Draft new Recommendation ITU-T Y.QKD-TLS

Quantum Key Distribution integration with Transport Layer Security 1.3

Summary

This Draft Recommendation specifies use cases, high-level requirements and reference models for quantum key distribution (QKD) integration with transport layer security 1.3 (TLS 1.3).

Keywords

Framework, QKDN, TLS 1.3, integration.

Table of Contents

Draft new Recommendation ITU-T Y.QKD-TLS

Quantum Key Distribution integration with Transport Layer Security 1.3

1 Scope

This Draft Recommendation specifies use cases, high-level requirements and reference models for quantum key distribution (QKD) integration with transport layer security 1.3 (TLS 1.3), the scope of this Recommendation is as follows:

- Overview of QKD integration with TLS 1.3.
- Use cases of QKD integration with TLS 1.3.
- High-level requirements of QKD integration with TLS 1.3.
- Reference models of QKD integration with TLS 1.3.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.3800] Recommendation ITU-T Y.3802 (2020), *Overview on networks supporting quantum key distribution.*

[ITU-T TR.QKDN-nq] Draft Technical Report ITU-T TR.QKDN-nq (202x), *Overview for integration of quantum key distribution network with non-quantum cryptographies*.

[ITU-T TR FG QIT4N D2.2] Technical Report ITU-T FG QIT4N D2.2 (2021), *Quantum information technology for networks use cases: Quantum key distribution network.*

[IETF RFC8446] Rescorla, E., *"The Transport Layer Security (TLS) Protocol Version 1.3", RFC8446, August 2018*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 transport layer security (TLS) [IETF RFC8446]: Cryptographic protocols used to provide a secure channel between two communicating peers. The secure channel should provide authentication, confidentiality and integrity.

3.2 Terms defined in this Recommendation

3.2.1 cryptographic application (App): Applications in the service layer to utilize the key pairs provided by the QKDN and perform encrypted communication between remote parties.

NOTE – The definition above is captured based on the description about cryptographic applications in [b-ITU-T Y.3805].

3.2.2 qkd-tls proxy: A functional entity to request Keys to a KM on behalf of TLS client/server.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

QKD Quantum Key Distribution

QKDN Quantum Key Distribution Network

KM Key Manager

TLS Transport Layer Security

5 Conventions

None.

6 Overview

The quantum key distribution network (QKDN) is expected to be able to provide optimized support for a variety of different quantum key distribution (QKD) services. It is assumed that the coverage of the QKD service is limited since it is delivered between network equipment such as OTN rather than between end-devices for example smart phone and servers.

One of the challenges of the QKDN is to support end-to-end the QKD service and integrating QKD with TLS is one of solutions for this.

In this regard, several issues are identified as follows;

- Whether QKD extension is necessary to integrate with TLS 1.3 [IETF RFC8446]
- How to integrate QKD with TLS 1.3 (e.g., pre-shared key encryption) [b-IETF RFC9258]

[Editor's note] If any TLS extension is necessary for integration QKD-TLS, the scope of TLS extension would be identified from Q16/13 perspective. IETF would decide whether TLS extension will be done or not.

The integration of QKD with TLS 1.3 would not require the use of extensions in the Client Hello message for the establishment of keys. The client and the TLS server must fetch the keys from the QKD module using the secure application interface.

First use cases of QKD-TLS integration are derived and then high-level requirements of QKD-TLS integration are described based on the use cases. The reference models of QKD-TLS integration are specified in order to resolve the issues.

7 Use cases of QKD-TLS integration

Figure 1 shows the QKDN's relation to end-to-end cryptography service. The end-to-end encryption can be realized between the cryptographic applications in the user network by applying QKDN or applying the integration of QKDN and non-quantum cryptographies.

