Your systems.
Working as one.

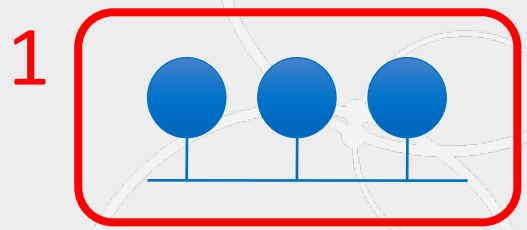
Security Hands On

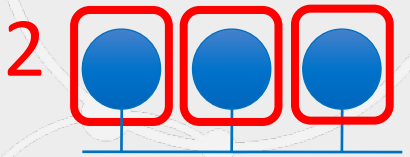
Gerardo Pardo-Castellote, Ph.D. CTO
Fernando Crespo Sanchez, Product Architect

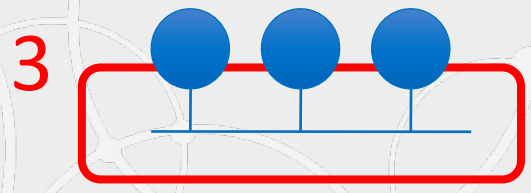
Intro to DDS Security

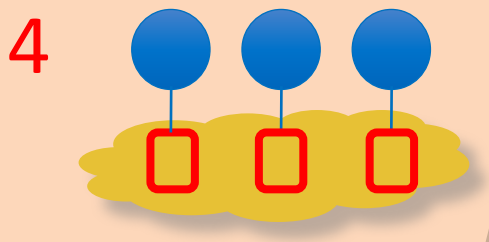
Security Must Protect Dataflow, Too

1. System Boundary
2. Host
3. Network Transport
   - Media access (layer 2)
   - Network (layer 3) security
   - Session/Endpoint (layer 4/5) security
4. Data & Information flows



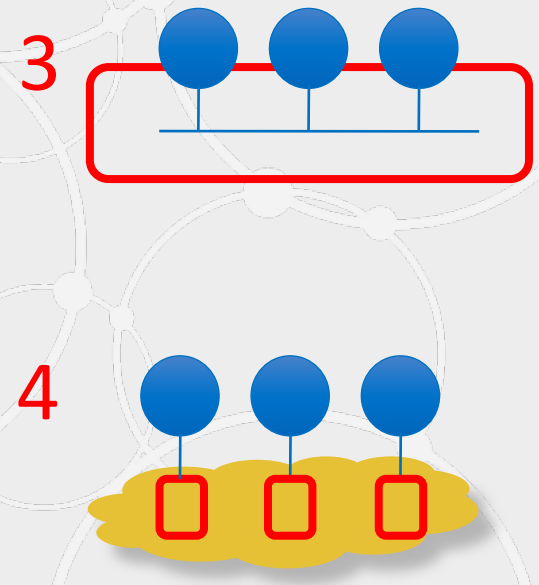






Approaches to Protect DDS

- Transport Layer Security
- Fine-Grained Security



Transport-Level Secure Data Transfer

Uses TLS or DTLS

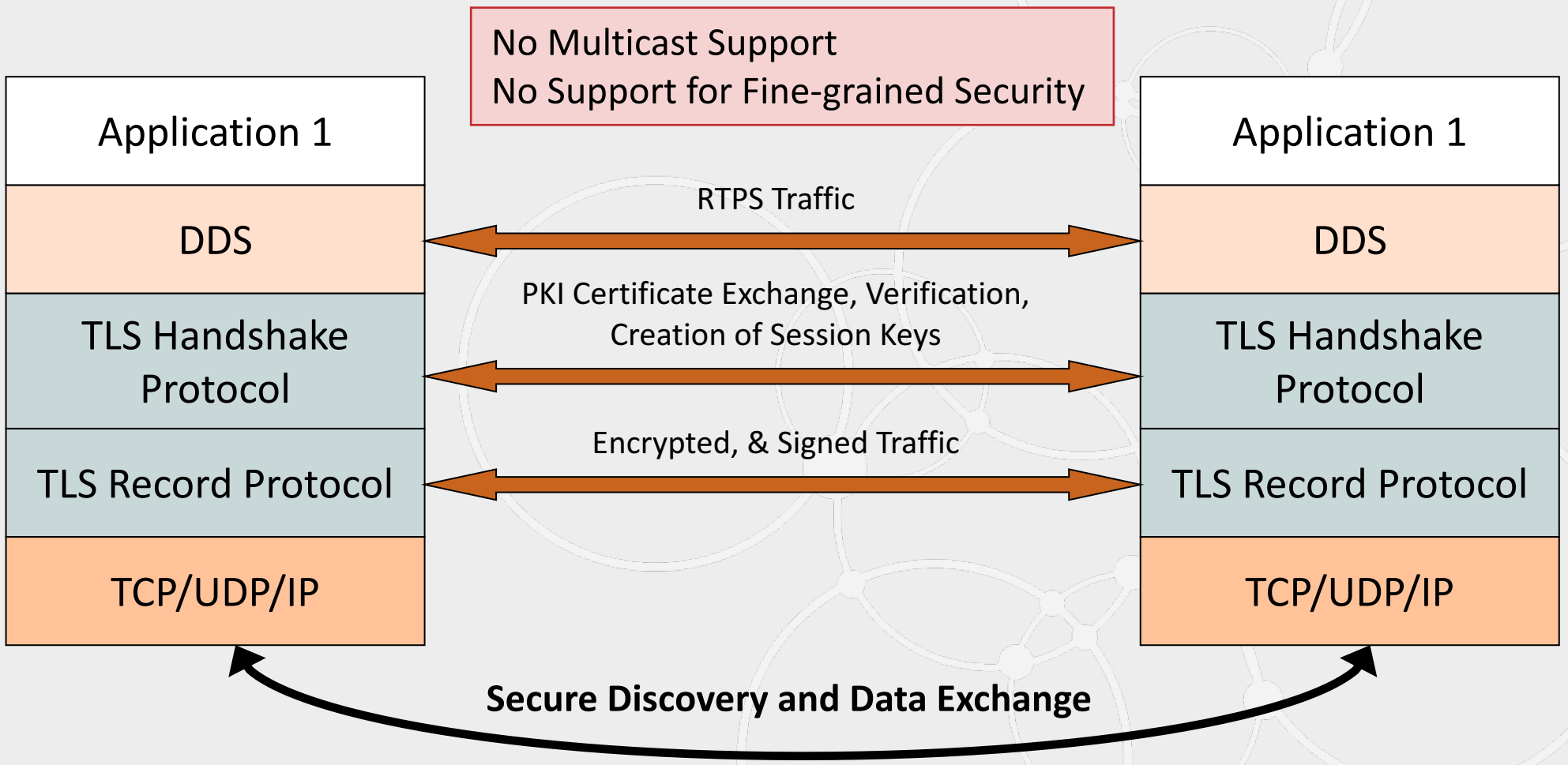
1. Authenticate
   - Verify your identity
2. Securely exchange cryptographic keys
3. Use keys to:
   - Encrypt data
   - Add a message authentication code

