ODVA 2017
Industry Conference
and 18th Annual Meeting

A Practical Guide for CIP Security Device Developers

Ron Floyd, Pyramid Solutions
Mike Mann, Pyramid Solutions
Jack Visoky, Rockwell Automation
Joakim Wiberg, HMS Industrial Networks

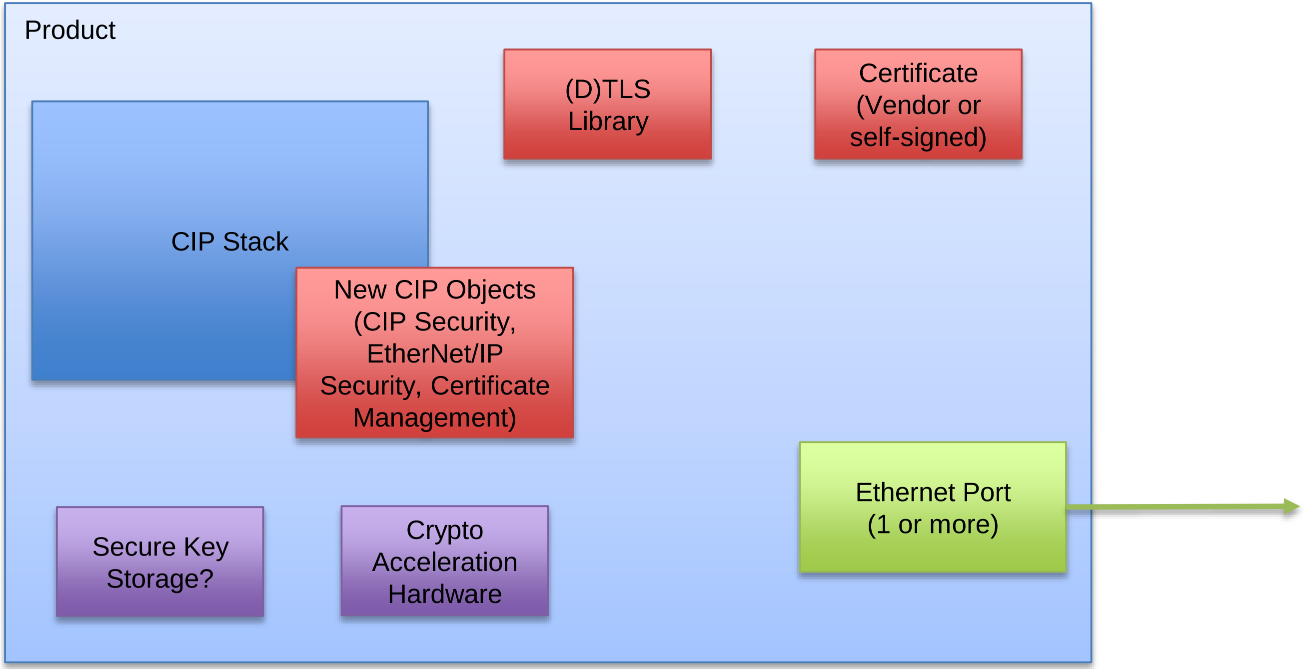
February 22, 2017

What is “A Practical Guide to CIP Security For Developers”?

- We want to give developers some hints and tips on what to do when implementing this functionality
- Although there is a lot of information in the spec, there is also some use in “non-normative” information
- None of the recommendations would be necessary for compliance
  - In some places perhaps no recommendation is made, just important considerations are noted

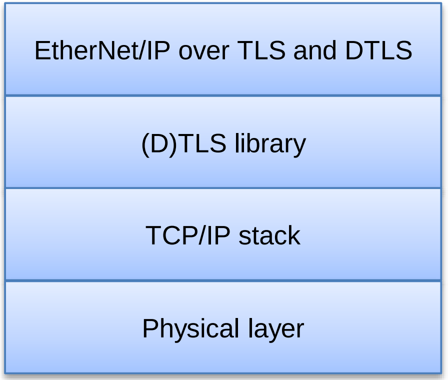
Before CIP Security

After CIP Security



Topic: Library Selection

- The core component in CIP Security over EtherNet/IP
  - Transport Layer Security (TLS)
  - Datagram Transport Layer Security (DTLS)
- A large and complex piece of software
- It’s probably a lot better to get a (D)TLS library than to try and write this code yourself ☺



Topic: Library Selection

- Many different vendors and projects
  - OpenSSL
  - wolfSSL
  - mbed TLS (formerly PolarSSL)
  - MatrixSSL
- At least a few vendors are using WolfSSL, that is a commercial library that is working for these purposes
  - Others?

