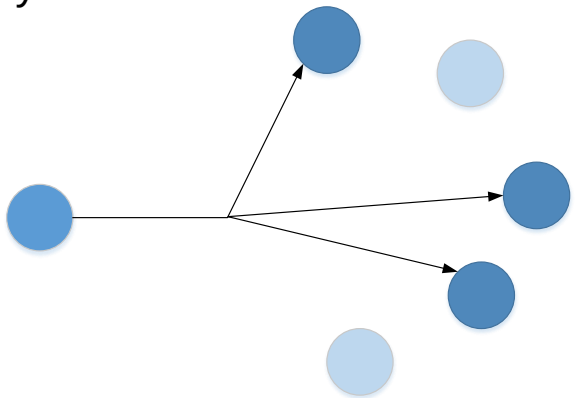
Secure Multicast

Jack Visoky and Joakim Wiberg
Rockwell Automation and HMS

October 10, 2018

Why Multicast

- Multicast has been a mainstay of EtherNet/IP for years
- Depending on the variables involved multicast transmission can provide a significant improvement to network transmission
  - Multicast can be particularly beneficial for I/O communications
  - In a case with several “listeners” the network infrastructure can take care of efficient delivery via multicast

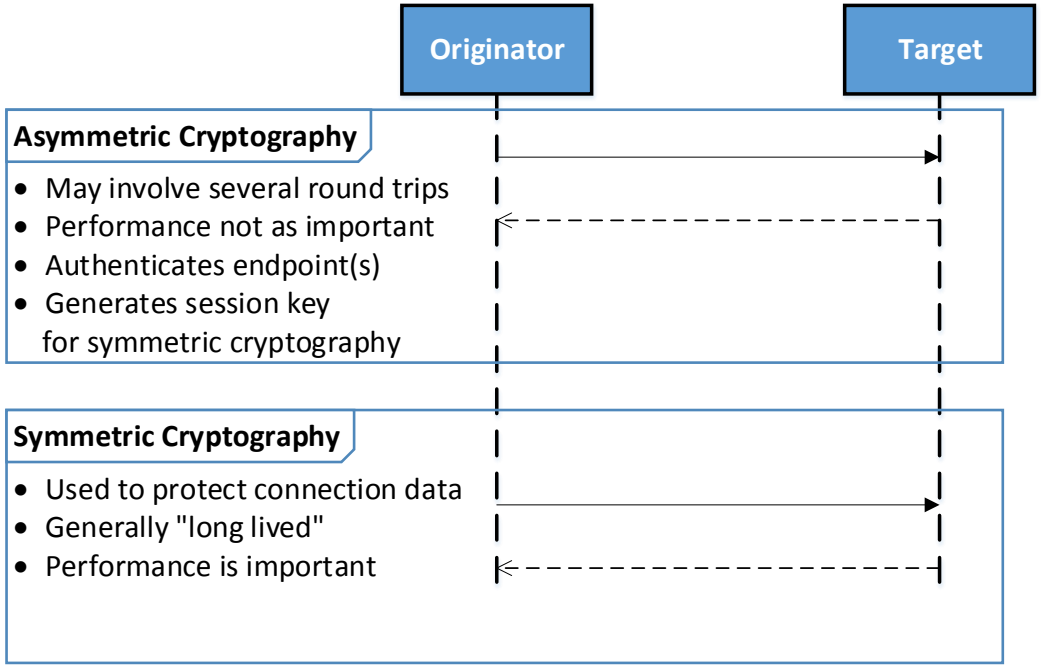


What Makes Multicast Security Difficult

- TLS and DTLS (the most widely supported transport security technologies) don't have any support for multicast
  - These are also the technologies on which CIP Security is based
- Large groups of senders/receivers make key management particularly difficult
  - Key servers need to authenticate each entering member and securely transmit key materials
  - Key servers are a single point of failure that can impact runtime
  - Members leaving the group mean that an action must be taken (like issuing new key material to all remaining group members) to ensure the departing members can no longer access data

