OPC UA Enables Secure Data Transfer and System Integrations in Private and Public Networks

Jouni Aro, Heikki Tahvanainen

Prosys PMS Oy

jouni.aro@prosysopc.com, heikki.tahvanainen@prosysopc.com

Automaatio XXI seminar 17.-18.3. 2015

Abstract

OPC UA (Unified Architecture) is a standard communication protocol (IEC 62541), designed to enable secure and reliable connections for interconnecting various systems in industrial automation and other communication areas where measurement data or event notifications are transferred. This enables soft real-time transfer of current and historical data and events, including alarm management. OPC UA is also selected as the backbone of the German government driven Industry 4.0 program, which targets to increase flexibility in production automation. In this paper, we will outline the security threats in an industrial environment, and then present how OPC UA targets these threats. ISA/IEC62443 security strategy is presented as reference. Details of the OPC UA Security model are presented to show the flexibility to current and future requirements. Performance test results will be presented, concerning OPC UA communication and the impact of the security. Finally a case study is presented, regarding Valio Oy, where OPC UA has been established widely for improving the overall security of the production sites. They are applying OPC UA for communication, although most applications are still supporting OPC Classic protocols only.

Introduction

OPC has become the de facto standard in factory automation for integrating different production automation related software in multivendor environments. Since its introduction in 1995 OPC has provided a reliable communication of measurement, alarms and history data. This communication is denoted nowadays as OPC Classic, and consists of several different protocol specifications, OPC Data Access (DA), Alarms & Event (AE), Historical Data Access (HDA) and a few others.

OPC Foundation, which is the organization developing OPC, released a new version of the protocol, called OPC Unified Architecture (UA) in 2009. The new version improves OPC in various important ways, making it one of the most comprehensive communication protocols defined for the information technology in general. OPC was accepted as an IEC standard (IEC 62541) in 2011.

In addition to the secure communications channel, OPC UA also includes Information Modeling capabilities, which makes it a suitable platform for exchanging semantic information between various systems. Standard OPC UA information models have already been defined, for example, for IEC 61131-1 and ISA95 standards. This makes OPC UA extremely useful in integrating applications at higher-level than pure measurement data value or event notification basis. The Information Modeling capabilities are left out of the scope of this article.

Communication protocol

OPC UA provides several unique capabilities as a communication protocol. It is based on TCP/IP, but the transport channel can be changed flexibly between UA binary, SOAP and HTTPS protocols. It is also open for new protocol alternatives in future. Each protocol also enables full security features, including application and user authentication and data encryption. This removes the need for additional VPN networks and improves the overall security also in scattered and public networks.

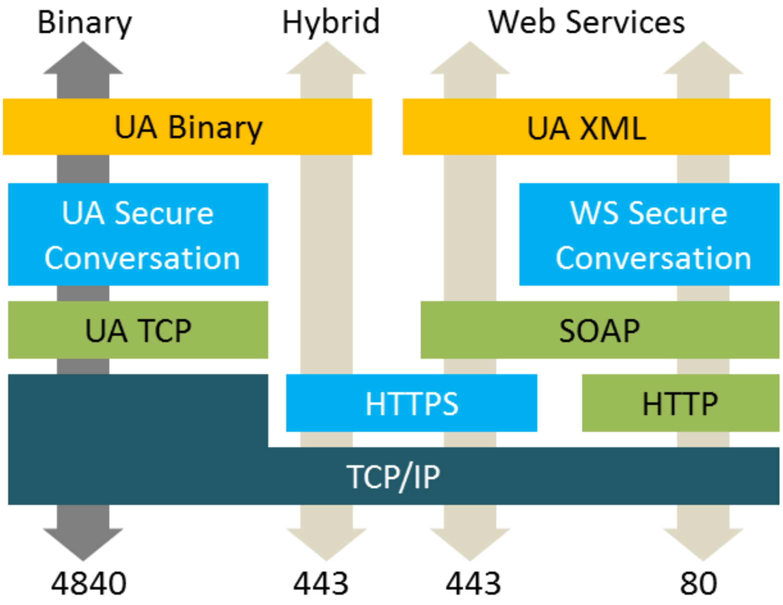

Figure 1. OPC UA transport protocol options. Security is implemented on the transport or application layer, depending on the selected protocol. Approximately same level of security is always available. The numbers denote standard IP port numbers assigned to the respective protocols. OPC UA applications can use different port numbers in practice. Only one port is used for the communication per application and selected protocol, making it ideal for firewall configurations.

