ETSI GS QKD 014 V1.1.1 (2019-02)

GROUP SPECIFICATION

Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API

Disclaimer

The present document has been produced and approved by the Quantum Key Distribution (QKD) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

DGS/QKD-014KeyDeliv

Keywords

API, protocol, quantum cryptography, Quantum Key Distribution

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Quantum Key Distribution (QKD).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The present document describes a communication protocol and data format for a quantum key distribution (QKD) network to supply cryptographic keys to an application. It is intended to allow interoperability of equipment from different vendors. A REST (REpresentational State Transfer) API is specified as a simple, scalable, widely deployed approach that is familiar to a large developer community. The REST-based API specifies the format of the URIs, the communication protocols (HTTPS), and the JSON (JavaScript Object Notation) data format encoding of posted parameters and responses, including key material.

Introduction

QKD networks deliver cryptographic keys to applications. In order to ensure the interoperability of QKD networks, QKD equipment, and applications from different vendors, a specification for a key delivery API from QKD networks to applications is important.

Another Group Specification ETSI GS QKD 004 [i.1] defines an object-based remote function call-style API between applications and QKD key management layer and provides key data streams with QoS functionalities for applications. On the other hand, the present document defines a simpler key delivery API, which is a REST-based API using the HTTPS protocol and data encoded in the JSON format to deliver block keys with key IDs to applications.

REST-based APIs are simple and easy for developers to understand and are popular in many application domains. They have a large developer community and many libraries, implementations, and guidance documents are available to the community. REST-based APIs are lightweight and scale to the "Internet" level regarding both the number of nodes and the number of applications.

It is hoped that this REST-based API specification for key delivery can encourage new entrants/developers into the QKD market, to promote new applications of QKD, and to develop a business ecosystem for QKD.

1 Scope

The present document specifies a communication protocol and data format for a quantum key distribution (QKD) network to supply cryptographic keys to an application.

It is in the form of an API (Application Programming Interface) that allows application developers to make simple method calls to a QKD network and to be delivered key material. It is intended to allow interoperability of equipment from different vendors.

The QKD network can consist of a single link between a single QKD transmitter and a single QKD receiver, or it can be an extended network involving many such QKD links. The API defines a single interface for the delivery of key material to applications in both scenarios. It is beyond the scope of the present document to describe how a QKD network generates key material shared between distant nodes.

A REST (REpresentational State Transfer) API is specified as a simple, scalable, widely deployed approach that is familiar to a large developer community. The REST API specifies the format of the URIs, the communication protocols (HTTPS), and the JSON (JavaScript Object Notation) data format encoding of posted parameters and responses, including key material.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] IETF RFC 7230 (June 2014): "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing".

NOTE: Available at https://www.rfc-editor.org/info/rfc7230.

[2] IETF RFC 7231 (June 2014): "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".

NOTE: Available at https://www.rfc-editor.org/info/rfc7231.

[3] IETF RFC 7235 (June 2014): "Hypertext Transfer Protocol (HTTP/1.1): Authentication".

NOTE: Available at https://www.rfc-editor.org/info/rfc7235.

[4] IETF RFC 5246 (August 2008): "The Transport Layer Security (TLS) Protocol Version 1.2".

NOTE: Available at https://www.rfc-editor.org/info/rfc5246.

[5] ETSI RFC 8446 (August 2018): "The Transport Layer Security (TLS) Protocol Version 1.3".

NOTE: Available at https://www.rfc-editor.org/info/rfc8446.

[6] IETF RFC 8259 (December 2017): "The JavaScript Object Notation (JSON) Data Interchange Format".

NOTE: Available at https://www.rfc-editor.org/info/rfc8259.

[7] IETF RFC 4648 (October 2006): "The Base16, Base32, and Base64 Data Encodings".

NOTE: Available at https://www.rfc-editor.org/info/rfc4648.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS QKD 004 (V1.1.1): "Quantum Key Distribution (QKD); Application Interface".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

Application Programming Interface (API): interface implemented by a software program to be able to interact with other software programs

key: random digital data with an associated universally unique ID

key container: JSON data format containing key data

NOTE: As specified in clause 6.3.

