QKD-TLS Integration Model for End-to-End Secure Communication

Jeongyun Kim
Network Research Division
ETRI
Daejeon, South Korea
jykim@etri.re.kr

Taesang Choi
Network Research Division
ETRI
Daejeon, South Korea
choits@etri.re.kr

Sang-kil Park
Network Research Division
ETRI
Daejeon, South Korea
wideideal@etri.re.kr

***Abstract*— An integration of Quantum Key Distribution and Transport Layer Security is an important research area aimed at addressing the vulnerabilities of traditional cryptographic systems posed by the advancement of quantum computing. QKD leverages the principles of quantum mechanics to provide absolute security, while TLS is a widely used protocol that ensures the confidentiality and integrity of internet communications, offering end-to-end security. The combination of these two technologies is considered a method to enhance TLS by incorporating QKD-generated keys, thereby mitigating potential security risks arising from quantum computing. This paper proposes the QKD-TLS integration models that can support end-to-end secure communication.**

Keywords—integration, quantum key distribution, transport layer security

I. INTRODUCTION

Today's Internet applications are widely adopting Transport Layer Security (TLS) for providing end-to-end security. Internet traffic can be encrypted and securely delivered by using TLS, which is standardized in RFC8449 [x].

With the advent of quantum computers, several of the cryptographic constructs, based on the security model of TLS, will be broken. Quantum-resistant replacements will have to be an alternate approaches to solve this problem.

Quantum Key Distribution (QKD) [1] is a technology, based on the quantum laws of physics, instead of the assumed computational complexity of mathematical problems, to generate and distribute secure cipher keys over unsecured networks such as Internet. It is done by using single photon technology and can detect potential eavesdropping via the quantum bit error rates of the quantum channel. Sending randomly encoded information on single photons produces a shared secret that is a random string and the probabilistic nature of measuring the photon state provides the basis of its security.

QKD network [2] connects a number of point-to-point QKD systems together so that one can develop shared secrets between users. It is a technology that extends the reachability and availability of QKD. The introduction of QKD network into current communication networks and cryptographic infrastructures brings new challenges to the design of the network architecture and security considerations.

Even though QKD can strongly replaces the TLS for quantum-resistant, there is a limitation applying at transport network, not end-point such as mobile object.

In this regard, a integration model combining TLS with QKD, is regarded as one of compromising solutions for end-to-end security.

This paper introduced QKD, QKD network, and TLS. The integration of QKD with TLS is considered a method to enhance end-to-end security, thereby mitigating potential security risks arising from quantum computing. In addition, this paper proposes the QKD-TLS integration models and describes PSK importing in QKD-TLS integration model. Based on this, end-to-end secure communication can be achieved.

II. KEY EXCHANGE OF QUANTUM KEY DISTRIBUTION

A. QKD and QKD Network

A QKD protocol is based on the laws of quantum mechanics. Keys generated by QKD modules implementing the QKD protocol can be consumed by any cryptographic applications using symmetric keys. The basic elements of a QKD are a transmitter (QKD-Tx) and a receiver (QKD-Rx), each of which is referred to as a QKD module. A QKD link connects the QKD modules. The keys are shared via the QKD link. The QKD link usually consists of a quantum link and a classical link. The QKD network is used to extend the range of the QKD system, and it consists of several static quantum nodes that have complete quantum capabilities. As shown in Figure 1, the quantum nodes execute the QKD protocol (e.g., the BB84 protocol) to distribute secure keys (also called QKD keys) between the neighboring nodes, and then, the hop-by-hop manner is adopted in distributing secure keys.

The QKD network (QKDN) consists of a quantum layer, a key management layer, a QKDN control layer and a QKDN management layer [2]. The user network is described by a service layer and a user network management layer.

In the quantum layer, each pair of QKD modules connected by a QKD link generates symmetric random bit strings, which is called as QKD-key. Each QKD module

pushes the QKD-key up to a Key Manager (KM) that is located in the same QKD node.

Key management layer includes KMs and KM links. Each KM is located in a QKD node. The KM performs key management. The KMs are connected via KM links. The KM receives random bit strings from QKD module(s) located in the same QKD node. The KM synchronizes and re-formats these bit strings, and stores them as keys in the storage, which are called KM-keys.

