

Post Quantum Cryptography for SSHv2 sessions

Table 1: Feature History Table

Feature name	Release information	Feature description
Post Quantum Cryptography for SSHv2 sessions	Release 26.2.1	Cisco IoT routers can combine the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) with classical elliptic-curve key-exchange algorithms to establish keys for SSHv2 sessions.

- Implementation of Post Quantum Cryptography, on page 1
- Overview, on page 2
- Factors to consider for deploying PQC, on page 2
- Protocol requirements, on page 2
- Restrictions for PQC, on page 2
- PQC negotiation scenarios , on page 3
- Configure PQC for SSHv2 sessions, on page 3
- Verify PQC for SSHv2 sessions, on page 4
- Key elements of PQC, on page 4
- Supported platforms, on page 6

Implementation of Post Quantum Cryptography

Cisco utilizes a hybrid approach to implement Post Quantum Cryptography (PQC). The hybrid approach to PQC combines a quantum-resistant algorithm, such as the module-lattice based ML-KEM (e.g., mlkem768), with a classical elliptic curve key exchange algorithm like NIST P-256 or X25519, along with a hash function such as SHA-256.

This combination provides enhanced security by protecting against both classical and quantum attacks while maintaining compatibility and performance with existing cryptographic protocols.

Cisco implements this hybrid approach in protocols like SSH allowing configurations that enforce the use of ML-KEM algorithms for quantum resistance without falling back to classical-only key exchanges.

Overview

PQC refers to cryptographic algorithms designed to be secure against attacks from quantum computers which can break many classical encryption methods. Current widely used encryption methods—such as Diffie-Hellman (DH), and Elliptic Curve Diffie-Hellman (ECDH) — are vulnerable to being broken as quantum computing technology advances.

Implementing quantum-safe encryption now is critical for the long-term confidentiality and integrity of sensitive communications, particularly for government, military, and financial sectors.

Factors to consider for deploying PQC

Review these factors before deploying PQC.

Backward compatibility with classical algorithms
If a device is only configured for PQC and the SSH client does not support PQC, the SSH session will fail and no connection will be established.
Impact on computational time
The key size in PQC implementations may have an impact on the computational time due to the mathematical complexity and processing requirements of quantum-resistant algorithms. Larger key sizes generally increase the computational workload. PQC implementations must consider hardware acceleration and software optimizations to mitigate the increased computational time due to larger key sizes.

Protocol requirements

Protocol requirements

SSHv2 :

To establish secure, encrypted, and remote command-line access to servers and network devices over unsecured network use SSH version 2.0. PQC cannot be implemented on SSHv1 due to severe security vulnerabilities and it does not support the modern, complex key exchange mechanisms required to resist quantum computer attacks.

Key Pairs :

Ensure that RSA key pair or EC key pair for the SSH server is generated using crypto key generate command.

Restrictions for PQC

Implementing PQC is not supported on devices that operate in Federal Information Processing Standards (FIPS) mode.

PQC negotiation scenarios

During the SSH protocol connection setup, the client and server negotiate which cryptographic algorithms to use for securing the session. This negotiation process is a basic example of how algorithm negotiation works in SSH and is applicable to any similar negotiation process.

The algorithms are considered in the order they are configured on the device, meaning the priority or preference of algorithms is determined by the sequence in which they are listed in the configuration. The negotiation proceeds by trying the algorithms in that configured order until a mutually supported algorithm is found and agreed upon for use in the session. The server must support at least one algorithm from the client's list for successful negotiation.

Table 2: sfds

Client configuration	Server configuration	Negotiated algorithm
Classical, Hybrid	Classical, Hybrid	Classical
Hybrid, Classical	Hybrid, Classical	Hybrid
Classical, Hybrid	Hybrid	Hybrid
Hybrid, Classical	Classical	Classical
Classical	Classical, Hybrid	Classical
Hybrid	Classical, Hybrid	Hybrid
Classical	Hybrid	None
Hybrid	Classical	None
Mixed (No overlap)	Mixed (No overlap)	None

Configure PQC for SSHv2 sessions

Use these commands to set up devices as an SSH server and an SSH client. This allows the device to accept connection requests from remote SSH clients, enabling secure remote management of the device through SSH connections.

