Safeguard Your Network in a Post-Quantum World

Rakesh Kandula, TME

Raja Kolagatla, Product Manager

November 15, 2023

Agenda

1. Post-quantum threat to security
2. Areas of impact
3. Transport security solutions
4. Product support and roadmap
5. Demo

Post-quantum threat to security

People are making incremental efforts in developing a quantum computer.

Once they have one sufficiently large and reliable, they could use it to break current encryption (public key algorithms).

Areas of impact

Scope of post-quantum threat

Firmware/software integrity

- Firmware and NOS image signing
- Secure Boot
- IMA* keys, software image posting, etc.

Identity

- Server certificates
- Individual identities
- Device identity (like SUDI**)
- SSH

Transport security

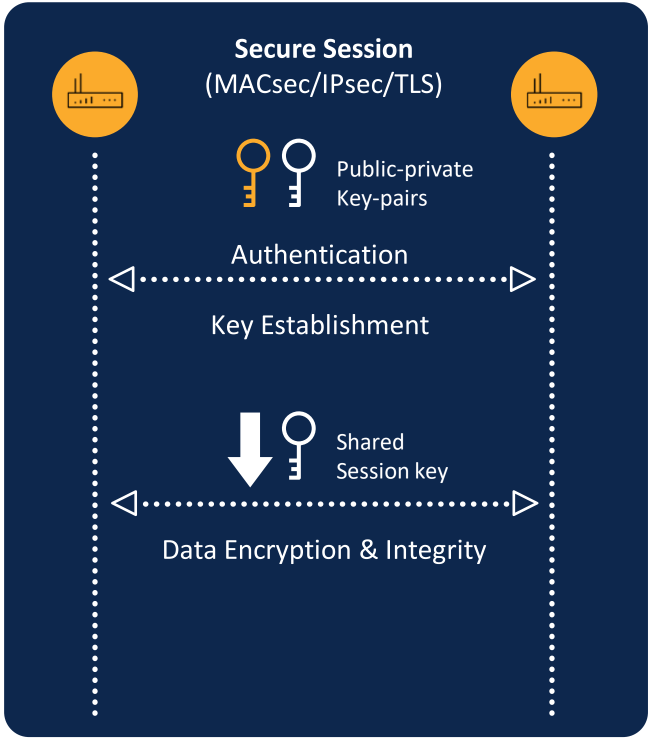
- MACsec
- IPsec
- TLS

*IMA – Integrity Measurement Architecture

**SUDI – Secure Unique Device Identifier

Transport security impact

Quantum computing impact on cryptography



Quantum-resistant?

Asymmetric cryptography

- Based on **mathematically related** public-private key-pairs
- Used for control plane operations
  - Authentication, key establishment
- Examples: RSA, DH, ECC

✗

Large, reliable quantum computers can break RSA, DH, ECC

Symmetric cryptography

- Based on shared key
- Used for bulk data encryption and integrity
- Protection level based on key strength
  - Key size and entropy
- Example: AES-GCM

✓

Symmetric crypto with large and high-entropy keys is resistant to quantum computer attacks

Why care about quantum threats now?

1. Attackers can tap flows today and store them to be decrypted in the future.

2. Any sensitive deployments that need forward secrecy for 5+ years must act now.

   - Military or other defense networks
   - Federal or other government agencies
   - Financial institutions and banks
   - Service provider networks catering to enterprises with sensitive data