Transport-Level Secure Data Transfer In RTI Connext DDS

Three Connext DDS transports available in Connext DDS

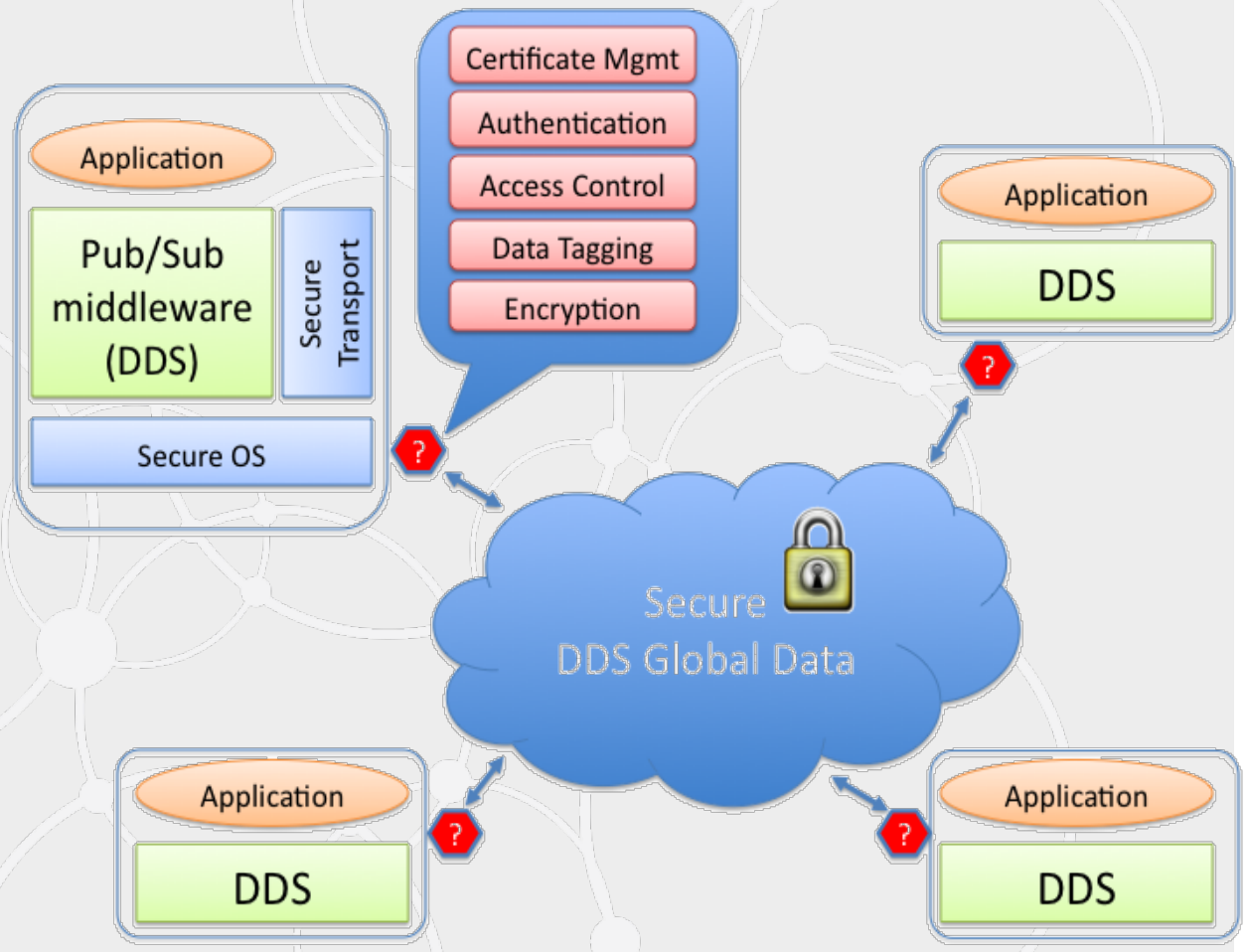
- RTI Secure WAN Transport
  - WAN UDP transport that uses UDP hole punching to traverse NATs
  - Optional transport authentication and encryption using DTLS
- RTI Secure DTLS Transport
  - LAN UDP transport
  - Transport authentication and encryption using DTLS
- RTI Secure TCP Transport
  - WAN/LAN TCP transport
  - Optional transport authentication and encryption using TLS

Transport-Level Security



DDS Security Standard

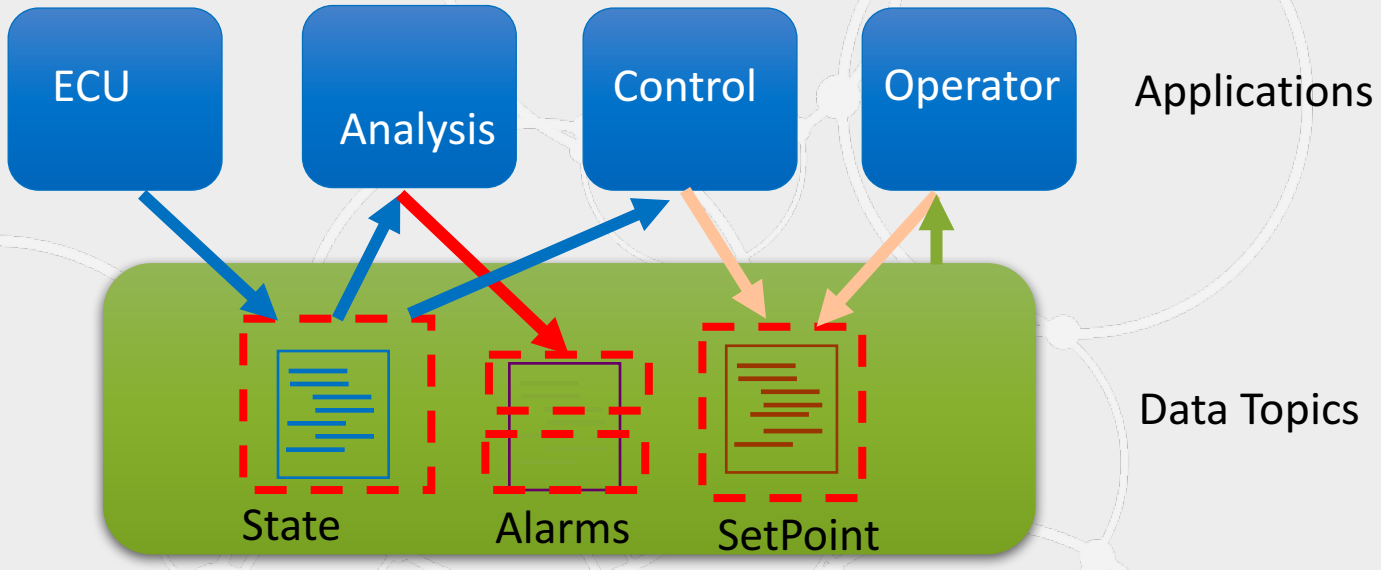
- DDS entities are authenticated
- DDS enforces access control for domains/Topics/…
- DDS maintains data integrity and confidentiality
- DDS enforces non-repudiation
- DDS provides availability through reliable access to data



