October 2020

SECURE AND OPTIMIZED METHOD OF PROVIDING TRUSTWORTHINESS FOR IOT SENSORS IN LOW-POWER WAN DEPLOYMENTS

Niranjan M. M

Nagaraj Kenchaiah

SECURE AND OPTIMIZED METHOD OF PROVIDING TRUSTWORTHINESS FOR IOT SENSORS IN LOW-POWER WAN DEPLOYMENTS

AUTHORS:
Niranjan M M
Nagaraj Kenchaiah

ABSTRACT

Currently there are multiple ways of verifying the identity and integrity of Internet of Things (IoT) sensors based on, for example, the Trusted Computing Group's (TCG's) Guidance for Securing Network Equipment, software-centered approaches such as using a checksum, and using an in-band and out-of-band approaches for integrity validation. In each of these approaches, trustworthiness may be based on limited artifacts. As well, none of these approaches employ quantum resistant secure key exchange methods between a Long Range (LoRa) Wide Area Network (LoRa) (LoRaWAN) Gateway and sensors. To address these challenges techniques are presented herein that apply an attestation method to the Constrained Application Protocol (CoAP), which is used between a LoRa Gateway and sensors, to provide "proof of integrity" and "freshness of proof of integrity" (in other words, trustworthiness) to IoT sensor devices. An Attestation ID that is derived through an attestation method is shared in data traffic (i.e., in-band) securely using a Post-Quantum Cryptography (PQC) method.

DETAILED DESCRIPTION

IoT consists of 'things' (e.g., sensors) that are enabled with some form of compute, network, and storage capability which, in turn, makes them subject to attack or modification by malicious parties. Hence there needs to be a way to perform investigation and cyber security forensics to gather and preserve attack evidence from a particular IoT computing device to check if it has been compromised.

Currently there are multiple ways of verifying the identity (e.g., using manufacturer usage descriptions (MUD)) and the integrity of IoT sensors, based either on the TCG's Guidance for Securing Network Equipment (e.g., Trusted Platform Module (TPM) v1/TPM v2, see https://trustedcomputinggroup.org/tcg-guidance-securing-networkequipment/) or on software-based approaches such as using a checksum (see, for example, “Software-Based Remote Code Attestation in Wireless Sensor Network” under https://ieeexplore.ieee.org/document/5425280). As well, there are mechanisms which employ an out-of-band approach for integrity validation and mechanisms which employ an in-band approach for integrity validation.

In each of these approaches, determining the trustworthiness of an IoT sensor by validating the proof of integrity, freshness of the proof of integrity, canary Stamps, Platform Configuration Registers (PCR) values, measurement lists, etc. is not considered.

Additionally, under certain in-band approaches to integrity validation, an encrypted Attestation ID that is shared between a sensor and a fog router is not quantum resistant and thus prone to cryptanalysis attacks using Shor's and Grover's algorithm (as will be further described below).

To address these challenges, techniques are presented herein that support use of an attestation method for providing trustworthiness to IoT sensors and use of a Post-Quantum Pre-Shared Key (PQPSK) for encrypting an Attestation ID which is sent in-band with the Internet Protocol version 6 (IPv6) packets for doing a first level of validation at the fog/edge router and attestation verification at the LoRa Gateway.

Various of the techniques that are presented herein leverage aspects of four foundational technologies. In support of the upcoming detailed discussion of the techniques that are presented herein, each of those four foundational technologies are briefly introduced below.

1. Constrained Application Protocol (CoAP). CoAP is a User Datagram Protocol (UDP) based protocol having request and response based messaging semantics. See, for example, Internet Engineering Task Force (IETF) Request For Comments (RFC) 7252 (“The Constrained Application Protocol” under https://tools.ietf.org/html/rfc7252), RFC 7959 (“Block-Wise Transfers in the Constrained Application Protocol” under https://tools.ietf.org/html/rfc7959), and RFC 8613 (“Object Security for Constrained RESTful Environments (OSCORE)” under https://tools.ietf.org/html/rfc8613).

2. LoRa Wireless Area Network (LoRa WAN) with IPv6. The transmission of IPv6 Packets over LoRaWAN is defined by the IETF in "Transmission of IPv6 Packets over LoRaWAN" (under https://tools.ietf.org/id/draft-vilajosana-lpwan-lora-hc-00.txt).

3. Cryptanalysis attacks and quantum resistant algorithms. For public-key encryption the algorithms that are currently employed are based on Rivest–Shamir–Adleman (RSA) public-key cryptosystems or elliptic-curve cryptography (ECC) and will be broken by quantum computers using, for example, Shor's algorithm, Grover's algorithm, etc.

Code-based cryptography (e.g., McEliece) has been studied since 1978 and has withstood attacks very well, including attacks using quantum computers.

For example, McEliece with binary Goppa codes using length n = 6960, dimension k = 5413 and adding t = 119 errors is thought to be safe (see https://eprint.iacr.org/2017/1180.pdf). However, this method is not commonly used because of the compute power needed to generate keys and their sheer size (as well, rekeying McEliece is too expensive).

The Stehle-Steinfeld version of the N-th degree Truncated polynomial Ring Units (NTRU) lattice-based crypto system is also believed to be quantum resistant (see https://pdfs.semanticscholar.org/f372/3b29d42e1e6b5869b478677761ff0a69860c.pdf).

Quantum Key Distribution (QKD) is a way to distribute commonly shared secrets which can be leveraged to derive the same Advanced Encryption Standard (AES) key on both sides. But QKD requires additional bandwidth (e.g., dedicated fiber) and a limitation on the distance (e.g., approximately 100KM). Furthermore, it is not suited for large scale enterprise deployments with numerous sites/locations.

4. Attestation and PQC methods.. Attestation is a trusted computing technology which may be applied to the instant problem class. TPM functionality is now embedded in a wide variety of devices including mobile phones, PCs, and routers. Attestation as defined by the TCG describes how the TPM may be used as a hardware root of trust and offer proof of integrity of a node. PQC methods are discussed below.