Topic: Library Considerations

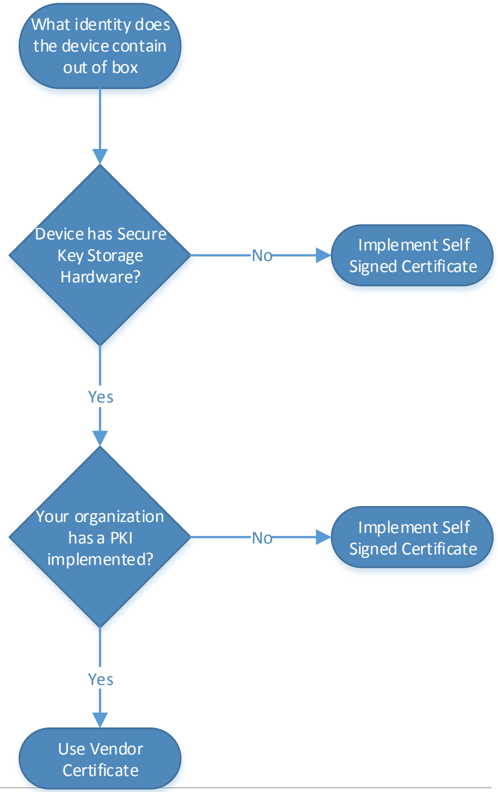
- Cost/Licensing
  - What is the budget for a (D)TLS library?
  - Is it open source?
  - Royalty based or licensed outright?
- Support
  - What happens when there are questions/work requests?
  - What level of documentation is available?
  - How intuitive is the API?
- Reputation
  - Is the library/vendor respected in industry?
  - Do the library developers have security expertise?
- Vulnerability Management
  - How are updates produced and consumed?
  - Are people actively testing the library for security issues?

Topic: Library Considerations

- Footprint
  - Memory constraints, what size is acceptable (both RAM and non-volatile)?
  - How configurable is the library; can unneeded features be compiled out of the binary?
- Capabilities
  - Does it support everything that is needed for CIP Security (e.g. NULL Ciphersuites)?
- Performance
  - Can it be optimized?
  - Does it integrate with hardware?
- Technology
  - Does it work well in the given environment (e.g. a Java library won’t work in a C environment)?
  - Is the API standard and fits in with the product’s architecture?

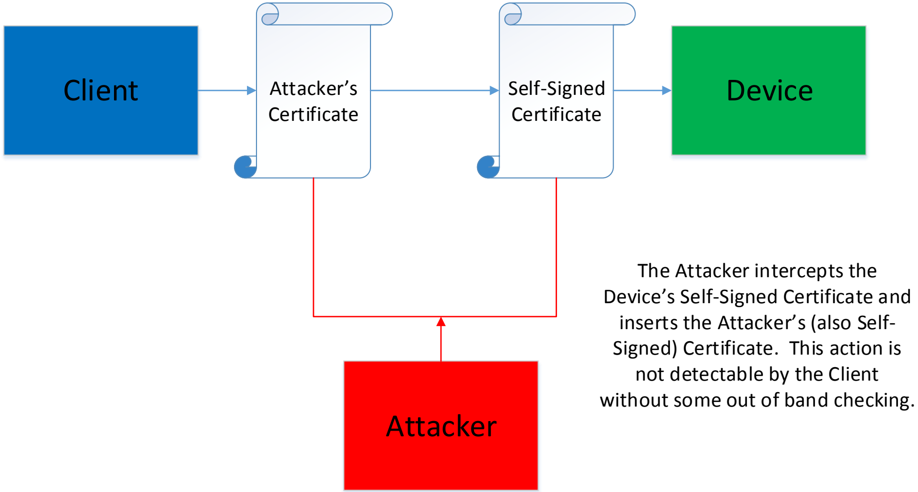
Topic: Key Management and Secure Identity