Secure (Unicast) Communication Background

- As background, security protocols (like TLS, DTLS, IPsec, etc…) generally are unicast, point-to-point protocols
- The same basic structure is seen across these protocols
  - Asymmetric cryptography generally used to set up the connection
  - Symmetric cryptography generally used to protect the data in transit

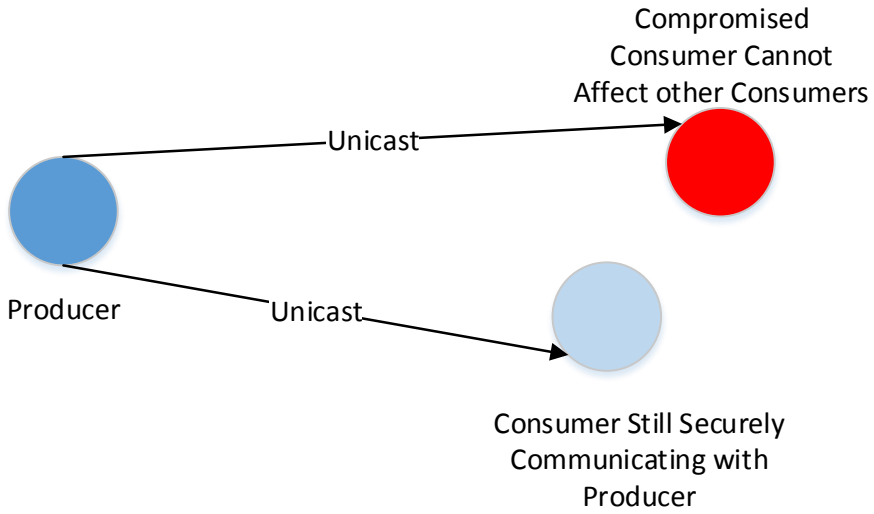


Secure Unicast Communication

- This basic pattern has served digital communications well for many years
- It's been proven in use across a number of industries, including securing industrial communications
- Despite this, there are some issues with mapping this pattern on to multicast
- Two issues stand out because they are fundamental to how the cryptography works
  1. Lack of originator authenticity
  2. Consumers dynamically joining and leaving

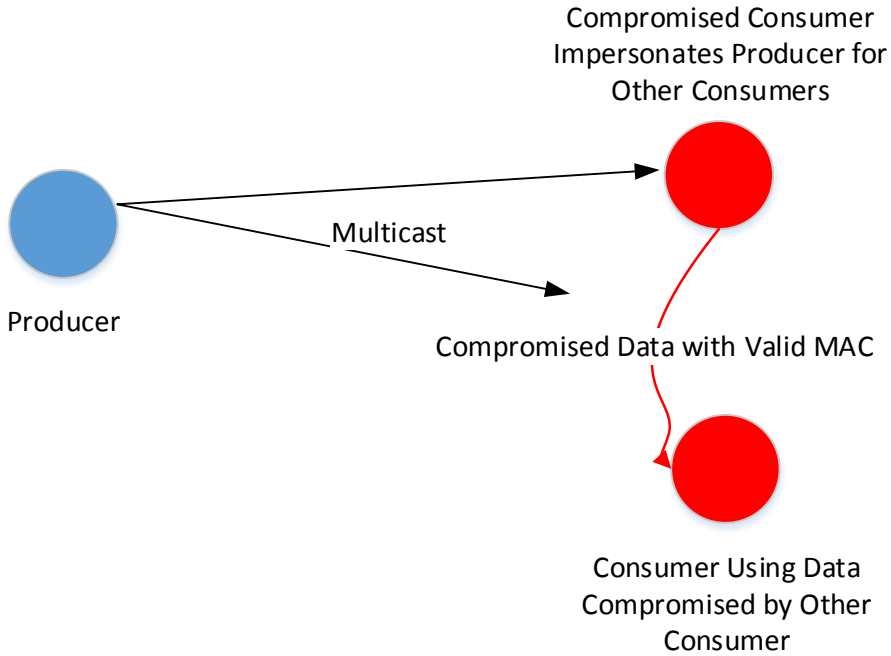
Issue #1: Lack of Originator Authenticity