Key Management Entity (KME): entity that manages keys in a network in cooperation with one or more other Key Management Entities

master secure application entity: Secure Application Entity that initiates a request to a Key Management Entity for one or more new keys that can subsequently be requested by a Slave Secure Application Entity specified in the request

QKD Entity (QKDE): entity providing key distribution functionality including acting as an endpoint for the distribution of keys to at least one other QKD Entity using QKD protocols

QKD link: link connecting a pair of QKD Entities

QKD network: network comprised of two or more Trusted Nodes

Quantum Key Distribution (QKD): procedure or method for generating and distributing symmetrical cryptographic keys with information theoretical security based on quantum information theory

Secure Application Entity (SAE): entity that requests one or more keys from a Key Management Entity for one or more applications running in cooperation with one or more other Secure Application Entities

slave secure application entity: Secure Application Entity that initiates a request to a Key Management Entity for one or more keys based on one or more key IDs that were previously delivered to a Master Secure Application Entity

Trusted Node (TN): node containing trusted equipment including one or more Key Management Entities and one or more QKD Entities situated within a security boundary

web API: Application Programming Interface that can be accessed using HTTP or HTTPS protocols

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
HTTP	Hyper Text Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDL	Interface Definition Language
IETF	International Engineering Task Force
JSON	JavaScript Object Notation
KME	Key Management Entity
OMG	Object Management Group
QKD	Quantum Key Distribution
QKDE	QKD Entity
QoS	Quality of Service
REST	REpresentational State Transfer
SAE	Secure Application Entity
TLS	Transport Layer Security
TN	Trusted Node
URI	Uniform Resource Identifier
URL	Unified Resource Locator
UTF-8	UCS Transformation Format 8 bits
UUID	Universally Unique Identifier

4 Key delivery API Specification Overview

This key delivery API is a REST-based API, a simple request and response style API between a SAE and a KME. SAEs request KMEs to deliver keys and KMEs deliver the keys. Calls to the API on a KME are intended to be made by SAEs with a point of presence within the same secure site as the KME they are connecting to. The QKD network may deliver common shared keys to SAEs in different sites.

Optical switches, encryption modules, and security management systems are examples of SAEs.

Keys are generated and shared securely with QKD technology by KMEs. Key management methods used by KMEs and how KMEs relay keys securely in a QKD network is outside the scope of the present document.

A QKD network can consist of a single QKD link, or it can be an extended network involving many such QKD links. An example of a QKD network is shown in Figure 1. Installing and configuring a QKD network, registering a new trusted node or QKD link into a QKD network and removing them from a QKD network are QKD network management issues and outside the scope of the present document.

Each KME shall have one or multiple QKDEs to connect with other KMEs via QKD links. KMEs shall be able to distribute keys to other KMEs. In each Trusted Node, there shall be at least one KME. One or multiple SAEs may connect with a KME within a Trusted Node. It is assumed that each Trusted Node is securely operated and managed. Each trusted node shall be located in its site. SAEs shall be located with its connected KMEs in its site. The API between SAE and KME shall be used within a security boundary in each site.

KMEs shall provide Web API server functionality to deliver keys to SAEs via HTTPS protocols.

Each KME shall have a unique ID (KME ID). A KME ID shall be unique in a QKD network. The format and the assignment of KME IDs is outside the scope of the present document.

SAEs make HTTPS requests to KMEs to get keys and status information.

Each SAE shall have a unique ID (SAE ID). A SAE ID shall be unique in a QKD network. The format and the assignment of SAE IDs is outside the scope of the present document.

All communications between SAE and KME shall use the HTTPS protocols (with TLS version 1.2 or higher) (IETF RFC 7230 [1], IETF RFC 7231 [2], IETF RFC 7235 [3], IETF RFC 5246 [4], IETF RFC 8446 [5]).

KMEs shall authenticate each request and identify the unique SAE ID of the calling SAE.

