March 2022

SECURING IN-SITU OAM USING OPTIMISED AND QUANTUM RESISTANT METHOD

Niranjan M M

SECURING IN-SITU OAM USING OPTIMISED AND QUANTUM RESISTANT METHOD

AUTHOR:

Niranjan M M

ABSTRACT

In-situ Operations, Administration, and Maintenance (iOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. There are many use cases for OAM/iOAM which is used for fault detection, network telemetry, best path selection, NFV optimisation etc., But they are prone to different types of attacks (DoS, Eavesdropping, resource exhaustion etc.,). All the existing techniques provide security to iOAM protocol uses pre-shared key or uses asymmetric key encryption methods. Pre-shared keys are immune to cryptanalysis attacks, but require manual configuration, hence not scalable and error prone. Asymmetric key methods use PKI algorithms and are prone to cryptanalysis attacks. There are no methods to provide quantum resistant key exchange mechanism for iOAM protocol. The techniques presented herein is one such method to provide post quantum secure method to iOAM protocol for providing authenticity and confidentiality.

DETAILED DESCRIPTION

In-situ Operations, Administration, and Maintenance (iOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. iOAM data fields are encapsulated in different protocols such as NSH, Segment Routing, Geneve, IPv6, IPv4 etc., There are many use cases for OAM/iOAM which is used for fault detection, network telemetry, best path selection, NFV optimisation etc., But they are prone to different types of attacks (DoS, eavesdropping, resource exhaustion etc.,).

Following are some of the attacks on the OAM/iOAM:

- Maliciously modification of iOAM options.
- Injecting packets with maliciously generated iOAM options.

- Denial of service (DoS) attack - A malicious attacker can add an iOAM header to packets, to consume the resources of network devices. In some case, attacker adds iOAM data such that packet size is more than MTU and causing fragmentation/packet to drop.
- As iOAM/OAM can be used for network reconnaissance (information about addresses, port numbers, network topology, performance), same can be captured (passively eavesdropping or actively sending OAM packets) and used maliciously to attack the network.
- iOAM options may include timestamps (if network devices are synchronised) and any attack on the time protocol can compromise the integrity of the timestamp-related data fields.

There are existing techniques to prevent above attacks:

- Proof of Transmit Option-Type (POT) is used for verifying the path of data packets. It uses Shamir's Secret Sharing Scheme (SSS). Here iOAM POT consists of "iOAM POT option header" and "iOAM POT option data fields". This method require couple of secret values kept by individual nodes and need to periodically update and share the new secret. It uses crypto-graphical algorithms (using prime number etc.,).
- Having integrity, authentication and encryption protection methods thereby preventing attackers from forging or tampering with OAM packet.
- Using Bidirectional Forwarding Detection (BFD) with authentication mechanism (SHA1, MD5 or simple password).
- Using One-way Active Measurement Protocol (OWAMP), where in HMAC as SHA1 for authentication and AES for encryption.
- Using Two-way Active Measurement Protocol (TWAMP), where in HMAC as SHA1 for authentication and AES for encryption.

In some cases, iOAM is deployed in specific network domains, thus confining the attacks within the network domain (site). If iOAM is deployed across the sites (especially with virtualisation, cloud, and containerised deployments), current methods suggest using secure links (e.g., by using IPsec in tunnel mode) to avoid external threats. All the existing techniques provide security to iOAM protocol uses pre-shared key or uses asymmetric key encryption methods. Pre-shared keys are immune to cryptanalysis attacks, but require manual configuration, hence not

scalable and error prone. Asymmetric key methods use PKI algorithms and are prone to cryptanalysis attacks. There are no methods to provide quantum resistant key exchange mechanism for iOAM protocol.

The techniques presented herein is one such method to provide post quantum secure method to iOAM protocol for providing authenticity and confidentiality. This method provides quantum resistant security to iOAM protocol packets exchanged between the devices to provide enhanced and optimised functionalities (such as fault detection, telemetry etc.,), where-in the packet entry point to/from the network (such as Router) acts as Key Server to distribute post-quantum identifier (PQPSK ID) to other downstream devices. Once the PQPSK ID is available with the devices, they use Quantum Key Source (QKS) to generate PQPSK Key from the PQPSK ID (Note: here, PQPSK Key is not exchanged and/or not negotiated by any means).

The Quantum Key Source (QKS) requires, all the devices to be initialised with the secret seed value so that it can generate unique pair of PQPSK (ID, Key). The secret seed value is shared using asymmetric key encryption method. Further, SKS generates unique PQPSK Key corresponding to PQPSK ID without having any further negotiation with the peer. This PQPSK Key is used to encrypt the iOAM metadata information which will be added/appended to the data packets (viz., IPv4, IPv6, Geneve etc.,).

Further, the iOAM metadata is added/appended to the data packets to inform/update the downstream devices about the event instead of sending explicit event notification messages. This method uses iOAM metadata payload itself to exchange initial seed secret value which is encrypted using asymmetric key algorithms (PKI methods). Also, it embeds PQPSK ID generated by the Router (Key server) in the iOAM metadata payload and share it with other downstream devices. As mentioned above, this PQPSK ID is used to generate unique PQPSK Key to encrypt/decrypt the iOAM metadata payload. Whenever Key Server wants to change the key, it generates new PQPSK ID and share it with the other devices, hence key revocation and new key generation is automated anytime (based on timeout, based on number of packets exchanged etc.,).

