CIP Security Pull Model from the Implementation Standpoint

Jack Visoky
Security Architect and Sr. Project Engineer
Rockwell Automation

Joakim Wiberg
Team Manager Technology and Platforms
HMS Industrial Networks

Presented at the ODVA
2018 Industry Conference & 18th Annual Meeting
October 10, 2018
Stone Mountain, Georgia, USA

Abstract

The Pull Model was added to the CIP Security specification in May of 2018 and it introduces a new, automatic method for provisioning CIP Security credentials into a device. However, the configuration of a server for granting the credentials is not defined by CIP Security and left to the implementer and end user. There are several potential models that can be used as a security policy. This paper explores policy models for credential granting ranging from purely automatic to highly manual. Security considerations and ease of use concerns for each explored model will be discussed in detail both in the context of implementers and end users.

Keywords

CIP Security, Cybersecurity, TLS, Provisioning, Security Policy

Definition of terms (optional)

Acronym/Term	Description
CA	Certificate Authority: the service that signs certificates to vouch for their validity. This includes keeping the private keys used to sign the certificates private, as well as publishing the public keys used to verify the certificate
CSR	Certificate Signing Request: a document which contains all of the necessary information for a certificate, but has yet to be signed by a Certificate Authority
DNS-SD	Domain Name System Service Discovery: A protocol that allows an endpoint to discover local services
EST	Enrollment over Secure Transport: A protocol that allows an entity to request a certificate securely over HTTP

IDS	Intrusion Detection System: a solution which detects various types of cyber-events and/or precursors to cyber-events, generally providing aggregation and notification of this data.
MAC ID/MAC Address	Media Access Control ID/Address: a unique identifier for an endpoint at the link layer of a computer network.
mDNS	Multicast Domain Name System: a protocol to discover DNS services using IP multicast communication.
PC	Personal Computer
PKI	Public Key Infrastructure: is a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption. In cryptography, a PKI is an arrangement that binds public keys with respective user identities by means of a certificate authority (CA).
Pull Model	Defined in CIP Volume 8: CIP Security, this is a mechanism for a CIP endpoint to autonomously request a certificate via DNS-SD and EST
Push Model	Defined in CIP Volume 8: CIP Security, this is a mechanism for a client to request a device to generate a CSR and then write the signed CSR back to the device as a certificate. The device does not act autonomously in the Push Model like it does in the Pull Model
SMS	Short Message Service: a type of “text message” that is widely supported on mobile devices
TLS	Transport Layer Security: the widely used and widely deployed solution for securing data at the transport layer. CIP Security is built on TLS, as are the vast majority of the websites that leverage transport security on the Internet.

Introduction

The CIP Security Pull Model was introduced in May of 2018, and allows a device to automatically request a certificate for use as a secure identity. Prior to the definition of the Pull Model, the only way a certificate was deployed to a device was through a series of interactive CIP services that requested the device create a Certificate Signing Request (CSR) and then “pushed” a signed certificate to the device. Although this is still an option for devices, the Pull Model offers several advantages. A major advantage is the automatic aspect of the Pull Model; once a device attempts to communicate on the network it will attempt to automatically discover a server and requests a secure identity. This reduces the complexity of provisioning a device for a user. Another advantage comes in the device replacement scenario. When replacing a device, most of the configuration data can simply be sent to the new device. Therefore, the same data that was on the old device will also exist on the new device. However, a device’s certificate is a notable exception. The certificate is unique to a device, so even in the case of a device replacement a new certificate must be issued. The Pull Model allows the device to automatically request and receive a new certificate. At this point other configuration data can be sent to the device. The Pull Model allows a minimum set of security credentials (the certificate) as well as some automatically set security configuration to be automatically provisioned to the device. With a secure identity the device can now participate in secure communications, which allows for sending of other needed configuration information in a protected manner.

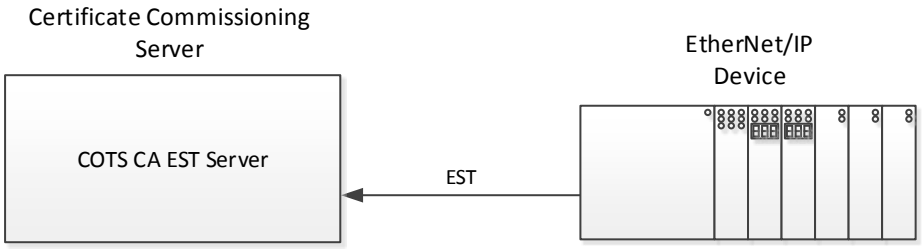

Figure 1: Pull Model Certificate Deployment (Image courtesy of Volume 8 of the CIP Networks Library)

The technologies used for the Pull Model are mDNS/DNS-SD for discovery, and EST for the secure request of the certificate. However, neither Volume 8 of the CIP specification, nor the EST RFC defines policies around what conditions must be met for a certificate to be granted. This allows for a great deal of flexibility for policy around CIP Security certificate deployment within the Pull Model. However, it is likely that a few basic policies will be popular. Some likely common policies are explored in this paper, as well as benefits and drawbacks of each.

