CIP Security Phase 1
Secure Transport for EtherNet/IP

Brian Batke, Principal Engineer
Rockwell Automation (babatke@ra.rockwell.com, +1-440-646-5277)
Joakim Wiberg, Team Manager Technology and Platforms
HMS Industrial Networks (jow@hms.se, +46-35-172936)
Dennis Dubé, Cyber Security Architect
Schneider Electric (dennis.dube@schneider-electric.com, +1-978-975-2891)

Presented at the ODVA
2015 Industry Conference & 17th Annual Meeting
October 13-15, 2015
Frisco, Texas, USA

Abstract

Industrial network protocols, EtherNet/IP included, continue to come under scrutiny for their lack of security measures. In its present form, EtherNet/IP provides no means for a device to know that the sender or receiver of a message is a trusted entity, or that the message has not been tampered with or maliciously inserted into a connection.

The ODVA Security Working Group, operating within the EtherNet/IP System Architecture SIG, has defined mechanisms that provide a secure communications transport for EtherNet/IP. Using standard security protocols – TLS and DTLS – EtherNet/IP devices and software applications now have the ability to perform secure communications between trusted entities, denying communications from untrusted entities.

In this paper we present an overview of the secure EtherNet/IP transport, which will be published in the CIP Networks Specification.

Keywords

CIP Security, Cyber Security, Certificates, Authentication, Integrity

The Need for CIP Security

Industrial automation protocols were initially defined during a period in which user application networks were largely disconnected from the wider enterprise, and when the perceived level of security risk was much lower than at present. As a result protocols such as CIP and EtherNet/IP were not designed to include inherent security mechanisms.

In recent years security researchers have highlighted and publicly reported a number of vulnerabilities in CIP and EtherNet/IP. In actuality these vulnerabilities are “by design”: standard capabilities of CIP, such as the Reset service of the Identity Object. The root cause of the vulnerability is the lack of security mechanisms in CIP, which allows any client – whether trusted or untrusted – to send a disruptive service such as Set-Attributes-Single to change the device’s IP address.

Malicious employees or others who are able to gain access to the control network can take advantage of the lack of protocol security to tamper with systems, disrupt operations, or potentially damage equipment or people.

Control system security has typically been addressed by adoption a defense-in-depth security architecture has been recommended for many years (see figure below). This architecture is based on the idea that multiple layers of security would be more resilient to attack. The expectation is that any one layer could be compromised at some point in time while the automation devices at the innermost layer would remain secure.

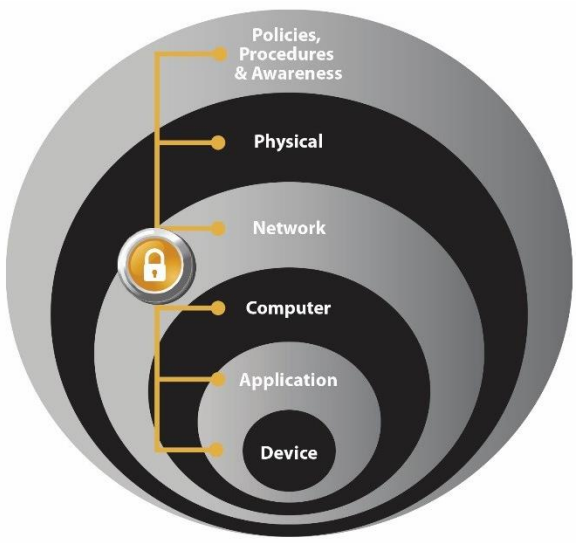

Figure 1. Defense in Depth Layers

As attackers become more sophisticated, it becomes more important for the CIP-connected device – the final layer of defense – to defend itself. Consider the situation where a piece of malware is, unknown to control system personnel, delivered to a compromised PC via USB drive. The malware could contain code to issue malicious CIP services to devices (as noted above). However if the device were able to reject such services from untrusted sources, the threat would be mitigated.

The goal of CIP Security is to enable the CIP-connected device to protect itself from malicious CIP communications. A fully self-defending CIP device would be able to:

- Reject data that has been altered (integrity)
- Reject messages send by untrusted people or untrusted devices (authenticity)
- Reject messages that request actions that are not allowed (authorization)

CIP Security Approach

CIP Security is defined in Volume 8 of the CIP Networks Specification, and includes the definition of security-related requirements and capabilities for CIP devices. Volume 8 at present is focused on EtherNet/IP, as EtherNet/IP-connected devices represent the largest risk due to enterprise network connectivity. CIP Security will eventually include material that is network-independent, and potentially include security mechanism for CIP networks other than EtherNet/IP.

Recognizing that every CIP device does not need to provide the same level of support for all defined security features, CIP Security defines the notion of a Security Profile. A Security Profile is a set of well-defined capabilities to facilitate device interoperability and end-user selection of devices with the appropriate security capability.

At present, two profiles are defined for EtherNet/IP devices, and two potential future profiles are identified for CIP-level security capability.

