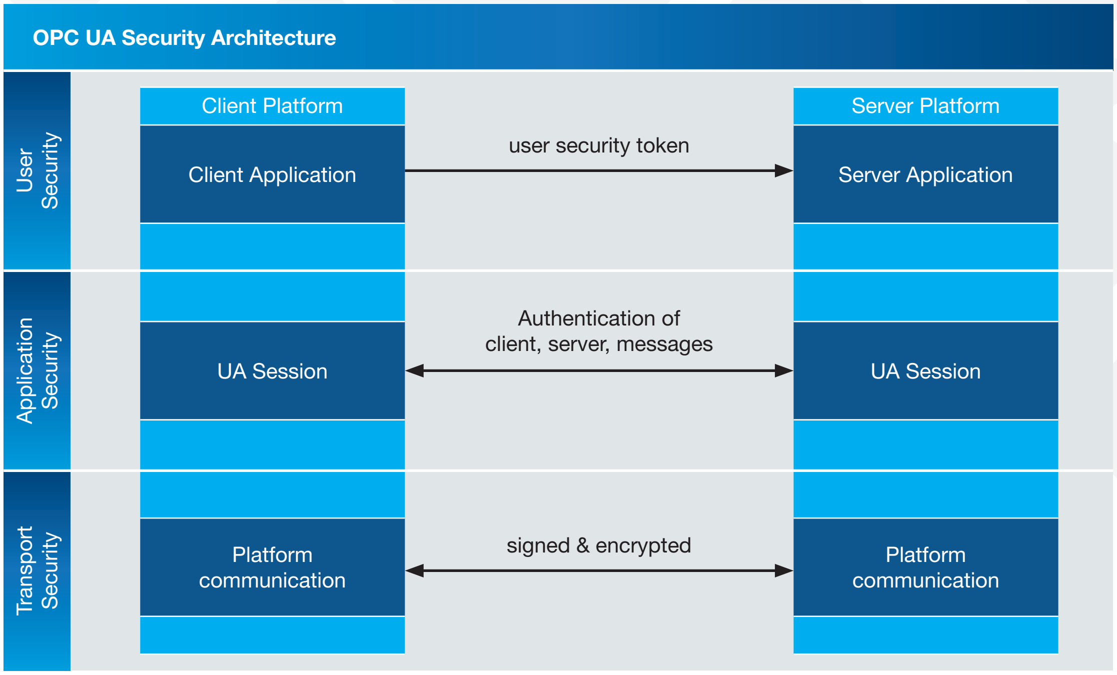


Figure 4: Scalable Security Concept

X.509 Certificate Management

A digital certificate is a structure that links an identity with a distinct entity such as a user, a product or an application instance. A digital certificate has an associated asymmetric key-pair. A certificate contains a public key and identity information of the owner. Furthermore, the owner of the certificates has a private key. The private key is used to sign and decrypt messages and the public key of the remote partner is used to encrypt and validate a message. A certificate is only valid for a defined period of time. The certificate format for OPC UA is defined by the X.509 standard and the required fields are defined in the OPC UA specification.

Self-signed certificates

A certificate created and signed by users themselves using a tool like OpenSSL is called a self-signed certificate. Accepting self-signed certificates can be dangerous if the operator currently in charge of making the trust decision is not well trained in X.509 certificate management. This is because it is not clear from the information provided whether one can trust the properties of the certificate as no trusted 3rd-party approved the correctness of the properties. Yet, a self-signed certificate is an inexpensive solution because one does not have to pay or go through additional effort for being trusted. In contrast, using a trusted certification authority (CA) can build a chain of trust from the remote party all the way to a trusted root (i.e., a root the communicating parties trust already).

CA-signed certificates in a PKI

A certificate authority (CA) is an entity that issues digital certificates [21]. It must be a trusted party, which is trusted by the owner of certificates and by the users that should accept the certificate. In a Public Key Infrastructure (PKI) there is at least one CA or there are even more hierarchically organized CAs that build a chain of trust. By providing a certificate revocation list a CA can revoke the trust of a certificate or of another CA that is at a lower level. A CA must meet high-security requirements and the private key matching the public key contained within its certificate must be kept in a safe place. By using multiple issuing CAs, it is possible to revoke the trust of one issuing CA without harming the other issuing CAs. A CA can distribute issued certificates by the following distribution channels [4]:

→ Manual Distribution Mechanism: The certificates are transported on a storage medium or via secure email communication. The certificates are installed manually. This requires a large amount of manual labor, especially for large deployments.

→ Custom Distribution Mechanism: The requesting application uses a well-known public repository, where it uses its credentials to authenticate, download and install the certificate from. A custom solution usually has the disadvantage that it can be more easily compromised by a hacker.

→ Automatic Certificate Management: The certificates are distributed via a **Global Discovery Server.** This option is explained in the Section “GDS Security Features”.

Storage of Private Keys

Private cryptographic keys are among the most sensitive digital assets one can think of. A private key empowers its possessor to exercise all rights associated with the key (e.g., adopting an identity, signing certificates, etc.). Thus, private keys should be stored in a secure way, i.e., such that no one except the rightful owner can ever get hold of it.

File-System Storage

Unless dedicated hardware or a special service is available, private keys may be stored in the standard file system storage. It is important to ensure in this case, however, that appropriate read- and write-permissions are set. Only legitimated system processes should be able to access the private key(s).
It is even better to use a separate system process for key operations and export its services to a limited number of legitimate application processes. In that way, the private key may still be used by a compromised application but is not accessible for permanent misuse. In all cases, the encryption of private keys should be considered in order to impede attacks.

Local Certificate Stores

Several operating systems and software packages provide a dedicated software tool for managing not only certificates but also corresponding private keys. The benefit of using local certificate stores is a coherent and secure storage within encrypted files.

PKCS#12-based Keystore

PKCS#12 is an archive file format used to store cryptography objects, like the bundle of a certificate and its private key. The file extension is normally “p12” or “pfx”. Typically, OpenSSL is used to create a PKCS#12 file by importing the private key and the corresponding certificate. An export password can be set during the creation of the PKCS#12 file.