Since the transport protocols are based on standard protocols, OPC UA communication can be implemented in different platforms. This has enabled already very small scale embedded devices that can act as independent OPC UA Servers in the network. Fraunhofer IOSB has demonstrated an implementation of OPC UA with a 15kb memory footprint. This makes it possible to place OPC UA "on chip" with almost any kind of industrial or consumer device.

On the other hand, the built-in security capabilities, which are also taking use of standard public key encryption mechanisms, enable secure usage in large networks.

Since SOAP is only implemented in a limited number of OPC UA applications, WS Security is left out of consideration in this article.

Security Architecture

OPC UA applications can run in different networks, which may be targeted by different vulnerabilities. ISA/IEC 62443 (formerly ISA99) defines strategies for managing the threats by dividing the operational network to segments, called *zones* in ISA62443, which are connected to each other through certain connection points, *conduits* (ISA62443). The conduits can be equipped with Intrusion Detection Systems (IDS) or Intrusion Prevention Systems (IPS or IDPS) to detect or prevent different malware entering the vulnerable parts of the plant network, respectively (Scarfone 2010). Since the network is eventually connected to the Internet, this is an important concept and it is also broadly adopted in the industrial plants. There are, however, still several small scale substations, which are open to the Internet without any security measures (Shodan, Tiilikainen 2013).

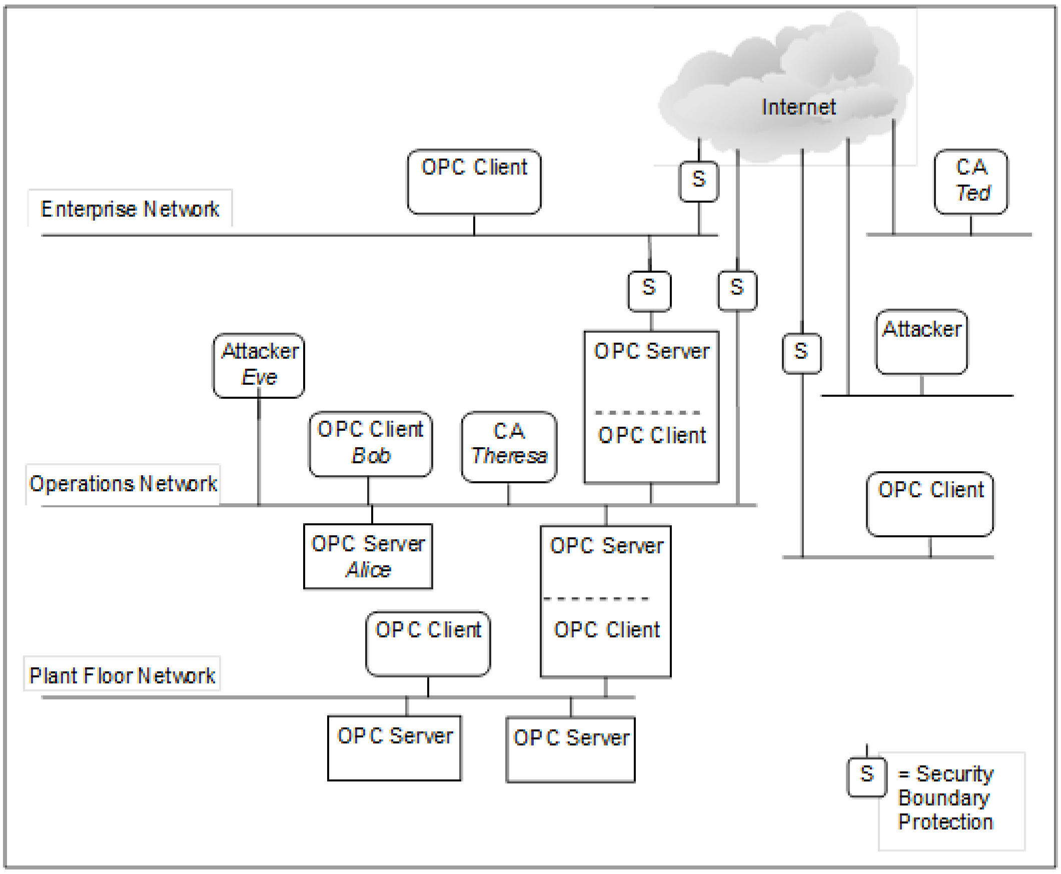

Figure 2. Various network segments and the security boundaries (called conduits in IEC 62443) between them. (OPC UA Part 2)