3. Less critical or short-lived sessions without long-term significance can wait.

Transport security solutions

Available options

Symmetric cryptography

✓ Long symmetric keys are quantum-safe

⚠ Issues with distributing keys and trust

Quantum key distribution

✓ Use quantum mechanics to protect the data

⚠ Technology limitations

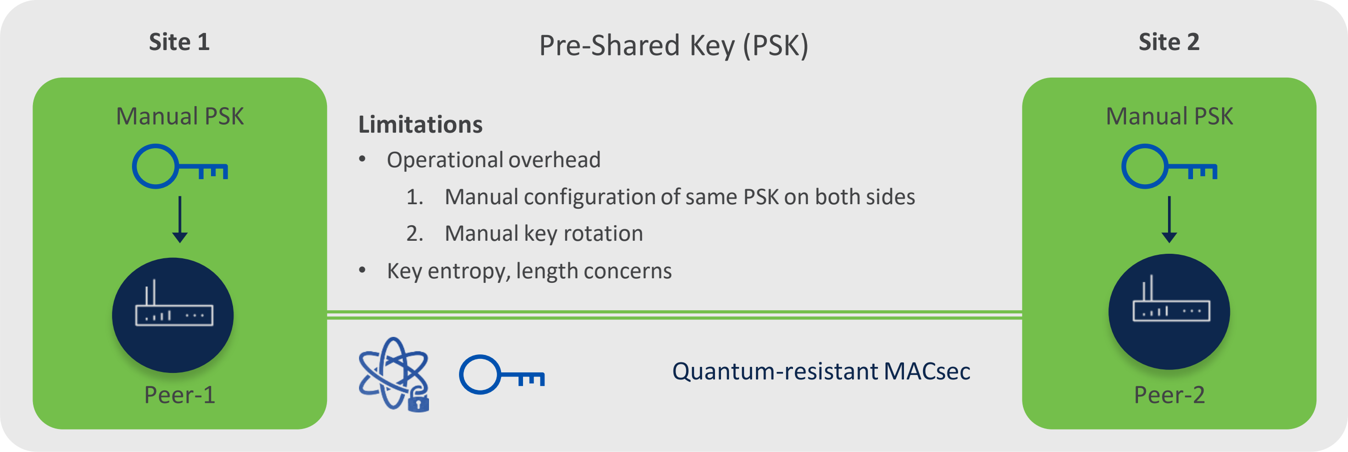
Post-quantum cryptography

✓ Replace current public key algorithms with new ones

⚠ Still need to vet the algorithms and update the protocols

Quantum-safe MACsec

Pre-shared key (PSK) option

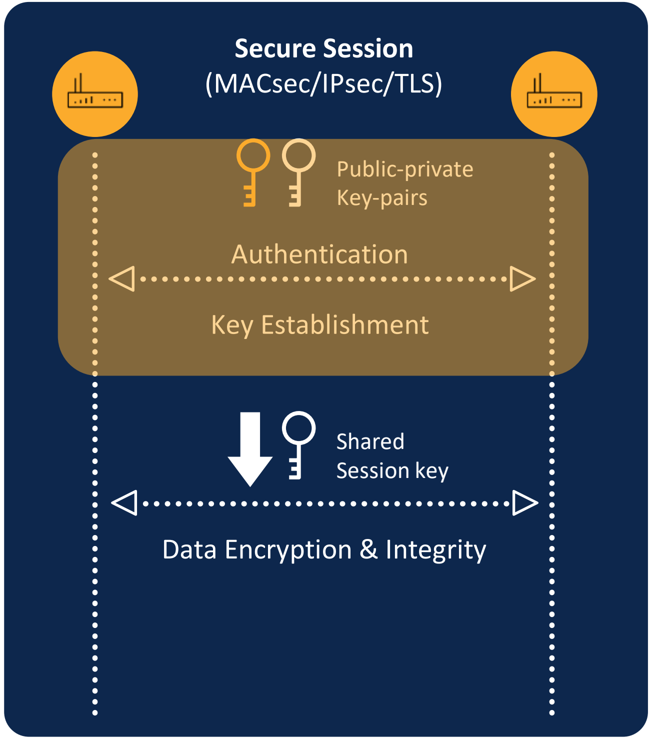


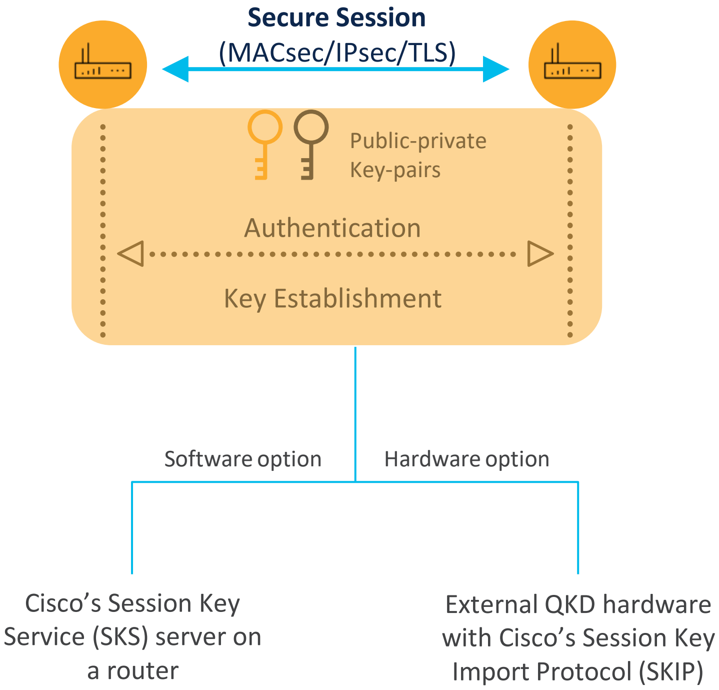
Limitations

- Operational overhead
  1. Manual configuration of same PSK on both sides
  2. Manual key rotation
- Key entropy, length concerns

1. MACsec with PSK option is already supported and used by customers.
2. There is no need for additional hardware (like Quantum Key Distribution - QKD) or software upgrade.
3. Quantum-safe as this is based on symmetric cryptography which is quantum-resistant.

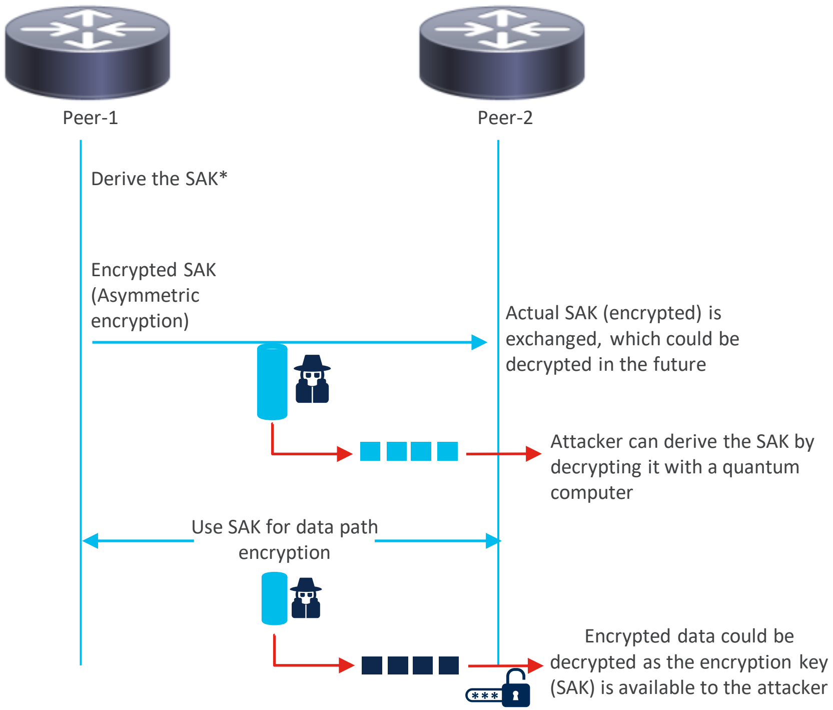
Quantum key distribution options




Secure Session
(MACsec/IPsec/TLS)
Public-private
Key-pairs
Authentication
Key Establishment
Software option
Hardware option
Cisco's Session Key Service (SKS) server on a router
External QKD hardware with Cisco's Session Key Import Protocol (SKIP)

