INTERNATIONAL TELECOMMUNICATION UNION
TELECOMMUNICATION STANDARDIZATION SECTOR
STUDY PERIOD 2022-2024

SG11-TD712/GEN
STUDY GROUP 11
Original: English

Question(s): 2/11 Geneva, 10-20 October 2023

TD

Source: Editors

Title: Consent – revised baseline text of draft Recommendation ITU-T Q.4164 (ex. Q.QKDN_Ck): Protocols for Ck interface for quantum key distribution network

Contact:	Kaoru KENYOSHI NICT Japan	E-mail: kaoru.kenyoshi@nict.go.jp
Contact:	Mariko Honda NICT Japan	E-mail: mariko.honda@ntt-at.co.jp
Contact:	Xuefu Wang QuantumCTek Co., Ltd. China	E-mail: xuefu.wang@quantum-info.com
Contact:	Zhangchao Ma CAS Quantum Network Co., Ltd. China	E-mail: mazhangchao@qtict.com
Contact:	Junsen Lai CAICT, Ministry of Industry and Information Technology (MIIT) China	E-mail: laijunsen@caict.ac.cn

Abstract: This TD is the output of draft Recommendation ITU-T Q.4164 (ex. Q.QKDN_Ck): Protocols for Ck interface for quantum key distribution network (for consent) at the Q2/11 meeting (Geneva, 10-20 October 2023).

Summary

This TD is the outcome of revised draft Recommendation ITU-T Q.4164 (ex. Q.QKDN_Ck): Protocols for Ck interface for quantum key distribution network (for consent), based on the discussion results on input documents C229 and C337 with modifications at the Q2/11 meetings (Geneva, 10-20 October 2023).

Draft Recommendation ITU-T Q.4164 (ex. QKDN_Ck)

Protocols for Ck interface for quantum key distribution network

Summary

Recommendation ITU-T Q.4164 specifies protocols for Ck interface in quantum key distribution network (QKDN).

Keywords

Protocol, QKD (quantum key distribution), QKDN (QKD network), signalling procedure, message parameters.

Table of Contents

1. Scope 5
2. References 5
3. Definitions 5
3.1. Terms defined elsewhere 5
3.2. Terms defined in this Recommendation 6
4. Abbreviations and acronyms 6
5. Conventions 7
6. Ck interface 7
7. Signalling procedure 7
7.1. Signalling procedures for key relay request in a distributed QKDN 7
7.2. Signalling procedures for key relay request in a centralized QKDN 8
7.3. Signalling procedures for session creation request 8
7.4. Signalling procedures for session creation notification 9
7.5. Signalling procedures for key reservation request 9
7.6. Signalling procedures for key allocation request 10
8. Signalling messages and parameters 10
8.1. Key relay next hop request message 10
8.2. Response to key relay next hop request message 11
8.3. Key relay request notification message 11
8.4. Key relay request message 12
8.5. Response to key relay request message 13
8.6. Session creation request message 13
8.7. Response to session creation request message 13
8.8. Session creation notification message 14
8.9. Response to session creation notification message 15
8.10 Key reservation request 15
8.11 Response to key reservation request 15

8.12 Key allocation request 16

8.13 Response to key allocation request 16

9. Security considerations 17

Appendix I Protocol implementation using TCP 18

Appendix II Protocol implementation using gRPC 20

Ⅱ. 1 Key relay next hop request and response message ..... **Error! Bookmark not defined.**

Ⅱ.2 Completion of key relay notification and response message 20

Ⅱ.3 Key relay request and response message 20

Ⅱ.4 Key reservation request and response message 21

Ⅱ.5 Key allocation request and response message 22

Bibliography 23

Draft Recommendation ITU-T Q.4164

Protocols for Ck interface for quantum key distribution network

1. Scope

This Recommendation specifies protocols at Ck interface for quantum key distribution network (QKDN) especially in the following areas.

- signalling procedures for Ck interface for QKDN;
- signalling messages and parameters for Ck interface for QKDN;
- security considerations

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 Q.4160] Recommendation ITU-T Q.4160 (2023), *Quantum key distribution networks - Protocol framework.*

[ITU-T X.1712] Recommendation ITU-T X.1712 (2021)/Cor.1 (02/2022), *Security requirements and measures for quantum key distribution networks - key management*.