Turning to the techniques that are presented herein, a known fog-based attestation approach considers a number of flows but has various limitations. More particularly:

- A checksum value is calculated based on the program memory space and used as an Attestation ID (e.g., hash ID) and shared with the server (e.g., LoRa) during initial attestation verification. The server then downloads the attestation ID of sensors to a fog router (e.g., a IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Border Router (6LBR)) to store it in the local database.
    - Limitation: Integrity validation of IoT sensors by a server (e.g., LoRa) using a checksum value is based on the program memory space, but no mention is made of attestation methods such as a canary stamp or a hardware fingerprint stored securely in Aikido/Anti-Counterfeit Technology (ACT2)/Trust Anchor module (TAM) chip which is tamper proof.
- Data packets from the sensor carry an Attestation ID in the IPv6 extension header (EH) to a fog router. For avoiding man-in-the-middle and replay attacks, it uses an encryption key which is periodically (e.g., based on time or based on number of packets) exchanged in control frames between a fog router and a sensor to encrypt the Attestation ID/hash ID which is sent in a IPv6 EH from a sensor to a fog router.
    - Limitations:
        - Keeping the encryption key in-sync across fog routers is challenging and requires use of some protocol to accomplish.
        - An encryption key is exchanged in control packets and hence should be sent over some secure tunnel or it should be conveyed using public-key cryptography (no mention is made of this) for sharing the secret. The secure tunnel establishment itself should also be based on a pre-shared key or certificate-based authentication. Pre-shared key configuration on all of the sensors and on all of the fog routers is not feasible (especially with scale) and thus motivates the use of Public Key Infrastructure (PKI) methods. But the use of public-key

cryptographic methods are prone to cryptanalysis attacks as described previously.

- A fog router uses an Attestation ID/hash ID to verify the locally stored information. Any change in program memory on a sensor will result in a different hashing and hence a different Attestation ID in an IPv6 EH which can be detected by a fog router.
    - Limitation: Just a checksum value is considered, as opposed to other information such as Known Good Values (KGV) including, for example, a microloader measurement list (ML), a Basic Input/Output System (BIOS) measurement list (BL), runtime functionality, working condition and validation of sensors, etc.
- Upon detecting an anomaly, a fog router may trigger the attestation verification request to a server (e.g., LoRa) and, in turn, the server initiates attestation verification with the sensor.

Under aspects of the techniques that are presented herein, instead of relaying only on a checksum value an attestation method for validating trustworthiness may be employed using the following information elements to generate an Attestation ID/hash ID:

- A hardware fingerprint for "device identity."
- Measurement lists (for a microloader, BIOS, libraries, applications - in short, KGV) and PCR values for "proof of integrity."
- A canary stamp, nonce, or timestamp for "freshness of proof of integrity."

In brief, the techniques that are presented herein provide trustworthiness to the sensors. Under other aspects of the techniques that are presented herein, instead of encrypting an Attestation ID/hash ID with the encryption key shared by the fog router over control packets (it should be shared over a secure tunnel or using a PKI), since it is known that PKI methods are prone to cryptanalysis attacks by Shor's and Grover's algorithm a post-quantum secure key exchange mechanism using a PQC method may be employed, as illustrated in Figure 1, below.

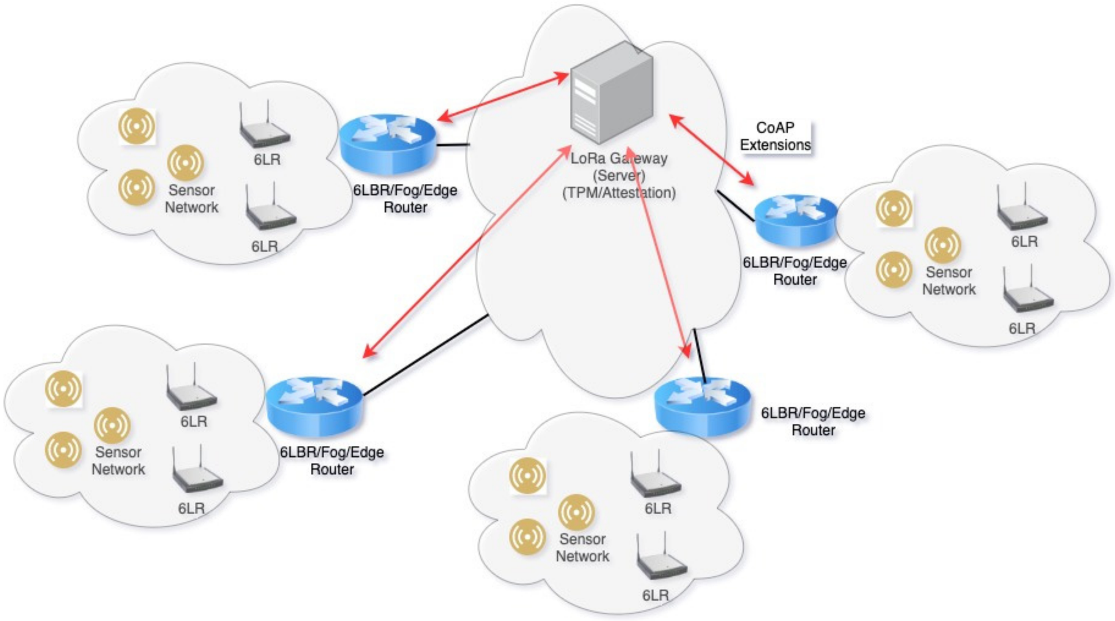


Figure 1: Exemplary Environment

PQC is critical for IoT security in a quantum era. The National Institute of Standards and Technology (NIST) is working to come up with PQC standards for Third Generation Partnership Project (3GPP) Fifth Generation (5G) environments and the same is applicable for low-power IoT (as mentioned in https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=927805).

NIST is designing a PQC algorithm as a "drop-in replacement" for RSA public-key cryptosystems and ECC, but there are many challenges with respect to public key size, signature size, alternative auxiliary functions, certificate authority (CA) infrastructure, etc. Candidates for a PQC algorithm include lattice-based, coding-based, multivariate, stateless hash-based signatures, and super singular elliptic curve isogeny-based approaches.

Out of these methods, NIST is planning to standardize stateful hash based signatures based on standards developed by the IETF. After further analysis and evaluation of the above methods, NIST is planning to release draft standards for public comments in 2022 or 2023.

Until the standardized PQC methods are in-place for IoT alternate approaches to the above described challenges are needed in support of, for example, different environments such as low-power IoT devices in smart city deployments, IoT sensors in low-power wide-area network (LPWAN) deployments, etc.

For generating quantum resistant keys (aka PQPSK) a PQC method may be employed. However, there are other methods, such as QKD, that may be used as a replacement for a PQC method for providing quantum security to IoT/sensor deployments.

Under aspects of the techniques that are presented herein an attestation method is applied to a CoAP which is used between a LoRa Gateway and sensors to provide "proof of integrity" and "freshness of proof of integrity" (in other words, trustworthiness) to IoT sensor devices. An Attestation ID that is derived during the attestation method is securely shared in data traffic (i.e., in-band) using a post-quantum secure method (e.g., a PQC method).