Quantum key distribution – Basic principle

Existing method



Derive the SAK*

Encrypted SAK (Asymmetric encryption)

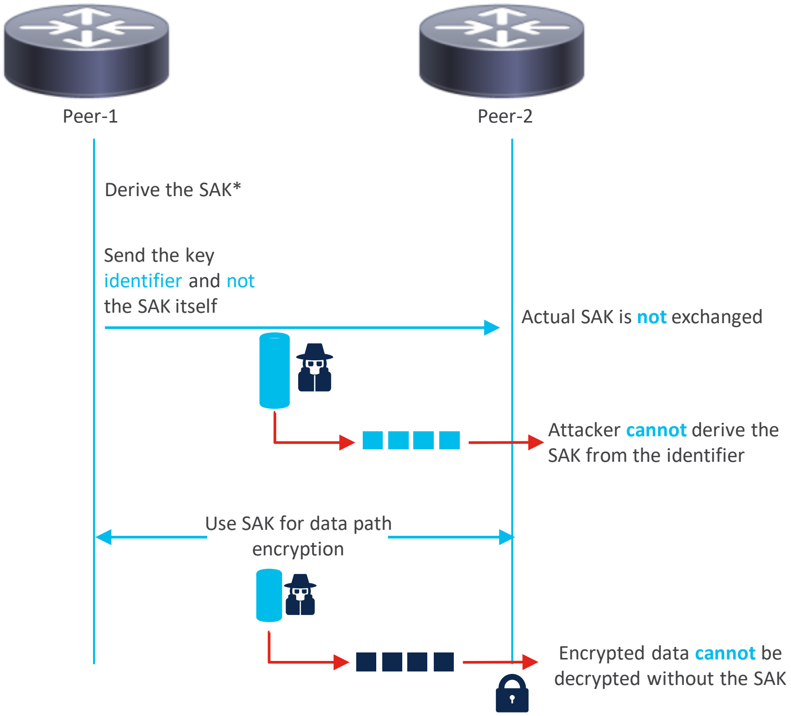
Actual SAK (encrypted) is exchanged, which could be decrypted in the future

Attacker can derive the SAK by decrypting it with a quantum computer

Use SAK for data path encryption

Encrypted data could be decrypted as the encryption key (SAK) is available to the attacker

QKD method



Derive the SAK*

Send the key identifier and not the SAK itself

Actual SAK is **not** exchanged

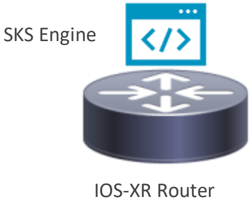
Attacker **cannot** derive the SAK from the identifier

Use SAK for data path encryption

Encrypted data **cannot** be decrypted without the SAK

*Security Association Key (SAK)

Cisco Session Key Service overview



1. SKS engine on the router generates the keys.

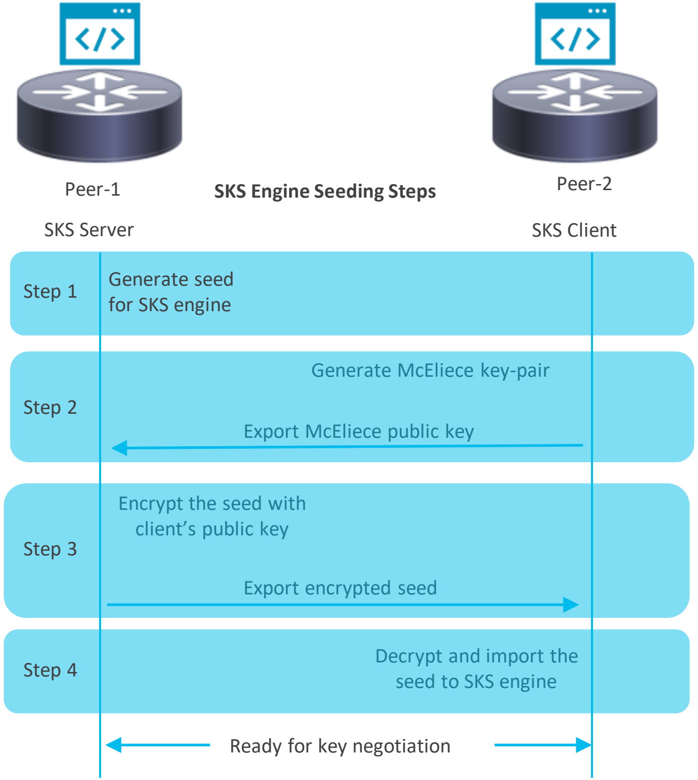
2. No additional hardware required.