Windows Certificate Store

Windows has different certificate stores for the current user and the local machine. Installed certificates can be viewed or deleted with the Windows Certificate Management Console (certmgr.msc). Windows can import or export “pfx”-files. Furthermore, applications can create custom certificate stores. By default, Windows allows only the owner and the System account to access the private key of a certificate.

Java Keytool

Java Keytool is a management utility for cryptographic keys and certificates. In its default configuration, Java Keytool uses the standard file system as an underlying storage service. However, as a further layer of security beyond the file system’s access permissions, Java Keytool allows using passwords to protect the cryptographic assets in its keystore. Generally, Java Keytool blocks all means for exporting private keys.

Hardware-based Storage

The best way to store cryptographic private keys is inside dedicated hardware modules. Such modules typically feature protection against physical intrusion and never expose the private key itself to the outside world. Rather, the use of the private key is offered as a service; i.e., cryptographic operations involving the private key are performed inside the module. The downside is the need for additional hardware components.

TPM

A Trusted Platform Module (TPM) is a special system component (software/hardware) that can monitor the state of its host platform. Only if the platform's integrity is successfully validated the TPM discloses its sensitive data for the use by the host. Storing private keys in a TPM can thereby prevent a compromised system from illegitimately using a private key. A TPM is defined by the corresponding TPM specification [20].

HSM

A Hardware Security Module (HSM) is a special hardware component (chip, extension board, or appliance) that can – among other things – generate and securely store cryptographic keys. In contrast to a TPM, the usage of its keys is not necessarily coupled with platform integrity. However, an HSM prevents key extraction by a compromised system.

Cloud-based Storage

Azure Key Vault

Azure Key Vault is an example of a cloud-based storage service. It helps safeguard cryptographic keys and secrets used by cloud applications and services. By using Key Vault, one can encrypt keys and secrets (such as authentication keys, storage account keys, data encryption keys, "pfx"-files, and passwords) using keys protected by HSMs. If one chooses to do this, Microsoft processes the private keys in validated HSMs for added assurance.

GDS Security Features

A Global Discovery Server (GDS) is an OPC UA server that provides services allowing servers to register themselves and also allows clients to search for servers to connect to. Furthermore, it provides X.509 security certificate management services for clients and servers. A video [19] gives a detailed introduction to the GDS. Figure 5 shows the structure and use cases of a connected factory that uses a GDS.

Claim-Based Security

A GDS provides the master database including roles, like security admin observer. The role management integrates with existing user and role management systems. Roles have access permission for nodes within the OPC UA Information Model. Users provide credentials to authenticate and to get a granted role and the corresponding access rights for a UA session. The identity information and access rights are handled via a claims-based authorization mechanism, which, e.g., Kerberos or OAuth2 provides.

Automatic Certificate Management

Automatic certificate management means that the GDS maintains the X.509 certificate provisioning and renewal for a list of UA applications that are available in an administrative domain. The GDS provides a certificate manager to request and to update certificates and trust lists. The certificate manager supports pull and push-based distribution models. Managing certificates by using the certificate manager scales better than handling certificates manually.

Pull Management

An OPC UA application communicates with a GDS that provides certificate management to use pull management. The application acts as a client and uses methods of the certificate manager of the GDS. Thereby, it checks if its certificates need to be updated and may request to sign a new certificate or to issue a new public and private key. Furthermore, trust lists with information about issuer CAs and other OPC UA applications can be updated regularly. The GDS checks if the user of the OPC UA application has the rights to perform the action and if the current configuration of the certificate manager allows the request. A request has to end with an explicit FinishRequest message.

Push Management

A GDS management client handles the communication between the GDS server and an OPC UA application that supports push management. Therefore, the OPC UA application acts as a server and has to provide the address space called server configuration nodes for push management. Thereby, the GDS management client asks the OPC UA application server to create a signing request. Additionally, it may ask to update the certificate of the server with a signed CA certificate or asks to update the trust lists. Any configuration change has to be applied explicitly.