Security Profile	General Description
EtherNet/IP Integrity Profile	Provides secure communications between EtherNet/IP endpoints to assure data integrity and device authenticity
EtherNet/IP Confidentially Profile	Provides secure communications between EtherNet/IP endpoints and ensures data confidentiality for transport class 0/1 traffic. Includes the EtherNet/IP Integrity profile as a subset
CIP Authorization Profile (future)	Provides secure communications between CIP endpoints to ensure device and user authenticity
CIP Integrity Profile (future)	Provides secure communications between CIP endpoints to ensure data integrity

Table 1. CIP Security Profiles

Table 2 shows the security properties provided by each of the profiles:

Security Property	EtherNet/IP Integrity Profile	EtherNet/IP Confidentially Profile	CIP Authorization Profile (future)	CIP Integrity Profile (future)
Device Authentication	√	√	√	√
Trust Domain	Broad – group of devices	Broad – group of devices	Narrow - individual device/user	Narrow - individual device/user
Device Identity	√	√	√	√
Data Integrity	√	√		√
Data Confidentiality		√		
User Authentication			√	
Change Detection (Audit)			√	
Policy Enforcement (Authorization)			√	

Table 2. Supported Security Properties

CIP Security Phase 1 Overview

CIP Security Phase 1, published as Volume 8 in the 2015-2 publication cycle of the CIP Networks Specification, provides a secure transport mechanism for EtherNet/IP devices.

CIP Security for EtherNet/IP devices makes use of the IETF-standard TLS (RFC 5246) and DTLS (RFC 6347) protocols in order to provide a secure transport for EtherNet/IP traffic. TLS is used for the TCP-based communications (including encapsulation layer, UCMM, transport class 3), and DTLS for the UDP-based transport class 0/1 communications. This approach is analogous to the way that HTTP uses TLS for HTTPS.

The secure EtherNet/IP transport provides the following security attributes:

- Authentication of the endpoints – ensuring that the target and originator are both trusted entities. End point authentication is accomplished using X.509 certificates or pre-shared keys.
- Message integrity and authentication – ensuring that the message was sent by the trusted endpoint and was not modified in transit. Message integrity and authentication is accomplished via TLS message authentication code (HMAC).
- Message encryption – optional capability to encrypt the communications, provided by the encryption algorithm that is negotiated via the TLS handshake.

The following diagram shows the protocol layering:

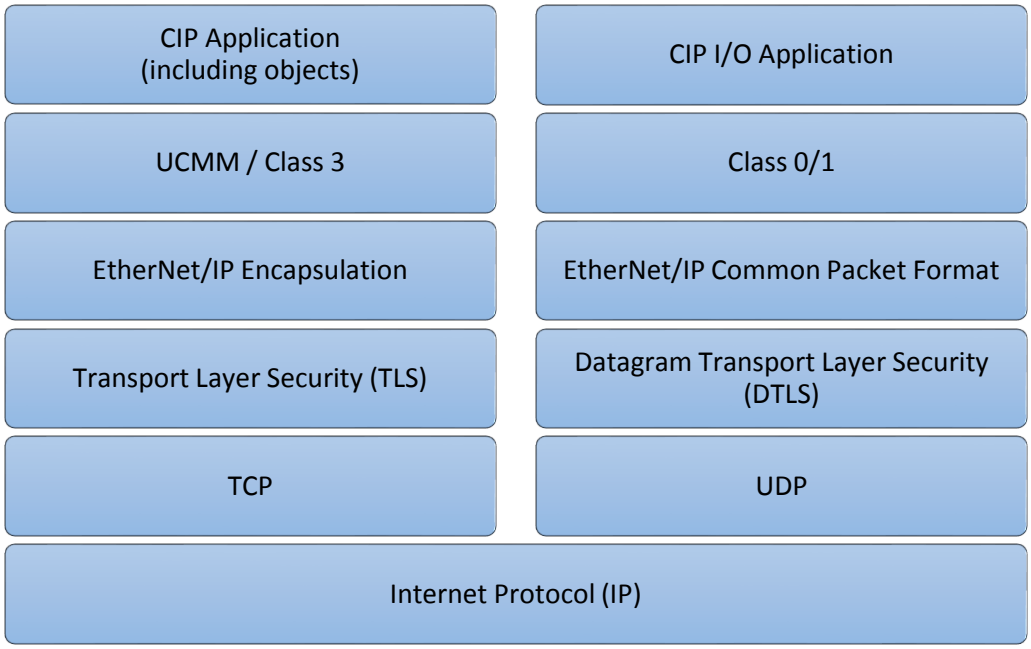

Figure 2. EtherNet/IP over TLS and DTLS Layering.

The following example illustrates how the secure EtherNet/IP transport would mitigate a security threat.

