# Integrating Quantum Key Distribution with IEEE 802.1X

Manish Talwar
Nimit Gupta
Gert Grammel

Nov 1, 2022

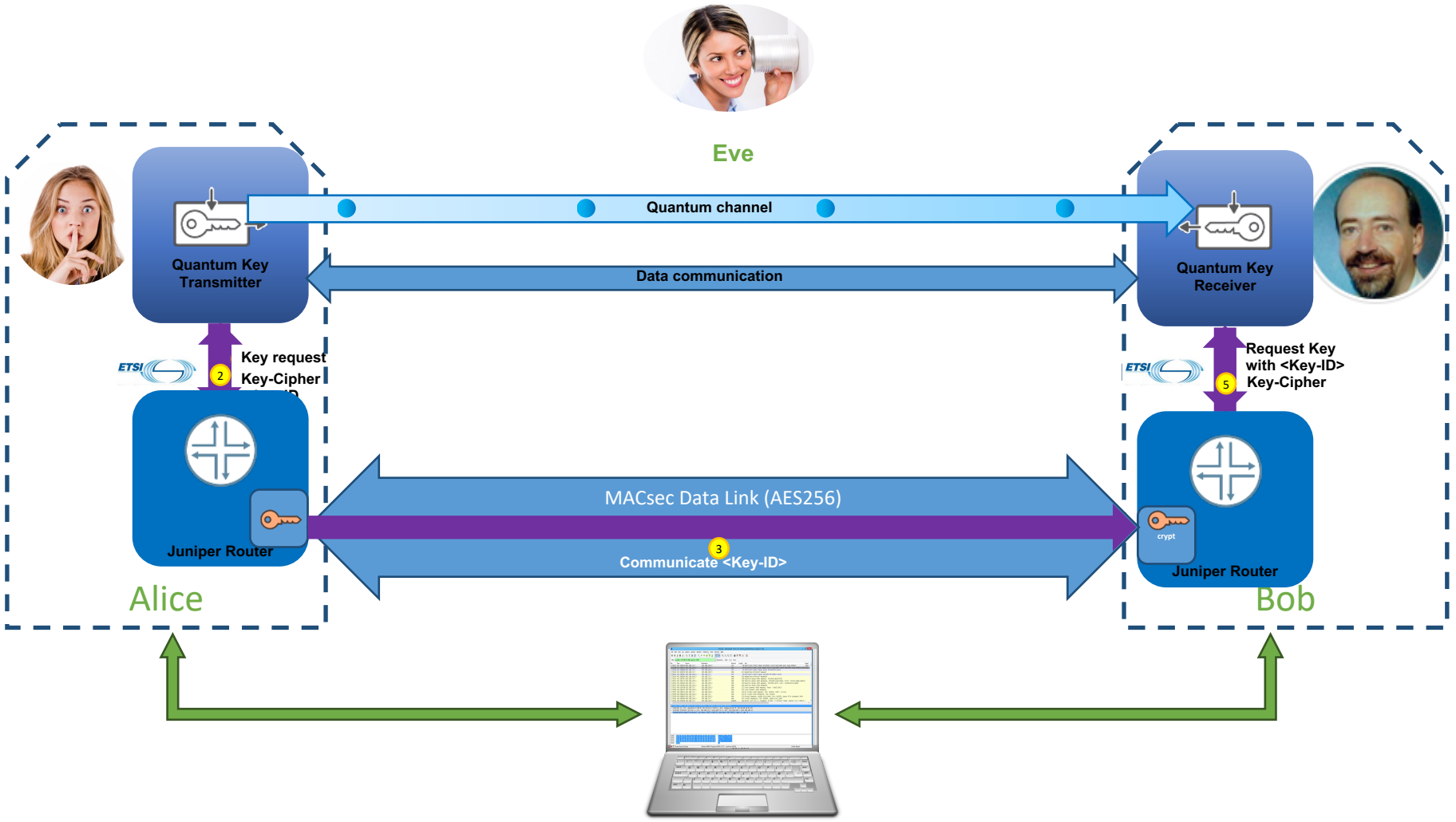
# Agenda

- Overview
- Background
- Problem Definition
- Proposed Solution
- Conclusion

# Overview

- A proposal to integrate QKD with MACsec Key Agreement Protocol.
- QKD utilizes the unique properties of quantum mechanical systems to generate and distribute cryptographic keying material.
- A Router obtains QKD key(s) from a QKD-server using REST APIs (using https).
- The peer Router needs quantum key-id and peer system-id to obtain the quantum key(s) from its QKD-server.
- Actual key is not transmitted on the wire and its security is guaranteed by quantum physics.
- Use MKA protocol to distribute the parameters needed to obtained quantum key-id from QKD-server.
- Solves the vulnerabilities in key management and makes MACsec quantum secure.

# QKD Key exchange



# Problem Definition

- MACsec guarantees confidentiality and integrity and relies on MKA for key-exchange.
- MKA Pre-shared keys were always vulnerable to human factors and using a PKI infrastructure for EAP-TLS authentication has been proved to be vulnerable by using quantum computers.
- QKD generates and distributes cryptographic keying material.
- First router needs peer system identifier (Secure Application Entity identifier) to get quantum key.
- Peer router needs quantum key-id and peer system identifier to get common QKD key.
- To provide end-to-end solution, QKD needs integration in MKA which would enable the peers on a link to obtain keys from QKD-server.

# Proposed Solution

- Use Key Management domain (KMD) to carry SAE-id(s) of the peer systems.
  - KMD length up to 253 UTF-8 characters
  - SAE-id string identifying the system

- Use CAK-Name (CKN) to carry quantum key-id(s)
  - CKN length up to 32 octets
  - Quantum key-id type is UUID

- SAE-id(s) are not secret and identify the peer system (MAC address, logical port).

- Quantum keys are not exchanged on wire.

- Only quantum key-id is sent which is not a secret.

# Proposed Solution …

Step 0: Each of the systems are configured to use quantum KME and one of the system is configured to be the key initiator.