3. The SKS engine must be seeded with the same seed on both the peers.

4. The seed is protected by McEliece cryptosystem which is quantum-resistant.

5. Only key-id is sent on the wire, and the peer derives the key from its local SKS engine.

Cisco Session Key Service workflow



Peer-1

SKS Server

SKS Engine Seeding Steps

Peer-2

SKS Client

Step 1

Generate seed
for SKS engine

Step 2

Generate McEliece key-pair

Export McEliece public key

Step 3

Encrypt the seed with
client’s public key

Export encrypted seed

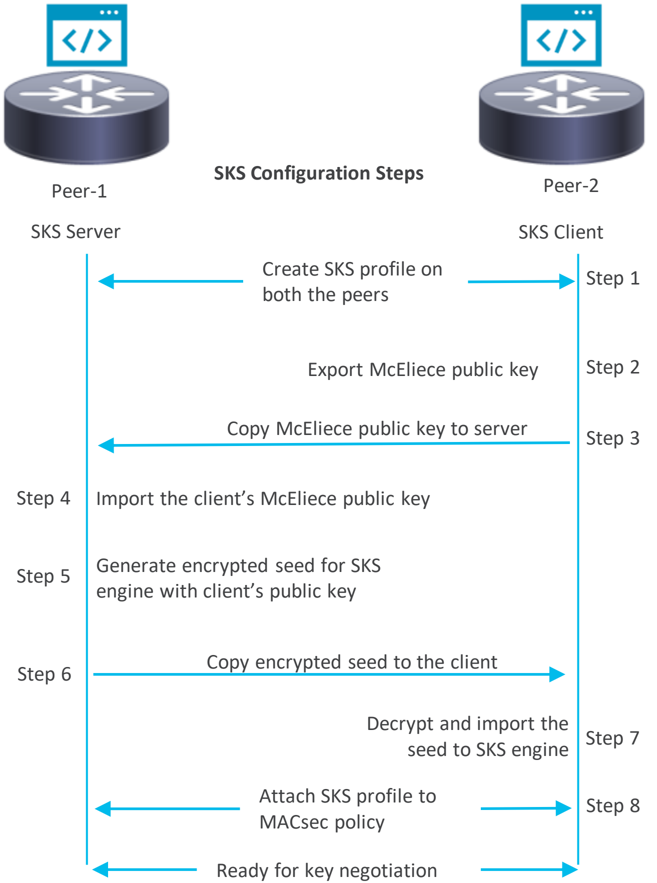
Step 4

Decrypt and import the
seed to SKS engine

Ready for key negotiation

Detailed configuration steps

Session Key Service configuration steps



SKS Configuration Steps

Peer-1

SKS Server

Peer-2

SKS Client

Step 1 Create SKS profile on both the peers

Step 2 Export McEliece public key

Step 3 Copy McEliece public key to server

Step 4 Import the client's McEliece public key

Step 5 Generate encrypted seed for SKS engine with client's public key

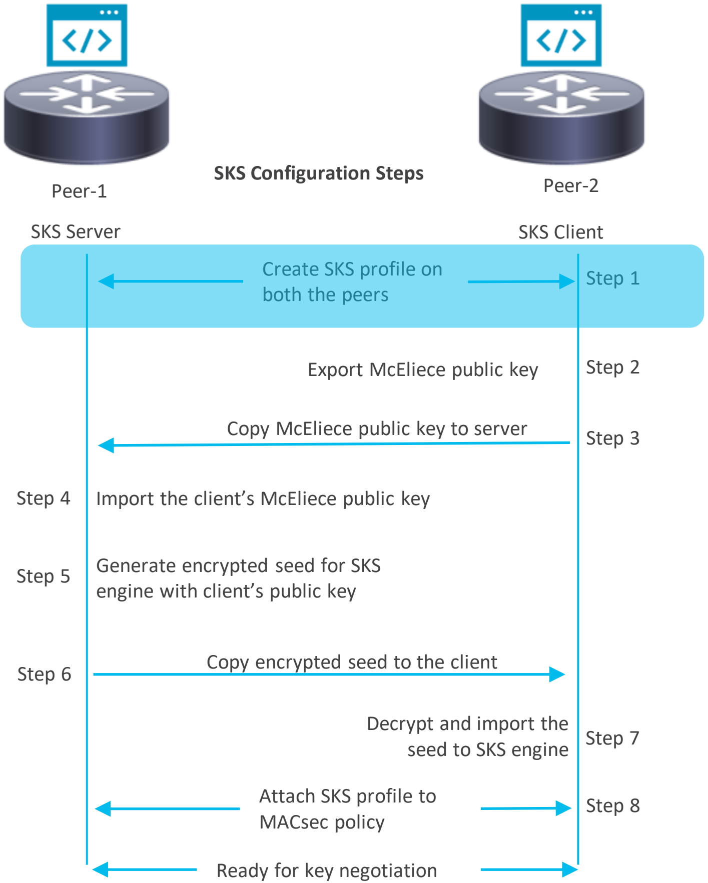
Step 6 Copy encrypted seed to the client