Data in the message body of HTTPS requests from SAE to KME and HTTPS responses from KME to SAE shall be encoded in JSON format as per IETF RFC 8259 [6].

The SAE making an initial "Get key" request is referred to as the Master SAE for the key(s) returned. An SAE making a subsequent "Get key with key IDs" request is called the Slave SAE for the key(s) returned.

Applications shall communicate Key IDs between SAEs as necessary for their operations but how they do so is outside the scope of the present document.

The present document makes the following security assumptions about the use of this API with a QKD network:

- each Trusted Node is securely operated and managed;
- this API is used between SAEs and KMEs within a secure site;
- each SAE is secure;
- each KME is secure.

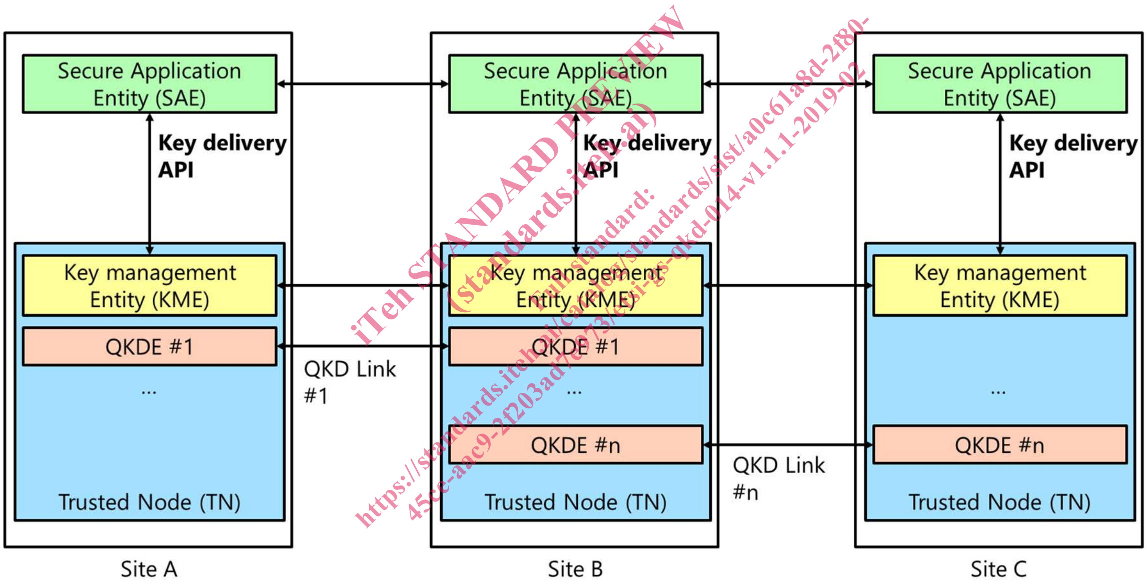

Figure 1: Example of QKD network

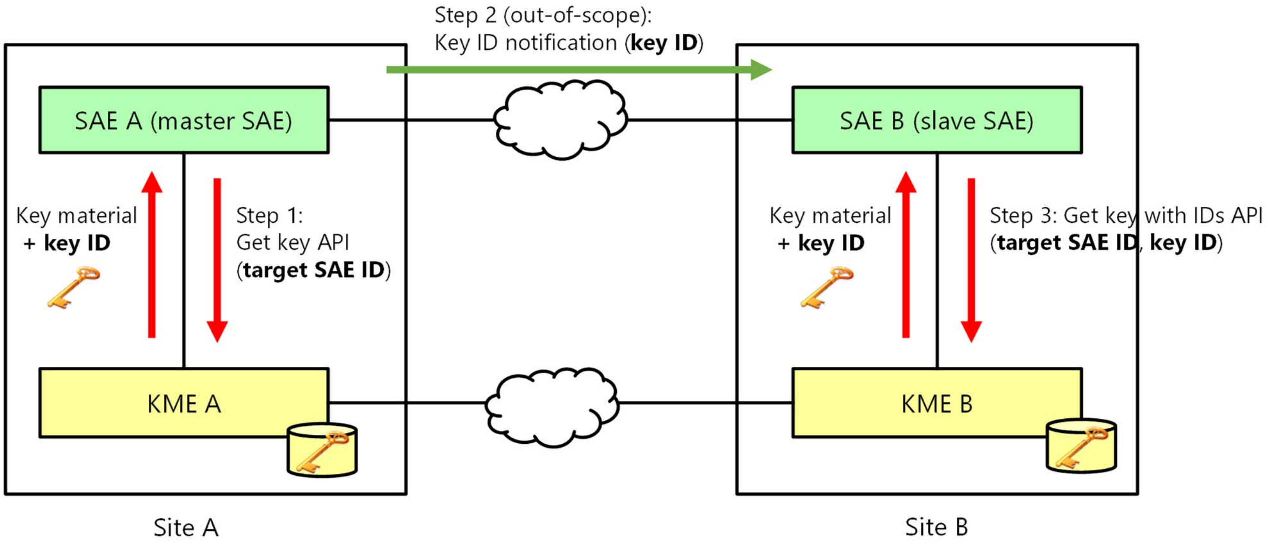

Figure 2: Use-case of key delivery API

Figure 2 shows a use-case illustrating a way the key delivery API can be used. KME A and KME B are either connected by a direct QKD link or a QKD network comprising multiple QKD links. SAE A is connected to KME A. SAE B is connected to KME B.