- In unicast connections there are only two parties
  - Symmetric cryptography (MAC) works well; data is either sent by party 1 or party 2
  - It is trivial for either the sender or received to know who sent this data
  - A compromised consumer cannot affect other consumers if they are all using independently secured unicast connections



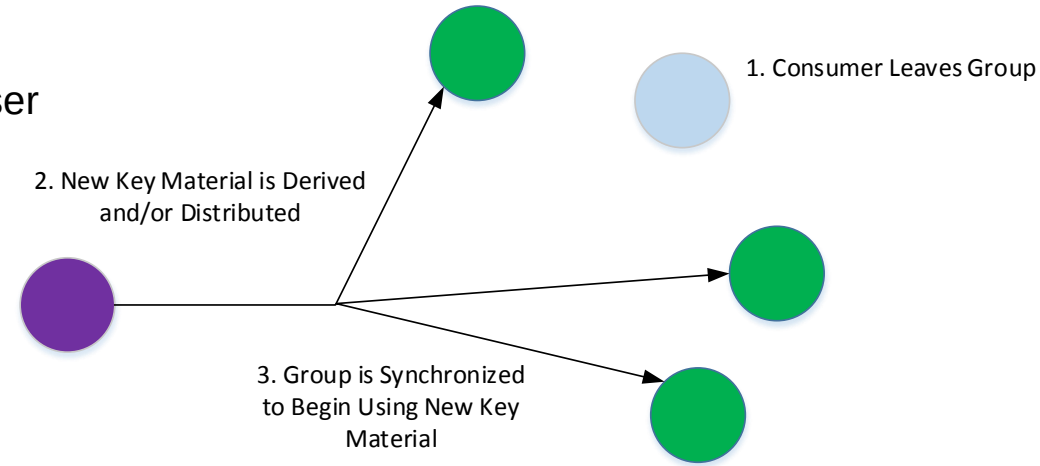
Issue #1: Lack of Originator Authenticity

- In multicast, there is more than one consumer of the data
  - Example: trivial case, 1 producer and two consumers
  - Symmetric key is shared between all parties
  - A given consumer cannot tell if the data came from the producer or the other consumer
  - Put another way, one compromised consumer can potentially compromise all other consumers



Issue #2: Consumers Dynamically Joining and Leaving

- Each time a consumer leaves the group, a new group key must be generated, distributed, and synchronized
  - This also generalizes to someone joining the group
  - Depends on threat model if the user cares about new consumers accessing old data, or old consumers accessing new data
- These operations can be quite computationally expensive as the could involve asymmetric cryptography, and at minimum involve group synchronization