Step 7 Decrypt and import the seed to SKS engine

Step 8 Attach SKS profile to MACsec policy

Ready for key negotiation

Session Key Service configuration steps



SKS Configuration Steps

Peer-1

SKS Server

Peer-2

SKS Client

Step 1 Create SKS profile on both the peers

Step 2 Export McEliece public key

Step 3 Copy McEliece public key to server

Step 4 Import the client’s McEliece public key

Step 5 Generate encrypted seed for SKS engine with client’s public key

Step 6 Copy encrypted seed to the client

Step 7 Decrypt and import the seed to SKS engine

Step 8 Attach SKS profile to MACsec policy

Ready for key negotiation

Configuration steps – SKS profile

Configure SKS profile on both the peers

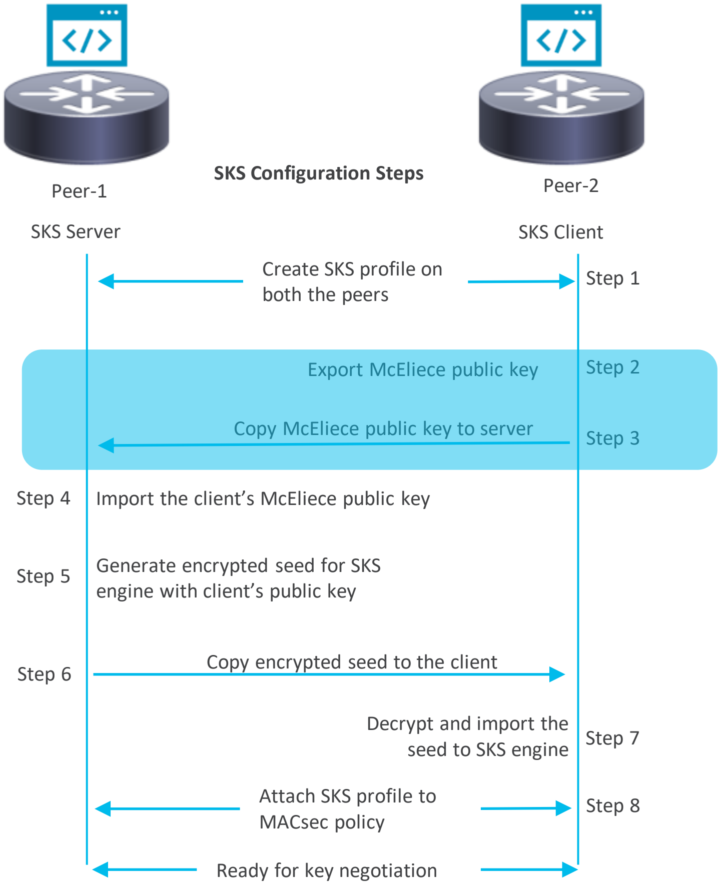
Peer-1 (Server)

```
sks profile prof-A device-identifier peer-1
        live-keys XX (Based on number of macsec sessions)
        peer-identifier peer-2
```

Peer-2 (Client)

```
sks profile prof-B device-identifier peer-2
       live-keys XX (Based on number of macsec sessions)
       peer-identifier peer-1 master
```

Session Key Service configuration steps



SKS Configuration Steps

Peer-1

SKS Server

Peer-2

SKS Client

Step 1 Create SKS profile on both the peers

Step 2 Export McEliece public key

Step 3 Copy McEliece public key to server

Step 4 Import the client’s McEliece public key

Step 5 Generate encrypted seed for SKS engine with client’s public key

Step 6 Copy encrypted seed to the client

Step 7 Decrypt and import the seed to SKS engine

Step 8 Attach SKS profile to MACsec policy

Ready for key negotiation

Configuration – McEliece key support

Export default McEliece public key using

Peer-2 (Client)

```
    crypto sks key export mceliece
    Path (default) : /disk0\:/MeCe_the_MC_default_pub
```

Copy McEliece public key to the server

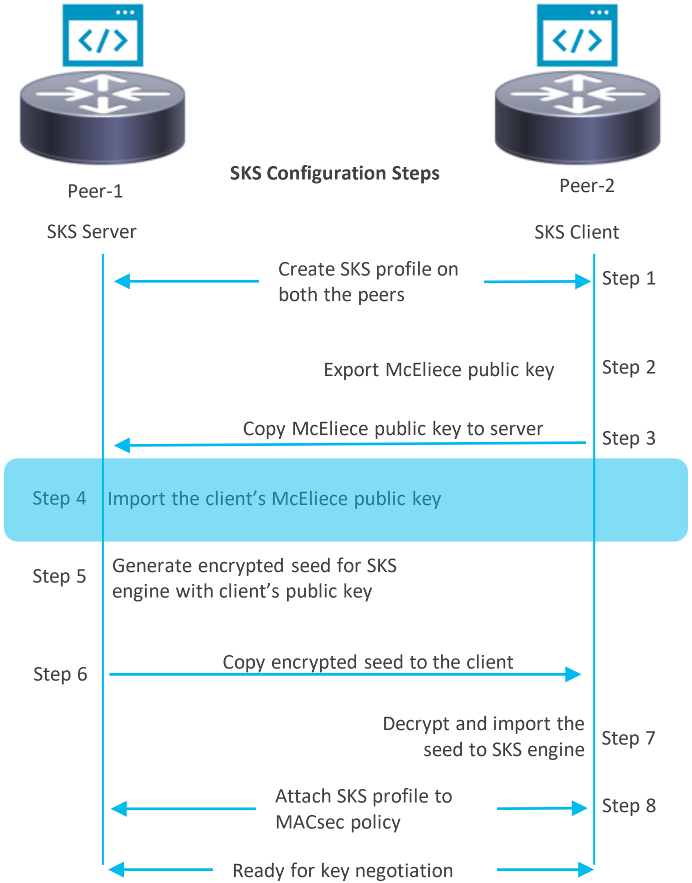
Peer-2 (Client)

```
scp /disk0\:/MeCe_the_MC_default_pub cisco@1.2.42.3:/disk0:/
```