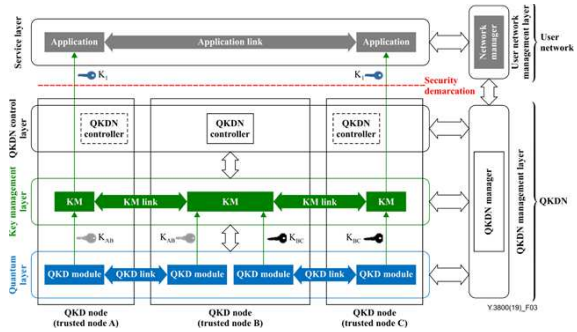

Figure 1 - Conceptual structures of a QKD network

The QKDN controller in control layer includes routing control for key relay, control of QKD links and KM links, session control for QKD services, authentication and authorization control, as well as QoS and charging policy control. A QKDN manager located in management layer monitors and manages the QKD network. Cryptographic applications are located in this layer, supplied with keys from the QKD network and conduct secure communications in application links.

B. QKD Key Exchange Model

Source application starts requesting a list of KM-keys by using a Get_key API. It is assumed that each QKD module in QKD nodes distributed QKD-keys between QKD nodes. In addition, KM-keys are also distributed each KM in QKD nodes.

From Get_key API, Source application is supplied Keys from the KM. The Keys include a list of KM-key identifiers and a list of KM-keys having certain values. The Key_ID_notification only contains some KM-key identifiers, not some KM-keys. Destination application has some Keys from Get_key_with_IDs API, and then both applications finally have same Keys.

III. KEY EXCHANGE OF TRANSPORT LAYER SECURITY

A QKD protocol is based on the laws of quantum mechanics. Keys generated by QKD modules implementing the QKD protocol can be used for symmetric encryption, securing data transmission, and providing a high level of security. QKD allows two parties to generate a secret key, making it secure for communication and protecting data from eavesdropping.

Transport Layer Security (TLS) is a cryptographic protocol that provides end-to-end security of data sent between applications over the Internet, and TLS 1.3, latest version of which, is standardized in RFC 8446 by IETF [4]. It is mostly familiar to users through its use in secure web browsing, and it can also be used for other applications such as e-mail, file transfers, etc.

TLS encrypts data sent over the Internet to ensure that eavesdroppers and hackers are unable to see what you transmit, which is a transaction security standard providing secure connections between two communicating entities.

TLS uses a combination of symmetric and asymmetric cryptography. In the former, data is encrypted and decrypted with a secret key known to both sender and recipient. The latter uses key pairs – a public key and a private key. This allows the public key of the recipient to be used by the sender to encrypt the data they wish to send to them.

It includes five sub-protocols, but handshake protocol is only introduced in the paper. The client and the server in TLS handshake protocol authenticate each other using certificates or pre-shared keys (PSK), instantiate the negotiation of security parameters and compute the session key that is used to encrypt exchanged data. As shown in Figure 2, the handshake can be consisted of three phases:

- Key Exchange: Establish shared keying material and select the cryptographic parameters. Everything after this phase is encrypted.
- Server Parameters: Establish other handshake parameters (whether the client is authenticated, application-layer protocol support, etc.).
- Authentication: Authenticate the server (and, optionally, the client) and provide key confirmation and handshake integrity.

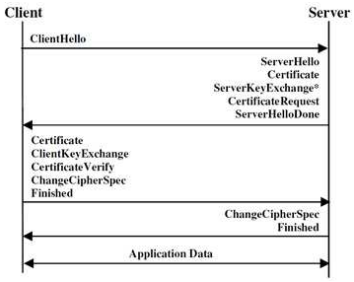

Figure 2 - Message Flow for TLS Handshake

The TLS Handshake process enables the sharing of the "symmetric encryption key" between the client and server so that both parties have the same key. This is where the whole TLS Handshake process comes in and it establishes the encrypted communication between the client and server.

The handshake protocol allows peers to negotiate a protocol version, select cryptographic algorithms, optionally

authenticate each other, and establish shared secret keying material. Once the handshake is complete, the peers use the established keys to protect the application-layer traffic, which is called as session key.

IV. QKD-TLS INEGRATION MODEL

One of compromising solutions for end-to-end security and security enhancement is to use a QKD-TLS integration model, which is a combination of TLS and QKD [3].