…while maintaining DDS interoperability & high performance

DDS: Data-Centric, Fine-Grained Security

- Per-Data-Topic Security
  - Control r,w access for each function
  - Ensures proper dataflow operation
- Complete Protection
  - Discovery authentication
  - Data-centric access control
  - Cryptography
  - Tagging & logging
  - Non-repudiation
  - Secure multicast
  - 100% standards compliant

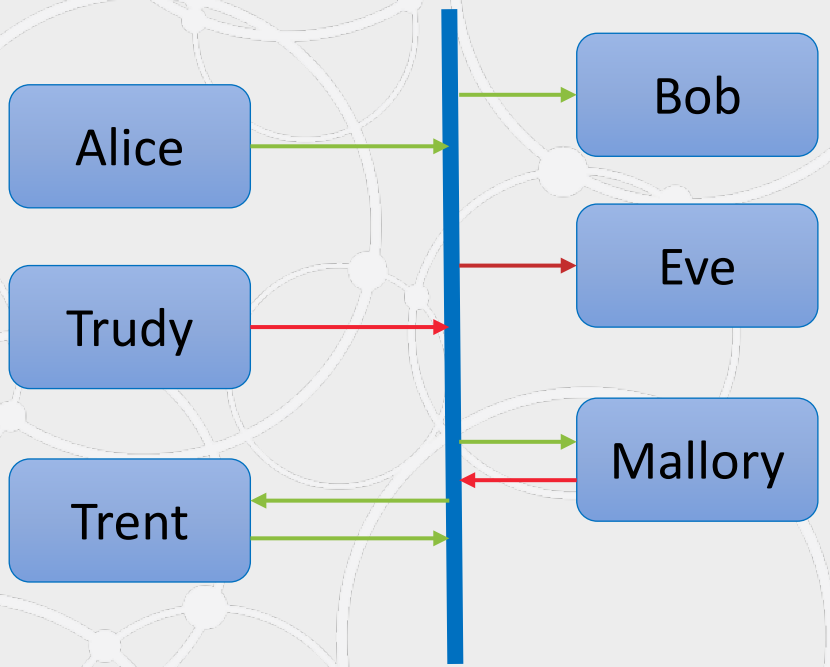


Data Topic Security model:

- ECU: State(w)
- Analysis: State(r); Alarms(w)
- Control: State(r), SetPoint(w)
- Operator: *(r), Setpoint(w)

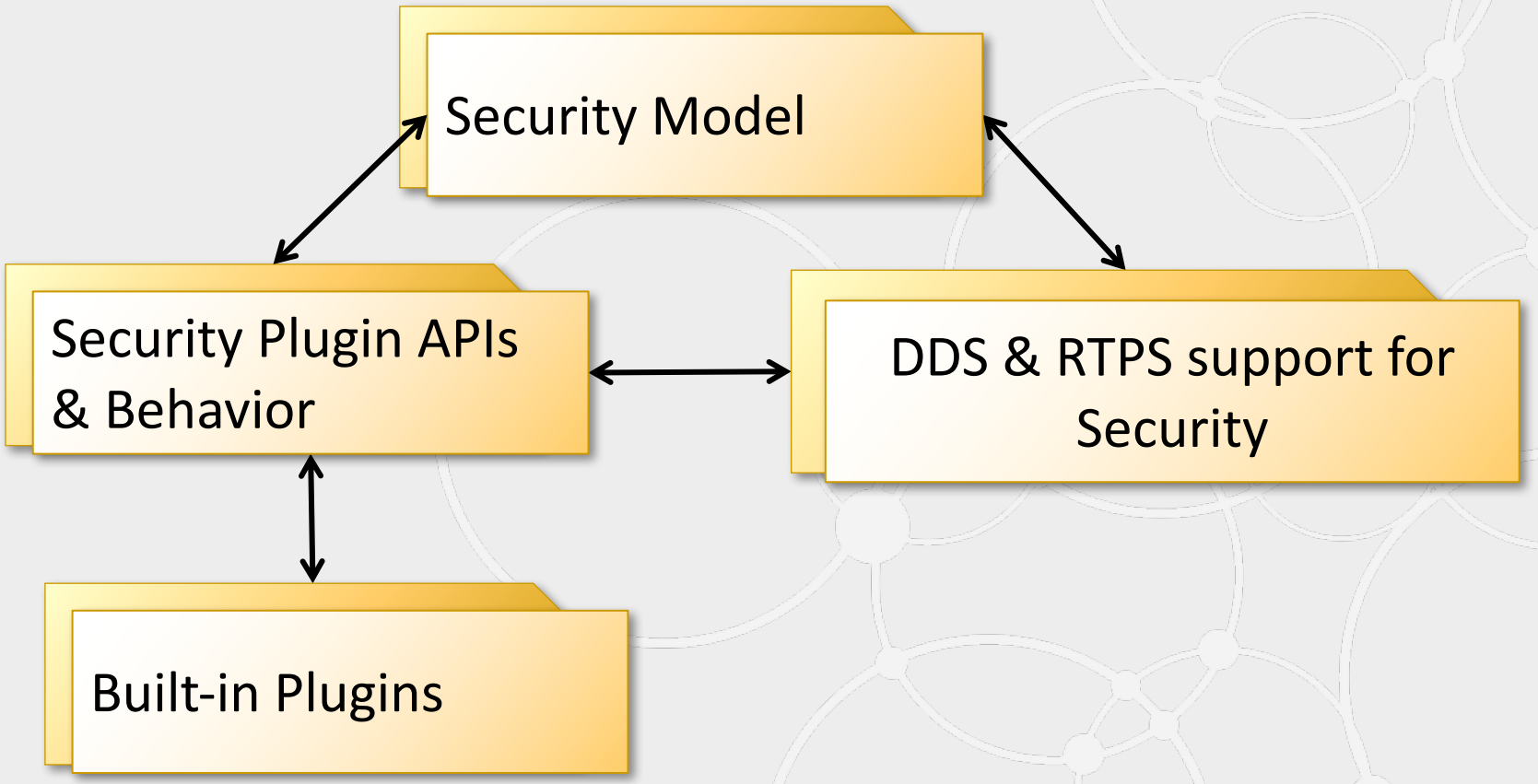
Threats

- Unauthorized Subscription
- Unauthorized Publication
- Tampering & Replay
- Insider Attack