Consider a simple end-user application that consists of an EtherNet/IP-connected programmable controller (PLC) and several EtherNet/IP-connected I/O devices. At initial configuration time, the user configures the PLC and each I/O module with a pre-shared key (PSK) and disables the non-

secure EtherNet/IP TCP and UDP ports. Subsequent EtherNet/IP communications take place over TLS and DTLS, and require that each endpoint possess the PSK that has been configured.

Assume further that an employee has unknowingly downloaded malware that sends programming commands to the PLC's IP address via EtherNet/IP. If the malware attempts to connect to the PLC without using TLS, the PLC will not accept the connection. If the malware attempts to connect via TLS, but doesn't know the PSK, the TLS connection will not be established. In either case, the malicious programming commands will not be sent to the PLC.

EtherNet/IP over (D)TLS

What are TLS and DTLS?

TLS, or Transport Layer Security, is a standard, general-purpose, security protocol that is commonly used to provide secure application communications. TLS is commonly-known for being the secure transport mechanism used by HTTPS. TLS version 1.2 is defined in RFC 5246.

TLS operates above the TCP layer, providing security for application-level communications. An application that uses TLS first opens a TCP connection with its communications partner, establishes a TLS session, then sends and receives the secure application traffic (EtherNet/IP in this case) over that TLS session.

DTLS, or Datagram Transport Layer Security, takes the same design approach as TLS and applies it to a datagram transport (e.g., UDP). DTLS version 1.2 is defined in RFC 6347.

TLS and DTLS use the concept of a "cipher suite", which determines the algorithms that are used for initial key exchange, encryption of application messages, and authentication of application messages. The set of cipher suites that are allowed for use with TLS and DTLS are defined via RFCs and are listed at the Internet Assigned Numbers Authority web site (www.iana.org).

An example cipher suite is:

TLS_RSA_WITH_AES_128_CBC_SHA

This cipher suite specifies that RSA is used for key exchange, AES (128-bit, CBC mode) for message encryption, and SHA-1 for HMAC on application messages.

Once a (D)TLS session is established using the (D)TLS Handshake protocol, application messages are exchanged using the (D)TLS record format, and are secured per the negotiated cipher suite.

New TCP and UDP Port Numbers

EtherNet/IP over (D)TLS uses new TCP and UDP port numbers, in addition to those already defined for EtherNet/IP:

- 2221/tcp is used for transport class 3 and unconnected messages over TLS.
- 2221/udp is used for transport class 0/1 over DTLS (including the class 0/1 Forward_Open and Forward_Close request/reply)

The above port numbers, like the existing EtherNet/IP port numbers, are registered with the Internet Assigned Numbers Authority (IANA).

The use of distinct port numbers for the TLS and DTLS traffic allows devices to easily distinguish secure traffic without requiring a method for the originator and target first negotiate whether the secure transport will be used.

Required Cipher Suites

The IANA (D)TLS cipher suite registry contains a large number of cipher suites that have been defined via RFCs. In order to ensure interoperability between EtherNet/IP devices, while not placing undue burden on implementations, Volume 8 specifies a limited set of cipher suites that devices are required support. The set of required cipher suites is based on the following underlying requirements:

- Allow the use of pre-shared keys or X.509 certificates for endpoint authentication
- Allow of use of either RSA or Elliptic Curve public/private key pairs
- Provide data encryption (in addition to data integrity), or data integrity only (null encryption)

The following tables show the required cipher suites specified in Volume 8. Several items should be noted:

- Originators are required to support both the RSA and EC suites, as the target device may either use an RSA or EC public/private key pair.
- Devices the EtherNet/IP Integrity Profile are not required to support the encryption suites for their transport class 0/1 data.

Cipher Suite	Description
TLS_RSA_WITH_NULL_SHA256	RSA for key exchange; null encryption; SHA256 for message integrity. Note that encryption is not provided.
TLS_RSA_WITH_AES_128_CBC_SHA256	RSA for key exchange. AES 128 for message encryption, SHA256 for message integrity.
TLS_RSA_WITH_AES_256_CBC_SHA256	RSA for key exchange. AES 256 for message encryption, SHA256 for message integrity.
TLS_ECDHE_ECDSA_WITH_NULL_SHA	ECDHE_ECDSA for key exchange; null encryption; SHA1 for message integrity. Note that encryption is not provided.
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	ECDHE_ECDSA for key exchange. AES 128 for message encryption, SHA256 for message integrity.
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	ECDHE_ECDSA for key exchange. AES 256 for message encryption, SHA256 for message integrity.

Table 3. (D)TLS Certificate Cipher Suites

Cipher Suite	Description
TLS_ECDHE_PSK_WITH_NULL_SHA256	ECDHE in conjunction with PSK for key exchange; null encryption; SHA256 for message integrity. Note that encryption is not provided.
TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256	ECDHE in conjunction with PSK for key exchange. AES 128 for

	message encryption, SHA256 for message integrity.