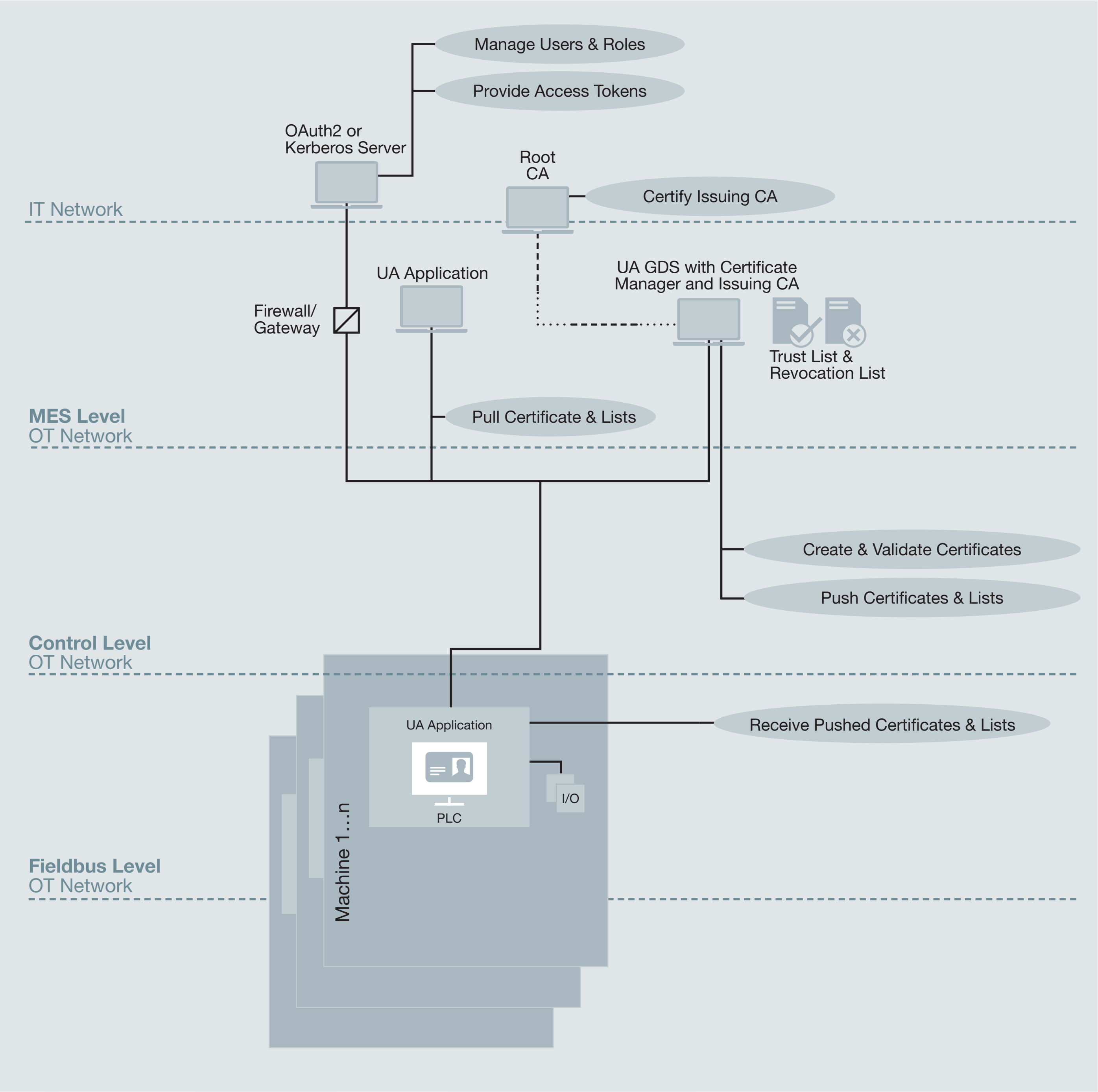
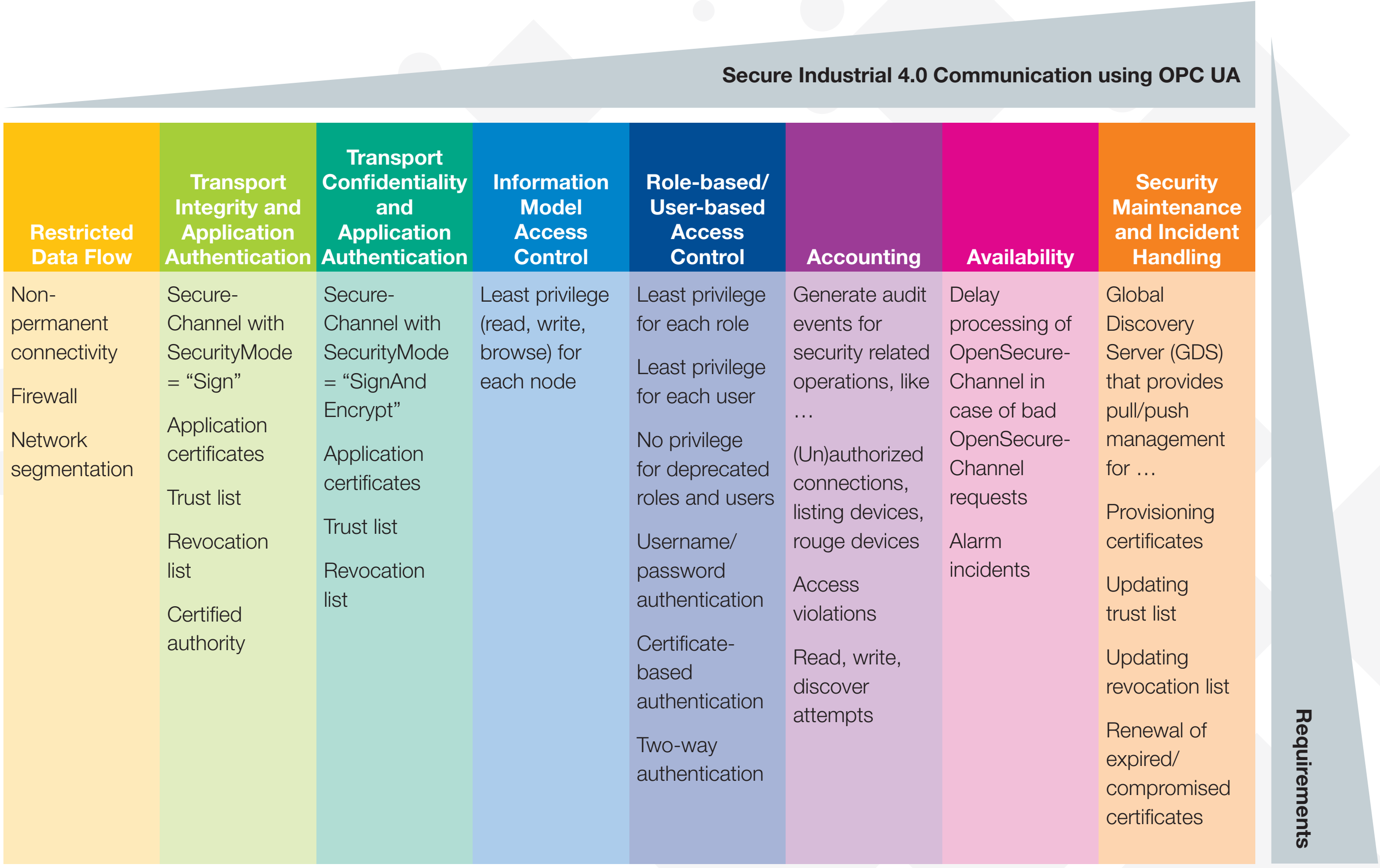


Figure 5: Connected Factory with new UA Features and Use Cases

Defense in Depth

The security concept “defense in depth” realizes the information assurance by using multiple layers. As a result, an attacker must break through several barriers before compromising the whole system. Figure 6 shows the security features that OPC UA offers within the different layers. Within each layer, several requirements can be fulfilled by using the corresponding OPC UA feature to improve the overall security.