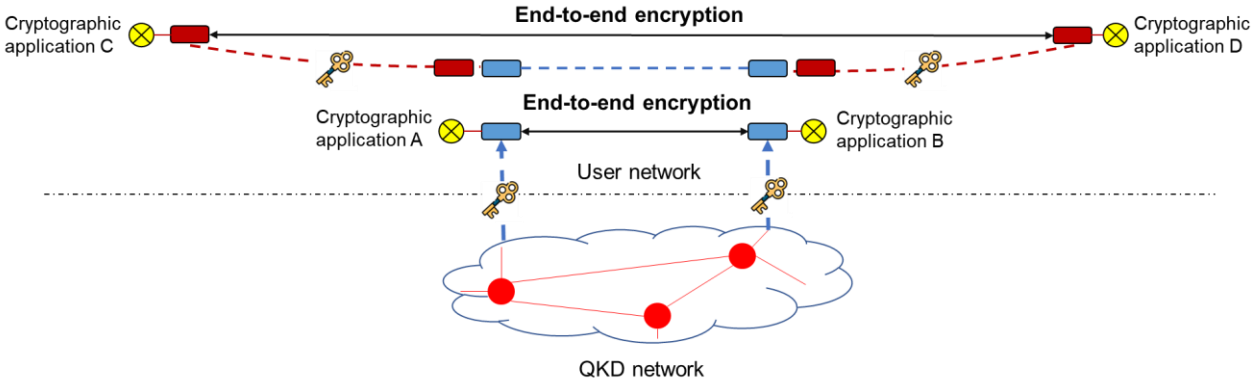

Figure 1. QKDN's relation to end-to-end cryptography service

Furthermore, in a use case of E2E QKD-encrypted model, KSA-keys are delivered to TLS 1.3 (Transport Layer Security 1.3) client and server symmetrically. TLS communication between client and server can be encrypted and decrypted thorough the keys. Therefore, a public-key exchange procedure may not be required, during the initiation process, so called TLS handshake.

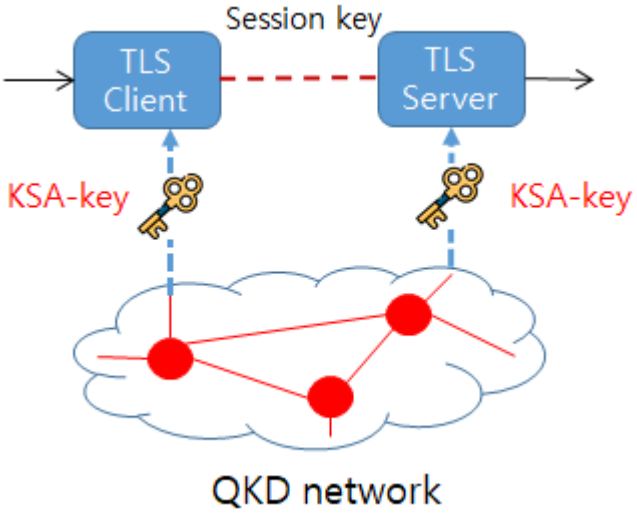

Figure 2. Use case of integration between QKDN and TLS protocol

It is assumed that TLS client/server and QKD module/key manger are located in the same trusted node together. Based on this assumption, the delivery connectivity from QKDN to TLS client/server is considered to be IT-secured.

If the TSL client/server and QKD module/Key Manager are not located in the same Trusted Node together, the connectivity from QKD network to TLS client/server should be secured. For example, Public Key cryptography using PQC algorithms can be used for securing the link between QKD network and TLS client/server against quantum computing attack.

The use cases are categorized into two; QKD and TLS are co-located in a trusted node and separately located.

8 High-level requirements of QKD-TLS integration

(Editor's Note) The meeting has decided to capture the proposed requirements from C-212 and C235 submitted to the interim meetings. However, the meeting needs to further review the requirements for agreement to be included in this clause.

- QKD-TLS integration is required to support the functional entity to integrate QKDN and TLS client/server.

Note: The functional entity refers to the QKD-TLS proxy in Clause 9.

- QKD-TLS integration is required to provide a secure way to deliver the KSA-keys generated by the QKDN to the TLS client/server.
- QKD-TLS integration is required to provide mechanisms for TLS client/server to establish keys with the symmetric keys from QKDN.

9 Reference models of QKD-TLS integration

For QKD-TLS integration, two kinds of reference models are considered. The TLS client/server and the key manager (KM) locate either in the same trusted node or in the different trusted node.

A cryptographic application enables to request Key to a KM. Regarding to Key requestor, the current QKDN Recommendation considered the cryptographic application itself, not TLS client/server. From the TLS client/server perspective, the Key requesting functionality is not included to their functionalities. Therefore, a certain functional entity is necessary to request Key to a KM. The functional entity is called to QKD-TLS proxy in this draft Recommendation.

9.1 Reference model of QKD-TLS integration in the same trusted node

The TLS client/server and the KM locate in the same trusted node. Figure 2 illustrates that the reference model of QKD-TLS integration includes a QKD-TLS proxy, which locates between TLS client and KM. The interfaces between the TLS client/server, QKD-TLS proxy and the KM are considered IT-secure. Therefore, Ak interface protocol is applied for Key request and Key supply at the interface between the QKD-TLS proxy and the KM. The details on the interface between the TLS client/server and the QKD-TLS proxy is out of scope in this draft Recommendation.