OPC UA helps to improve security measures in the conduits, in practice by enabling communication to be limited as much as possible to the secure OPC UA protocol. In comparison to standard VPN solutions, OPC UA enables more fine-grained security in practice. By opening a VPN tunnel between two network segments, those segments are in practice connected to each other without any guarded conduits in between. IDPS systems also cannot detect any encrypted traffic going through the VPN tunnel. IDPS must in these cases be installed on the unencrypted part of the tunnel entrance, or to the host computers. The IDPS systems may also have trouble analyzing all traffic, when the network load is high. There may also be various attacks targeted at the IDPS systems (Scarfone 2010). Regarding all these issues, it is advisable to avoid VPN tunnels as a generic security solution.

Security Objectives

There are various different objectives that are generally considered in regard to system security. OPC UA is targeted at delivering data securely from one application to another. In this context the following objectives are to be considered:

- Authentication: the ability to recognize applications and users that are connecting to each other and performing operations
- Authorization: the ability to control who may read and write data and execute various tasks in an operational system
- Confidentiality: the ability to protect data from unauthorized applications and users
- Integrity: the ability to ensure that data is not changed during the transfer
- Auditability: the ability to validate that selected security measures are applied and also track changes to the configuration of the operational system
- Availability: the ability to ensure that the operational system is usable and accessible in all situations

Security Threats

There are various threats that target the operational systems in general. In the OPC UA application context, the following threats are to be considered:

- Message Flooding: An attacker may send a large volume of requests
- Eavesdropping: Sensitive information is used by unauthorized parties
- Message Spoofing: Use forged messages to get unauthorized access
- Message Alteration: Capture and modify data in transfer
- Message Replay: Capture and resend data later unmodified
- Malformed Messages: Use invalid messages to confuse applications and gain unauthorized access
- Server Profiling: Deduce information about the target system to discover vulnerabilities by sending certain messages and validating the result
- Session Hijacking: Sniff network traffic and inject manipulated messages to use an established session
- Rogue Server: Pretend to be a valid server and make the client disclose information about itself
- Compromising User Credentials: Obtain user credentials written in paper, screen or from communication data to gain unauthorized access to the system

Security Architecture

The OPC UA security is based on three layers: The *Transport Layer*, *Communication Layer* and *Application Layer*.

The transport is always done over TCP. If HTTPS protocol is used, then TLS security is used to encrypt the traffic already in the Transport Layer. The Communication Layer consists of a Secure

Channel which can optionally perform message signing and encryption. When a secure communication policy is selected, this layer ensures the confidentiality and integrity of the messages that are sent. It also enables authentication of the applications, which may communicate with each other. The Application Layer consists of the session, which is used to authenticate and authorize users. All operations are performed on a session, therefore ensuring that unauthenticated users may not access or modify data in the target system. The session is always bound to a Secure Channel, which is also renewed frequently.

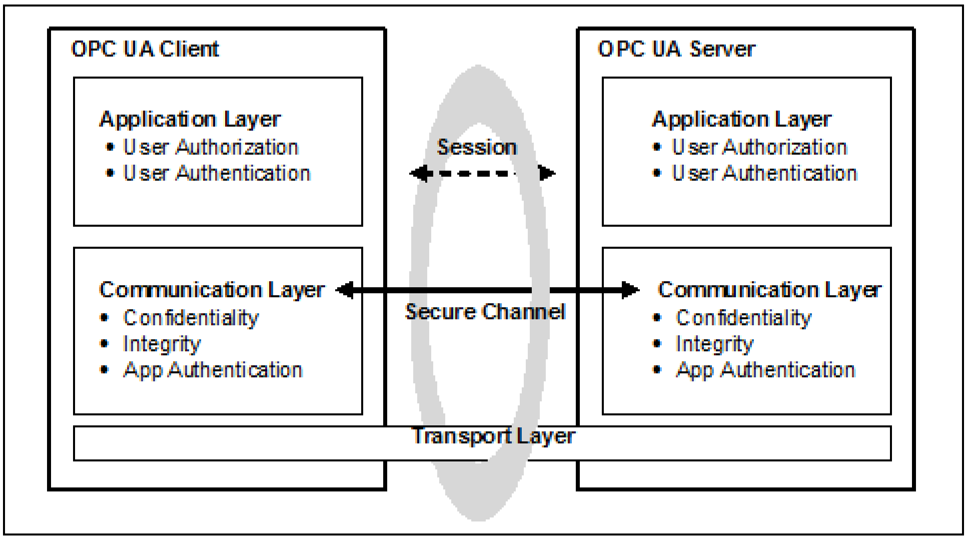

Figure 3. The layers of OPC UA communication. (OPC UA Part 2)