Elements of particular interest and note within the techniques that are presented herein are discussed below.

A first element of the techniques presented herein comprises an attestation method to provide trustworthiness to sensors. CoAP runs over UDP, wherein CoAP request and response semantics are carried in CoAP messages which include either a Method Code or Response Code, respectively. Optional request and response information, such as the Uniform Resource Identifier (URI) and payload media type, are carried as CoAP options.

Aspects of a first element of the techniques that are presented herein add attestation information to the CoAP request and response packets as an extension that embeds:

- A hardware finger print (e.g., derived from a Secure Unique Device Identifier (SUDI) or similar value).
- Software (e.g., KGV such as operating system (OS), BIOS, kernel, version, application binaries/libraries, etc.).
- A canary stamp (e.g., counters, time-ticks, PCR values, etc.).
- An Attestation ID that is generated by hashing the above information.

A first sub-element to the first element of the techniques presented herein comprises proof of integrity. In particular, TPM functionality is used as a root of trust and as a proof of integrity of sensors.

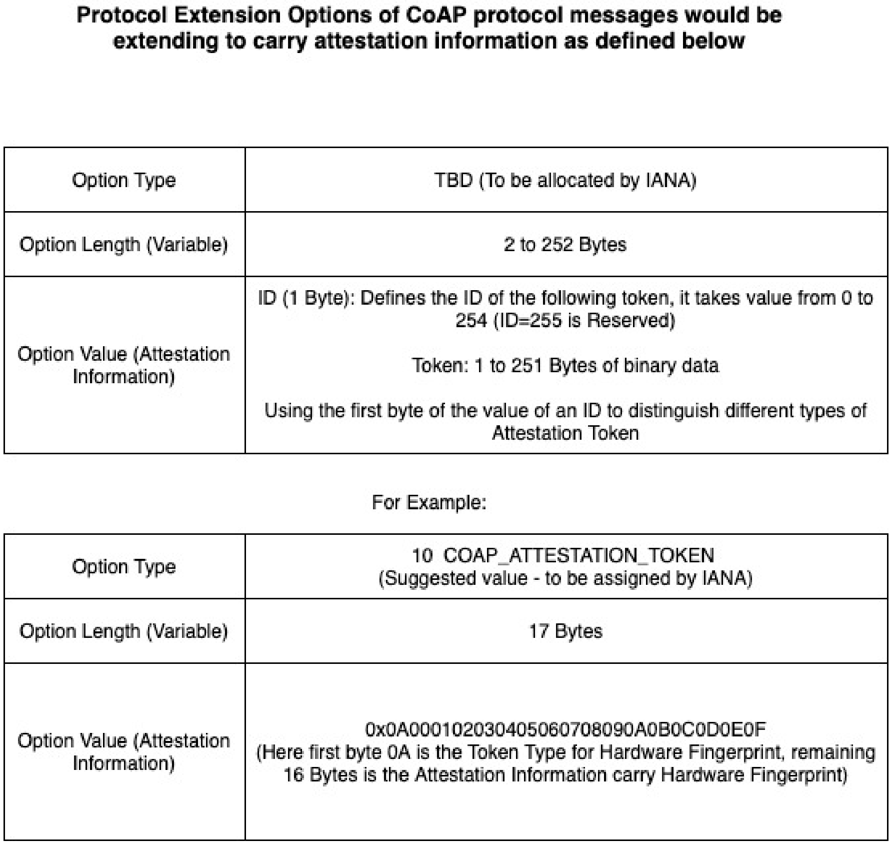
As mentioned in RFC 7252 (see https://tools.ietf.org/html/rfc7252#section-5, “Request/Response Semantics”), CoAP provides support for carrying a number of options.

These options are used as additions to the protocol without breaking backward compatibility with earlier versions. Aspects of the techniques presented herein add attestation information as another option to be carried in request/response (CoAP) packets to provide proof of integrity.

A second sub-element to the first element of the techniques presented herein comprises freshness of the proof of integrity. Proof of integrity can also be accompanied with a signature to prove freshness of the proof of integrity – i.e., by adding a signature over a random nonce (aka entropy) presented by the LoRa Gateway to the sensor in the CoAP packet. This would help in detecting the replay of old evidence via a "nonce." A "nonce" is a random number provided by the LoRa Gateway making the request. Such a nonce is passed into the TPM. The results coming out of the TPM include a signature based on the nonce. That result is the output from the TPM which could not have been generated before that nonce was provided.

Consider an example as follows, and as illustrated via Figure 2, below:

- If a CoAP packet from the LoRa Gateway contains random data/nonce it is extended to carry an intention to validate proof of integrity.
- A CoAP packet from the sensor carries an extension to its proof of integrity along with a signature over random data/nonce received in CoAP packet from the LoRa Gateway.
- A LoRa Gateway and a sensor participating in a CoAP protocol exchange will use this attestation information to verify whether the sensor is trustworthy or not to make a decision to consider as a trusted sensor.



Protocol Extension Options of CoAP protocol messages would be extending to carry attestation information as defined below

Option Type	TBD (To be allocated by IANA)
Option Length (Variable)	2 to 252 Bytes
Option Value (Attestation Information)	ID (1 Byte): Defines the ID of the following token, it takes value from 0 to 254 (ID=255 is Reserved) Token: 1 to 251 Bytes of binary data Using the first byte of the value of an ID to distinguish different types of Attestation Token

For Example:

Option Type	10 COAP_ATTESTATION_TOKEN (Suggested value - to be assigned by IANA)
Option Length (Variable)	17 Bytes
Option Value (Attestation Information)	0x0A000102030405060708090A0B0C0D0E0F (Here first byte 0A is the Token Type for Hardware Fingerprint, remaining 16 Bytes is the Attestation Information carry Hardware Fingerprint)

Figure 2: CoAP Protocol Extensions

A third sub-element to the first element of the techniques presented herein comprises in-band validation using an Attestation ID. Once a sensor is considered to be trusted, an Attestation ID shared by the sensors during attestation verification is downloaded to the fog routers (e.g., a 6LBR) which will be used to validate the Attestation ID sent by the sensor in the IPv6 EH data packets (to be further explained in a "How it Works" narrative below). To avoid man-in-the-middle and replay attacks (i.e., malicious activity), an Attestation ID will be encrypted using a PQPSK which is shared using a post-quantum secure key exchange method (as explained in the following paragraphs).