KME A and KME B exchange and store keys and each key delivered is assigned a universally unique ID. SAE A (master SAE) can initiate secure communication with SAE B (slave SAE) according to the following steps:

- Step 1: SAE A calls the key delivery API method "Get key" with the SAE B ID of the slave SAE to get keys from KME A. KME A delivers to SAE A key materials with the associated key IDs that are (to be) shared with KME B.
- Step 2: SAE A notifies SAE B of the key IDs. This communication between master SAE and slave SAE is outside the scope of the present document.
- Step 3: SAE B calls the key delivery API method "Get key with key IDs" with the SAE A ID of the master SAE and the notified key IDs information to get the identical keys from KME B. KME B delivers to SAE B the identical key materials with the identical associated key IDs that are shared with KME A.

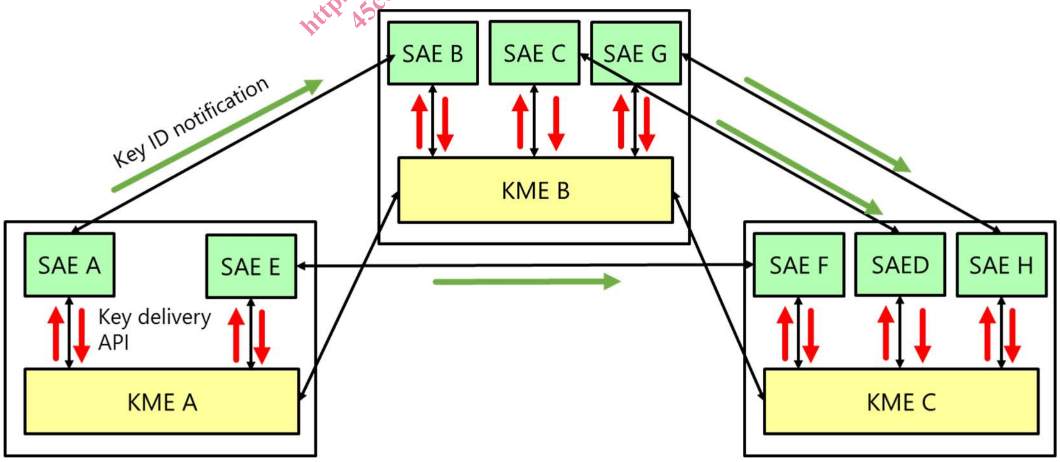

Figure 3: Use-case illustrating multiple SAEs connecting to each KME

Figure 3 shows another use-case illustrating how the key delivery API can be used. Multiple SAEs are connected to a single KME.