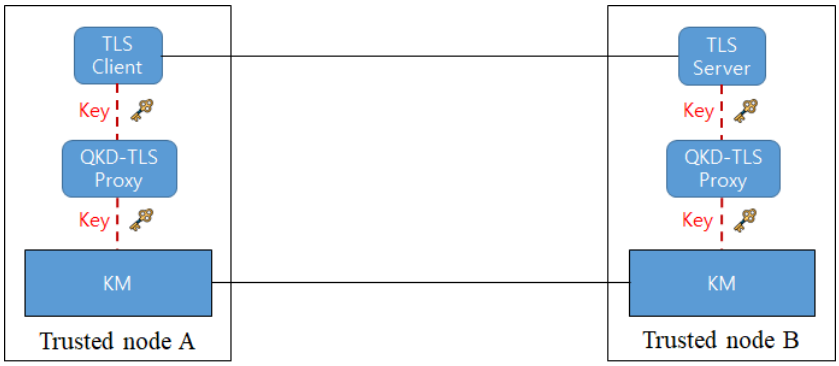

Figure 2. Reference model of QKD-TLS integration in the same trusted node

9.2 Reference model of QKD-TLS integration in the different trusted node

A TLS client/server and a KM locate in the different trusted node. Figure 3 illustrates that the reference model of QKD-TLS integration includes a QKD-TLS proxy, which locates together with TLS client/server. In Figure 3, Keys are delivered from the KM to the QKD-TLS proxy through highly private channels. These highly private channels can be realized by using OTP encryption. On the other hand, Public Key cryptography using PQC algorithms can be used for secure Key delivery as well.

Editor's note: Regarding Public Key cryptography using PQC algorithms, relevant contributions are invited for further study.

The details on Key delivery between the QKD-TLS proxy and the KM is further study.

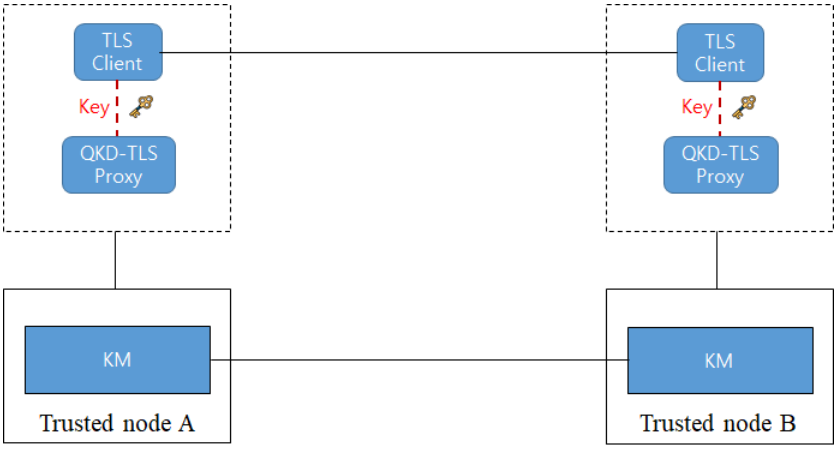

Figure 3. Reference model of QKD-TLS integration in the different trusted node

10 Integration Model for QKD and TLS

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

10.1 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 [IETF RFC8446]:

- (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 KSA-Keys to a KM. For integrating TLS and QKD, the former model utilizes existing PSK and then KSA-Keys are mapped to PSK. Get_key API is applied for KSA-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.

In the latter model, the KSA-Keys requesting functionality is not included to TLS client/server functions. Therefore, a certain functional entity is necessary to request KSA-Keys to the KM, which is called to QKD-TLS proxy. Get_key API is applied for KSA-Keys request between the QKD-TLS proxy and the KM.

10.2 PSK Importing in QKD-TLS Integration Model

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

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 KSA-keys to TLS client/server is carefully provided. Some requirements can be considered as followings.

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

11 Security considerations

TBD

Appendix I

Operational procedures

(This appendix does not form an integral part of this Recommendation.)

This appendix comprises the key distribution and TLS handshake procedures in QKD-TLS integration.