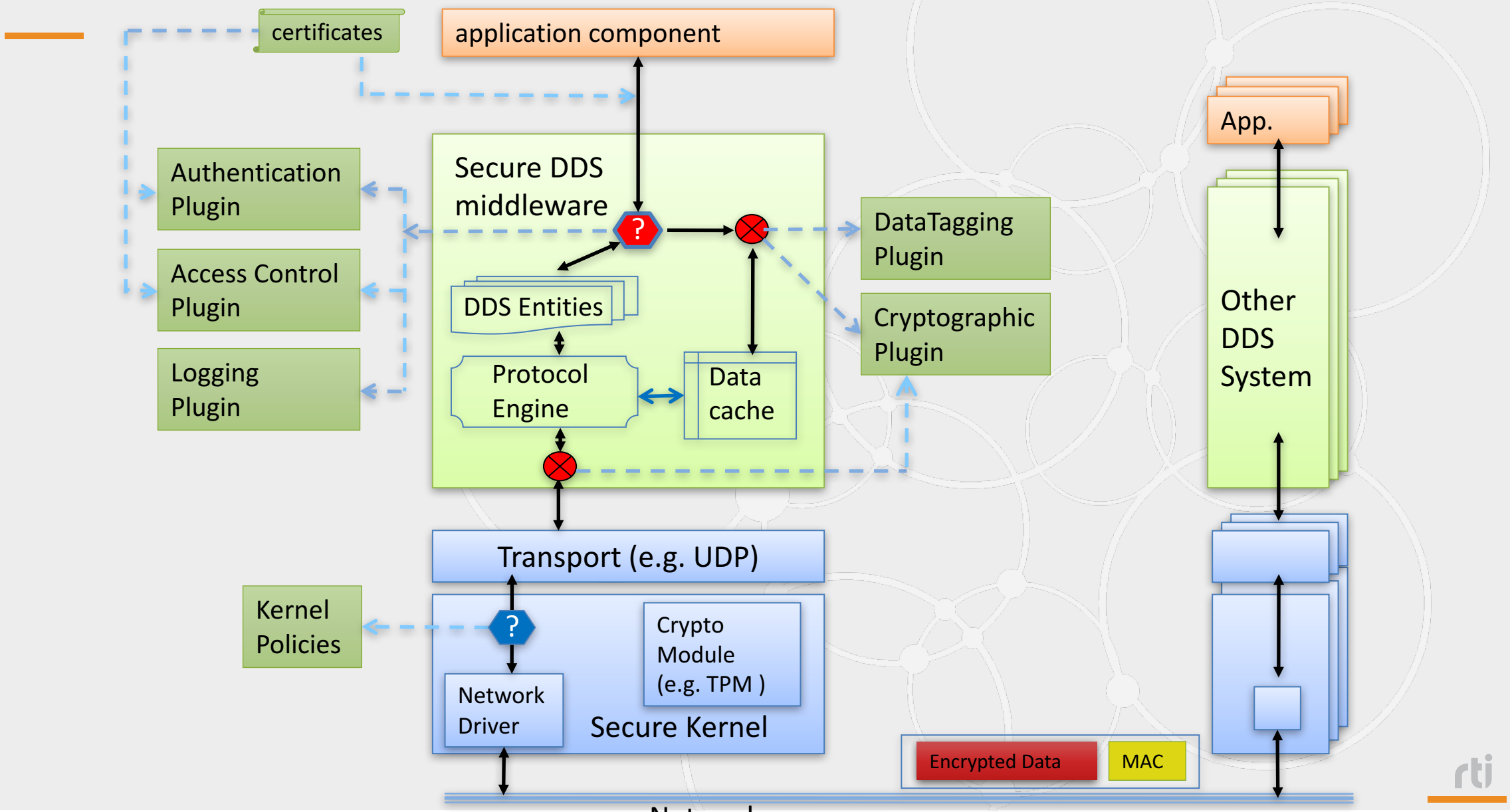


Local machine is assumed to be trusted

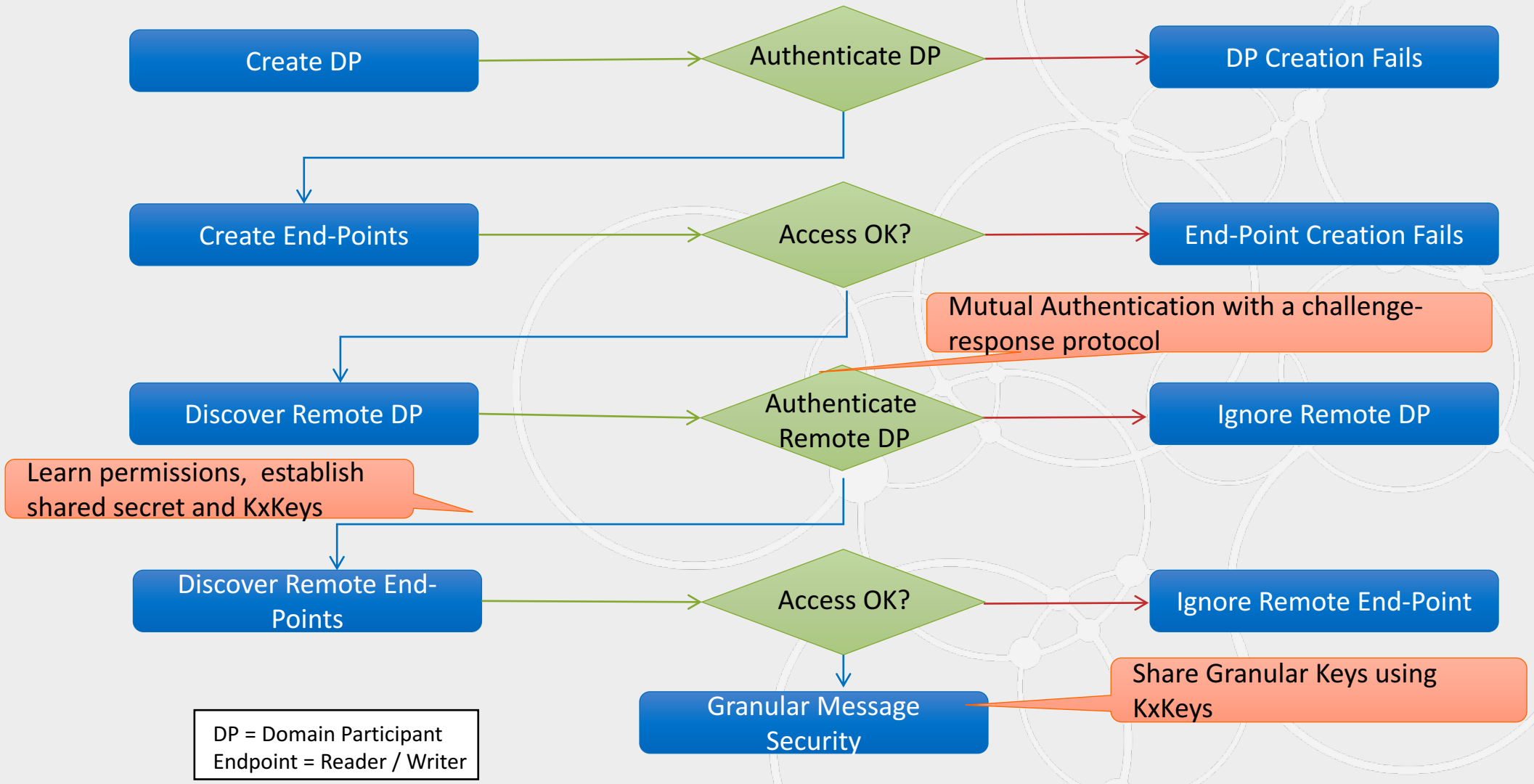
DDS Security Standard Covers Four Related Concerns