Restricted Data Flow	Transport Integrity and Application Authentication	Transport Confidentiality and Application Authentication	Information Model Access Control	Role-based/User-based Access Control	Accounting	Availability	Security Maintenance and Incident Handling
Non-permanent connectivity Firewall Network segmentation	SecureChannel with SecurityMode = “Sign” Application certificates Trust list Revocation list Certified authority	SecureChannel with SecurityMode = “SignAnd Encrypt” Application certificates Trust list Revocation list	Least privilege (read, write, browse) for each node	Least privilege for each role Least privilege for each user No privilege for deprecated roles and users Username/password authentication Certificate-based authentication Two-way authentication	Generate audit events for security related operations, like … (Un)authorized connections, listing devices, rouge devices Access violations Read, write, discover attempts	Delay processing of OpenSecureChannel in case of bad OpenSecureChannel requests Alarm incidents	Global Discovery Server (GDS) that provides pull/push management for … Provisioning certificates Updating trust list Updating revocation list Renewal of expired/compromised certificates

Figure 6: Defense in Depth Using the OPC UA Security Architecture

Recommendations for Using OPC UA in a Secure Way

When securing the communication with the OPC UA protocol,the following settings are of central importance:

→ **SecurityMode:** The SecurityMode [5] should be 'Sign' or 'SignAndEncrypt'. This ensures that, among other things, authentication at the application level is enforced. The SecurityMode 'None' does not provide any protection! SecurityMode 'SignAndEncrypt' must be used if not only integrity but also confidentiality of data has to be ensured. [6]

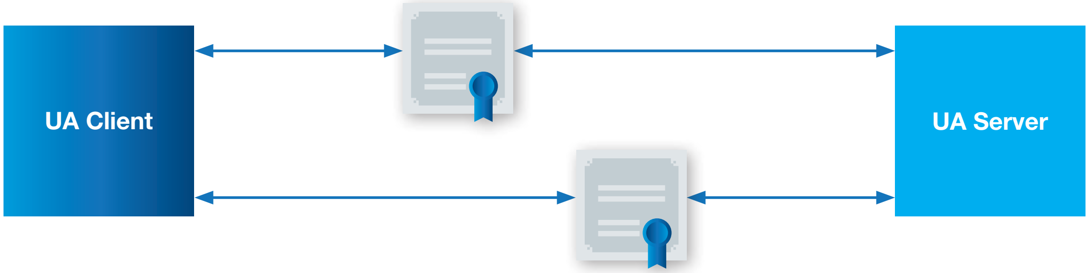
→ **Selection of cryptographic algorithms:** At a minimum, the SecurityPolicy [7] 'Basic256Sha256' should be chosen, provided that this is technically possible, i.e., all existing clients the server needs to interact with also supports this policy. Note that a good client connection strategy must start with the most secure profile, check that this is supported by the server and then try the next best thing until a common profile is found. Weaker security policies use outdated algorithms such as SHA-1 and should not be used. [6]

→ **User authentication:** The possibility of logging in with the identifier 'anonymous' should be used only for accessing non-critical UA server resources as it does not provide any protection (what data is deemed non-critical is at the discretion of the UA application developer). When this generic identifier is used it is not possible to trace who has changed, for example, the data or configuration on the server side. Also if no adequate restriction of the rights of the identifier 'anonymous' was configured then an attacker could use this identifier to read or write data in an unauthorized manner. [6]

→ **Certificate and private key storage:** Never store private keys or the corresponding certificate files (.pfx/p12) on an unencrypted file system. Use the dedicated certificate stores of your operating system and use operating system capabilities for setting the access rights. TPM modules or external secured hardware, like USB-based authentication tokens to store certificates and/or private keys improve the security level.

→ **Using certificates:** Don't accept connections which do not provide trusted certificates. Especially, self-signed certificates should not be trusted automatically, i.e., without an additional verification. If the certificates are not self-signed, a Certificate Authority (CA), e.g., for all OPC UA applications of a company is required. The certificates of the Certificate Authority are either self-signed or signed by another Certificate Authority. Certificate Authorities can be multilayered. (cf. [2])

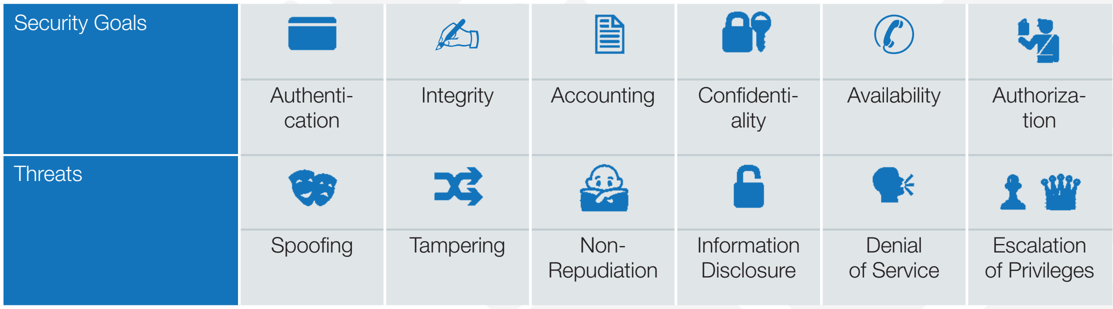
→ **Managing and maintaining certificates:** Use certificate trust lists and certificate revocation lists to manage valid certificates. Only trusted users or processes should be allowed to write these lists. The lists should be updated regularly and only authorized user should be able to change these lists.



Use Cases

Attackers that have no physical access to devices and machines often use their communication interfaces to start attacks. The VDMA Industry 4.0 Security Guidelines [9] recommend to use strong, standardized, and state-of-the-art protocols, e.g. the TLS family for TCP/IP-based communication. However, there are many nuances to establishing secure communications for various networks. For instance, the TLS protocol is only a tool integrated into various communication protocols, which means they should be correctly configured and maintained. Moreover, some fieldbus networks do not use IP-based communications and require real-time communication.

In this chapter, we describe two use cases, their security goals, threats, and solutions based on the OPC UA security model. Figure 7 shows the used symbols for the considered security goals and their corresponding threats.