Connection Security

The establishment of OPC UA connection ensures the authentication and authorization of applications using standard security techniques. Each application instance has an *Application Instance Certificate*, which is a standard X.509v3 certificate with some extra fields for additional OPC UA validation. The respective RSA public and private keys are used to perform a secure hand-shake, when applications create the Secure Channel between them. Both applications will perform the authentication of the other party in the hand-shake, which in practice is an *OpenSecureChannel* service message. During the hand-shake the applications also exchange a symmetric encryption key, which is then used to secure all forth coming messages through the Secure Channel. The symmetric encryption is done with AES-128 or AES-256.

Once the Secure Channel is in place, the client application will create a Session in the server, over which all other service messages are sent and validated.

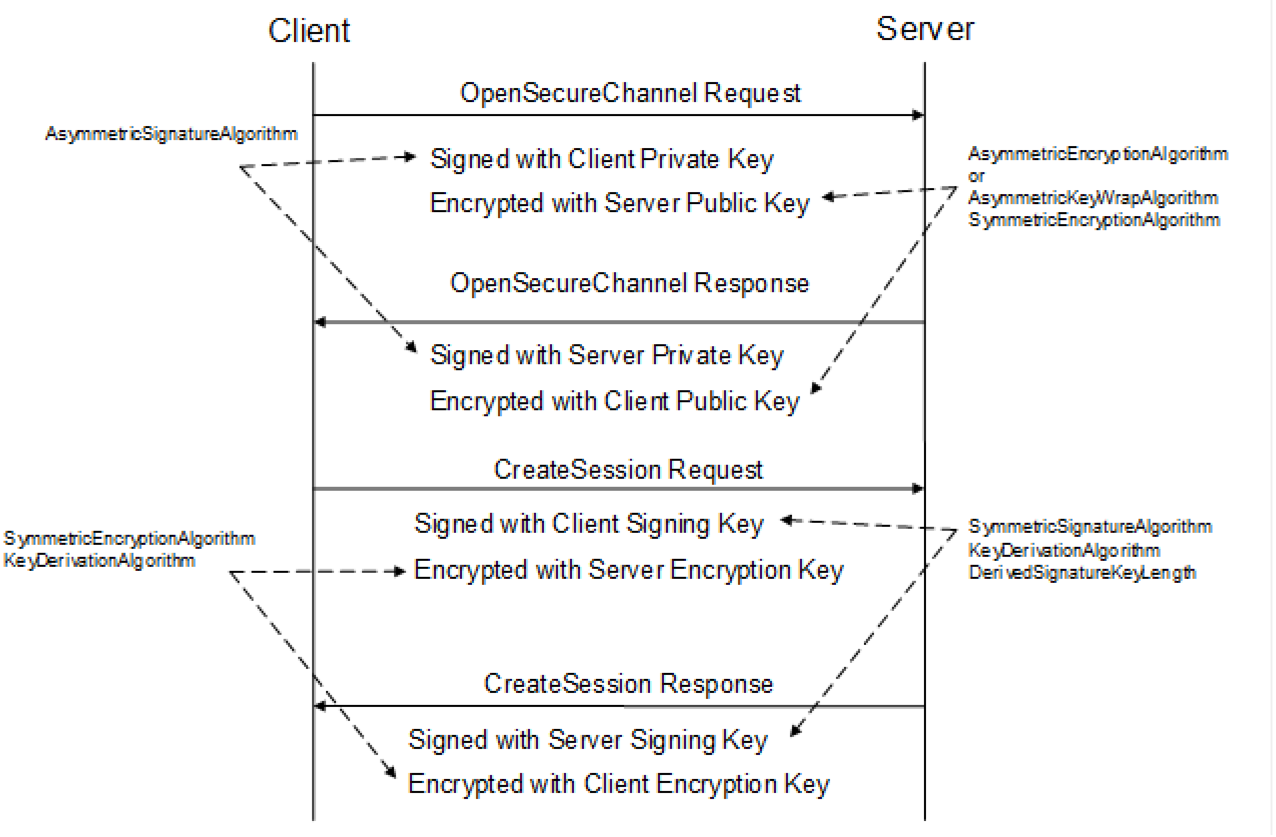
Client
Server
OpenSecureChannel Request
AsymmetricSignatureAlgorithm
Signed with Client Private Key
Encrypted with Server Public Key
AsymmetricEncryptionAlgorithm
or
AsymmetricKeyWrapAlgorithm
SymmetricEncryptionAlgorithm
OpenSecureChannel Response
Signed with Server Private Key
Encrypted with Client Public Key
CreateSession Request
Signed with Client Signing Key
SymmetricSignatureAlgorithm
KeyDerivationAlgorithm
DerivedSignatureKeyLength
SymmetricEncryptionAlgorithm
KeyDerivationAlgorithm
Encrypted with Server Encryption Key
CreateSession Response
Signed with Server Signing Key
Encrypted with Client Encryption Key