Pluggable Security Architecture



Overview of What Happens



Pluggable Architecture

Plugin	Purpose	Interactions
Authentication	Authenticate the principal that is joining a DDS Domain. Establish shared secret between participants	The principal may be an application/process or the user associated with that application or process. May exchange messages to do mutual authentication and establish shared secret
Access Control	Decide whether a principal is allowed to perform a protected operation.	Protected operations include joining a specific DDS domain, reading a Topic, writing a Topic, etc.
Cryptography	Perform the encryption and decryption operations. Create & Exchange Keys. Compute digests, compute and verify Message Authentication Codes. Sign and verify signatures of messages.	Invoked by DDS middleware to encrypt data compute and verify MAC, compute & verify Digital Signatures
Logging	Log all security relevant events	Invoked by middleware to log
Data Tagging	Add a data tag for each data sample	

SPI	Built-in Plugin	Notes
Authentication	DDS:Auth:PKI-DH	Uses PKI with a pre-configured shared Certificate Authority. DSA and Diffie-Hellman for authentication and key exchange Establishes shared secret
AccessControl	DDS:Access:Permissions	Governance and Permissions Document Each signed by shared Certificate Authority **Security configuration per Domain and Topic** **Access control per Domain and Topic**
Cryptography	DDS:Crypto:AES-GCM-GMAC	**Automatic key distribution** AES-128/192/256-GCM for Encryption and MAC SHA1 and SHA256 for digest **Separate keys per DW and DR** **Transparent secure multicast**
Logging	DDS:Logging:DDS_LogTopic	

Writer Message Security

- Encryption keys & MAC keys are generated per data writer
- These keys are securely distributed to data readers
- Distribution of these keys is done using other symmetric keys derived from the shared secret
  - Key distribution is transport independent
- Different parts of messages can optionally be protected per governance policy
- Data Delivery is independent of key distribution
  - May use any transport, including multicast

Access Control & Policy

- DDS Security allows for configuring & enforcing the privileges of each participant
  - Which domains it can join & what Topics it can read/write?
- It also allows specifying & enforcing policies for the whole domain, e.g.
  - Which topics are discovered using Secure Discovery?
  - Which Topics have controlled access?
  - Encrypt or Sign for Secure Discovery?
  - Encrypt or Sign for each secure Topic?
  - What to do with unauthenticated access requests?