A second element of the techniques presented herein comprises a PQPSK generation and sharing mechanism using a PQC method. The CoAP may be extended to allow the LoRa Gateway (e.g., a server) to distribute a post-quantum identifier (e.g., a PQPSK ID) to various sensors. Here, the LoRa Gateway (i.e., the server) acts as key server and a sensor acts as non-key server.

Subsequently, the LoRa Gateway (i.e., server) and a Sensor will use the negotiated PQPSK ID to get the same PQPSK key from a PQC method, also referred to as the Quantum Key Source (QKS), which will be used as an encryption key for encrypting the Attestation ID in the IPv6 packet as an EH in the data traffic (i.e., in-band data packet).

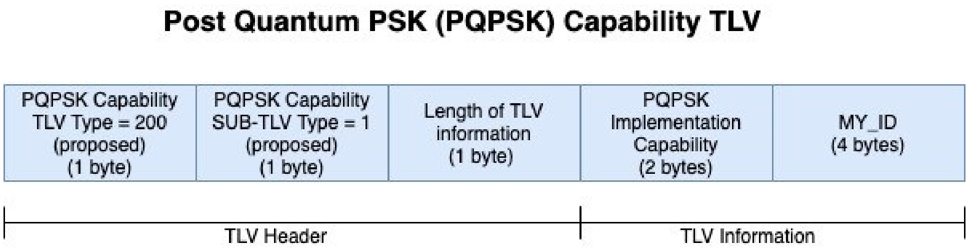
PQC requires the initial seed secret key to be installed on each of the PQC key provider devices so that the PQC method generates a unique PQPSK Key from the PQPSK ID (i.e., a unique {PQPSK Key, PQPSK ID} pair). Management of the initial seed key is presented below in the description of a third sub-element to this (i.e., second) element of the techniques presented herein (i.e., describing a secret seed value being shared between a LoRa Gateway and a sensor).

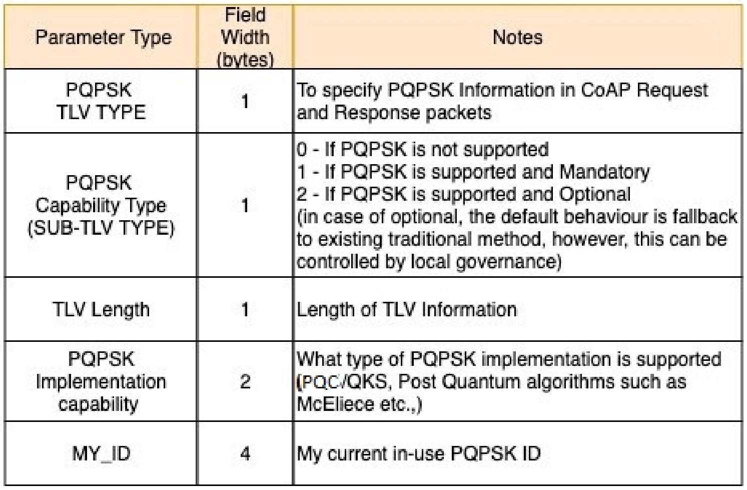
To avoid a denial-of-service (DoS) attack on a PQPSK ID, a Key Server (KS) should include some authentication information along with the PQPSK ID generated by the PQC method. For example:

- A signature is signed on a PQPSK ID by the KS using a private key (e.g., securely stored through a SUDI/ACT2/TAM chip which is tamper proof).
- A PQC method on the KS generates a {PQPSK Key, PQPSK ID} pair which is unique per device.
- The KS may generate a signature by signing on a PQPSK ID using its private key and include that signature also as part of a PQPSK ID type-length-value (TLV).
- The consolidated PQPSK ID TLV is shared to a peer.
- The peer validates the signature using the public key of the KS before accepting the PQPSK ID sent by the KS.
- A set of PQPSK TLVs are introduced to provide post-quantum security.
  - PQPSK Capability TLV - For exchanging PQPSK capabilities to identify the ability for post-quantum security.
  - PQPSK ID Information TLV - For sending a PQPSK ID from the LoRa Gateway (a KS) to the sensor (a non-key server)

A first sub-element to the second element of the techniques presented herein comprises a PQPSK Capability TLV (exchanged between a sensor and a LoRa Gateway).

IoT sensors which support a post-quantum pre-shared key (a PQPSK) announce that capability as part of a PQPSK capability TLV in a CoAP Request message (see https://tools.ietf.org/html/rfc7252#section-5). An example of one possible PQPSK Capabilities TLV is presented in Figure 3, below.


Post Quantum PSK (PQPSK) Capability TLV
PQPSK Capability TLV Type = 200 (proposed) (1 byte)
PQPSK Capability SUB-TLV Type = 1 (proposed) (1 byte)
Length of TLV information (1 byte)
PQPSK Implementation Capability (2 bytes)
MY_ID (4 bytes)
TLV Header
TLV Information



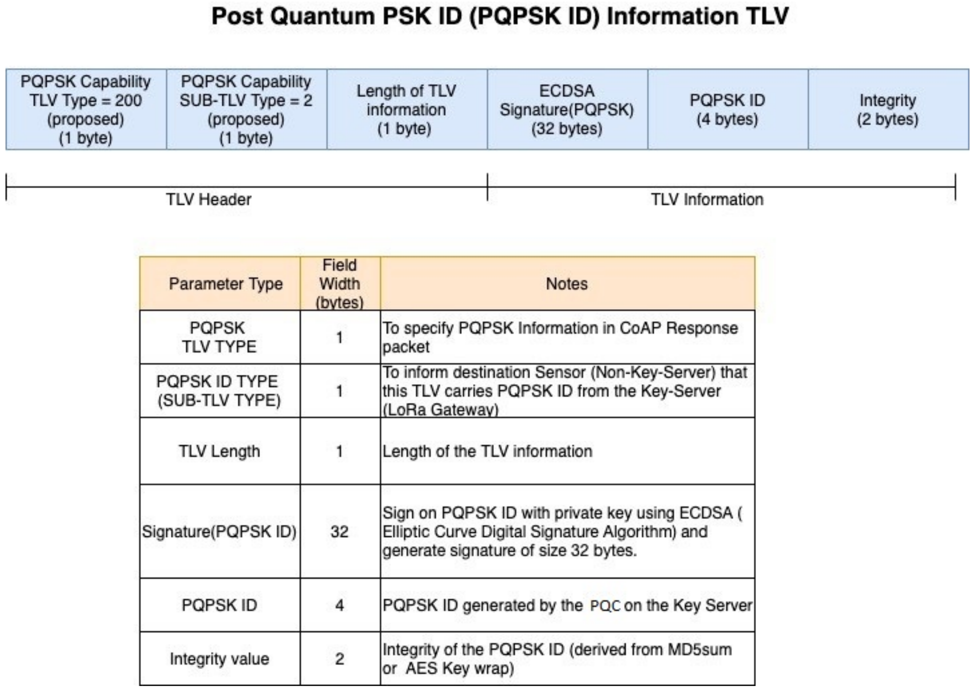
Parameter Type	Field Width (bytes)	Notes
PQPSK TLV TYPE	1	To specify PQPSK Information in CoAP Request and Response packets
PQPSK Capability Type (SUB-TLV TYPE)	1	0 - If PQPSK is not supported 1 - If PQPSK is supported and Mandatory 2 - If PQPSK is supported and Optional (in case of optional, the default behaviour is fallback to existing traditional method, however, this can be controlled by local governance)
TLV Length	1	Length of TLV Information
PQPSK Implementation capability	2	What type of PQPSK implementation is supported (PQC/QKS, Post Quantum algorithms such as McEliece etc.,)
MY_ID	4	My current in-use PQPSK ID