Vendor Certificate Based Approval

One of the most straightforward and potentially useful ways to configure trust in the Pull Model is to base the trust on a verified Vendor Certificate. CIP Security requires a device to have a default certificate, allowing for two options: a vendor-signed certificate or a self-signed certificate. An EST server could be set up with knowledge of the trust anchors for a small list of trusted vendors. When a device connects to an EST server, the server could request a client certificate, at which point devices with a valid Vendor Certificate would use that certificate to authenticate themselves. If the certificate was indeed signed by a trusted vendor's CA then the device could automatically be granted a certificate. As ODVA members are encouraged to create devices with Vendor Certificates, this would hopefully cover a wide range of potential devices. Furthermore, the automaticity of granting a certificate to a trusted vendor's device allows for seamless device replacement/commissioning.

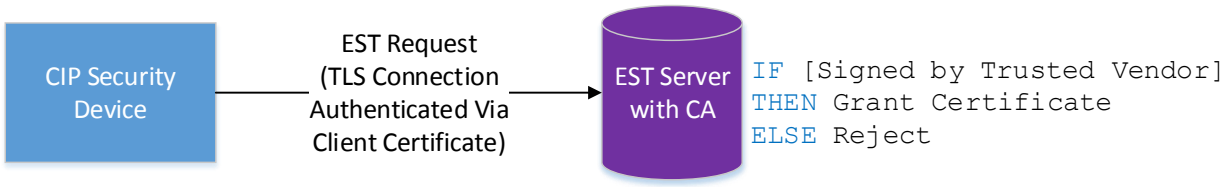

Figure 2: Example Vendor Certificate Based Approval

Of course, it is likely that large-scale deployments will include endpoints without Vendor Certificates. As noted, CIP Security compliant devices are not required to have a Vendor Certificate. Furthermore, for a software endpoint installed on a PC it would not be possible to include a Vendor Certificate. Therefore, it is unlikely that the trust configuration could cover all possible devices, and this trust configuration would likely need to be combined with others. At the same time this type of trust model would likely significantly ease the burden for device replacement and initial commissioning.

This model is not without risks. If a particular vendor's certificate authority is compromised then it would be possible to commission one or more attacker-controlled endpoints. However, vendors certainly have a vested interest in protecting their PKI, and would likely apply best practices for robust protection. Possibly easier than compromising a PKI would be to compromise a given vendor's device. As an example, a vulnerability in firmware leading to arbitrary code execution would allow an attacker access to a trusted instance of a Vendor Certificate, which could be used to provision an attacker-controlled device. However, similar to the vendor's PKI, vendors would likely expend energy to guard against vulnerabilities

and patch those that are discovered. As a worst case, even a trusted device without vulnerabilities might have some capability to launch an attack, such as a PLC that might execute user code. In this case an attacker might gain control of a valid PLC but load it with malicious user code. As such, this trust model is not without risk, and users would still need to apply best practices which might include IDS, security auditing, and network hardening. At the same time this model requires minimal user action to put into practice, and could be used in systems which need to get up and running quickly.

Administrator approval based on notification

In contrast to the automated case described above which requires quite a bit of infrastructure and effort to set up and maintain there are cases when an almost fully manual approval process is desirable; in other words there is always a human "in the loop" when granting a certificate. Cases when one would like to use a manual process to approve the request might be one of the following:

- In an installation with a limited number of devices
- When it's expected to be a low number of devices added to the network at a later point
- In a case where it's preferable to manually keep track of devices that have been granted access

Requiring an administrator to approve a device that attaches to a network subsequently requires that administrator to inspect any identity information the device in question is providing. When a device without valid credentials is attached to a network it will request new security credentials from the CA server, and in the case of the manual approval the request will show up on the CA server's user interface.

The CA server administrator would have to check in periodically to see if there are any pending requests to approve or decline. While the request is pending in the CA server, the device will continuously poll the CA server and wait for the administrators input. During this time the device won't have credentials to communicate, using TLS, with other devices on the network which already have been provisioned with security credentials.