Figure 4. OPC UA hand-shake. (OPC UA Part 6)

Security Configuration

OPC UA enables a flexible selection of the used security mode between each connection. OPC UA specification defines three Message Security Modes: *None*, *Sign* and *SignAndEncrypt*. These define the level of security applied to each message. It also defines several alternative Security Policies: *Basic128Rsa15*, *Basic256* and *Basic256Sha256*. New policies can be defined and old ones made obsolete as security requirements increase in future. These define the enabled key sizes of the Application Instance Certificates, the signature algorithm and the symmetric encryption algorithm that are used.

All OPC UA servers that implement the *Standard Server Profile* must provide signing and encryption capabilities with at least the Basic128Rsa15 policy level (OPC UA Part 7). Applications implementing only the *Micro* or *Nano Embedded Server Profile* don't need to provide security capabilities at all. The server administrator may configure which policies are available. The client application always makes the selection, which security mode is used for each connection. This makes sure that security is always available and can be easily switched on as necessary.

User Authentication

User authentication is performed on the session level. OPC UA defines alternative authentication methods: *Anonymous*, *User name and Password* combination, *X.509 certificates* and also ways to use external user authentication systems, such as Kerberos, via *External Tokens*. The server applications must again support different alternatives depending on the Server Profile that they implement. And the client application again selects the authentication type to use for the connection from the alternatives implemented and configured for the server.

Certificate Management

Since the security of OPC UA is based on X.509 certificates, the main concern in practice will be the certificate management. OPC UA specification does not enforce any specific strategy for managing the certificates.

All OPC UA applications maintain certificates in their own *Trust Store*. All encountered Application Instance Certificates from the connecting applications are classified either as *trusted* or *rejected*. This selection can be done on a certificate basis, which is usable, when the number of connections is small. The applications typically use self-signed certificates by default, which can only be trusted individually. In a standard security environment, trust is based on Certificate Authorities (CA), which sign certificates and also maintain Certificate Revocation Lists (CRL). CA helps to maintain the trust chain between applications: all certificates signed by a trusted CA can be trusted automatically, until they are revoked.

OPC Foundation has defined a so called Global Discovery Service (GDS) to help implement local CA services in practice. It is expected that commercial implementations of the GDS will become in the market. Standard CA systems may be used to manage OPC UA certificates as well.

The Application Instance Certificates are only used for the UA Binary communication. When HTTPS is used, similar, but slightly different X.509v3 certificates are used as normal part of the SSL Layer applied over HTTP.

Performance measurements

It is commonly seen that encryption may not be used in resource constrained devices because it is too resource intensive. How much overhead exactly do these security measures introduce?

Modern PC hardware usually contains hardware-accelerated encryption. However, not all devices support such features. What kind of performance could be expected from different kinds of hardware platform? We compared the following computers:

1) Dell laptop, OS Windows 8.1 64-bit, Intel Core i7 @2.70GHz, 8 GB RAM memory and SSD drive.

2) Raspberry Pi, OS Raspbian Linux, 700 MHz single-core ARM, 512 MB RAM memory and SD card as a storage.

The Dell laptop contains hardware-accelerated encryption whereas the Raspberry Pi does not have this feature. The average results of profiling the time used to just encrypt a single OPC UA chunk at a given platform are shown in Figure 5.