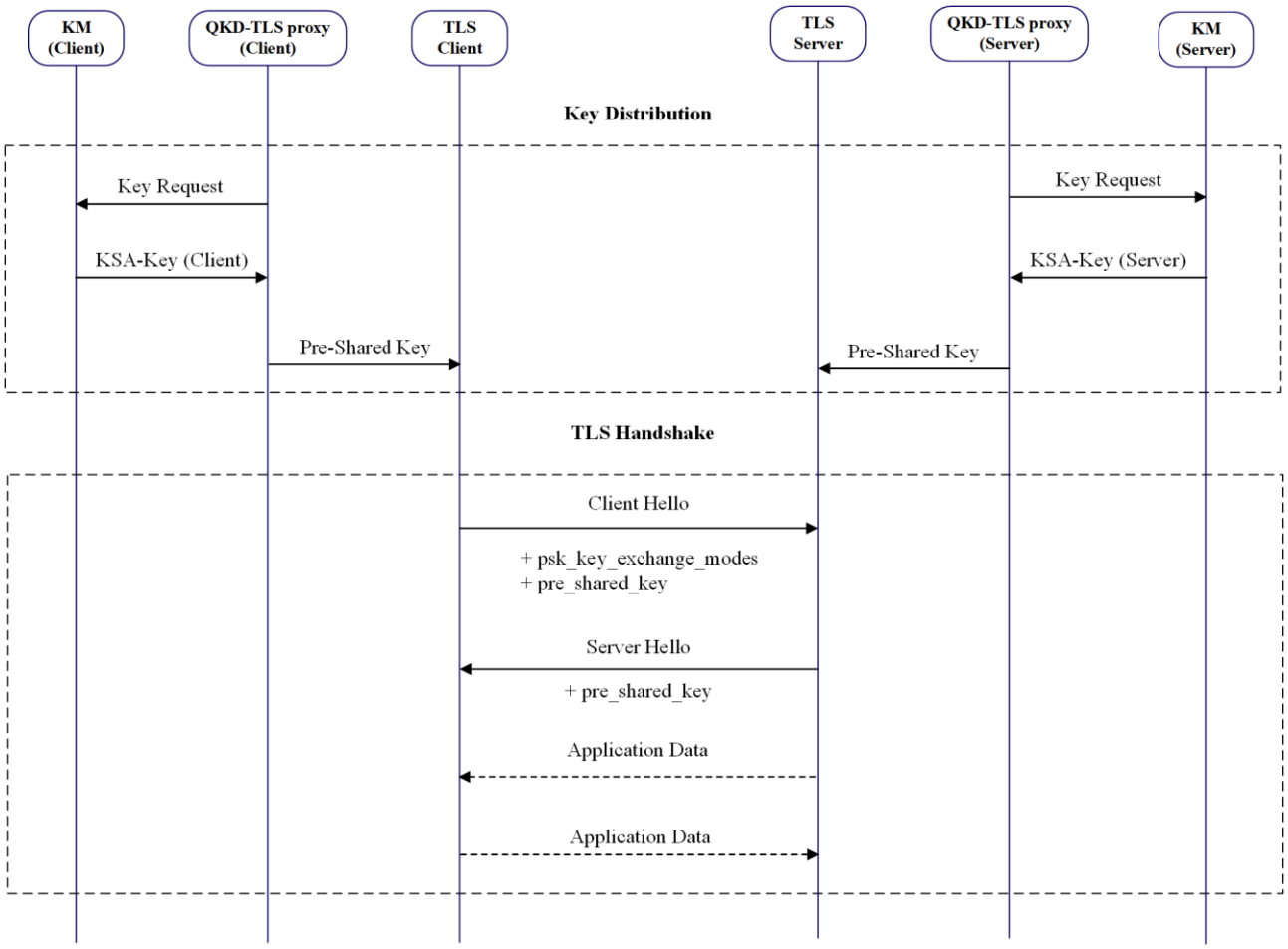

Figure I-1. Key distribution and TLS handshake procedures in QKD-TLS integration

In QKD-TLS integration, the QKDN can supply symmetric keys which will be used in the TLS handshake protocol. There are two main modules in the key exchange procedures of QKD-TLS integration, which are key distribution module and the TLS handshake module.

In the key distribution module, the procedures are shown as follows:

1) The QKD-TLS proxy requests keys from the corresponding KM.

2) The KSA-keys generated by the KM are delivered to the QKD-TLS proxy.

3) The QKD-TLS proxy supports the KSA-keys to the TLS client/server as the pre-shared keys.

Then the TLS client and server obtain a list of pre-shared keys which will be used in the TLS handshake protocol. The details of TLS handshake with PSK-only key establishment are shown as follows:

1) The TLS client sends “Client Hello” message to the TLS server. In “Client Hello” message, the “psk_key_exchange_modes” extension is provided with the value of “psk_ke” which means the PSK-

only key establishment. In the “pre_shared_key” extension, there is a list of PSK identities that the client is willing to negotiate with the server.

2) The TLS server sends “Server Hello” message to the TLS client. The server selects one PSK which will be used for key establishment in the “pre_shared_key” extension.

3) The symmetric key between TLS client and server is established. Then the application data is encrypted by the symmetric key.

Note: The symbol “+” in Figure I-1 indicates noteworthy extensions sent in the previously noted message.

Bibliography

[b-IETF RFC9258] IETF RFC9258 (2022), *Importing External Pre-Shared Keys (PSKs) for TLS 1.3.*

[b-ITU-T TR-XSTR-HYB-QKD] ITU-T Technical Report "Overview of hybrid approaches for key exchange with quantum key distribution" May 2022.

[b-ITU-T Y.3805] Recommendation ITU-T Y.3805 (2021), *Quantum key distribution networks – Software-defined networking control.*

_____