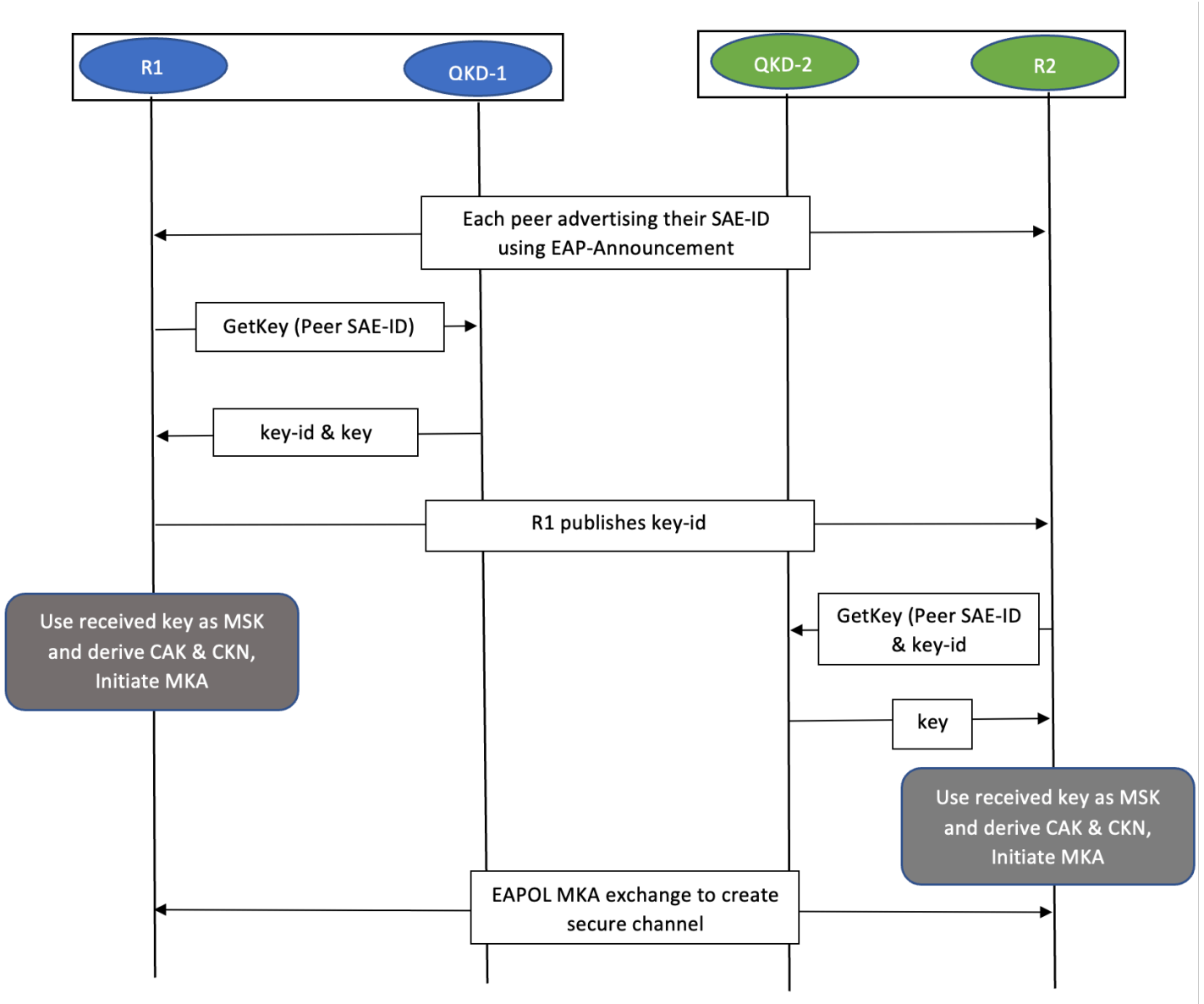
Step 1: Non-key initiator system sends its SAE-id (SAE-B) in KMD in EAPOL-Announcement to its peer.

Step 2: Peer KME(s) agree on a key using quantum principles. The other system (SAE-A) uses REST API calls to get one or more keys from its KME. SAE-A gets quantum key and quantum key-id from its KME.

Step 3: System SAE-A would send a MKPDU protected with one of the keys (integrity and data origin protected). This MKPDU will have its SAE-id (SAE-A) encoded in KMD (see 802.1X-2020: Table 11-7 Parameter set type 6) and quantum key-id in CKN.

Step 4: System SAE-B will use the received SAE-id (SAE-A) and quantum key-id to retrieve the key from its KME. SAE-B can compute ICV after it has received the quantum key-id (MSK).

# Proposed Solution – Message Flow for MKA-QKD



# Conclusion

- Integration of QKD in MKA provides end-to-end solution of secure key distribution and makes MACsec quantum secure.

- MKA integrates QKD seamlessly.

- Next Steps

- Juniper Contacts
  - Manish Talwar mtalwar@juniper.net
  - Nimit Gupta nimitg@juniper.net
  - Gert Grammel ggrammel@juniper.net

# Glossary

**MKA:** MACsec Key Agreement protocol
**EAP:** Extensible Authentication Protocol
**EAPOL:** EAP Over LAN
**QKD:** Quantum Key Distribution
**MACsec:** MAC level Security
**PKI:** Public Key Infrastructure
**KME**: Key Management Entity
**SAK:** Secure Association Key
**CAK:** Connectivity Association Key
**PSK:** Pre-Shared Key
**MSK:** Master Session Key
**CKN:** Connectivity Association Key Name
**KMD:** Key Management Domain
**EAP-TLS:** Extensible Authentication Protocol-Transport Layer Security
**SAE-ID**: Secure Application Entity identifier
**QKD key-id:** A unique identifier associated with a QKD-key

# References


1. IEEE MACsec
2. IEEE MKA
3. ETSI QKD