[ITU-T Y.3800] Recommendation ITU-T Y.3800 (2019)/Cor.1 (04/2020), *Overview on networks supporting quantum key distribution*.

[ITU-T Y.3802] Recommendation ITU-T Y.3802 (2020)/Cor.1 (04/2021), *Quantum key distribution networks - Functional architecture*.

3. Definitions

3.1. Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 **key management** [ITU-T Y.3800]: All activities performed on keys during their life cycle starting from their reception from the quantum layer, the storage, formatting, relay, synchronization, authentication, to supply to cryptographic application and delete or preserve depending on the key management policy.

3.1.2 **key management agent (KMA)** [ITU-T Y.3802]**:** A functional element to manage keys generated by a quantum key distribution (QKD) module/QKD modules in a QKD node (trusted node).

NOTE - KMA acquires keys from a QKD module/QKD modules, synchronizes, resize, formats, and stores them. It also relays keys through key management agent (KMA) links.

3.1.3 **key management agent link (KMA link)** [ITU-T Y.3802]**:** A communication link connecting KMAs to perform key relay and communications for key management.

3.1.4 **key manager (KM)** [ITU-T Y.3800]: A functional module located in a quantum key distribution (QKD) node to perform key management in the key management layer.

3.1.5 **key relay** [ITU-T Y.3800]: A method to share keys between arbitrary quantum key distribution (QKD) nodes via intermediate QKD node(s).

3.1.6 **key supply agent (KSA)** [ITU-T Y.3802]: A functional element to supply keys to a cryptographic application, being located between a key management agent (KMA) and the client.

NOTE - Application interfaces for cryptographic applications are installed into the key supply agent (KSA). The KSA synchronizes keys, and verifies their integrity via a KSA link before supplying them to the client.

3.1.7 **key supply agent link (KSA link)** [ITU-T Y.3802]: A communication link connecting KSAs to perform key synchronization and integrity verification.

3.1.8 **quantum key distribution (QKD)** [b-ETSI GR QKD 007]: Procedure or method for generating and distributing symmetrical cryptographic keys with information theoretical security based on quantum information theory.

3.1.9 **quantum key distribution link (QKD link)** [ITU-T Y.3800]: A communication link between two quantum key distribution (QKD) modules to operate the QKD.

NOTE – A QKD link consists of a quantum channel for the transmission of quantum signals, and a classical channel used to exchange information for synchronization and key distillation.

3.1.10 **quantum key distribution module (QKD module)** [ITU-T Y.3800]: A set of hardware and software components that implements cryptographic functions and quantum optical processes, including quantum key distribution (QKD) protocols, synchronization, distillation for key generation, and is contained within a defined cryptographic boundary.

NOTE – A QKD module is connected to a QKD link, acting as an endpoint module in which a key is generated. These are two types of QKD modules, namely, the transmitters (QKD-Tx) and the receivers (QKD-Rx).

3.1.11 **quantum key distribution network (QKDN)** [ITU-T Y.3800]: A network comprised of two or more quantum key distribution (QKD) nodes connected through QKD links.

NOTE – A QKDN allows sharing keys between the QKD nodes by key relay when they are not directly connected by a QKD link.

3.1.12 **quantum key distribution network controller (QKDN controller)** [ITU-T Y.3800]: A functional module, which is located in a quantum key distribution (QKD) network control layer to control a QKD network.

3.1.13 **quantum key distribution node (QKD node)** [ITU-T Y.3800]: A node that contains one or more quantum key distribution (QKD) modules protected against intrusion and attacks by unauthorized parties.

NOTE – A QKD node can contain a key manager (KM).

3.2. Terms defined in this Recommendation

None.

4. Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CNCF Cloud Native Computing Foundation

ID	IDentifier
KM	Key Manager
KMA	Key Management Agent
KSA	Key Supply Agent
QKD	Quantum Key Distribution
QKDN	Quantum Key Distribution Network
RPC	Remote Procedure Call
TCP	Transmission Control Protocol

5. Conventions

None.

6. Ck interface

Ck is a reference point connecting a QKDN controller control and management function in a QKDN controller and a KM control and management function in a key manager (KM). It is responsible for the QKDN controller to communicate control information with a key management agent (KMA) and a key supply agent (KSA).