Having the CA server administrator check the user interface on a periodic basis isn't a preferable way of dealing with the requests. If the administrator forgets to check the user interface a device will be stuck waiting for approval and won't be granted access to the network. The system administrator who's installing the device would then have to contact the CA server administrator for the approval. For these reasons it's generally recommended to have the CA server send some sort of notification to the CA server administrator once a certificate request is received. The kind of notification used is dependent on what's preferable for each application; the notification can be a push-message, e-mail, SMS, or something that show up on a dashboard. This notification allows the administrator to act promptly and access the CA server to inspect the certificate request and either approve or reject the request. Generally for a system with the appropriate infrastructure this model provides a high degree of control and assurance for secure device provisioning.

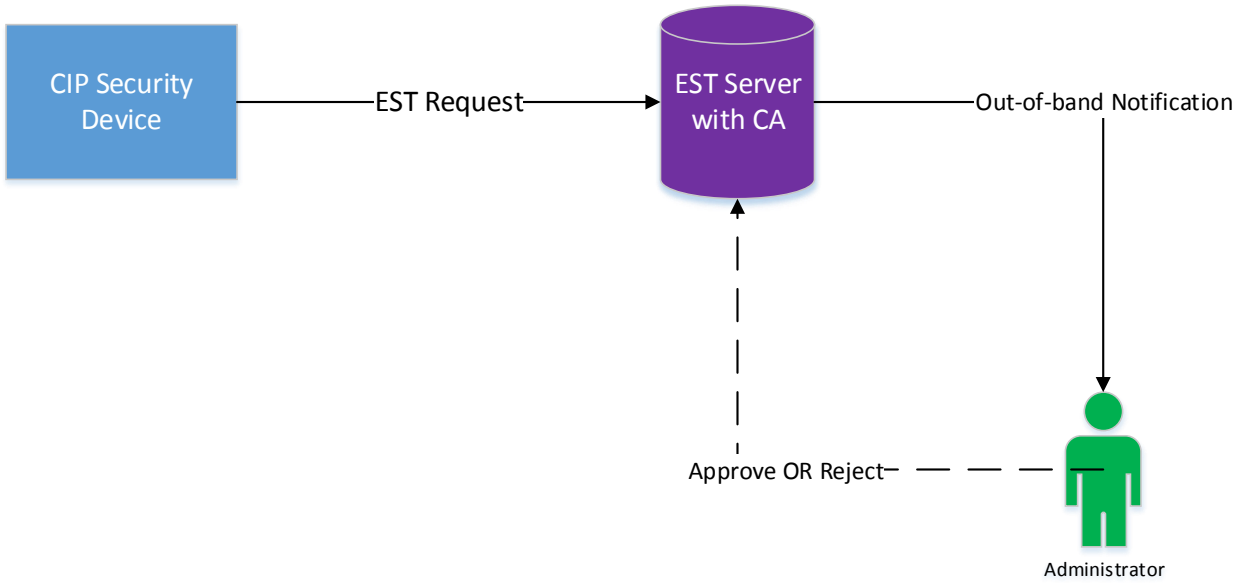

Figure 3: Example Administrator Based Approval

Approval based on a list of serial numbers

One option of approving certificate requests which is somewhat similar to the use of Vendor Certificates is using serial numbers of the devices. The approval of certificate requests can be automated using this approach. In this case the administrator of the CA server will have to install the serial numbers into the CA server. Upon reception of a certification request the CA server will then search the list of serial numbers. If a match is found the certificate request will be granted, conversely if there's no matching serial number within the inventory list the request will be rejected.

Naturally the serial numbers must be guaranteed to be a unique identifier amongst all possible devices, otherwise a device that shouldn't be granted a certificate will be provisioned with a certificate allowing it to communicate. For this reason, the CIP Serial Number alone can't be used since it's only guaranteed to be unique per vendor. However, this could be combined with the CIP Vendor ID to create a unique tuple, all of which is present in the default certificate. However, this is not the only option. Another approach could use the MACID as a serial number, as the MACID is guaranteed to be unique for all Ethernet devices. Note that Vendor Certificates contain both the CIP Serial number and CIP Vendor ID, so for a device with a Vendor Certificate this mechanism would be robust against a spoofing attack. Installations with a robust inventory management system will likely benefit from this model as they already have much of the necessary infrastructure in place to take advantage of this mechanism.