Configuring & Deploying DDS Security

Identity CA

Permissions CA

Participant2

Identity

Permissions

Participant1

Identity

Permissions

PrivateKey

PrivateKey

Governance

Participant3

Identity

Permissions

PrivateKey

Built-in Plugins: XML Governance Document

- Specifies how a domain should be secured

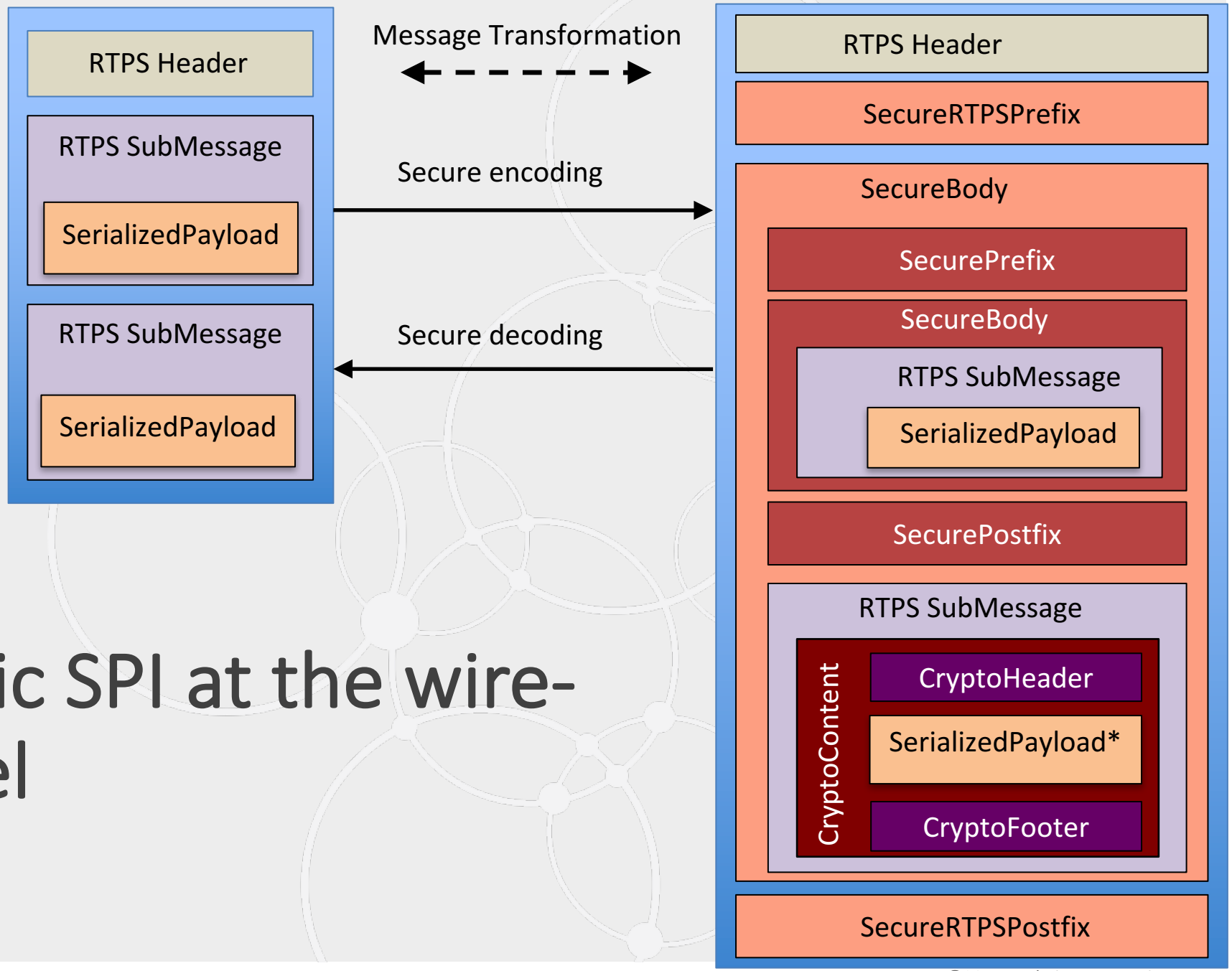
- Signed by the Permissions CA

- Provided to the plugins using the PropertyQosPolicy

on the DomainParticipantQos

```
<dds xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="http://www.omg.org/spec/DDS-SECURITY/...">
    <domain_access_rules>
        <domain_rule>
            <domains>
                <id>0</id>
                <id_range>
                    <min>10</min>
                    <max>20</max>
                </id_range>
            </domains>
            <allow_unauthenticated_participants>false</allow_unauthenticated_participants>
            <enable_join_access_control>true</enable_join_access_control>
            <discovery_protection_kind>ENCRYPT</discovery_protection_kind>
            <liveliness_protection_kind>SIGN</liveliness_protection_kind>
            <rtps_protection_kind>SIGN</rtps_protection_kind>
            <topic_access_rules>
                <topic_rule>
                    <topic_expression>Square*</topic_expression>
                    <enable_discovery_protection>true</enable_discovery_protection>
                    <enable_liveliness_protection>true</enable_liveliness_protection>
                    <enable_read_access_control>true</enable_read_access_control>
                    <enable_write_access_control>true</enable_write_access_control>
                    <metadata_protection_kind>SIGN</metadata_protection_kind>
                    <data_protection_kind>ENCRYPT</data_protection_kind>
                </topic_rule>
            </topic_access_rules>
        </domain_rule>
    </domain_access_rules>
</dds>
```

Cryptographic SPI at the wire-protocol level



Built-in Plugins: XML Permissions Document

- Contains the permissions of the Domain Participants

- Signed by the Permissions CA

- Provided to the plugins using the PropertyQosPolicy on the DomainParticipantQos

```
<dds xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="http://www.omg.org/spec/DDS-SECURITY/...">
    <permissions>
        <grant name="ShapesPermission">
            <subject_name>emailAddress=cto@acme.com, CN=DDS Shapes Demo, OU=CTO Office,
                          O=ACME Inc., L=Sunnyvale, ST=CA, C=US</subject_name>
            <validity>
                <!-- Format is CCYY-MM-DDThh:mm:ss[Z|(+|-)hh:mm]
                     The time zone may be specified as Z (UTC) or (+|-)hh:mm.
                     Time zones that aren't specified are considered UTC. -->
                <not_before>2013-10-26T00:00:00</not_before>
                <not_after>2018-10-26T22:45:30</not_after>
            </validity>

            <allow_rule>
                <domains>
                    <id>0</id>
                </domains>
                <publish>
                    <topics>
                        <topic>Cir*</topic>
                    </topics>
                </publish>
                <subscribe>
                    <topics>
                        <topic>Square</topic>
                    </topics>
                </subscribe>
            <default>DENY</default>
        </grant>
    </permissions>
</dds>
```

Configuration Possibilities

- Are “legacy” or un-identified applications allowed in the Domain?
  - Yes (if configured) unauthenticated applications will:
    - See the “unsecured” discovery Topics
    - Be allowed to read/write the “unsecured” Topics
- Is a particular Topic discovered over protected discovery?
  - If so it can only be seen by “authenticated applications”

Configuration Possibilities

- Is the access to a particular Topic protected?
  - If so only authenticated applications with the correct permissions can read/write
- Is data on a particular Topic protected? How?
  - If so data will be sent signed or encrypted+signed
- Are all protocol messages signed? Encrypted?
  - If so only authenticated applications with right permissions will see anything

Using a Trusted Platform Module (TPM)

- TPM can be used with Connext DDS Secure.
- Usage depends on specific provider
- RTI has demo working with a 3rd party library (Mocana) that is "API" compatible with OpenSSL and interfaces with the TPM
  - Approach is TPM-vendor independent
  - Uses TCG standard for communication with the TPM
  - Demo uses Infineon

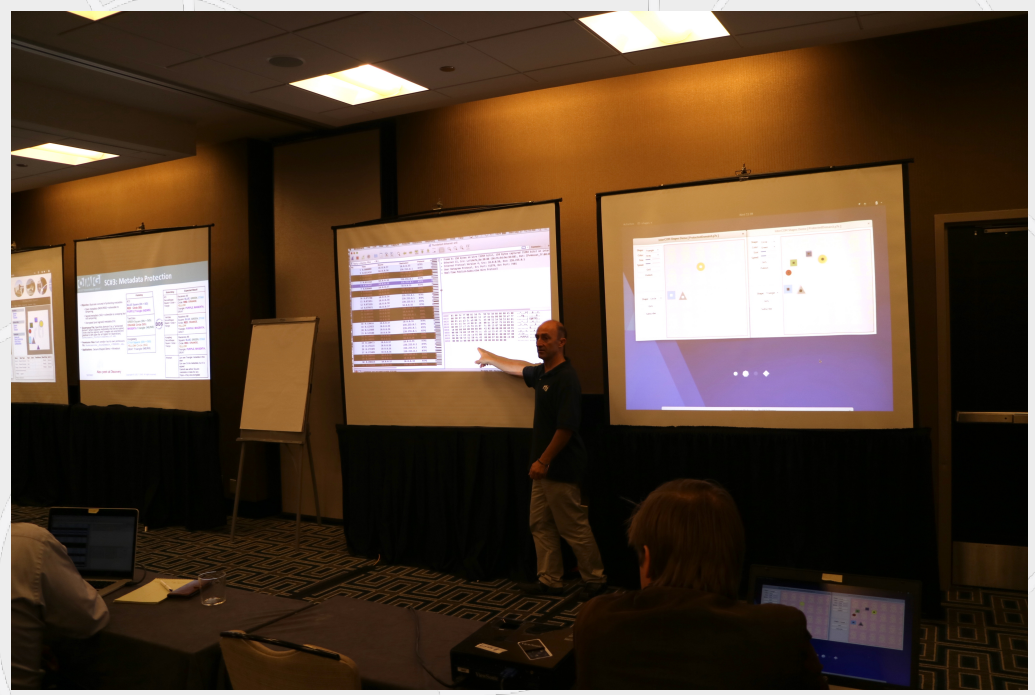
Key Benefits

More Powerful Than Other Secure Middleware Technologies

- Standard & Interoperable
- Scalable: Supports multicast
- Fine-grain: Control Topic-level aspect
- Flexible: Build your own plugins
- Generic: Works over any transport
- Transparent: No changes to Application Code!