7. Signalling procedure

Examples of signalling procedure of key request, key relay, and key supply in QKDN are described in the Appendix I of [ITU-T Q.4160]. The protocol suites applied for the signalling are specified in clause 7 of [ITU-T Q.4160]. Two kinds of signalling procedures are defined depending on the network architecture of distributed QKDN and centralized QKDN.

7.1. Signalling procedures for key relay request in a distributed QKDN

The distributed QKDN performs key relay with a series of hops between KMs to the destination KM. At the Ck interface of the distributed QKDN, a KM requests a QKDN controller for the information of the KM of neighbours for the next hop for key relay and the KM relays the key to the next KM based on the controller's response. Then the next KM requests again to the controller for the next hop. This request and hop procedure repeats until the key relay is completed to the destination.

Figures 1 shows signalling procedures for key relay request in a distributed QKDN.

The KM sends Key relay next hop request to the QKDN controller in order to request the KM identifier (ID) of the next hop. The QKDN controller responds possible KMs with the KM IDs to hop to the next KM.

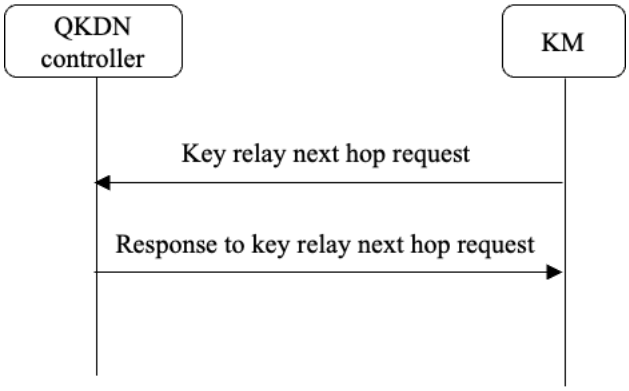

Figure 1 - Signalling procedures for key relay request in a distributed QKDN

7.2. Signalling procedures for key relay request in a centralized QKDN

In a centralized QKDN, if a key relay is needed after receiving a key request from the cryptographic application, the KM can request a key relay route to the QKDN controller, and the QKDN controller returns to the requesting KM with the whole route to the destination KM (list of KMs to be passed). When all the key relays are completed, the source KM returns notification that the key relay is completed.

Figures 2 shows signalling procedures for key relay request in a centralized QKDN.

After the cryptographic application sends a key request to the KM, if a key relay is needed and there is no available key relay route at the KM, the KM requests a key relay route to the QKDN controller with Key relay request notification. The QKDN controller specifies the whole route for the key relay to the destination KM and notifies the requesting KM of the list of transit KMs by a Key relay request. The source KM starts the key relay according to the list of transit KMs. When the key relay completes at the KM to which the destination cryptographic application is connected, the source KM notifies the QKDN controller of completion of key relay by Response to key relay request.

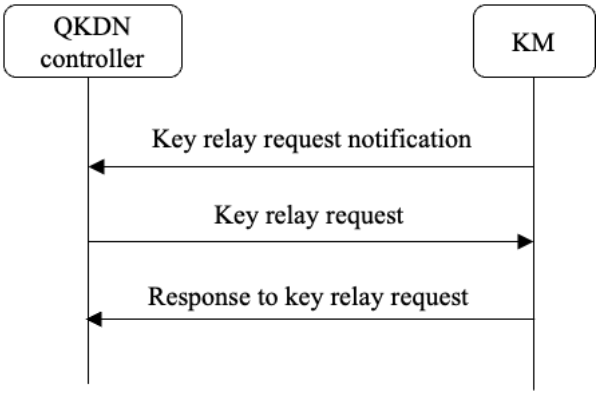

Figure 2 - Signalling procedures for key relay request in a centralized QKDN

7.3. Signalling procedures for session creation request

To facilitate the key supply between the cryptographic applications and the KMs at both sides, the source KM can send a session creation request to the corresponding QKDN controller, which then generates a session ID and notifies the destination KM with the session ID to create a session. After receiving the session creation result, the QKDN controller responds to the source KM with the session ID.

Figures 3 shows signalling procedures for session creation request.