- Vendor Certificate vs. Self-Signed Certificate
  - If a Vendor Certificate is used, private key must be stored securely
  - Both work equally well for CIP Security
  - Vendor Certificate has may be useful for other things (securely identify a given vendors products, bootstrapping other things, etc…)
  - Essentially, low cost option vs. a more expensive yet more flexible/extensible option



Vendor Certificate CIP Security Benefit

- Vendor Certificate can be used to protect against “Man In The Middle” attacks on initial provisioning
  - However, only if the Vendor’s root was built in to the product
  - And only if both sides have a Vendor Certificate (unlikely for a software tool to have this)



Topic: Key Management with Vendor Certificate

- Need a secure place to store the key
  - There are solutions for this; TPMs, Secure Key Store chips, some FPGAs have built-in capabilities, etc…
- Need a mechanism to sign the Vendor Certificate
  - PKI; this comes with all the issues that are normally associated with the PKI
    - Managing a Certificate Authority – protect the keys!
    - Managing a Registration Authority – how to validate identity of requestors
    - How to access the PKI (e.g. just over a network or other mechanisms?)

```
X509 Certificate:
Version: 3
Serial Number: 6e6e5112000000000f24
Signature Algorithm:
    Algorithm ObjectId: 1.2.840.10045.4.3.4 sha512ECDSA
    Algorithm Parameters: NULL
Issuer:
    CN=Rockwell Automation - Manufacturing Intermediary CA 5
    O=Rockwell Automation, Inc.
    C=US

 NotBefore: 12/16/2015 12:19 PM
 NotAfter: 12/6/2055 12:19 PM

Subject:
    CN=1756-L85E (00b48f94)
    O=Rockwell Automation, Inc.
    C=US

Public Key Algorithm:
    Algorithm ObjectId: 1.2.840.10045.2.1 ECC
    Algorithm Parameters:
    06 08 2a 86 48 ce 3d 03  01 07
        1.2.840.10045.3.1.7 ECDSA_P256 (ECDH_P256)
```

Topic: Connection Origination

- Lots of devices are just “targets”, don’t originate connections
- Connection origination has additional considerations
  - How would a device know to originate connections as secure?
    - In an environment that has a mix of CIP Security capable devices and non-CIP Security capable devices this can be challenging
    - Otherwise non-secure ports can be disabled and all communications can be over CIP Security sessions

Topic: Debugging

- Previously packets can be sniffed using Wireshark or a similar tool
- If confidentiality is enabled this becomes much harder
  - Suggestion is just to debug it using a NULL ciphersuite
  - Wireshark plugins for confidentiality are available, but session keys are needed (use Wireshark 2.1.0 https://2.na.dl.wireshark.org/win64/Wireshark-win64-2.1.0.exe)
  - Considerations of how to allow for this
    - Don’t want this enabled in the field!!!
    - But, developers would want to be able to use this relatively easily

Topic: Testing Tools

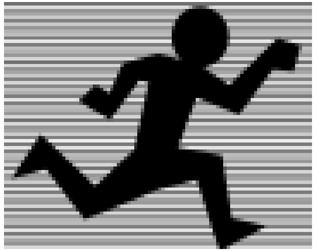
- OpenSSL
  - Useful for initial testing during early development
    - Together with Wireshark the initial TLS handshake can be debugged and tested
  - Perform the initial shake and key-exchange
    - Handy when performing performance evaluation and optimizations
  - Test and verify supported TLS versions
- Nmap
  - List all supported cipher suites

Topic: Testing Tools

- May want a tool that runs on a PC and originates connections
  - Send configuration to the device
  - Initiate connections to the device
  - Easily debug communications via a “transparent client” (simple to allow this tool to show what communications it sends or receives)

Topic: Performance Configurations

- There is (of course!) a cost to enabling CIP Security
- Can a given product handle the performance degradation?
  - Connection startup
    - Computational cost to handshaking (especially certificate verification)
    - Extra steps/data over the network for handshaking
  - Data flow during connection lifetime
    - Latency concerns; can performance targets be achieved?