Figure 3: PQPSK Capabilities TLV

If IoT sensors are not capable of PQPSK then they may fall back to a legacy means of encrypting the Attestation ID.

A second sub-element to the second element of the techniques presented herein comprises distribution of a PQPSK ID information TLV (comprising, for example a type, a signature, a PQPSK ID, a checksum, etc.) between a LoRa Gateway and a sensor. When both a LoRa Gateway and a sensor are PQPSK capable, the LoRa Gateway (a KS) generates the PQPSK ID with the help of a local PQC service and distributes same to the sensor in a PQPSK ID TLV in a CoAP Response message (see https://tools.ietf.org/html/rfc7252#section-5)

An example of one possible PQPSK ID information TLV is presented in Figure 4, below.



Post Quantum PSK ID (PQPSK ID) Information TLV

PQPSK Capability TLV Type = 200 (proposed) (1 byte)	PQPSK Capability SUB-TLV Type = 2 (proposed) (1 byte)	Length of TLV information (1 byte)	ECDSA Signature(PQPSK) (32 bytes)	PQPSK ID (4 bytes)	Integrity (2 bytes)

Parameter Type	Field Width (bytes)	Notes
PQPSK TLV TYPE	1	To specify PQPSK Information in CoAP Response packet
PQPSK ID TYPE (SUB-TLV TYPE)	1	To inform destination Sensor (Non-Key-Server) that this TLV carries PQPSK ID from the Key-Server (LoRa Gateway)
TLV Length	1	Length of the TLV information
Signature(PQPSK ID)	32	Sign on PQPSK ID with private key using ECDSA (Elliptic Curve Digital Signature Algorithm) and generate signature of size 32 bytes.
PQPSK ID	4	PQPSK ID generated by the PQC on the Key Server
Integrity value	2	Integrity of the PQPSK ID (derived from MD5sum or AES Key wrap)

Figure 4: PQPSK ID Information TLV

The second element of the techniques presented herein includes a first phase (Phase-1) of the PQC process – sharing a secret seed value between a LoRa Gateway and a sensor. Various of the activities that may occur during Phase-1 are described below.

As part of the manufacturing process, private and public keys are generated on a LoRa Gateway (e.g., a server) and a sensor using a quantum secure algorithm such as McEliece. Note that this is a one-time process.

Under a TPM or virtual TPM (vTPM) scenario a manufacturing team generates private and public keys using the same root CA (e.g., through a PKI). The generated keys are stored in a SUDI/ACT2/TAM chip on a LoRa Gateway (a KS) and a sensor (a non-key server). The public keys (with root CA certificate chain) are bundled as part of the image.

Under a third-party support scenario McEliece public keys can also be exchanged between peers, over an out-of-band secure connection, to support third-party certificates.

The private and public key of a LoRa Gateway and a sensor may, to aid exposition, be referred to as SK1 and PK1 for the LoRa Gateway and SK2 and PK2 for the sensor. As well, the LoRa Gateway acts as KS.

A HMAC Key Derivation Function (HKDF) may be employed to generate a unique "secret seed value" on the LoRa Gateway. The generated "secret seed value" may be encrypted with the public key PK2 of the sensor to generate a ciphertext. The ciphertext is then sent to the sensor. The sensor may decrypt the ciphertext using its private key SK2 to recover the "secret seed value." Now both the LoRa Gateway and the sensor have a common "secret seed," which may be used by a PQC method in phase two of the PQC process (as described below in connection with a fourth sub-element to the second element of the techniques presented herein).

Similarly, the same secret seed value is exchanged between a LoRa Gateway and a fog/edge router so that a fog/edge router can generate the same PQPSK key from the PQPSK ID of the sensor. Note that the PQPSK ID that is populated in phase two of the PQC process, as described below, is downloaded along with a per-sensor Attestation ID to a fog router by the LoRa Gateway.

Figure 5, below, illustrates aspects of common secret seed establishment using the McEliece algorithm.