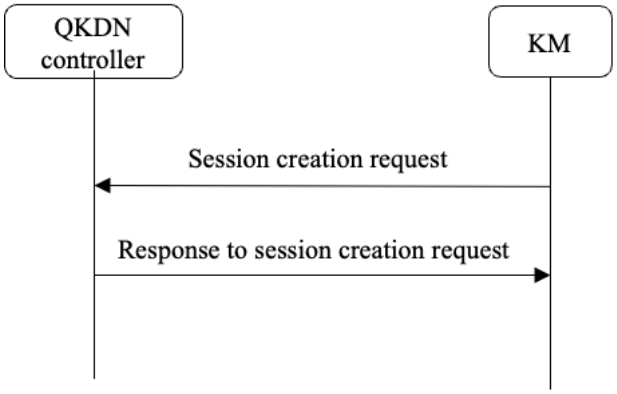

Figure 3 - Signalling procedures for session creation request

7.4. Signalling procedures for session creation notification

The corresponding QKDN controller can send a session creation notification with the session ID to the destination KM, which then notifies the destination cryptographic application with the received session ID for a session creation. After receiving the session creation result from the destination cryptographic application, the destination KM responds to the corresponding QKDN controller with the session ID.

Figures 4 shows signalling procedures for session creation notification.

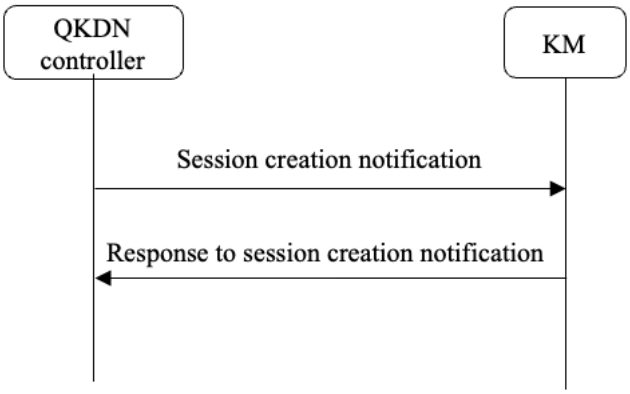

Figure 4 - Signalling procedures for session creation notification

7.5. Signalling procedures for key reservation request

The QKDN controller can send a key reservation request to the KM with KMA identifier (ID) to reserve the key that will be relayed to destination KM. After receiving the key reservation request, the KM responds to the corresponding QKDN controller by Response to key reservation request with KMA-key ID reserved and Result code.

Figures 5 shows signalling procedures for key reservation request.

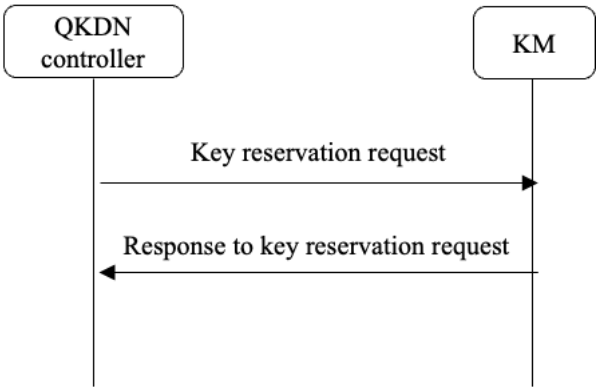

Figure 5 - Signalling procedures for key reservation request

7.6. Signalling procedures for key allocation request

The corresponding QKDN controller can send a key allocation request with KMA ID、KMA-key IDs to allocate key resource which has been reserved. After receiving the key allocation request, the KM responds to the corresponding QKDN controller with Result code. Figures 6 shows signalling procedures for key allocation request.

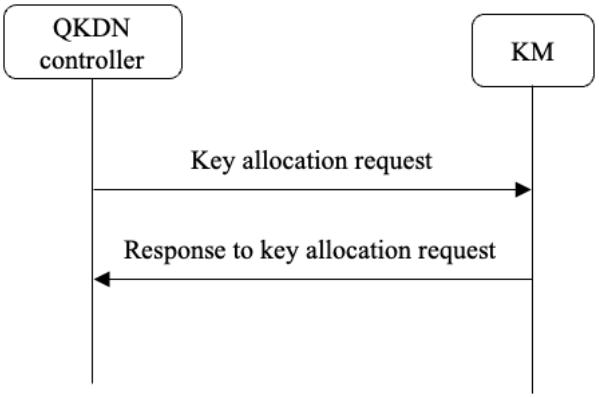

Figure 6 - Signalling procedures for key allocation request

8. Signalling messages and parameters

This clause specifies messages and their parameters for the Ck interface.