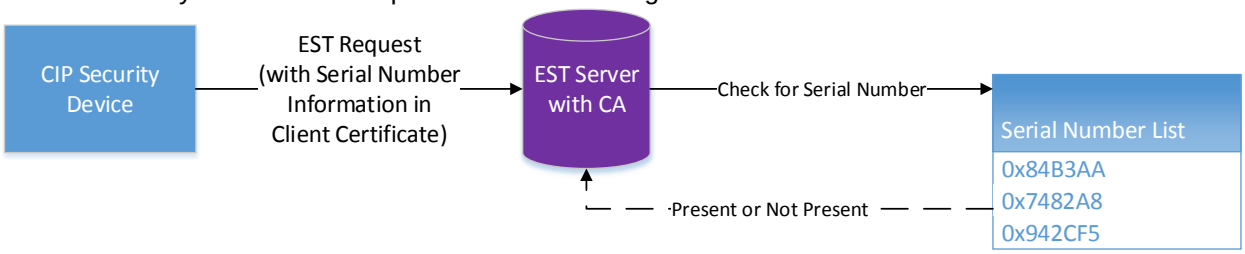

Figure 4: Example Serial Number Based Approval

Username/Password (Software Only)

The EST server does allow for a username/password to be provided for authentication in terms of granting a certificate. Although the CIP Specification does not discuss this workflow, it could potentially be used for devices that contain a human user interface. In particular, software clients that participate in the Pull Model are well-suited to using this type of authentication, as a user would be installing them onto a PC which could be used to enter a username and password. This could be especially useful because software endpoints could not contain Vendor Certificates, prohibiting them from being used with the vendor trust model discussed previously. However, if combined with this trust model the user has a robust system that deals with both trusted vendor devices and software. Note that the password is sent to the EST server over a TLS connection, therefore it has protection while in transit.

Beyond just using this model for software, this model could potentially be used even for devices. If the devices had a mechanism to enter a username and password, such as a removable media interface, then they could also participate in this type of trust configuration. However, there is no standard mechanism for this to be done at this time, so any mechanism for this would necessarily be vendor specific.

Risks within this trust model deal with the standard risks around usernames and passwords. Humans are notoriously bad at managing passwords, and this is often an exploited attack surface. Therefore, any use which implements this type of trust configuration would need to follow best practices for managing passwords within their environment. Systems with a large degree of software would benefit from this type of model.

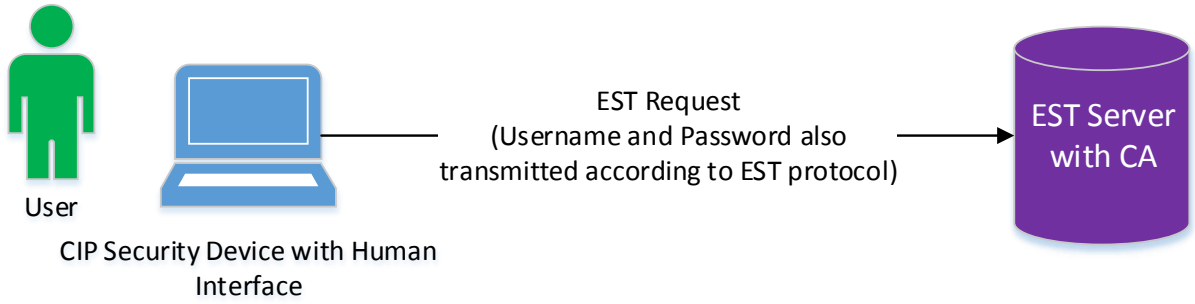

Figure 5: Example Username Password Based Approval

Approval via Provisioning Certificate

Another mechanism for configuration of trust would be to produce one or more certificates and key pairs signed by a CA that the EST server trusts. These certificates and key pairs could then be accessed by devices when authenticating to the EST server. One mechanism for this would be to use secure and removable smartcards to allow temporary access to the certificate and key pair by a trusted device. Since physically controllable smartcards are used, the access would be temporary, and controlled via physical access to both the device and the smartcard. Note however that like the username/password trust model discussed above, there is no standard mechanism for a device to access the certificate and key pair. Devices would likely need some sort of removable media port that can be accessed physically during commissioning. Despite this limitation, this type of trust configuration is a powerful mechanism for commissioning of devices using the Pull Model. Furthermore, this type of trust configuration would likely work for software, as PCs generally have a mechanism for accessing removable media.

This trust configuration has risk around the control and protection of the media which stores the trusted certificate and key pair. Some of these risks can be mitigated via storing this on secure hardware, like a smart card. However, even when employing hardware with secure key storage the user still risks losing control of the media with keys. These can be lost or stolen, and in a large system it would be difficult to

keep track of all of the keys. Of course, certificate revocation could be used to mitigate some of these risks, even going so far as to only allow a given certificate and key pair to be used at most one time. At this time a model like this would be limited due to lack of vendor support, but if this support grows then it would provide a high level of assurance for systems which can control the provisioning media.