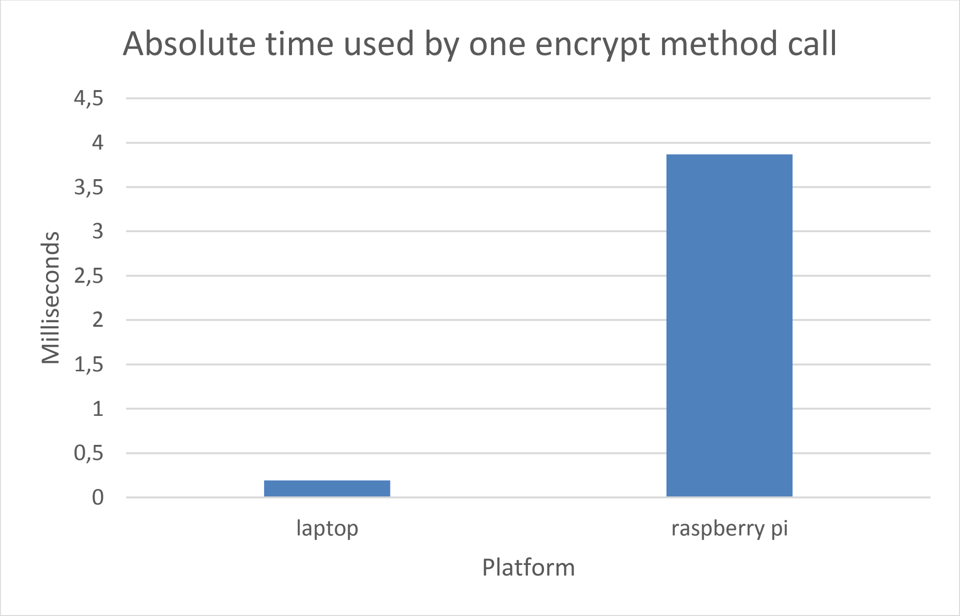

Figure 5. Time spent encrypting one chunk.

The most interesting thing in practice is not the absolute time but the relative difference between the devices. We see that on the Raspberry Pi platform the encryption is performed approximately 20 times slower than on normal modern PC hardware.

Figure 6. shows the time required to make a complete read service call, including the request and response messages. The read consists of a single variable of byte array with 10 elements.

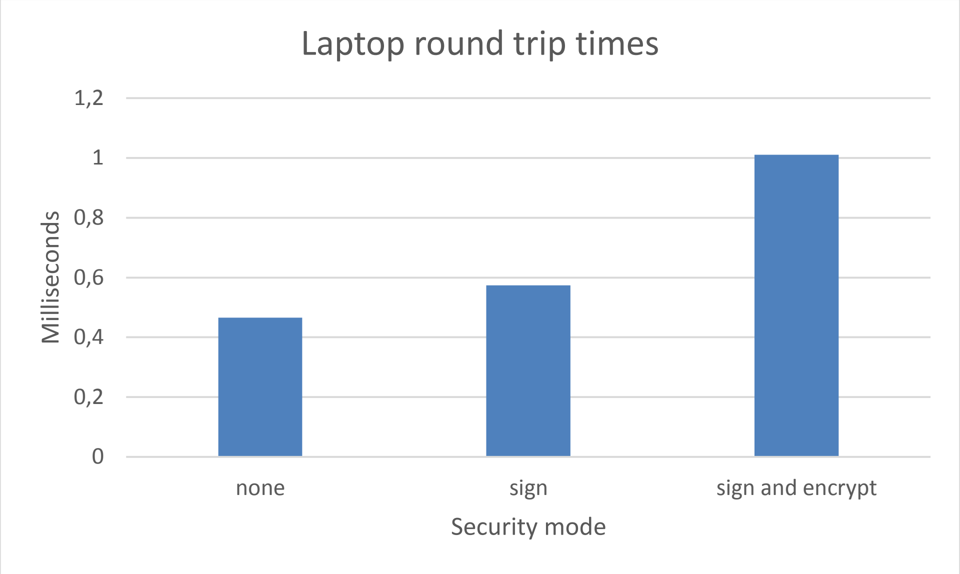

Figure 6. Round trip times measured on office laptop.