Source path on the client

Destination path of server

Session Key Service configuration steps



SKS Configuration Steps

Peer-1

SKS Server

Peer-2

SKS Client

Step 1 Create SKS profile on both the peers

Step 2 Export McEliece public key

Step 3 Copy McEliece public key to server

Step 4 Import the client’s McEliece public key

Step 5 Generate encrypted seed for SKS engine with client’s public key

Step 6 Copy encrypted seed to the client

Step 7 Decrypt and import the seed to SKS engine

Step 8 Attach SKS profile to MACsec policy

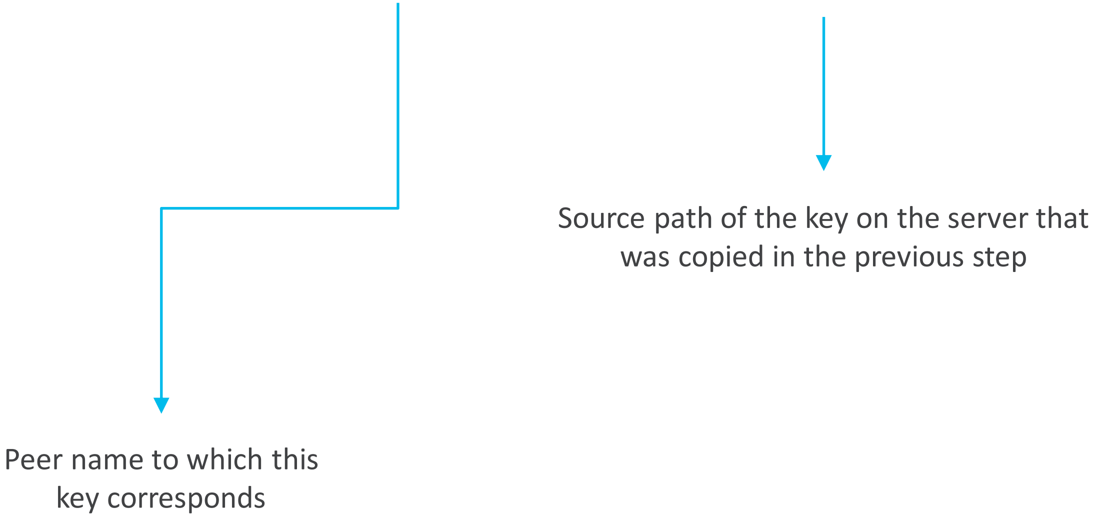
Ready for key negotiation

Configuration – McEliece key import

Import McEliece public key on the server

Peer-1 (Server)

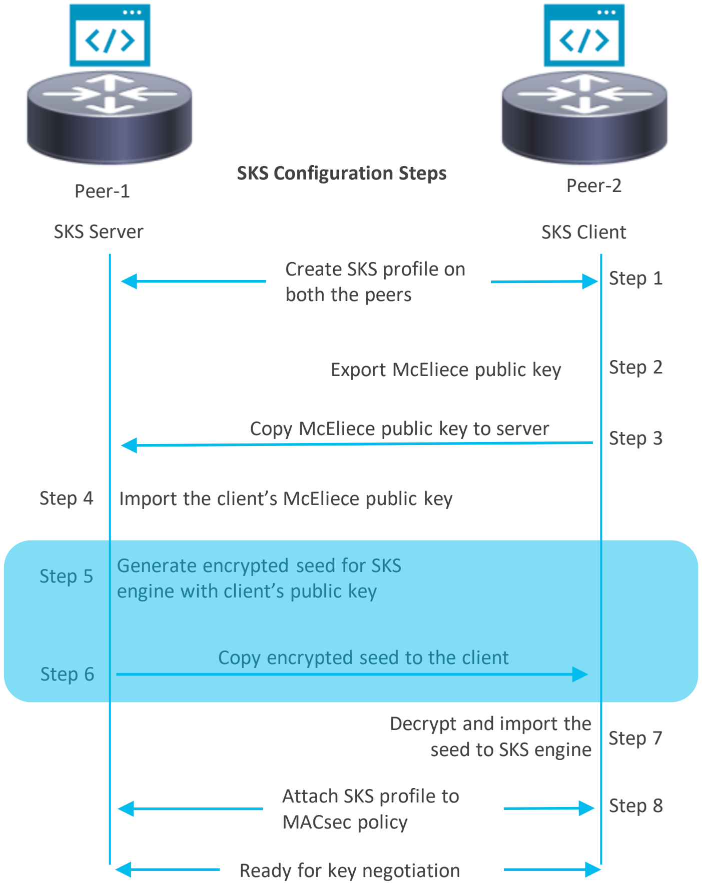
crypto sks key import mceliece peer-2 disk0:/MeCe_the_MC_default_pub



Source path of the key on the server that
was copied in the previous step

Peer name to which this
key corresponds

Session Key Service configuration steps



SKS Configuration Steps

Peer-1

SKS Server

Peer-2

SKS Client

Step 1 Create SKS profile on both the peers

Step 2 Export McEliece public key

Step 3 Copy McEliece public key to server

Step 4 Import the client’s McEliece public key

Step 5 Generate encrypted seed for SKS engine with client’s public key

Step 6 Copy encrypted seed to the client

Step 7 Decrypt and import the seed to SKS engine

Step 8 Attach SKS profile to MACsec policy

Ready for key negotiation

Configuration – Generate and export the seed

Generate and export the seed on the master

Peer-1 (Server)

```
crypto sks seed export mceliece peer-2
```

Generates an encrypted seed for Peer-2. The seed is encrypted with the McEliece public key of Peer-2 that was imported in previous steps.