Note The PQC algorithms are not enabled by default and must be explicitly enabled.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip ssh server algorithm kex** *algorithm* command to configure the PQC based key-exchange algorithms that the SSH server supports.

Example:

```
Router(config)# ip ssh server algorithm kex mlkem768nistp256-sha256
```

Step 3 Use the **ip ssh client algorithm kex** *algorithm* command to configure the PQC based key-exchange algorithms that SSH client supports.

Example:

```
Router(config)# ip ssh client algorithm kex mlkem768x25519-sha256
```

This command is used to define the order and selection of key exchange (KEX) algorithms that the SSH client will use during the SSH handshake process.

Verify PQC for SSHv2 sessions

Use this section to understand how to verify PQC implementation for SSHv2.

Command	Purpose	What to look for in output
show ip ssh	Displays the status of the SSH server, its configuration details.	In the output of this command, look for these details that indicate PQC implementation is successful: KEX algorithms: **mlkem1024nistp384-sha384,** **mlkem768nistp256-sha256,** **mlkem768x25519-sha256**

This example shows the response when a PQC algorithm command is entered on a platform that does not support PQC.

```
Router(config)# ip ssh server algorithm kex mlkem1024nistp384-sha384 ecdh-sha2-nistp256
Router(config)#
000205: Jun 18 16:41:43.086: %SSH-3-KEX_PQC_NOT_SUPP: PQ/T hybrid key exchange algorithms
are not supported on this platform.
Router(config)#
```

Key elements of PQC

PQC is implemented by using the algorithms available as part of Module-lattice based Cryptography. Module-lattice based Cryptography is a type of quantum-resistant public-key cryptography that uses structured, high-dimensional grid points (lattices) to secure data. Cisco implements PQC into protocols like SSH using ML-KEM.

These algorithms are resistant to quantum attacks and are used for secure key establishment, encryption, and digital authentication or signatures.

This table lists the areas in which each algorithm can be used:

ML-KEM algorithm	What this algorithm means	Security Level
mlkem768x25519-sha256	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. X25519 is a widely used classical elliptic curve Diffie-Hellman key exchange algorithm. SHA-256 is the hash function used for cryptographic hashing in the key exchange process. In summary, mlkem768x25519-sha256 is a hybrid PQC key exchange algorithm combining the ML-KEM 768-bit lattice-based scheme with classical X25519 and SHA-256, used to secure key exchanges against quantum and classical threats in Cisco's PQC-enabled network protocols.	The security level of this implementation is balanced and is best suited for deployments where performance metrics are critical.
mlkem768nistp256-sha256	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. NIST P-256 is a widely used classical elliptic curve Diffie-Hellman key exchange algorithm standardized by NIST. SHA-256 is the hash function used for cryptographic hashing in the key exchange process. In summary, mlkem768x25519-sha256 is a hybrid PQC key exchange algorithm combining the ML-KEM 768-bit lattice-based scheme with classical X25519 and SHA-256, used to secure key exchanges against quantum and classical threats in Cisco's PQC-enabled network protocols.	The security level of this implementation is good and is best suited for most deployments.

ML-KEM algorithm	What this algorithm means	Security Level
mlkem1024nistp384-sha384	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. 1024 is the highest security parameter set, equivalent to the security level of AES-256. NIST P-384 is the traditional (classical) component. It uses Elliptic Curve Diffie-Hellman (ECDH) with the NIST P-384 curve to maintain security against current non-quantum threats. In summary, mlkem1024nistp384-sha384 represents a hybrid cryptographic proposal combining a quantum-resistant ML-KEM algorithm with classical elliptic curve cryptography (NIST P-384) and SHA-384 hashing. This hybrid approach ensures security against both classical and quantum attacks by negotiating keys using both classical and post-quantum algorithms.	The security level of this implementation is the highest and is best suited for deployments where maximum security is needed.

Supported platforms

PQC can be configured only on these platforms in autonomous mode:

- Cisco Catalyst IR1101 Rugged Series Routers,
- Cisco IR1000 Rugged Series Secure Routers
- Cisco Catalyst IR1800 Rugged Series Routers,
- Cisco Catalyst IR8140 Heavy Duty Series Routers,
- Cisco Catalyst IR8340 Rugged Series Routers,
- Cisco Embedded Service 6300 Series Routers.