Security Goals	Authentication	Integrity	Accounting	Confidentiality	Availability	Authorization
Threats	Spoofing	Tampering	Non-Repudiation	Information Disclosure	Denial of Service	Escalation of Privileges

Figure 7: Legend of Security Symbols

Custom Certificate Distribution and Operation for a Condition Monitoring System

Example Use Case Developed by Beckhoff

This application was done some years ago before the GDS certificate functionality was available. Therefore, it makes no use of automatic setup procedures for security. It demonstrates the certificate management processes, which need to be established during the whole lifetime of secured communication, thus, beyond the lifetime of certificates, which should not be more than approx. 5 years (depending on governmental suggestion regarding the key length and used algorithms).

Energy data of buildings needs to be collected in a central manner for analyzing and establishing an alarm management system, i.e., if a water leakage is detected. The scenario is a special use case of condition monitoring. Figure 8 shows the architecture of the whole system. Different organizations require to operate such a monitoring system. Their demand is to aggregate different energy data from various places with different requirements. Some places are connected via a VPN-tunnel, some places have a dialup connection, and some places have a classical internet gateway connection.

Beckhoff developed an energy data logger device, which is connected to the cloud via OPC UA to meet the requirements. The data logger device is a small PLC mounted in a small cabinet within a building or in the countryside. The logger collects and caches the data, which is produced by the devices within a building or by the devices in the countryside. The devices are connected by an MBUS interface to the logger PLC. The logger PLC pushes the cached data to a central cloud service via an OPC UA channel. The cloud provides configuration data to the logger PLC – like the MBUS configuration and intervals for collecting data.

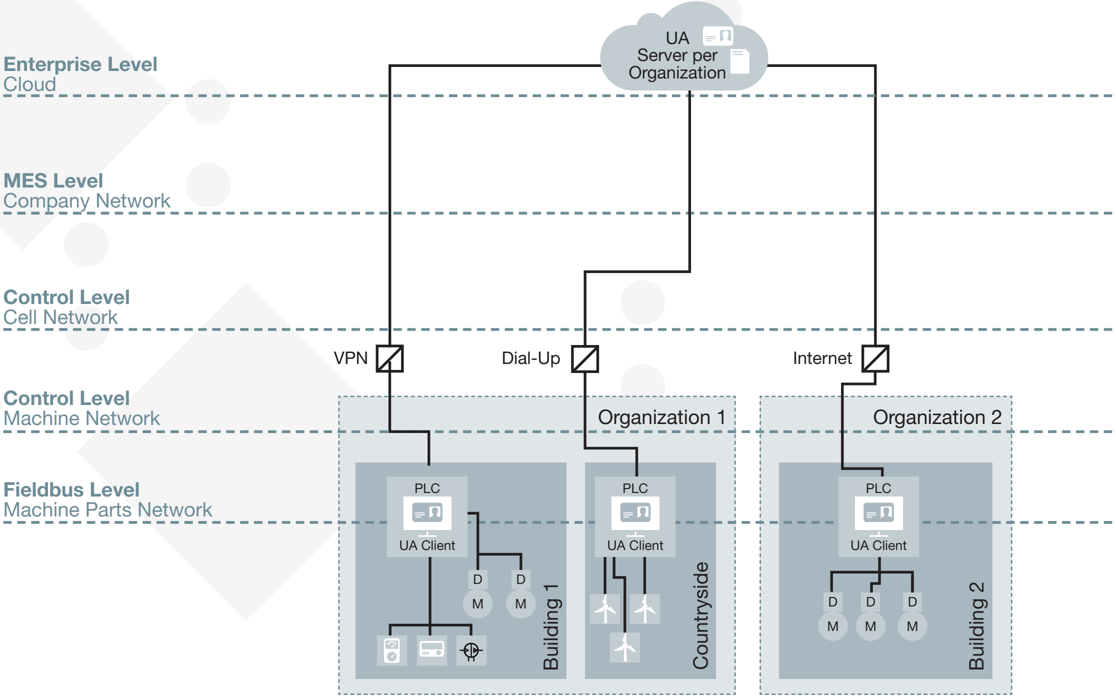


Figure 8: Secure Communication of PLCs with a Cloud for Condition Monitoring of Energy Data

Security Goals and Threats

Table 1 shows the considered security goals and threats of this use case.

The security goal authentication is the most relevant goal for this use case. Only authenticated OPC UA applications should be able to provide data and should be able to read data. Furthermore, the logged data of each organization should only be visible and accessible by the organization itself. Private keys should not be disclosed. Also, attackers should not be able to impersonate the identity of an organization. This case does not differentiate users/roles and their access rights. The authentication is done only via application certificates.

Integrity is important because each organization wants to ensure that the collected data is valid, correct, and that collected data belongs to its devices and PLCs. The logged data may be used for critical actions, like controlling or billing. The data is useless if it is altered by an unauthorized person. Therefore, tampering of the data is a certain security threat. The local insecure MBUS connection is considered to be protected by physical access restrictions and not by the software, i.e., the PLC is located in a secure area.

Accounting and the corresponding threat non-repudiation is not considered in the description of the presented solution. Nevertheless, accounting is done as a monitoring system within the receiving servers and enables to detect security incidents. Even logs have to be protected against unauthorized changes or deletion.

The security goal confidentiality is important for the use case because the data is sent through the public internet and the logged data of a certain organization should not be disclosed to another organization or an attacker. Therefore, it has to be encrypted.

Availability is not in the focus of the use case because the PLC collects and buffers the data independent of the availability of the OPC UA connection. The PLC is able to work standalone for a certain period of time. Therefore, the solution does not consider Denial of Service attacks.

Authorization must be ensured to separate the data accessibility of all organizations. However, we did not consider the elevation of privilege as a security threat because we trust that the used OPC UA client and server handle security policies correctly.

💳	✍	📄	🔐	✆	👮
yes	yes	partly	yes	no	partly
🎭	🔀	🙊	🔓	🗣	♟ ♛

Table 1: Considered Security Goals and Threats

Summarized Security Requirements

- → Encrypted and signed two-way communication – between data loggers and the central cloud system
- → Revoking access from compromised data loggers
- → Supporting seperate organizations, where each one hosts multiple data loggers
- → Processes for manufacturing data logger PLCs, which are ready to run securely by default and by an easy installation
- → Processes for renewing certificates as well as general software updates

Assumptions Regarding Security

- → Data logger PLCs are located in a secure location/cabinet, where only authorized persons have access. Thereby, the risk of an attack on the insecure MBUS connection is lowered
- → Security events are monitored
- → Private keys should not be disclosed

Technical Solution

For the encryption of the communication, X.509 certificates are used. These are located on the one hand at the PLC (as UA Client) and on the other hand on one UA Server per organization (cf. Figure 8).