DDS-Security Interoperability Demo

- 3 vendors (RTI, TwinOaks, Kogsberg)
- 7 Scenarios (see presentation)
- Code available on GitHub (https://github.com/omg-dds/)



Secure Services and Tools

Integration with Persistence Service, Routing Service, Queuing Service, …

Secure Persistence Service

- Needs “read/write” permissions to the persisted topics
- Uses Governance file to determine how each Topic is protected.
- Uses its own per-Writer Key material
  - Stores WriterKey material in the database (encrypted)
  - Stores data in encrypted form
  - Replays data encrypted with WriterKey material
- Requires “-password” command-line to execute

Secure Routing Service

- Has 2 DomainParticipants hence:
  - 2 Identities, 2 Governance, 2 Permission files
  - Needs “read” permissions on the “Input” participant for the routed topics
  - Needs “write” permissions on the “Output” participant for the routed topics
- Output data protected according to Governance on output domain
- If Durable Writer History then
  - Stores data encrypted
  - Stores WriterKeys (encrypted) along with durable data

Secure(*) Queuing Service

- Has 1 DomainParticipant
  - Configured with Identity, Governance, Permissions…
  - Needs read permissions to the input (queued) topics e.g. “MyQueueTopicName”
  - Needs write permissions to the output topics, e.g. “MyQueueTopicName@MySharedSubscriberName”
  - Output protected according to governance for “*@MySharedSubscriberName”
- Queue producer need write permissions to “MyQueueTopicName”
- Queue consumers need read permissions to “MyQueueTopicName@MySharedSubscriberName”
- Data stored unencrypted

Secure Recording & Replay Service

- Has 1 DomainParticipant
  - Configured with Identity, Governance, Permissions
  - Needs read permissions to recorded topics
  - Needs write permissions to replay topics
  - Can store data data different ways
    - File Encryption (after recording stops it encrypts)
    - User Data can per Topic can choose:
      - NONE, Data, Data+Metadata encryption
    - Discovery data per buitin Topic can choose:
      - NONE, Data, Data+Metadata encryption

Cloud Discovery Service

- Only used to bootstrap
- Works with DDS Security without special configurations
- Can use secure transport e.g. (D)TLS

Secure Web Integration Service

- Has DomainParticipant on DDS side
  - Configured with Identity, Governance, Permissions
- Uses HTTPS on the web-client side
  - Clients identified by a Client-API-Key
  - Only clients with valid Client-API-Key can connect
    - All clients can access the Topics and Domains that have been granted DDS permissions

Secure Database Integration Service

- Has one DomainParticipant on DDS side
  - Configured with Identity, Governance, Permissions
  - Needs read permissions to topics stored
  - Needs write permissions to topics monitored
- Decrypts DDS data before storing. The database itself may provide its own encryption if so configured.

Tools

- Can participate in secure domains
- Need Identity, Governance, Permissions to join secure DDS domain
  - Need read permissions for user Topics to display data
- Admin Console
  - Via UI, single security identity and set of configuration for all secure domains it joins
    - Governance and Permissions can vary per Domain
  - Via XML, there is the option of using different a security configuration (identity, permissions) per domain ID
- Monitor UI, Ping, Spy
  - Via XML can specify a configuration for each domain ID

Built-in Monitoring and Administration Topics

- DDS Core Monitoring Topics
- Distributed Logging: 2 Topics
- Service Monitoring and Administration Topics (see respective user's manuals)
  - Routing Service Monitoring & Admin
  - Recording Service Admin
  - Persistence Service Admin
  - Queuing Service Monitoring & Admin

Can be individually secured using XML configuration

Secure DDS Toolkit for LabVIEW

- DDS Toolkit 2.0.0.105 currently supported with Connext DDS 5.2.7 and LabVIEW 2015 SP 1 and later
- Configuration per DomainParticipant by using XML or in-memory Profile (created using security Panel or security subVI)
  - Separate identities and authorities for all DomainParticipants
  - Separate governance and permissions specific to each DomainId
- Needs read/write permissions according to the Topics used in each domain

Prototyper and Connector

- Uses XML application Creation to configure DomainParticipants
- Security configuration per DomainParticipant
  - Authorities, Identity, Governance, Permissions can vary per DomainParticipant