Topic: Hardware Architecture

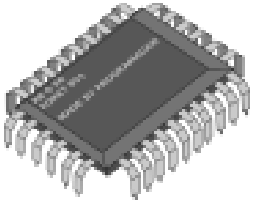
- Including specialized hardware on a CIP Security product can be very helpful (although not strictly necessary)
- Three general types of hardware:
  - Cryptographic Accelerator
  - Secure Key Storage
  - Entropy Generator

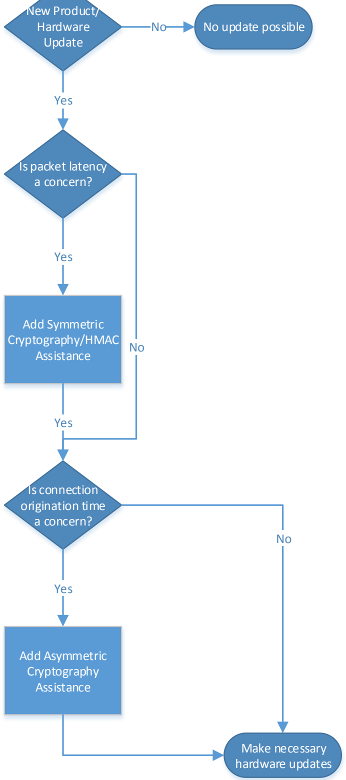
Hardware Considerations

- Regardless of what hardware is included, there are some common considerations
  - Trust Boundaries: is the hardware in an ASIC, on a PCB, on a USB stick, etc…
  - Performance: does the hardware achieve target performance
  - Capabilities: algorithms supported, interfaces, etc…
  - Cost: can the cost of adding the extra hardware be justified?
  - Contention: do multiple parts of the system need to access the hardware at the same time? If so, what mechanism can be used to arbitrate this

Topic: Cryptographic Accelerator Hardware

- Hardware can be used to make increase performance of cryptographic operations
  - Anywhere from a modest assist to near/at line speed
  - Of course this requires investment; whether or not it is worthwhile depends on many factors
  - However, given the importance of CIP Security, it is probably a good thing to at least consider for any new products





Topic: Other Hardware

- Secure Key Storage Hardware
  - As mentioned previously, this is needed for a Vendor Certificate
  - Other keys can be stored here (like key provisioned as part of the user granted identity)
  - It is important to consider key lengths and algorithms supported

- Entropy Generating Hardware
  - Including a True Random Number Generator is very helpful for secure generation of keys
  - Generation of cryptographic entropy is very difficult without specialized hardware

Topic: Entropy

- A good library will have at least one PRNG algorithm for generating random data
- However, those algorithms need to be seeded with truly random data
- This has to come from something physical
  - Cryptographic hardware often includes a TRNG
  - If you don't have a TRNG then you need to get creative
    - Look for things in the system that are non-deterministic
    - There's been work done on this, several papers published
    - Guidance could be provided for a few standard mechanisms

Topic: Cipher suites

- There are a lot of ciphersuites available, what should be used
  - CIP Security Spec defines some required ones
  - There are many others
- Asymmetric – generally 2 choices
  - Elliptic Curve offers same or better security at a smaller key size
  - RSA is more widely deployed

NIST SP 800-57 Pt. 1 Rev. 4

Recommendation for Key Management: General

Table 2: Comparable strengths

Security Strength	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
≤ 80	2TDEA[21]	$L = 1024$ $N = 160$	$k = 1024$	$f = 160\text{-}223$
112	3TDEA	$L = 2048$ $N = 224$	$k = 2048$	$f = 224\text{-}255$
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256\text{-}383$
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384\text{-}511$
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

Topic: Cipher suites

- Confidentiality, AES is essentially the gold standard
  - There are a lot of variations to this though
  - Generally the ones defined in the CIP Security Specification should be reasonable
  - Most TLS libraries will support many others; if space is not an issue other options can be given
    - CCM and GCM are both authenticated algorithms, give some additional benefit at the cost of complexity
- HMAC
  - SHA-2 is widely deployed and supported, SHA-1 still accepted by NIST for HMAC
  - SHA-3 recently released, yet to be widely adopted

Topic: System time

- X.509 v3 certificates have a field defining its validity period
  - notBefore and notAfter
- Likely the EtherNet/IP device doesn't have an RTC
  - Thus the validity period of the certificate can't be verified
- The EtherNet/IP device could implement NTP or IEEE-1588
  - Though none of them are secure
- Roughtime might be an alternative in the future

THANK YOU