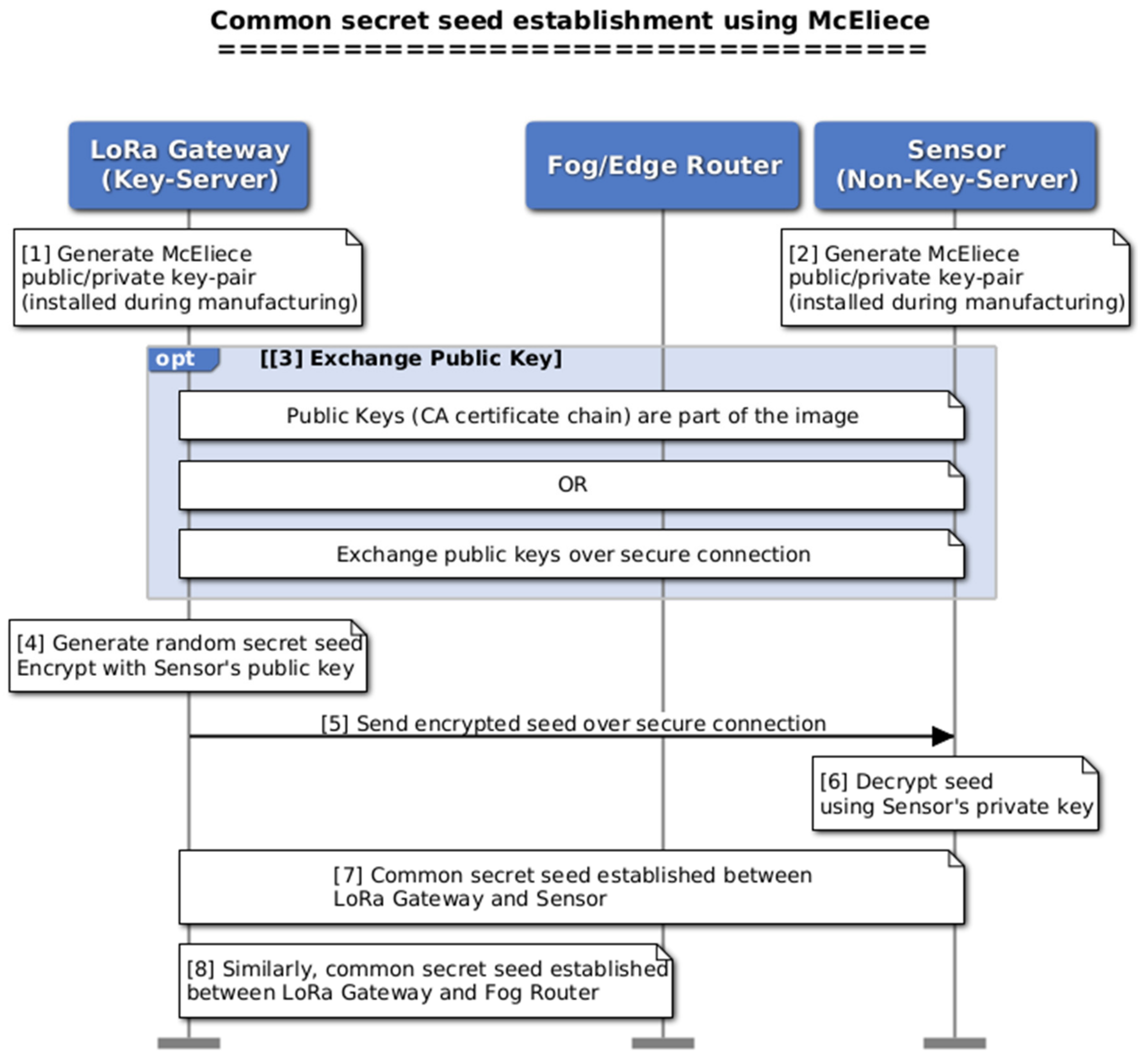

Common secret seed establishment using McEliece
LoRa Gateway (Key-Server)
Fog/Edge Router
Sensor (Non-Key-Server)
[1] Generate McEliece public/private key-pair (installed during manufacturing)
[2] Generate McEliece public/private key-pair (installed during manufacturing)
opt
[[3] Exchange Public Key]
Public Keys (CA certificate chain) are part of the image
OR
Exchange public keys over secure connection
[4] Generate random secret seed Encrypt with Sensor's public key
[5] Send encrypted seed over secure connection
[6] Decrypt seed using Sensor's private key
[7] Common secret seed established between LoRa Gateway and Sensor
[8] Similarly, common secret seed established between LoRa Gateway and Fog Router

Figure 5: Common Secret Seed Establishment

The second element of the techniques presented herein further includes a second phase (Phase-2) of the PQC process – generation of the same pre-shared key using a PQC method. Various of the activities that may occur during Phase-2 are described below.

The LoRa Gateway and the sensor use a service which will generate a unique {PQPSK Key, PQPSK ID} pair based on the common secret seed exchanged in phase one (as described above). This service may be called a PQC process, which is a software entity implemented inside the LoRa Gateway/fog router/sensors.

The LoRa Gateway and the sensor initialize their local PQC instance with the common secret seed that was exchanged in phase one (as described above). The LoRa

Gateway (KS) requests its local PQC instance to generate a PQPSK secret key (which is unique to each sensor) and the PQC generates the secret key PQPSK key.

A unique identity PQPSK ID is paired with each generated PQPSK key. The LoRa Gateway receives the PQPSK key and the corresponding unique identity PQPSK ID from the local PQC instance. The LoRa Gateway then sends the identity PQPSK ID corresponding to the PQPSK key to the sensor. It is important to note that the PQPSK key itself is never exchanged between the LoRa Gateway and the sensor.

The sensor requests its local PQC instance to fetch the PQPSK Key corresponding to the PQPSK ID that was received from the LoRa Gateway. Since the PQC instances of the LoRa Gateway and the sensor have with the same seed secret, they will have the same PQPSK key corresponding to the PQPSK ID. Now, the LoRa Gateway and the sensor will have the same PQPSK Key which they can use as an encryption key for encrypting an Attestation ID in the IPv6 EH packet. Optionally, sensors can use the PQPSK key as a post-quantum symmetric key to provide data confidentiality (e.g., encryption/decryption of data traffic).

Any time that a LoRa Gateway wishes to refresh the encryption key the LoRa Gateway and the sensor may run the above procedure. Hence, a LoRa Gateway and a sensor can refresh their keys at will, without any administrative intervention and without ever exhausting their shared keys (since they are generated once). A PQPSK ID populated for each sensor is downloaded along with a per-sensor Attestation ID to a fog router by the LoRa Gateway. Figure 6, below, depicts aspects of a post-quantum secure connection establishment using PQC.