Table 4. (D)TLS Pre-Shared Key Cipher Suites

Originator and Target Behavior

The message format for EtherNet/IP over TLS is the same as is used for EtherNet/IP over port 44818. The Encapsulation Layer as described in Chapter 2 of the EtherNet/IP specification is used to encapsulate the application messages. CIP unconnected and transport class 3 behavior also remains unchanged with the exception that the messages are transported using the TLS application data protocol. Figure 3 shows a (simplified) message sequence for secure unconnected / class 3 messaging over TLS:

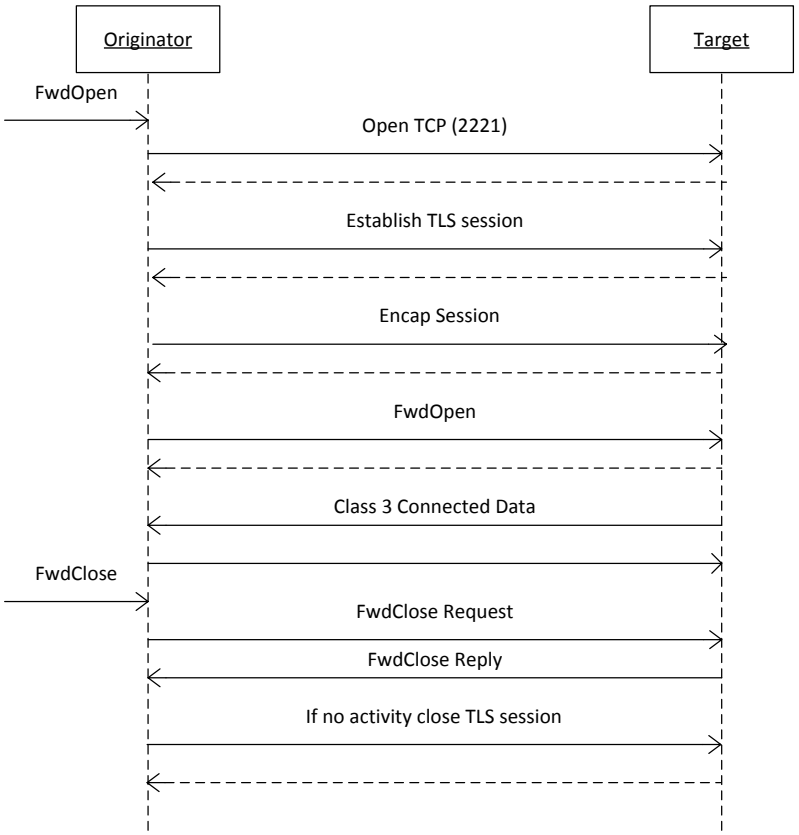

Figure 3. Message Sequence for EtherNet/IP over TLS

For secure CIP class 0/1 connections, both the Class 0/1 connected data as well as the Forward_Open, Large_Forward_Open, and Forward_Close services are sent over DTLS. Note that this is in contrast to class 0/1 connections that are not secured, where the Forward_Open (and related) services are sent over TCP, and the class 0/1 connected data sent over UDP. Using UDP-only with secure class 0/1 connections allows the use of a single security channel, and therefore a single security handshake, to establish the secure session.

In order to transport the Forward_Open / Forward_Close services over UDP, a new Common Packet Format item is defined in Volume 2, Chapter 2. The format of the class 0/1 connected data is the same as when sent via UDP without DTLS.

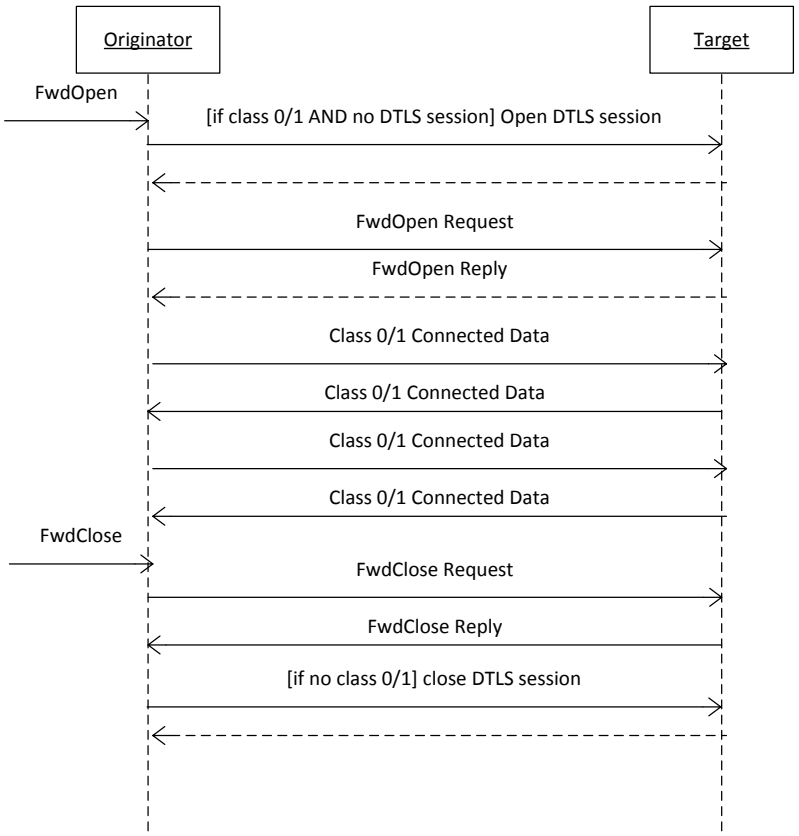

Figure 4. Class 0/1 over DTLS Message Sequence

CIP Objects

Volume 8 of the CIP Networks Specification introduces three new objects for access control, configuration, and certificate management. The three new objects are:

- CIP Security Object
- EtherNet/IP Security Object
- Certificate Management Object

CIP Security Object

The CIP Security Object provides an interface for controlling access to a device during initial security commissioning. The CIP Security Object ensures that a single client has exclusive access once a configuration session has been started.

The commissioning approach using the CIP Security Object is summarized as follows:

- The commissioning tool establishes an initial CIP connection to the device.
- For commissioning over EtherNet/IP, the commissioning tool should use a TLS connection in order to ensure security on that configuring connection. The device's default certificate is used for the initial commissioning connection.
- The commissioning tool sends the Begin_Config service to the CIP Security Object. If the device has other TCP connections for EtherNet/IP, or other CIP connections (any class) on other CIP ports, then the service is rejected.
- Once the Begin_Config service is accepted, the device does not allow any further TCP connections for EtherNet/IP or CIP communications (any class) from other CIP ports. If the device has other, non-CIP mechanisms for configuration (e.g., web server), those mechanisms should also be disallowed. This ensures exclusivity of the configuring connection.
- The commissioning tool then performs any necessary security-related configuration, e.g., to the EtherNet/IP Security Object and applies the new settings.
- Once all commissioning steps have been completed, the commissioning tool issues the End_Config service to the CIP Security process.
- The device may now accept new CIP communications (using newly configured settings).

The sequence described above ensures that one single client will have exclusive access to the whole device, thus all configuration can be performed in a secure and atomic manner.

EtherNet/IP Security Object

The EtherNet/IP Security Object provides an interface for the configuration of the various settings used by EtherNet/IP over TLS and DTLS, including:

- Whether to use X.509 certificates, PSK, or both
- PSK value and/or which certificates to use
- Trusted certificate authorities and certificate revocation list (if certificates are used)
- Allowed cipher suites
- Other behavior related to certificate verification such as client certificate verification, certificate chain handling, and certificate expiration check

The EtherNet/IP Security Object makes use of the Certificate Management Object and File Object as the containers for certificates to be used with TLS and DTLS, summarized as follows:

The EtherNet/IP Security Object makes use of the Certificate Management Object and File Object as the containers for certificates to be used with TLS and DTLS, as shown in Figure 5. By default the Active Device Certificates attribute points to the Certificate Management Object instance associated with the device's default certificate. Users may elect to use certificates generated from their own PKI, stored via dynamic Certificate Management Object instances.

Note that the actual certificate files are stored as File Object instances.