A certificate authority (CA) per organization is used to separate access to the different organizations. A "Root CA" of the CloudProvider is used to sign the CAs of the organizations.

Whenever a PLC is compromised (detected to be stolen or the building/housing of the PLC was harmed) the certificate is added to the Certificate Revocation List (CRL) of the corresponding organization's UA server, which inhibits the compromised UA client to connect and send further data or read configurations. The use case of a compromised server is covered by corresponding management processes.

Maintenance Solution

Additionally to the technical solution, the scenario requires maintaining solutions to cover the capability of replacing outdated certificates but also to upgrade software, which could also be required for supporting new security algorithms.

The cloud system provider sends these signed certificates and private keys encrypted to the manufacturer via a separate channel. Additionally, the cloud provides the public key of the organization's CA. When an organization orders a data logger, the manufacturer takes a certificate and the organization's public key CA from the batch and places it on the PLC. Furthermore, the PLC manufacturer sends a notification to the cloud provider including the thumbprint of the used certificate. Thereby, the cloud provider knows that the certificate is now going to be online.

The PLC sends a notification to the cloud system provider, which provides the thumbprint of the certificate and its location when the operator organization mounts and installs the PLC by adding power, internet connection, and the building communication cable. As a result, the cloud system provider has the information, which PLC and which corresponding certificate is installed at which location.

Half a year before a certificate reaches its expiration date after 5 years, the cloud provider sends a reminder to make an order for a replacement certificate and private key. The manufacturer sends the new signed certificate, the private key, and potentially a new CA certificate to the operator organization via a separate channel. Furthermore, the manufacturer sends updates. The operator organization replaces the old certificate, private key and installs updates.

Automatic Certificate Distribution using a GDS

Example Use Case Developed by Microsoft

In the course of providing a more flexible Industry 4.0 compliant production, i.e., producing customer-specific product variants, factories become more and more flexible. As a result, new machines have to be integrated into the production system on demand. Therefore, the security requirements concerning the management of certificates are stronger than within the former use cases. Using local trust and revocation lists in such a flexible environment results in high maintenance costs and does not scale very well.

This use case uses the automatic setup functionality of OPC via a UA Global Discovery Server (GDS) and a central certificate manager in combination with cloud functionality, e.g., realized by the Microsoft Azure cloud system. It demonstrates the certificate management processes, which need to be established during the whole lifetime of secured communication.

Figure 9 shows the architecture of the whole system. The OPC UA applications (server/client) are connected to an edge server that implements the GDS functionality for registration and certificate handling. This service also implements a GDS management client to update certificates and trust lists on connected OPC UA servers with push support.

The edge server is connected via a secure transport protocol, e.g., an encrypted AMQP, MQTT, or HTTPS connection, to the Azure IoT hub in the cloud network. Note that the GDS is located within the company network and is also available with reduced functionality without a cloud connection.

On the enterprise level, the IoT hub is connected to an OPC Twin microservice for OPC client- and server registration. This microservice is connected to a management dashboard running as a web app, where users can manage the OPC UA applications via a browser. Furthermore, the IoT hub connects to a GDS Vault microservice to store certificates and trust lists.

The GDS Vault microservice handles new keypair requests, certificate signing requests (CSRs), and trust lists. It leverages Azure Key Vault to store private keys and to sign certificates in a secure and protected area.

Security Goals and Threats

Table 2 shows the considered security goals and threats of this use case.

Authentication ensures that only known applications are able to publish telemetry data to the cloud. The security threat spoofing is considered in the use case because it should not be possible to masquerades as another user to access data of that attacked user.

The integrity of the data is important. The data is useless if it is altered by an unauthorized person. An attacker should not be able to tamper with the data.

Accounting of security events is important to detect attacks but is not discussed within the shown solution. Therefore, non-repudiation is not considered as we do not describe accounting functionality. Nevertheless, the OPC UA security events should be stored and monitored, e.g., by a security information and event management system.

Confidentiality is an important goal because the telemetry data of the OPC UA servers should be transferred to the cloud without being disclosed. Therefore, the data has to be encrypted.

Availability is an important goal but this use case does only discuss tasks for certificate renewing, which avoids downtime. We did not consider Denial of Service attacks and countermeasure like load balancing to improve the availability.

Authorization is important to ensure that only authorized commands are sent to the OPC UA applications. Furthermore, only authorized OPC UA application should be able to send data to the cloud. We assume that either the Azure Web App, the Azure cloud, or the OPC UA applications check the user credentials. Additionally, we assume that all involved parties have the required access rights for performing their tasks. Furthermore, we do not consider the elevation of privilege as a security threat because we assume that all communication partners handle their security policies correctly. The used access tokens are only valid for a short period.

💳	✍	📄	🔐	✆	👮
yes	yes	no	yes	partly	partly
🎭	🔀	🤐	🔓	🗣	♟ 👑

Table 2: Considered Security Goals and Threats

Summarized Security Requirements

→ Encrypted and signed bothway communication between all applications
→ Valid X.509 certificates for each OPC UA application
→ Central certificate management
→ Revoking access from compromised UA applications
→ Process for certificate provisioning and renewal

Assumptions Regarding Security

→ User credentials and authorization are checked on the cloud level and by each OPC UA application
→ OPC UA application trust the certificates provided by the certificate manager of the GDS
→ Each application handles its security policies correctly and uses only access tokens with a short lifetime to avoid an elevation of privilege
→ The operating system clock of all machines is secured against attacks and is synchronized regularly using a network time protocol
→ Private keys and trust/revocation lists are stored securely
→ Signed and encrypted channel
→ Security policy basic256Sha256

Technical Solution

In this use case, we use a central GDS to secure the communication between the OPC UA application within the company network and between the company network and the cloud network. The GDS implements a central certificate manager that provides X.509 certificates to the other OPC UA applications or signs their certificates. Therefore, the GDS is a self-signed CA or uses a subCA. Figure 9 shows the scenario where several machines and their UA server running on a PLC are connected the GDS.