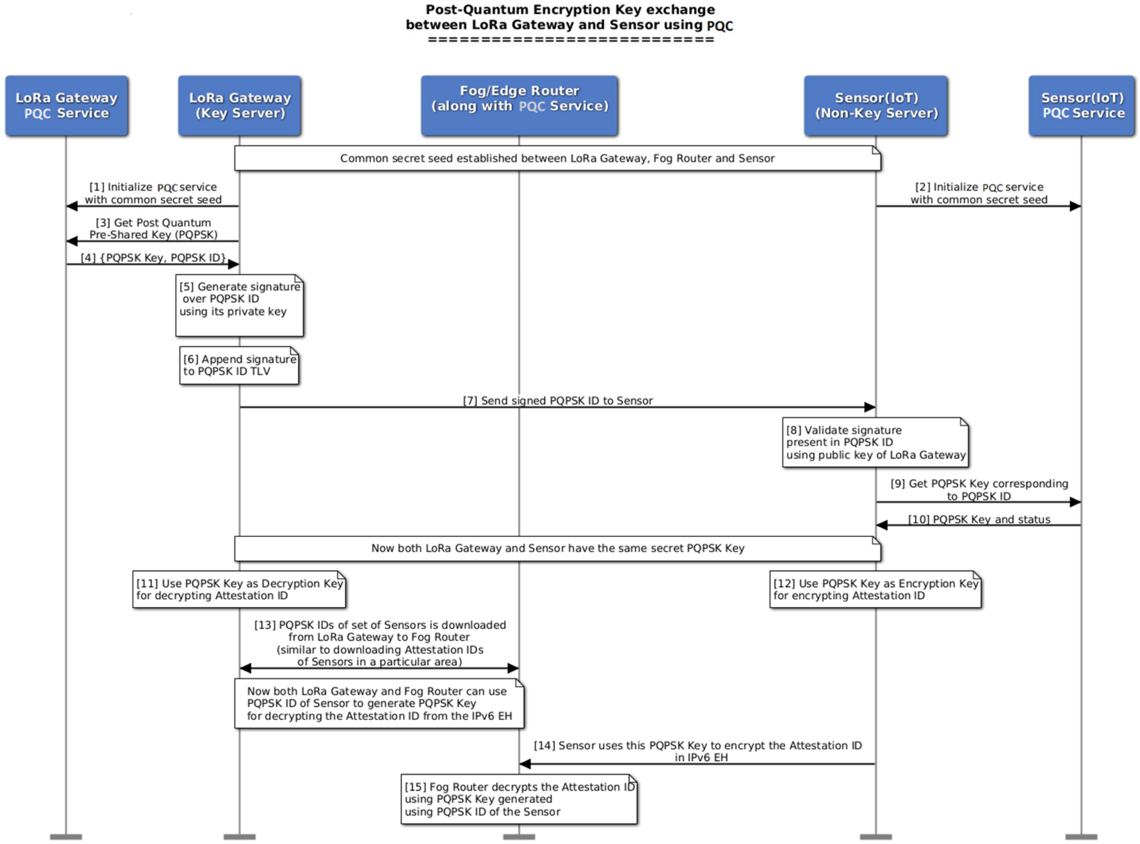

Post-Quantum Encryption Key exchange
between LoRa Gateway and Sensor using PQC
LoRa Gateway PQC Service
LoRa Gateway (Key Server)
Fog/Edge Router (along with PQC Service)
Sensor(IoT) (Non-Key Server)
Sensor(IoT) PQC Service
Common secret seed established between LoRa Gateway, Fog Router and Sensor
[1] Initialize PQC service with common secret seed
[2] Initialize PQC service with common secret seed
[3] Get Post Quantum Pre-Shared Key (PQPSK)
[4] {PQPSK Key, PQPSK ID}
[5] Generate signature over PQPSK ID using its private key
[6] Append signature to PQPSK ID TLV
[7] Send signed PQPSK ID to Sensor
[8] Validate signature present in PQPSK ID using public key of LoRa Gateway
[9] Get PQPSK Key corresponding to PQPSK ID
[10] PQPSK Key and status
Now both LoRa Gateway and Sensor have the same secret PQPSK Key
[11] Use PQPSK Key as Decryption Key for decrypting Attestation ID
[12] Use PQPSK Key as Encryption Key for encrypting Attestation ID
[13] PQPSK IDs of set of Sensors is downloaded from LoRa Gateway to Fog Router (similar to downloading Attestation IDs of Sensors in a particular area)
Now both LoRa Gateway and Fog Router can use PQPSK ID of Sensor to generate PQPSK Key for decrypting the Attestation ID from the IPv6 EH
[14] Sensor uses this PQPSK Key to encrypt the Attestation ID in IPv6 EH
[15] Fog Router decrypts the Attestation ID using PQPSK Key generated using PQPSK ID of the Sensor

Figure 6: Post-Quantum Key Exchange

Various of the techniques presented herein offer a number of benefits, including:

- Since IoT devices (e.g., IoT sensors, industrial IoT devices, etc.) are prone to several attacks and vulnerabilities, providing trusted IoT deployments in an optimized way by triggering the validation of integrity only whenever an edge/fog devices detect anomaly with the IoT sensors.
- Contributing to providing trusted LoRaWAN deployments.
- Leveraging PQC for providing quantum resistant method to share encryption keys for encrypting attestation information (e.g., Attestation ID/hash ID). Sensors can use a PQPSK key as a post quantum symmetric key to provide data confidentiality (encryption/decryption of data traffic).

Aspects of the techniques presented herein may be further understood through the following "How it Works" overall flow narrative:

1. In the LoRaWAN gateway topology, a LoRa Gateway uses a CoAP-based attestation method to validate the trustworthiness of the IoT Sensor (as explained above in the description of a first element of the techniques presented herein – an attestation method to provide trustworthiness to sensors). As part of that process, an Attestation ID/hash ID is generated and exchanged between the IoT sensor and the LoRa Gateway.

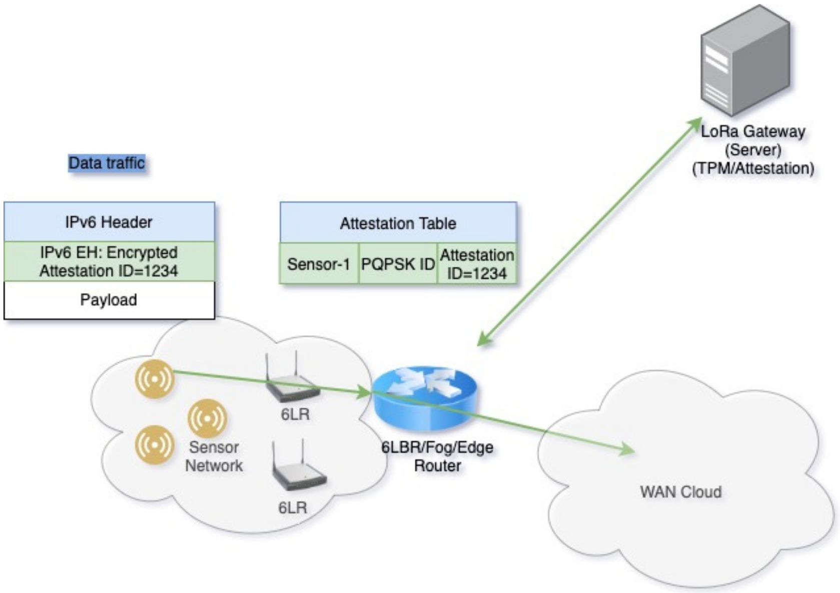
2. The Attestation ID/hash ID will be downloaded to the border nodes (e.g., fog routers, 6LBR, etc.) from the LoRa Gateway using a CoAP extension.

3. The data offload is done selectively so that information that is relevant just to the set of personal area network (PAN) (e.g., a mesh ID) is downloaded to the fog/edge routers.