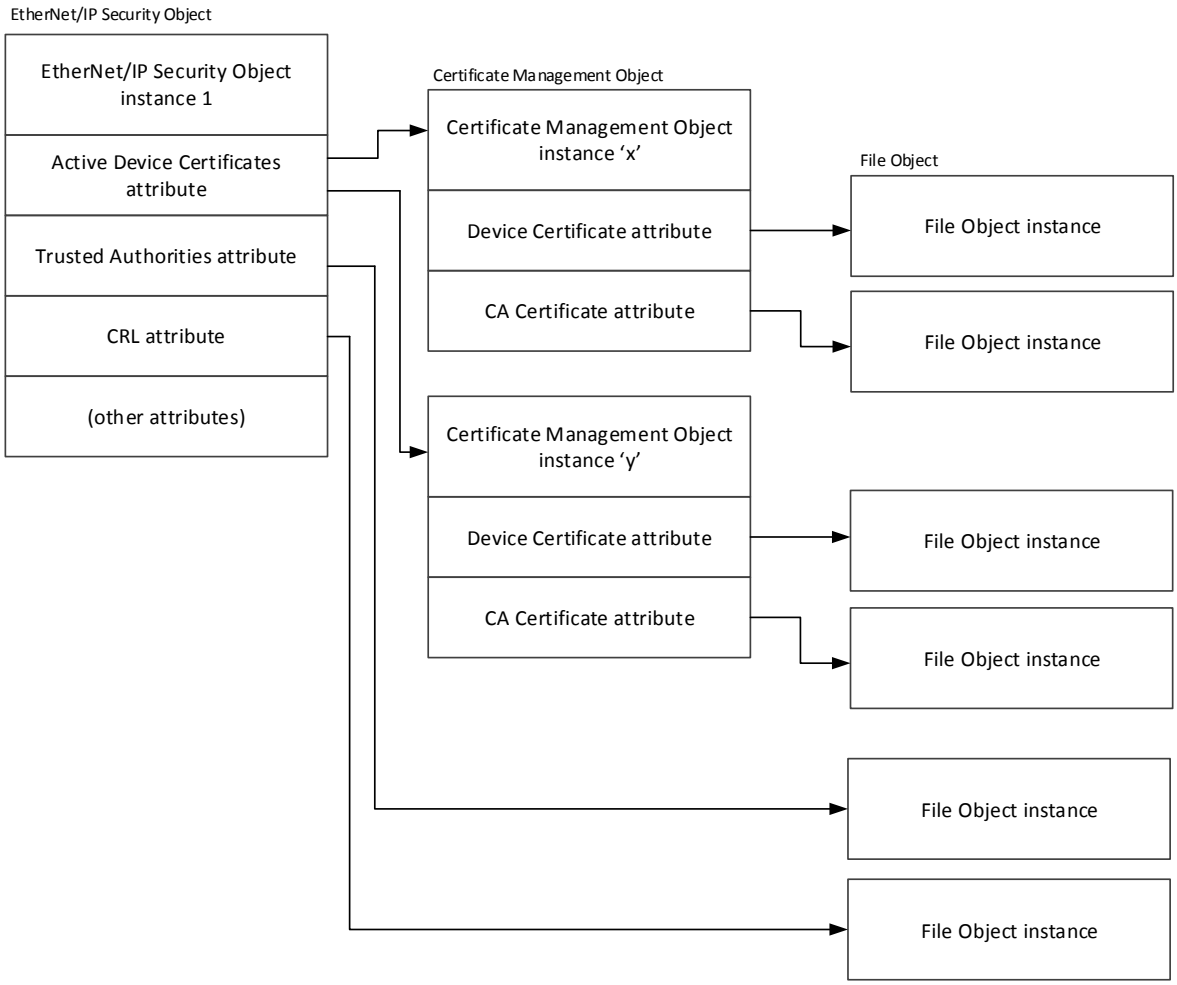

Figure 5. EtherNet/IP Security, Certificate Management, and File Object Relationship

Certificate Management Object

The Certificate Management Object provides an interface to manage a device's X.509 certificates for use in applications such as EtherNet/IP over (D)TLS. Its purpose is to organize, obtain, and expose X.509 certificates so that the certificates may be used by different applications.

Each Certificate Management Object instance represents one device certificate. The Certificate Management Object allows both static (always present in the device), and dynamic (created via the Create service) instances. Instance 1 (static) always contains the device's default certificate and associated CA certificate(s). Additional dynamic instances may be created by the user in order to provision the device with certificates generate by the user's PKI.

As noted above, the certificates contained by a Certificate Management Object instance are stored in the device as File Object instances to allow a defined CIP interface to read and write the certificate files.

The Certificate Management Object provides two models by which the device can obtain additional certificates:

- Using the *push* module, a configuring client users Certificate Management and File Object services to write a signed certificate to the device.

- Using the *pull* model, the device obtains a certificate from an enrollment server using a standard protocol such as EST or SCEP.

At present only the push model is defined, subsequent revisions of Volume 8 will add define the pull model.

Using the push model, a configuration client obtains a Certificate Signing Request (CSR) from the device, interacts with the end user's Certificate Authority (CA) to obtain a signed certificate on behalf of the device, then writes the device certificate to the device as a File Object instance.

The push model allows the use of different CA platforms and protocols without requiring the device to have explicit knowledge of those platforms and protocols.

Configuration and Commissioning

The security provisioning of a CIP device includes activities at four stages of the device lifecycle, including:

- Factory Default Configuration,
- Initial Configuration,
- Configuration Changes, and
- Decommissioning.

From a security perspective, the configuration and commissioning state of a CIP device is represented by the states of a set of security related CIP Objects as shown in Table 5.

CIP Object or Data Structure	Description	Where defined
CIP Security Object	Controls overall behavior of CIP Security	Vol 8 Chapter 5
EtherNet/IP Security Object	Provides interfaces for the configuration of EtherNet/IP over TLS and DTLS	Vol 8 Chapter 5
Certificate Management Object	Provides interfaces to manage the devices x.509 certificates	Vol 8 Chapter 5
File Object	One or more instances provides references to files containing x.509 Certificates and CRL	Vol 1 Chapter 5
TCP/IP Interface Object	Provides interfaces to specify whether individual device ports and protocols will be Open/Closed	Vol 2 Chapter 5
Public/Private Key Pair	Identifies the characteristics of and provides the values of the devices public/private key pair	internal and vendor defined
Certificate Signing Request (CSR)	The encoded CSR Request to be used by either the configuration tool (now) or the device (future) to obtain a CA-signed device certificate from an enrollment server.	internal vendor defined

Table 5. Security Related Object and Data Structures

Factory-Default Configuration

Factory-Default Configuration is the state of the device after power up initialization. The Factory-Default state is represented by:

- the state of the set of security related objects,
- its public/private key pair, and
- its default certificate(s).