The URL of the GDS is known to the UA applications using an offline configuration or alternatively by using the LDS-ME discovery. Furthermore, the certificate of the cloud provider that signs the certificate of the GDS is contained within the applications trust list and each UA application has preconfigured user credentials and a valid vendor certificate, which allows it to request new certificates. The UA applications use pull management via an OPC UA secure channel to request certificates renewing and to update trust lists. The UA applications always check if the domain in the URL of the GDS matches one of the domains in the certificate. This configuration provides the best protection against accidental registration with rogue certificate managers.

The procedure for getting a new certificate from the GDS via pull management is as follows: Once an OPC UA application is registered with the GDS, it opens an OPC UA channel with its certificate and user credentials. The certificate authenticates the UA application and the credential authorizes to request a new certificate from the GDS. Thereafter, it requests a certificate signing or an update of the trust lists. The GDS uses the GDS Vault microservice via the IoT hub to create a new valid keypair or a signed certificate that is valid for the configured lifetime, provides the credentials to the requestor and puts this certificate on its trust lists. The UA server installs the received certificate and closes the secure channel. Afterwards, the UA server opens a new channel, using the new certificate and requests a new trust and revoke list from the GDS. The GDS provides these. Finally, the UA application replaces its local trust and revoke list with the received lists. The pull management has the advantage that no central control is required and that the OPC UA application can handle their certificate management autonomously by itself. However, as a drawback, no central security management for all OPC UA applications can be implemented.

Therefore, as an alternative to the pull management a push management can be used to realize the use case. In this case, all OPC UA applications require the implementation of the server configuration address space, which allows a GDS management client to renew application certificates and to update trust lists in a central way. Thereby, it is also possible to trigger the renewal from the management dashboard via the OPC Twin microservice and the IoT hub.

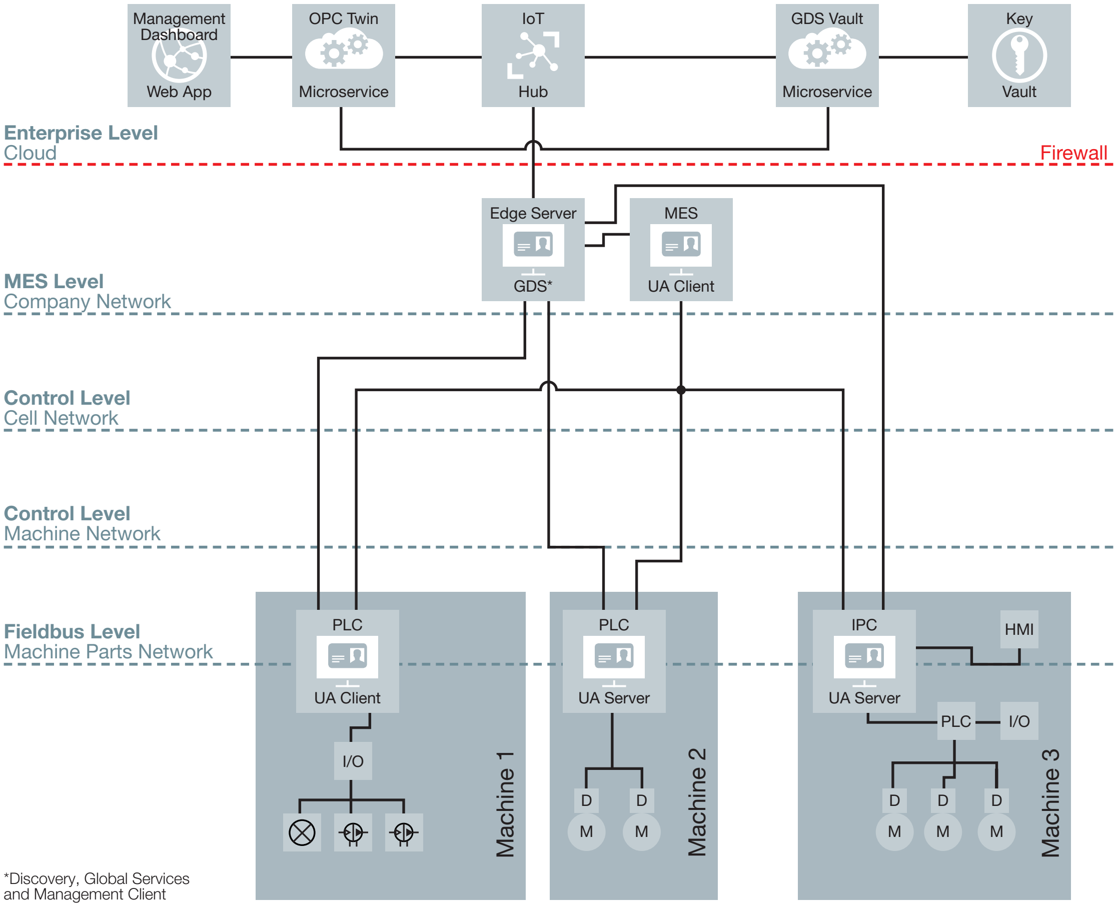
The private keys are stored in a local key store on each computer. Only the operating system users with permission to run the UA application as an administrator can access the server configuration nodes to update the private keys, certificates and trust lists in the key store. Anyone else who is not using an account authorized to run the UA application has no rights to access the server configuration.

Maintenance Solution

Reasonable validity periods for the certificates of the OPC UA applications are between 1 and 2 years. The validity period for the CA certificate that is located within the GDS has to be longer. Therefore, reasonable validity periods are between 2 to 5 years. In the case of a threetier CA architecture, the cloud CA has a longer validity period of 3 to 10 years. In any case, the validity periods should not exceed recommended lifetimes for the used cryptographic algorithms and key lengths.