4. The above can be further downloaded from the 6LBR to a 6LoWPAN Router (6LR) depending on the node capability.

5. Any sensor which forwards data to a remote sensor or cloud outside of the fog network will include the Attestation ID in the IPv6 EH of the data traffic.

6. In support of scale (e.g., volume), different options may be enabled. One such option may indicate when to include the Attestation ID in the IPv6 EH – e.g., on all packets sent from the sensor, selectively on different packets to reduce the bandwidth and power consumption, or an on demand request (using, for example, an IPv6 Neighbor Solicitation (NS) extension) from a fog/edge router to a sensor. See, for example, Figure 7, below.

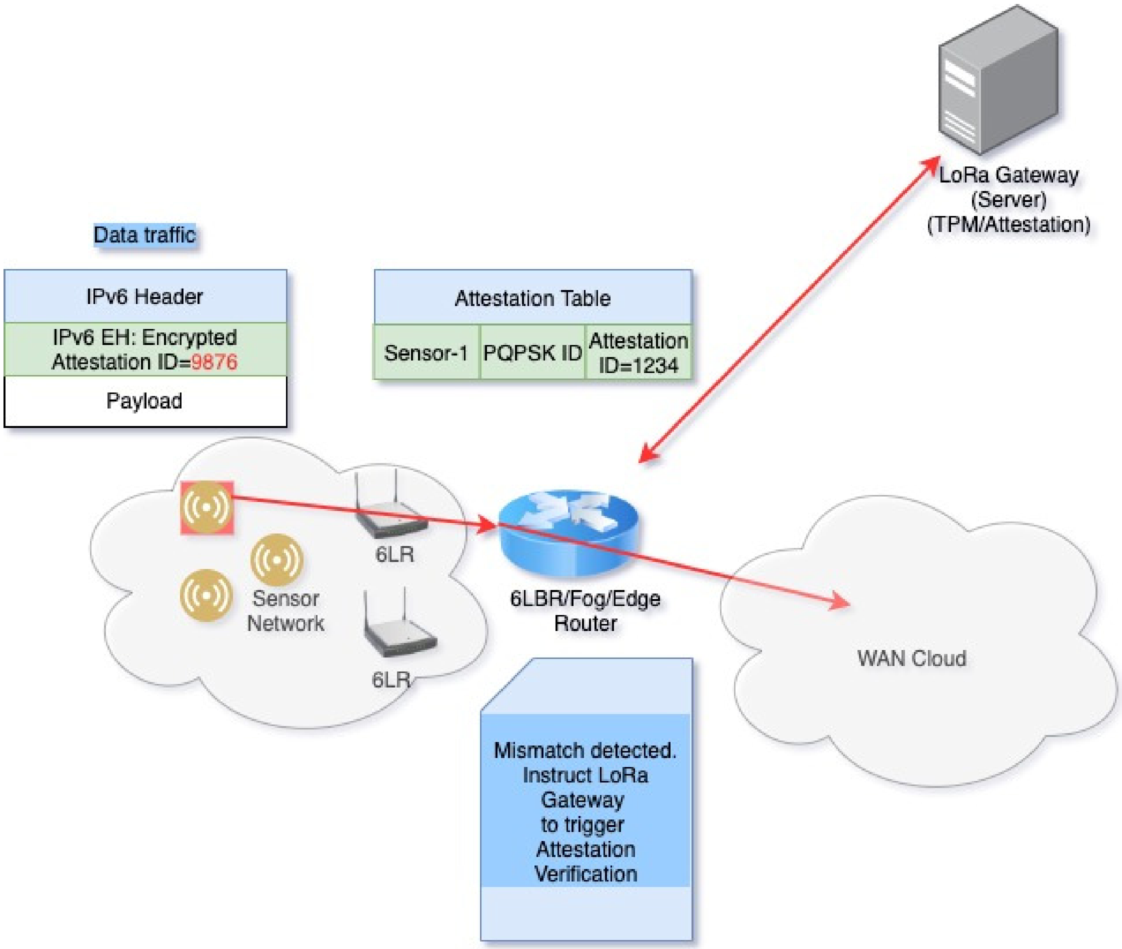


Attestation Table		
Sensor-1	PQPSK ID	Attestation ID=1234

Figure 7: Attestation ID Inclusion in an IPv6 EH

7. Since an IPv6 EH is used as the mechanism to carry the attestation information (Attestation ID/hash ID) man-in-the-middle and replay attacks are quite possible in which the malicious activity in the network can snoop on the EH to retrieve the Attestation ID with malware itself being inserted into the sensor to replay the Attestation ID to remain undetected. To avoid such a situation, the Attestation ID computed should be based on the PQPSK key generated using a quantum resistant method such as a PQC method (as explained above in the description of a second element of the techniques presented herein – a PQPSK generation and sharing mechanism using a PQC method). A PQC method is one way to generate and share quantum resistant keys, but any of the quantum resistant methods (as described above) may be incorporated.

8. In any of the cases, if the sensor device is compromised (e.g., if malicious code is downloaded to the sensor) falsified data may flow through the IoT deployment and/or the functionality of other sensors may be impacted. By employing aspects of the techniques presented herein the Attestation ID/hash ID will change resulting in a mismatch between the value received in an IPv6 data packet (i.e., an EH) and the cached value in fog/edge routers (i.e., nodes). See, for example, Figure 8, below.



Attestation Table		
Sensor-1	PQPSK ID	Attestation ID=1234

Mismatch detected. Instruct LoRa Gateway to trigger Attestation Verification

Figure 8: Attestation ID Mismatch

9. If an edge/fog router detects any such anomaly it will trigger the LoRA Gateway (i.e., a server) to initiate validation of the "proof of integrity" and "freshness of proof of integrity". Until the sensor validity verification is confirmed by the LoRa Gateway, none of the traffic from the sensor will be forwarded to any other IoT devices (i.e., nodes) and it will be dropped by the 6LR/6LBR (as described above in connection with a first element of the techniques presented herein).

As described above, aspects of the techniques presented herein support a scalable solution that, possibly among other things, integrates integrity validation (using an attestation method) in-line with data traffic using a quantum resistant method and detects any compromised sensors quickly to avoid falsified data or a negative impact on other sensors.

In summary, techniques have been presented hat apply an attestation method to the CoAP, which is employed between a LoRa Gateway and sensors, to provide "Proof of Integrity" and "Freshness of Proof of Integrity" (in other words, trustworthiness) to IoT sensor devices. As well, an Attestation ID that is derived during an attestation method is shared in data traffic (i.e., in-band) securely using a post-quantum secure method such as a PQC method.