Once the Factory-Default Configuration is available, the device is ready for Initial Configuration.

Initial Configuration
Initial Configuration is the provisioning of the device with the data, parameters, and credentials that it needs to perform its security functions. Device provisioning includes the selection of trust models and related cryptography as well as the credentials, e.g. certificates or PSK that will be used for end point authentication. When the Initial Configuration is complete, the device is CONFIGURED and is ready to execute its control function activities.

Configuration Changes
After a device has attained the CONFIGURED state, the end-user can effect changes to the security configuration data. The configuration changes to the attributes of the set of security related objects are achieved using secure communications. The application of the changes is done in a controlled manner so as not to cause unintended disruption to the ongoing device operation.

Decommissioning
Decommissioning is the activity of returning the device to its Factory-Default Configuration state. As a consequence of this activity, configuration data and credentials are deleted. This includes any files that provide credentials such as the device certificate, the CA certificate, etc. The credentials will need to be provisioned again, if the device is placed back into service after Decommissioning.

Secure Connection for Initial Configuration
Initial Configuration is achieved using secure communications which is provided by sending EtherNet/IP messages over TLS at TCP port 2221. In this TLS connection, the configuration tool is the client TLS endpoint while the device is the server TLS endpoint. See Volume 8 Chapter 3 for details.

At this stage in the lifecycle of the device, the only credential that the device has is its default certificate, either self-signed or vendor-signed. A TLS connection whose server authentication is based on the device's default certificate is technically achieved. However the level of trust using the default device certificate is less than a TLS connection based on a CA-signed certificate.

Note that what this approach provides is that the configuration session can't be compromised, e.g., by a Man-in-the-Middle attack. However, the device doesn't know if the configuring client is in fact a trusted entity. Therefore it would be up to the user to verify the ultimate configuration. The end user should be aware of this situation and should take additional precautions such as performing the configuration work on a closed network, etc.

As described in Volume 8, Chapter 5, the establishment of the TLS connection for the Initial Configuration work is allowed only if there are no other TCP connections established. And once the TLS connection for Initial Configuration is established, no other TCP connections are allowed until the Initial Configuration work is done and its TLS connection is closed.

An overview of the Initial Configuration TLS Session is shown in Figure 6.