Some examples

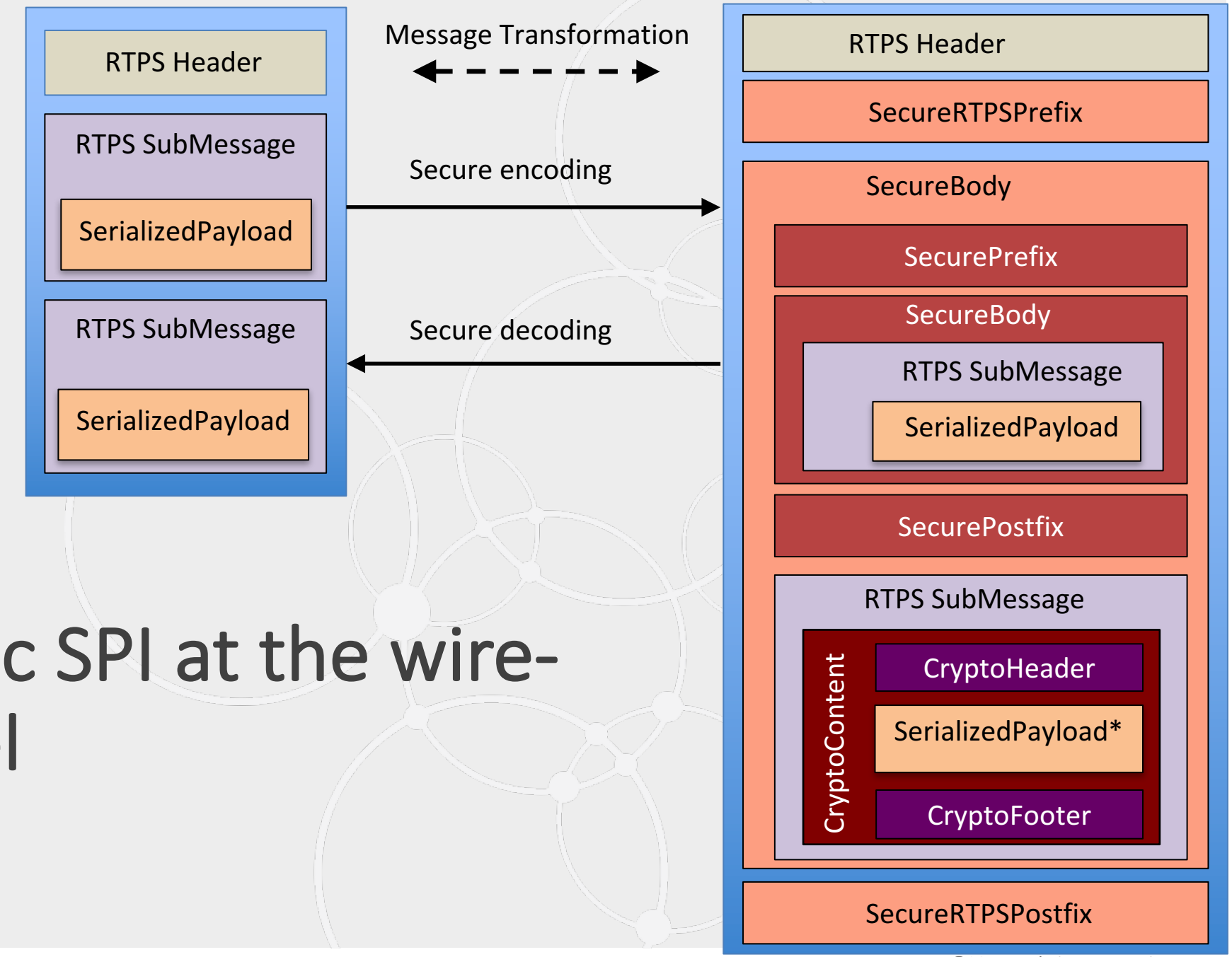
Attack scenarios

1. Unauth Pub using DDS
2. Attack on Data (network tamper)
3. Attack on Meta-Data (network tamper)
4. Unauth Subs with DDS app
5. Unauth Network Data Snooping (using Wireshark)
6. Unauth Network Meta Data Snooping (using Wireshark)
7. Discovery snooping

Protecting against the attacks

1. Unauth Pub using DDS
   - Require write permissions
2. Attack on Data (network tamper tcpwrite)
   - Require message signing (DATA, Submessage, RTPS)
3. Attack on Meta-Data (network tamper tcpwrite)
   - Require message signing (Submessage, RTPS)
4. Unauth Subs with DDS app
   - Require read permissions
5. Unauth Network Data Snooping (using Wireshark)
   - Encrypt (Data or Submessage)
6. Unauth Network Meta Data Snooping (using Wireshark)
   - Encrypt (Submessage)
7. Discovery snooping
   - Encrypt discovery

Cryptographic SPI at the wire-protocol level



Non Security

Unauth Subscription & Publication Using DDS

- Start ShapesDemo Publisher (OpenDomain::NonSecure) and publish BLUE Square
- Start ShapesDemo Subscriber (OpenDomain::NonSecure) and subscribe Square
- ***[Subcriber receives data]***
- Start RTI DDS SPY
- ***[Spy receives data]***
- Start ShapeType_publisher to publish some “bad data” (random data)
- ***[Subscriber receives “bad data”]***
- Stop ShapeType_publisher
- Stop RTI DDS Spy

Security

Avoid Unauth Subscription Using DDS: AccessControl Enabled

- Restart ShapesDemo Publisher (AccCtrl::P4_PubSubAll_NoEnc) and publish BLUE Square
- Restart ShapesDemo Subscriber (AccCtrl::P4_OnlySubSquare_NoEnc) and subscribe Square
- ***[Subcriber receives data]***
- Publish RED Circles
- Subscribe to RED Circles
- ***[Subscriber fails to create DataReader]***
- Start RTI DDS SPY
- ***[Spy does not receive data]***
- Stop Spy

Avoid Unauth Publication Using DDS: AccessControl Enabled

- Start ShapeType_publisher to publish some “bad data”
- ***[Subscriber does not receive “bad data”]***

Attack on Data/Metadata with Wireshark

- Start Wireshark and capture some DATA RTPS messages
- Save a DATA RTPS message representing a BLUE Square
- Replay DATA RTPS message changing shape size
- ***[Attack does not work because of SN]***
- Replay DATA RTPS message changing shape size and sequence number
- ***[Attack succeeds]***
- ***[Subscriber does not get data from Publisher until target SN is reached]***

Avoid Attack on Data/Metadata: Message Integrity Protection

- Restart ShapesDemo Publisher (P1_PubSubAll_MAC) and publish BLUE Square
- Restart ShapesDemo Subscriber (P4_OnlySubSquare_MAC) and subscribe BLUE Square
- Start Wireshark and show Integrity protection
- ***[Data and metadata still visible]***
- Run SequenceNumber attack
- ***[Attack does not work because message is signed]***

Unauth Network Snooping

- Show RTPS DATA packets (including user payload and metadata)
- Show DISCOVERY packets (including user payload and metadata)

Avoid Unauth User Data Network Snooping: Data Encryption

- Restart ShapesDemo Publisher (P1_PubSubAll_EncryptData) and publish BLUE Square
- Restart ShapesDemo Subscriber (P4_OnlySubSquare_EncryptData) and subscribe BLUE Square
- Start Wireshark and capture some RTPS DATA messages
- ***[User Data is Encrypted]***
- ***[Meta Data is still visible]***

Avoid Unauth Meta Data Network Snooping: Meta Data Encryption

- Restart ShapesDemo Publisher (P1_PubSubAll_EncryptSubmsg) and publish BLUE Square
- Restart ShapesDemo Subscriber (P4_OnlySubSquare_EncryptSubmsg) and subscribe BLUE Square
- Start Wireshark and capture some RTPS DATA messages
- ***[User Data and Meta Data are Encrypted]***

Avoid Unauth Discovery Network Snooping: Discovery Encryption

- Stop ShapesDemo Publisher and ShapesDemo Subscriber
- Start Wireshark and capture RTPS traffic
- Restart ShapesDemo Publisher (P1_PubSubAll_EncryptDisc) and publish BLUE Square
- Restart ShapesDemo Subscriber (P4_OnlySubSquare_EncryptDisc) and subscribe BLUE Square
- ***[Endpoint Discovery Data is not available anymore Unencrypted]***