Peer name to which this seed corresponds

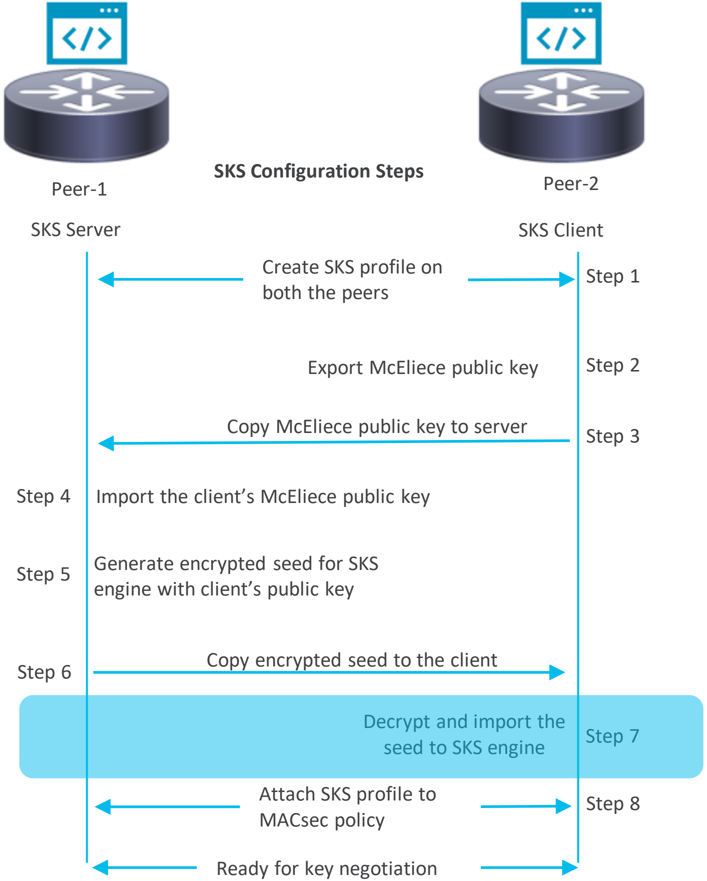
End result - An encrypted seed will be exported at path /disk0\:/enc_self_peer-2 in the master

Copy the seed to the Client

Peer-1 (Server)

```
scp /disk0\:/enc_self_peer-2 cisco@1.2.43.3:/disk0:/
```

Session Key Service configuration steps



SKS Configuration Steps

Peer-1

SKS Server

Peer-2

SKS Client

Step 1 — Create SKS profile on both the peers

Step 2 — Export McEliece public key

Step 3 — Copy McEliece public key to server

Step 4 — Import the client's McEliece public key

Step 5 — Generate encrypted seed for SKS engine with client's public key

Step 6 — Copy encrypted seed to the client

Step 7 — Decrypt and import the seed to SKS engine

Step 8 — Attach SKS profile to MACsec policy

Ready for key negotiation

Configuration – Copy and import the seed on client

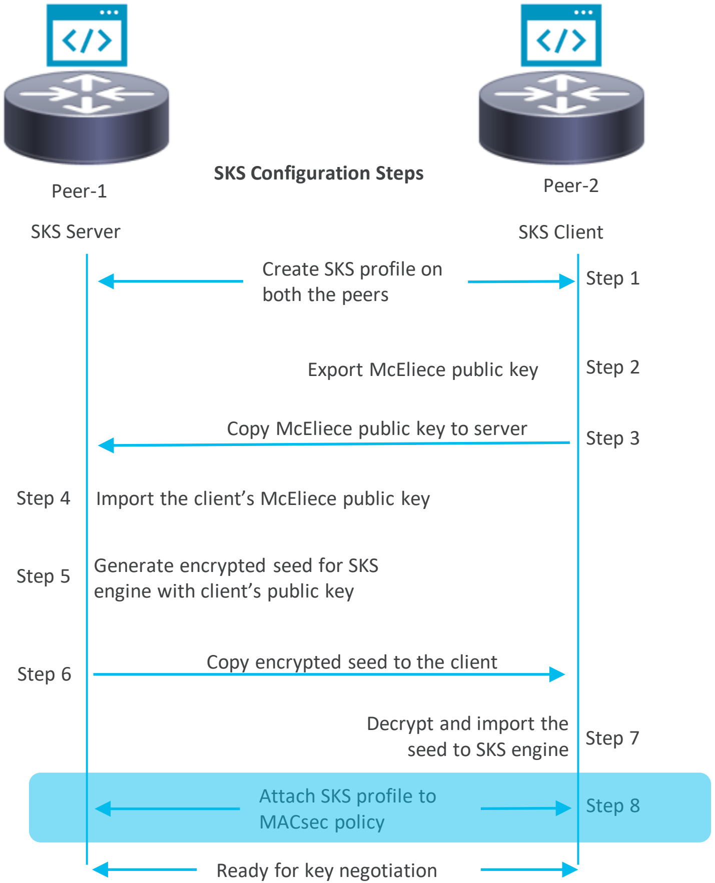
Import the seed on the Client

Peer-2 (Client)

```
crypto sks seed import mceliece peer-1 disk0:/enc_self_peer-2
```

Peer server’s name that
generated this seed

Session Key Service configuration steps



SKS Configuration Steps

Peer-1

SKS Server

Peer-2

SKS Client

Create SKS profile on both the peers — Step 1

Export McEliece public key — Step 2

Copy McEliece public key to server — Step 3

Step 4 — Import the client’s McEliece public key

Step 5 — Generate encrypted seed for SKS engine with client’s public key

Step 6 — Copy encrypted seed to the client

Decrypt and import the seed to SKS engine — Step 7

Attach SKS profile to MACsec policy — Step 8

Ready for key negotiation

Configuration – MACsec profile

Attach the SKS profile on Peer-1

Peer-1 (Server)

```
 macsec-policy p1
            ppk
                 sks-profile prof-A
```