Depending on push or pull model for certificate handling, the renewal and update is either triggered by the UA application or by the GDS management client. The renewal of the certificates should be performed a reasonable time before a certificate expires, e.g., after ¾ of its lifetime but at least a day before it expires. Many UA applications require a restart after renewing its certificate. Therefore, it is recommended to reserve an explicit maintenance time slots to renew certificates. The renewal of certificates must be performed supervised in cases of a compromised certificate of a UA application. In this case, the GDS cannot decide easily if the UA application is the correct one or a rogue UA application. The GDS can check if the domain in the URL of the UA application matches one of the domains in the application's certificate and can validate of the fields of presented certificate matches to the registered certificate of the application. This check adds additional security. Besides renewing certificates, trust lists including the revocations lists should be updated regularly to protect against compromised certificates and to announce and revoke new trusted CAs and self-signed certificates.

Under normal circumstances, UA application reject expired certificates for communication which were not being renewed in time. As an exceptional case, OPC UA allows being configured to accept expired certificates. Thereby, it is possible to encrypt and sign the communication that is required to renew certificates. However, such a configuration should only be active for a short period of time in an exceptional case because it can be used by attackers that have access to old certificates, e.g., from discarded devices.



*Discovery, Global Services and Management Client

Figure 9: Certificate Management via a GDS and the Azure Cloud

Bibliography

[1] OPC Foundation, „OPC UA Security,“ [Online]. Available: https://opcfoundation.org/security

[2] OPC Foundation, "OPC Unified Architecture Specification, Part 2: Security Model, Release 1.03," Scottsdale, USA.

[3] R. Armstrong and P. Hunkar, „The OPC UA Security Model,“ OPC Foundation, Scottsdale, USA, 2010.

[4] A. Fernbach and W. Kastner, „Certificate Management in OPC UA Applications: An Evaluation of different Trust Models,“ in Proceedings of the 2012 IEEE 17th Conference on Emerging Technologies & Factory Automation (ETFA), 2012.

[5] OPC Foundation, „OPC Unified Architecture Specification, Part 4: Services, Release Candidate 1.04.11,“ Scottsdale, USA, 2017.

[6] Fiat, Störtkuhl, Plöb, Zugfil, Gappmeier and Damm, "OPC UA Security Analysis," Federal Office for Information Security, Bonn, Germany, 2017.

[7] OPC Foundation, „OPC Unified Architecture Specification, Part 7: Profiles, Release 1.03,“ Scottsdale, USA, 2015.

[8] OPC Foundation, „OPC Unified Architecture Specification, Part 12: Discovery, Release 1.03,“ Scottsdale, USA, 2015.

[9] [Online]. Available: http://www.vdmashop.de/refs/Leitf_I40_Security_En_LR_neu.pdf

[10] [Online]. Available: http://www.27000.org/ismsprocess.htm

[11] [Online]. Available: http://isa99.isa.org/ISA99%20Wiki/Home.aspx

[12] [Online]. Available: https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx

[13] [Online]. Available: https://www.first.org/cvss/

[14] [Online]. Available: https://www.commoncriteriaportal.org/

[15] Mano Paul, Official (ISC)^2 GUIDE TO THE CSSLP CBK, Boca Raton,
USA: CRC Press, 2014.

[16] [Online]. Available: https://opcfoundation.org/markets-collaboration/m2m-alliance/

[17] [Online]. Available: https://www.owasp.org/index.php/Main_Page

[18] OPC Foundation, „OPC Unified Architecture Specification, Part 14: PubSub, Release Candidate 1.04.24,“ Scottsdale, USA, 2017.

[19] [Online]. Available: https://www.youtube.com/watch?v=TCy8JlnWIXw

[20] [Online]. Available: https://trustedcomputinggroup.org/tpm-main-specification/

[21] Wylie Shanks, “Building and Managinga PKI Solution for Small and Medium Size Business”, 2013, Online Available: https://www.sans.org/reading-room/whitepapers/certificates/building-managing-pki-solution-small-medium-sizebusiness34445

Further information

This document gives only a condensed overview of security for OPC UA. Security is a must-have in connected systems. An overview of industrial security is given by the VDMA guideline [9]. One requires an overall security concept, which is based on accepted security standards.

An information security management as described by the ISO/IEC 2700x [10] series of security standards requires organizational policies, infrastructure policies, and development policies. Furthermore, personnel must be trained regularly and you must be prepared for security incidents. The IEC 62443 [11] series of security standards defines industrial communication networks requirements for the network and system security. One should also be aware of applicable threats and risks. STRIDE [15] defines a common security threat classification model. Furthermore, CVSS [13] defines a security threats evaluation model. Additionally, Common Criteria [14] defines a common methodology for information security evaluation. The book [15] gives one a good starting point for becoming a security expert. Being up to date and networking with security professional is also one of the key factors for getting the latest news. We advise to get into contact with community projects, like our OPC Foundation security user group [16] or the Open Web Application Security Project (OWASP) [17]. The OWASP publishes the top 10 security risks regularly and publishes security guidelines. If not having available the resources for building up security expertise by oneself, we advise to get in touch with external experts from security companies, OPC UA companies, universities, or research societies such as Fraunhofer.

Online Videos

LEARN MORE ABOUT ...

OPC Videos

https://opcfoundation.org/resources/multimedia

OPC UA Technical Introduction by Uwe Steinkrauss

https://youtu.be/nYMbQiRqK74

What is OPC? UA in a minute

https://www.youtube.com/watch?v=-tDGzwsBokY

OPC UA Security by Darek Kominek

https://www.youtube.com/watch?v=NFQfZeU90Kw

HEADQUARTERS / USA

OPC Foundation
16101 N. 82nd Street
Suite 3B
Scottsdale, AZ 85260-1868
Phone: (1) 480 483-6644
office@opcfoundation.org

OPC EUROPE

Huelshorstweg 30
33415 Verl
Germany
opceurope@opcfoundation.org

OPC JAPAN

c/o Microsoft Japan Co., Ltd
2-16-3 Konan Minato-ku, Tokyo
1080075 Japan
opcjapan@microsoft.com

OPC KOREA

c/o KETI
22, Daewangpangyo-ro 712,
Bundang-gu, Seongnam-si, Gyeonggi-do
13488 South Korea
opckorea@opcfoundation.org

OPC CHINA

B-8, Zizhuyuan Road 116,
Jiahao International Center, Haidian District,
Beijing, P.R.C
P.R.China
opcchina@opcfoundation.org

V3

www.opcfoundation.org