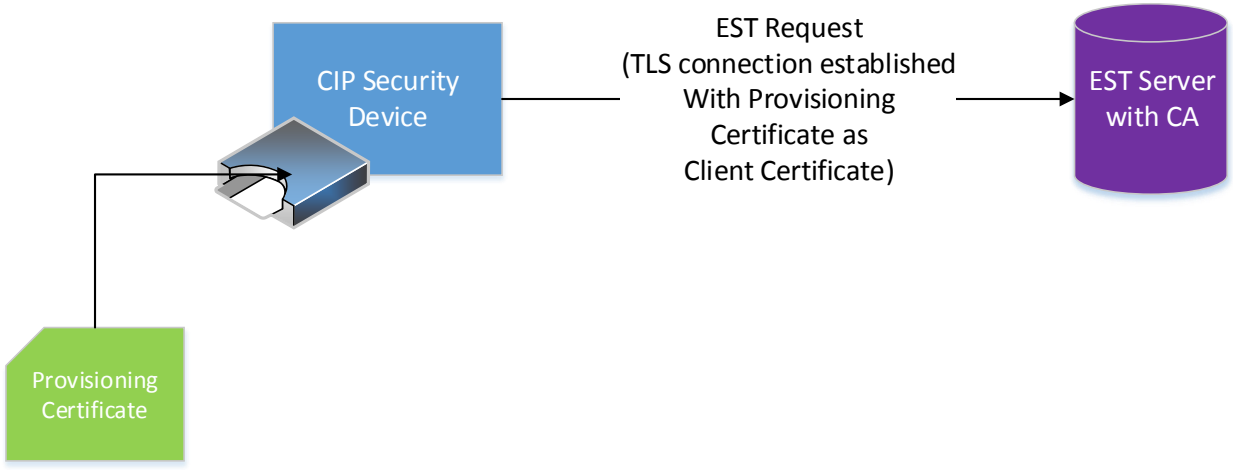

Figure 6: Example Provisioning Certificate Based Approval

Global Grant

A final provisioning method would be to simply grant certificates to any party which requests one. This strategy would have pretty obvious security implications in that anyone who was able to get onto the network could request a certificate and begin communicating, thereby circumventing many of the security benefits available. Despite this, there still may be uses for this type of model. A user might wish to test security within their system without fully deploying segmented trust anchors. Or a user might have other network hardening techniques and wishes to simply track the device communicating via certificate grant. It is unlikely that this model will be widely used, although it does remain a potential option for some use cases. The large inherent risk in deploying this type of model will likely preclude its use in most situations. Its use would likely be limited to laboratory and testing types of environments, not deployed in actual production.

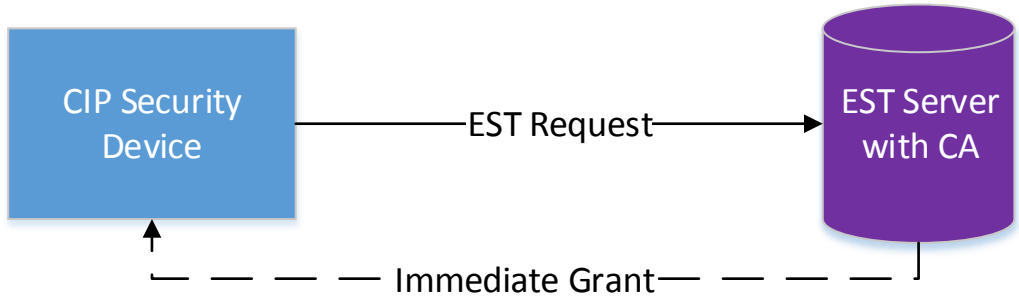

Figure 7: Example Global Grant Approval

Conclusion

This paper has explored several different models for the approval of secure certificate grant within the CIP Security Pull Model. None of these are a “one-size-fits-all” solution, rather each offer distinct advantages and disadvantages. Many of these models could be easily implemented with a commercial or open source CA, others would need some assistance from product vendors (e.g. the username/password or provisioning certificate grant). As such, this paper provides guidance on some of the characteristics of each model described. Users will need to go through a proper threat modeling process to determine which model is most suitable for their application and environment, and whether or not modifications need to be made to the given models.

References

[1] RFC5246, Transport Layer Security (TLS) Protocol Version 1.2, Aug 2008
[2] ODVA, Inc. The CIP Networks Library, Volume 8: CIP Security™, PUB00299
[3] RFC 7030, Enrollment over Secure Transport
[4] RFC 6763, DNS Based Service Discovery