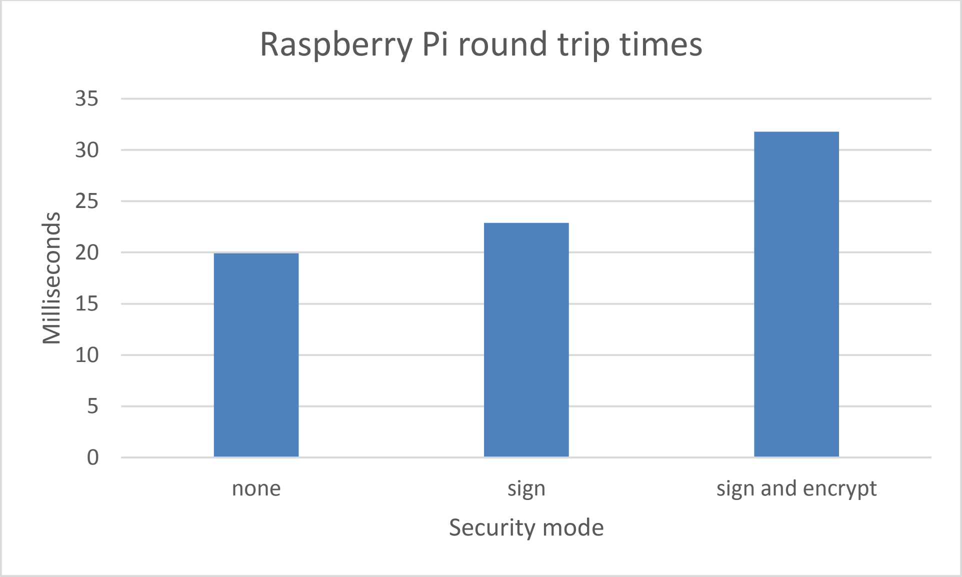

Figure 7. Round trip times measured on a Raspberry Pi computer.

The measurements were made with both server and client on same machine using the sample applications included in Prosys OPC UA Java SDK. The communication protocol was UA Binary and security policy was Basic128RSA15. The times were recorded by profiling synchronic read service calls and averaging it for 1000 calls. As a conclusion we can see that encryption adds overhead to the communication as expected. Round trip times with encryption takes 1.5 – 2 times compared to the times without security. However, in most application areas this could still be deemed negligible because the absolute time is in the range of 0.5 ms to 10 ms, only.

Figures 8 and 9 illustrate same measurements but now with a byte array variable containing 10 000 elements. It can be seen that when the amount of transferred data grows, also the overhead of encryption seems to grow.

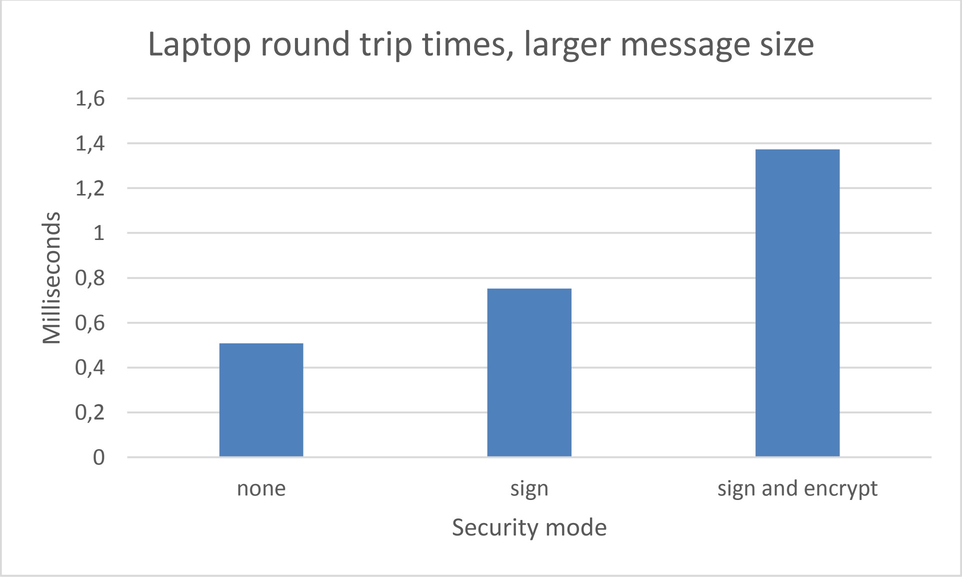

Figure 8. Round trip times measured on office laptop with a larger byte array.