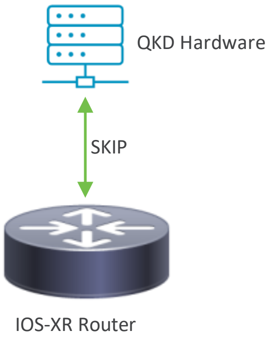
Attach the SKS profile on Peer-2

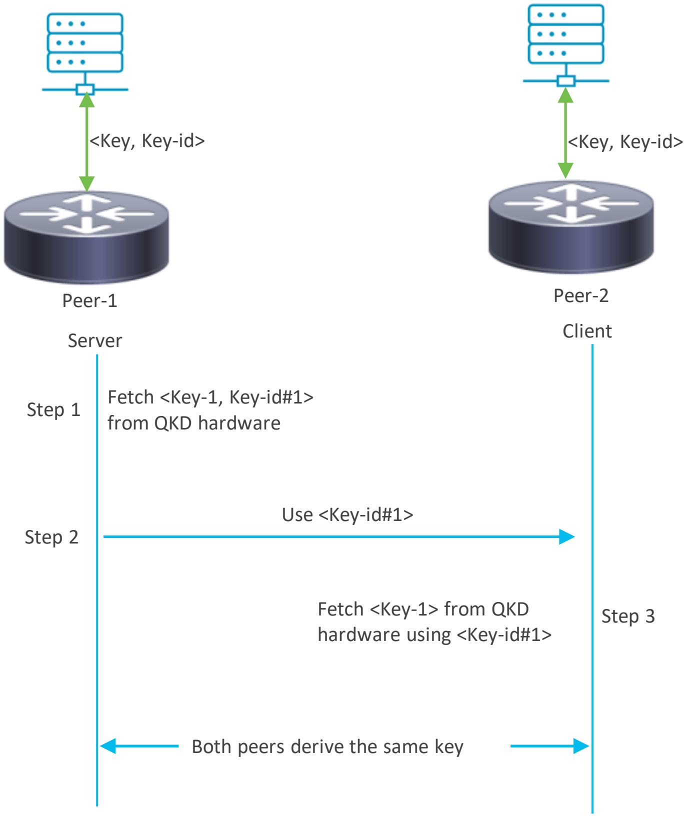
Peer-2 (Client)

```
 macsec-policy p2
            ppk
                 sks-profile prof-B
```

Quantum key distribution options

External QKD hardware with Cisco’s Session Key Import Protocol (SKIP)



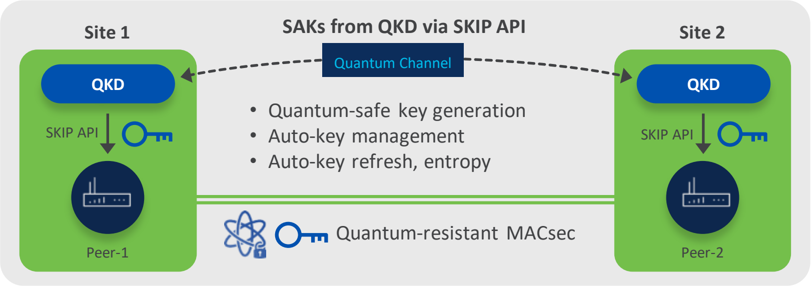

<Key, Key-id>
<Key, Key-id>
Peer-1
Server
Peer-2
Client
Step 1
Fetch <Key-1, Key-id#1> from QKD hardware
Step 2
Use <Key-id#1>
Fetch <Key-1> from QKD hardware using <Key-id#1>
Step 3
Both peers derive the same key

1. Dedicated hardware to generate the session keys and key-id’s
2. The QKD hardware for a given pair of devices would be in sync
3. Each peer fetches the key and key-id from the QKD hardware over a TLS connection
4. Only key-id is sent on the wire, and the peer fetches the key from the QKD hardware

Quantum-safe MACsec

Quantum key distribution options

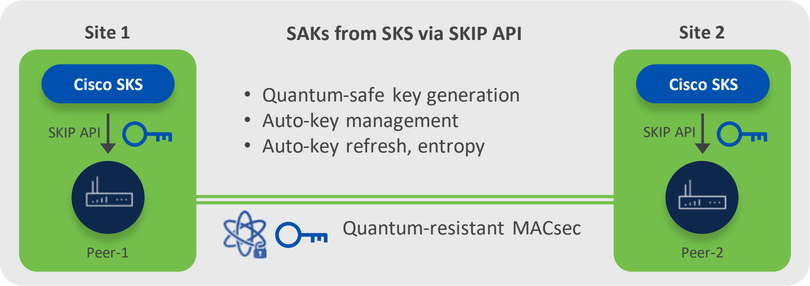
External QKD hardware



- Quantum-safe key generation
- Auto-key management
- Auto-key refresh, entropy

1. Hardware-based key source
2. Dedicated optical fiber (up to 100 km supported)
3. QKD hardware per-site/peer
4. Very expensive
5. Supported from IOS-XR 7.9.1 release

Cisco SKS server



- Quantum-safe key generation
- Auto-key management
- Auto-key refresh, entropy

1. Software-based key source
2. No dedicated circuit or distance limitations
3. No additional hardware requirement
4. No additional cost
5. Supported from IOS-XR 7.4.1 release

Product support and roadmap

SKS and SKIP support matrix

Platform	Release	SKS support	SKIP support
Cisco 8000 Series Router	7.4.1	Yes	No
Cisco 8000 Series Router	7.9.1	Yes	Yes
NCS 5700	7.9.1	Yes	Yes
NCS 5500	7.9.1	Yes	Yes
NCS 540 (N540-ACC-SYS, N540X-ACC-SYS, N540-24Z8Q2C-SYS)	7.9.1	Yes	Yes
NCS 540 (all other variants)	7.10.1	Yes	Yes
ASR 9000	7.10.1	Yes	Yes

Demo

Q&A

The bridge to possible