The technique presented herein is divided into different phases. For example, with NFV service chain, each NFV device/instance would authenticate and register with the centralised controller (Router/Gateway/any central entity). Once it is registered, centralised device would be aware of the service(s) running on the NFV in the service chain deployments.

Phase-1: Exchange public keys:

As part of manufacturing, Private and Public keys are generated on all the devices using a quantum secure algorithm such as McEliece. This is one time process:

- Manufacturing team generates private and public keys using same root CA.
- These generated keys are stored on the device/instance (in TPM/vTPM).
- Public keys (root CA certificate chain) are bundled as part of image.
- McEliece Public keys can also be exchanged between peers over the secure connection, to support 3rd party certificates.

Phase-2: Exchange common seed secret across devices as part of iOAM metadata payload:

- The packet entry point to/from the network (such as Router) acts as Key Server (KS) and other downstream devices acts as Non-Key-Server (NKS).
- Key Server generates random secret seed value and encrypt using downstream device's public key. Secret seed value will be "common for all the downstream devices in the NFV service chain".
- Key Server transmit the encrypted secret seed value to all other devices in the service chain individually (encrypted using downstream device's public key) inside iOAM metadata payload.
- Downstream devices process the iOAM metadata carrying encrypted secret seed value and decrypt the seed using its private key.
- Now all devices in the iOAM domain will have a common secret seed.
- All devices initialise their QKS with the common seed secret exchanged as above.

Phase-3: Exchange common PQPSK ID as part of iOAM metadata payload:

Key Server generates and shares PQPSK ID to the downstream devices. To avoid DDoS attack on PQPSK ID, Key Server should include some authentication information along with PQPSK ID generated by the SKS such as

Signature: Signed on PQPSK ID by Key Server using its private key (generated and distributed by Certificate Authority)

- QKS on Key Server generates (PQPSK Key, PQPSK ID) pair.

- Key Server would generate signature by signing on PQPSK ID using its private key and include that signature also as part of PQPSK ID TLV.
- Key Server shares this PQPSK ID TLV to the downstream devices as part of iOAM metadata payload.
- Downstream devices validate the signature using public key of Key Server before accepting the PQPSK ID sent by the Key Server.

The technique presented herein is explained with example as below, but not limited to container platforms, applicable for virtual/cloud/physical platforms as well.

Step-1: State of the network with NFV chain:

Let us consider production NFV chain with container components, which includes QoS Router, IPS/IDS, Load balancer, HTTP server (nginx). Each component will have the profiles configured with parameters/settings that impacts how it treats the inbound packet.

For example:

a. QoS Router does rate limiting or bandwidth allocation to support rate (with/without AVC)

b. Intrusion protection System (IPS)/Intrusion Detection System (IDS) does packet inspection to different regex depth (with/without ACLs) etc.,

Step-2: Increase in legitimate traffic detected by a component in the NFV chain

For example:

a. During office hours, employees would get connected to the network (wired/wireless), and hence would increase in legitimate traffic (Enterprise deployments)

b. During working hours, customers/consumers (shopping complex, coffee days etc.,) connect to guest network and it varies dynamically based on time (i.e., more traffic during evening time than in the afternoon) (ISP/Data center deployments).

QoS Router first detects such increase in the legitimate traffic and anticipates that this might continue for some time.

Step-3: Component attaches iOAM metadata information to the data packet to inform downstream components about the incoming traffic pattern:

iOAM metadata information would include:

- Average size of the packets and packet rate
- Statistics about the protocols observed so far (e.g., TCP(6), HTTP(80), HTTPS(443) etc.,
- Source and destination IP addresses.

This iOAM metadata information is encrypted using PQPSK Key generated by the QKS using PQPSK ID (shared earlier by the first component through which packets enters the NFV chain (in this case, it is QoS Router).

The encrypted iOAM metadata will be added to the data packet as TLV and transmitted on the network for processing by the downstream components.

Step-4: Downstream components leverage the iOAM metadata to tune their profiles (parameters) to handle the incoming traffic:

Downstream components detect the iOAM metadata, decrypt using PQPSK Key generated by QKS using PQPSK ID (shared earlier by the first component through which packets enters the NFV chain (in this case, it is QoS Router). Upon decrypting the iOAM metadata, downstream component adjusts its parameters so they can handle the incoming traffic in an efficient and optimal way.

For example:

a. The IPS would reduce the regex search depth per packet from 200 to 100 bytes to accommodate the high rate of incoming packets

b. vSwitch would replace DPDK to VPP (VPP might scale better for this traffic profile compared to other types)

c. Load balancer would instantiate a greater number of worker/member containers to process increase in traffic.

The techniques presented herein provides post quantum secure iOAM protocol. This method allows frequent key refresh with no administrative cost. Moreover, this method is applicable for any deployment with iOAM use case whether devices are virtual (cloud/containers) or hardware based.