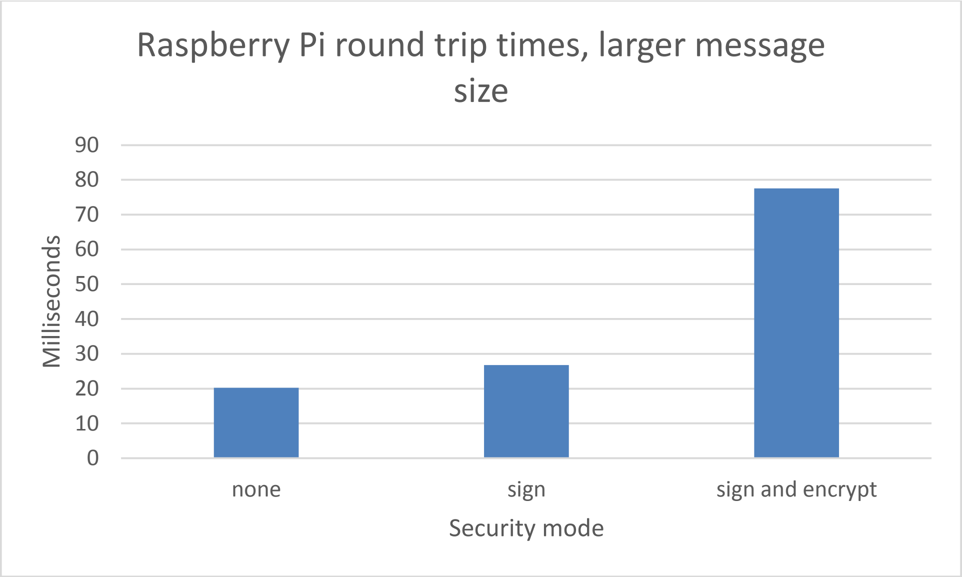

Figure 9. Round trip times measured on a Raspberry Pi computer with a larger byte array.

Case Valio

Valio has taken OPC UA in wide use as their internal system integration channel, and have been able to increase the overall security level of their operational networks. The secure connections enable better distribution of production information in the whole production system and also between production sites.

Valio is still using mostly OPC Classic applications, including OPC DA, AE and HAD, in all its production systems. However, OPC UA has been taken in use as a generic communication channel between the applications. This is accomplished with the help of OPC UA Gateway applications, which can convert between OPC UA and the OPC Classic protocols. Security has also been applied to the connections wherever necessary. All OPC UA applications have got their Application Instance Certificates signed by the company Certificate Authority.

They are also effectively using different network segments to limit access to the production site. All information between the office network and production network is directed via an intermediate, DMZ network (demilitarized zone). UA Gateway can also act as a single connection point in between the office and production network, installed in the DMZ network.

Discussion

OPC UA provides a modern secure communication protocol, enabling management of most security threats targeting modern production facilities. It can only provide security for the communication channel, so traditional security measures must still be used to ensure the security of the host computers and access to them in the sites. Certificate management will play a vital role in ensuring the overall security of the system. Malware can enter the production network by various means, so security measures on hosts should not be decreased even when better communication security is established.

The performance measurements show that encryption adds some overhead to the communication, but in most cases not very dramatically. Instead of performance, a bigger problem may be the management of certificates, which requires a good infrastructure to ensure security is not compromised in practice.

In future, as more and more devices will be available in open networks, security will be an important factor to ensure privacy and confidentiality of data. Especially in production environments, security will be a main concern, when ensuring the intermittent operation of production plants.

References

(ISA62443 2012) ISA-TR62443-3-1 (99.03.01) Security for industrial automation and control systems. Draft 1, Edit 1 December 2012. http://isa99.isa.org/Documents/Drafts/ISA-TR62443-3-1-WD-LTR.pdf

(OPC UA Part 2) OPC Unified Architecture Specification. Part 2: Security Model. Release 1.02. April 17, 2013

(OPC UA Part 6) OPC Unified Architecture Specification. Part 6: Mappings. Release 1.02. August 8, 2012

(OPC UA Part 7) OPC Unified Architecture Specification. Part 6: Profiles. Release 1.02. April 17, 2013

(Scarfone 2010) Scarfone, Karen; Mell, Peter (February 2007). "Guide to Intrusion Detection and Prevention Systems (IDPS)". Computer Security Resource Center (National Institute of Standards and Technology) (800–94). Retrieved 1 January 2010.

(Shodan) SHODAN search engine. http://www.shodanhq.com/

(Tiilikainen 2013) Tiilikainen, Seppo; Manner, Jukka (21.3.2013). Suomen automaatioverkkojen haavoittuvuus, Aalto-yliopisto. Raportti. https://research.comnet.aalto.fi/public/Aalto-Shodan-Raportti-julkinen.pdf