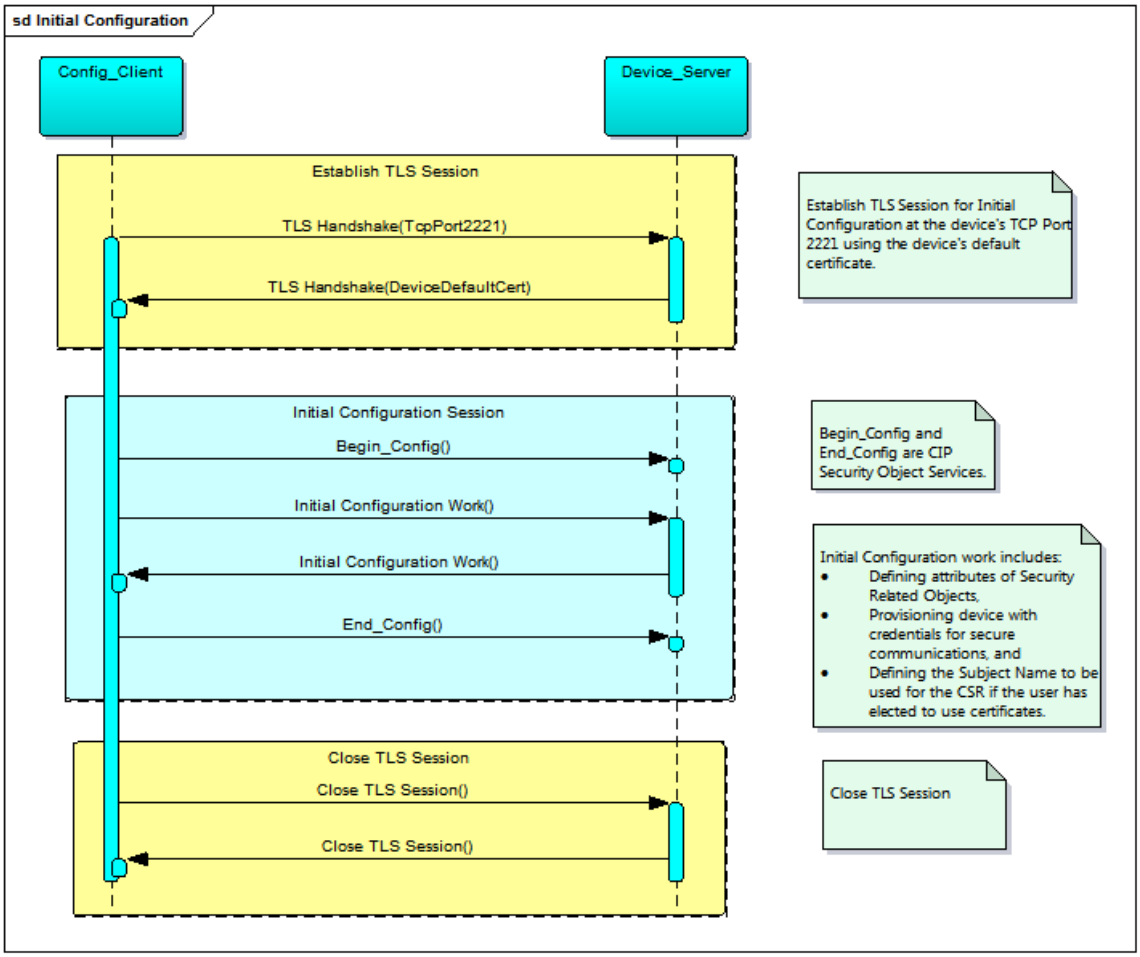

Figure 6. TLS Session for Initial Configuration

Provisioning of Credentials
A device can be configured to use Pre-Shared Keys (PSK) and/or x.509v3 Certificates for endpoint authentication credentials during the TLS Handshake exchanges.

Pre-Shared Keys
Phase 1 of Volume 8 provides for the configuration of one PSK. The PSK is an attribute of the EtherNet/IP object. The definition of the PSK value is accomplished using the secure TLS Session for Initial Configuration which provides confidentiality through encryption. In Phase 2 of Volume 8, the capability to define multiple PSK values will be specified.

X.509v3 certificates
A device can be provisioned to use X.509v3 certificates and artifacts which includes:

- device certificates,
- certificates from trusted authorities, and
- Certificate Revocation Lists.

Two models for the provisioning of certificates and certificate artifacts are defined, i.e

- the push model (Phase 1) in which a configuration client obtains the certificates from a Certificate Authority on behalf of the device and afterwards writes the signed certificates to the device using Certificate Management Object and File Object services, and
- the pull model (Phase 2) in which the device obtains the certificates from a Certificate Authority using Enrollment protocols such as EST or SCEP and writes the signed certificates to the device using Certificate Management Object and File Object services.

Certificate Management

Use of Certificates in EtherNet/IP Security

As specified in Volume 8, Chapter 3, EtherNet/IP makes use of TLS and DTLS to provide a secure transport for EtherNet/IP traffic. TLS and DTLS make use of X.509 certificates during the (D)TLS handshake in order to authenticate the communicating peers, and to establish shared session keys.

There are two scenarios in which certificates are required for EtherNet/IP over (D)TLS:

1. In order to perform initial commissioning, a TLS connection is made to the device which uses the device's default certificate. The default certificate can either be vendor-provided, or can be a self-signed certificate generated by the device.
2. The end user can elect to generate certificates via the user's PKI and install those certificates in the EtherNet/IP devices. Coupled with configuring the trusted Certificate Authority (or Authorities), devices can initiate and accept secure sessions only with those other devices that have certificates from the user's PKI.

Further Usage of Certificates in CIP Security

It is expected that certificates will have additional uses beyond the current use for EtherNet/IP over TLS and DTLS. For example, it is likely that CIP-based device and user authentication and authorization will make use of certificates.

Default Device Certificate

A device that supports CIP Security shall have a default X.509v3 device certificate when in its factory default state. Two options are provided for the default device certificate:

1. Vendor-supplied certificate installed in the device at manufacture time. (Recommended). For this option, it is recommended that vendors follow the Secure Device Identity standard (Std-IEEE 802.1AR-2009).
2. Self-signed certificate generated by the device, as shipped from the vendor or generated at its initial power-up. (Required if the vendor-supplied option is not provided).

The default device certificate shall be stored in Certificate Management Object instance 1. If providing a vendor-supplied certificate, the certificate of the root authority, and any intermediates, shall also be stored as a certificate chain in Certificate Management Object instance 1.

Certificate Enrollment with End-User PKI

End users have the option to configure devices with certificates generate by their own PKI, giving the device a locally-significant, cryptographically-secure identity. The user-generated certificates can be used with EtherNet/IP over (D)TLS, ensuring that only the end-users designated devices are able to establish secure communications.

The Certificate Management Object (see Volume 8, Chapter 5) defines a CIP interface for certificate management. Two models are defined by which a device may obtain certificates from an end-user PKI:

- Using the *push* module, a configuring client uses Certificate Management and File Object services to write a signed certificate to the device.
- Using the *pull* model, the device obtains a certificate from an enrollment server using a standard protocol such as EST or SCEP.

The models are described in more detail in the following sections.

Note: at present ONLY the push model is defined. The pull model will be defined in a subsequent specification revision.

References

[1] RFC5247, Transport Layer Security (TLS) Protocol Version 1.2, Aug 2008
[2] RFC6347, Datagram Transport Layer Security Version 1.2, Jan 2012
[3] RFC5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008