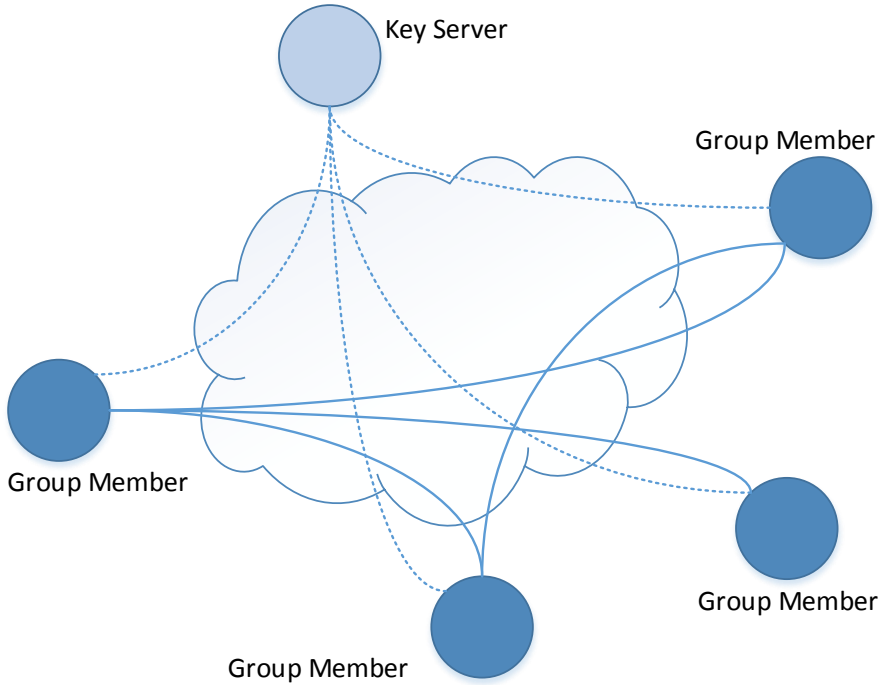


Existing Multicast Security Technologies

- Several technologies were examined for potential application to CIP Security
- No existing technology clearly and seamlessly fills the gap for securing multicast EtherNet/IP

IPsec and GDOI

- IPsec and GDOI
  - IPsec is (from a very high level) similar to TLS, although it works at the IP layer instead of the transport layer
  - GDOI (Group Domain Of Interpretation), along with IKE (Internet Key Exchange) can be used to distribute/revoke key materials to a multicast group
    - Note this is currently only defined to work with IPsec, although the RFC mentions that it could be extended to other security protocols



GDOI and IPsec Issues

- GDOI/IKE/IPsec are all very flexible but very complex; far more difficult to use than TLS and DTLS
- GDOI is a somewhat old technology that hasn't really kept up to date (still supports SHA-1 signatures)
- Although there are implementations, it isn't really widespread and certainly not within TLS/DTLS libraries
- Works with IPsec but far from a drop in to TLS and DTLS; significant work would be needed to fit this into these technologies
- Consumers joining/leaving dealt with via re-key policies, although impersonating producers is not dealt with

Secure Multicast with DTLS

- There was an effort within IETF, although it was abandoned years ago
- Relied on a server to distribute keys to the multicast group
- Looked somewhat promising but there was still significant definition needed for it to be completed
  - As this effort is not completed a full analysis is not possible
  - If there was enough industry pressure then this effort could potentially be restarted

Cellular multicast security

- 3GPP implements multicast security for cellular networks
- This is the technology on which 4G and LTE is built
- Again, uses a server to distribute keys to a multicast group
- Uses a lot of technologies which do not fit well into EtherNet/IP
  - SRTP instead of TLS
  - HTTP digest authentication
- Although could provide some “inspiration” for secure multicast in CIP Security, there are too many diverging technology choices for this to be workable

OPC-UA Pub Sub

- OPC-UA Pub Sub is a protocol that supports multicast communications
- Implements a Secure Key Server (SKS) to provide key materials to the multicast group
  - Policies can be implemented for key rotation
  - No protection for producer authenticity
- “Closed” solution developed by OPC Foundation
  - Not freely available like IETF materials
  - Further analysis requires access to specifications

Unicast Only

- One has to wonder, why is it that point-to-point security has been standardized so well but multicast security has not?
  - Perhaps the efficiency benefits of multicast aren't really that great in the long run, especially when security is taken into account
- Lack of energy around defining secure multicast is somewhat telling
  - Could be similar to the worries around using TLS to secure all web traffic years ago
  - Now HTTPS is the norm; networks, hardware, and software all improved to the point that this is barely noticeable

Conclusion

- Two potential options:
  1. Do Nothing – if it is determined that the market does not have a strong need for multicast then it is not worthwhile to define a security solution around it; multicast will likely fade from use over time
  2. Define Secure DTLS in IETF – if the market does require multicast, then an effort in IETF should be started/restarted. This would leverage the larger Internet community to define secure multicast in a way that is well vetted and workable for a large community, including IoT and Industrial Communications

THANK YOU