A. QKD-TLS Integration Model

For QKD-TLS integration, two kinds of models can be considered. It depends on whether the functions of TLS client/server are extended for QKD, or new functional entity is devised for QKD.

TLS supports three basic key exchange modes:

- (EC)DHE (Diffie-Hellman over either finite fields or elliptic curves)
- PSK(Pre-shared Key)-only
- PSK with (EC)DHE

A cryptographic application enables to request KM-Keys to a KM, as shown Figure 3. For integrating TLS and QKD, the former model utilizes existing PSK and then KM-Keys are mapped to PSK. Get_key API is applied for KM-Keys request between the TLS client/server and the KM, instead of the cryptographic application. In this model, the TLS client/server should have some extension for communicating with the KM, where Get_key API is used.

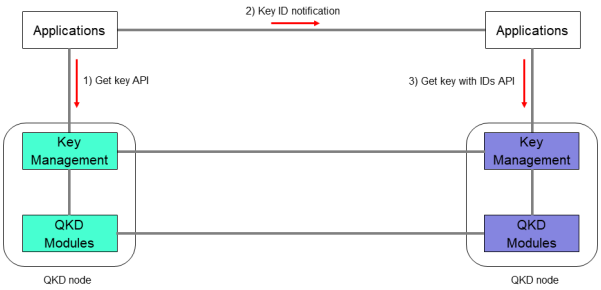

Figure 3 – Message Flow for QKD Key Exchange

In the latter model, the KM-Keys requesting functionality is not included to TLS client/server functions. Therefore, a certain functional entity is necessary to request KM-Keys to the KM, as shown in Figure 4. The functional entity is called to QKD-TLS proxy in this paper. Get_key API is applied for KM-Keys request between the QKD-TLS proxy and the KM.

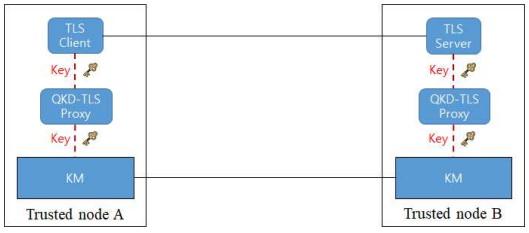

Figure 4 - QKD-TLS Integration Model with Proxy

B. PSK Importing in QKD-TLS Integration Model

Regardless of QKD-TLS integration models, the PSK importing is applied with same process defined in RFC9258 [5].

External PSKs are symmetric secret keys provided to the TLS protocol implementation as external inputs. The Key for QKD is one example of external PSK. Most major TLS implementations support external PSKs. Stacks supporting external PSKs provide interfaces that applications may use when configuring PSKs for individual connections.

For successfully deploying the integration QKD with TLS, the importing KM-keys to TLS client/server is carefully provided. Some guidance can be considered as followings.

- Based on KM-keys sent from TLS client, the KM-keys are finally determined after having negotiation between TLS server and TLS client.
- The KM-keys include a lifetime of each KM-key, priority, and
- The KM-keys is required to be securely delivered from QKD node to TLS client/server.

V. CONCLUSION

This paper proposed the QKD-TLS integration models and describes PSK importing in QKD-TLS integration model. For successfully deploying the integration QKD with TLS, some guidance are considered. Based on this, end-to-end secure communication can be achieved.

ACKNOWLEDGMENT

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00397958, Standardization Lab. for Quantum Information Communications, 50%) and (No. RS-2024-00406071, Development of a quantum channel and existing channel multiplexing system to reduce cost of building quantum cryptography communication equipment, 50%) .

REFERENCES

[1] B. A. Hubermann, B. Lund, and J. Wang, “Quantum Secured Internet Transport,” Information Systems Frontiers, vol. 22, pp. 1561–1567, November 2020

[2] Recommendation ITU-T Y.3800 (2019)/Cor.1 (2020), “Overview on networks supporting quantum key distribution”.

[3] Draft Recommendation ITU-T Y.QKD-TLS(2025), “Quantum Key Distribution integration with Transport Layer Security 1.3”.

[4] E. Rescorla. "The Transport Layer Security (TLS) Protocol Version 1.3" IETF, RFC 8446, 2018.

[5] D. Benjamin and C. A. Wood, “Importing External Pre-Shared Keys (PSKs) for TLS 1.3,” IETF, RFC9258, 2022.