In the M/O column of the tables in this clause, M indicates that the parameter is mandatory for signalling, and O indicates that the parameter is optional for signalling.

The messages and parameters defined in this clause are independent of a specific protocol and can have different implementations. The recommended protocol implementations are described in Appendix I and II.

NOTE – A message parameter described in the following tables is not necessarily mapped to a field in the message payload and might be a part of control parameters of one specific protocol. The Data type column of the tables may vary with specific protocols.

8.1. Key relay next hop request message

Table 1 shows parameters of Key relay next hop request message. Either Destination KMA ID or Application destination ID is mandatory to specify the destination.

Table 1 - Parameters of Key relay next hop request message

Parameter	Description	Data type	M/O	Remarks
Source KMA ID	ID of KMA that is the source in the entire key relay route	string	O	
Destination KMA ID	ID of KMA that is the destination in the entire key relay route	string	Either Destination KMA ID or Application destination ID is mandatory	
Application destination ID	ID of the cryptographic application with which the source cryptographic application (i.e. source application) requests to communicate	string	Either Destination KMA ID or Application destination ID is mandatory	
Extension	Array of extension parameters	Array of objects	O	

8.2. Response to key relay next hop request message

Table 2 shows parameters of Response to key relay next hop request message. For a distributed QKDN, the QKDN controller returns the IDs of the possible KMs to reach the destination KMA, with or without the destination KMA ID.

Table 2 - Parameters of Response to key relay next hop request message

Parameter	Description	Data type	M/O	Remarks
Source KMA ID	ID of KMA that is the source in the entire key relay route	string	O	
Destination KMA ID	ID of KMA that is the destination in the entire key relay route	string	Mandatory if Application destination ID is contained in the Key relay next hop request message.	
Next KMA IDs	IDs of KMAs available as a next relay hop to relay the keys to the destination KMA	Array of string	M	
Extension	Array of extension parameters	Array of objects	O	

8.3. Key relay request notification message

Table 3 shows parameters of Key relay request notification message. In a centralized QKDN, after receiving a key request sent by a cryptographic application, if a key relay is needed, the KM can request the whole route of the key relay to the QKDN controller. At this time, as same in the case of

the distributed QKDN, information is required for either the Destination KMA ID or the Application destination ID in order to specify the destination of the key relay.

Table 3 - Parameters of Key relay request notification message

Parameter	Description	Data type	M/O	Remarks
Source KMA ID	ID of KMA that is the source in the entire key relay route	string	O	
Destination KMA ID	ID of KMA that is the destination in the entire key relay route	string	Either Destination KMA ID or Application destination ID is mandatory	
Application destination ID	ID of the cryptographic application with which the source cryptographic application (i.e. source application) requests to communicate	string	Either Destination KMA ID or Application destination ID is mandatory	
Extension	Array of extension parameters	Array of objects	O	

8.4. Key relay request message

Table 4 shows parameters of Key relay request message. The QKDN controller returns all KMs in the route (Transit KMA IDs) to reach the destination KMA, with or without the destination KMA ID.

Table 4 - Parameters of Key relay request message

Parameter	Description	Data type	M/O	Remarks
Source KMA ID	ID of KMA that is the source in the entire key relay route	string	O	
Destination KMA ID	ID of KMA that is the destination in the entire key relay route	string	Mandatory if Application destination ID is contained in the key relay request notification message.	
Transit KMA IDs	List of IDs of KMAs that are the transition nodes of key relay route	string	M	
Key relay request ID	ID of key relay request	string	O	
Extension	Array of extension parameters	Array of objects	O	

8.5. Response to key relay request message

Table 5 shows parameters of Response to key relay request message.

Table 5 - Parameters of Response to key relay request message

Parameter	Description	Data type	M/O	Remarks
Response	Result of key relay	string	M	Success or failure reason
Key relay request ID	ID of key relay request	string	O	
Extension	Array of extension parameters	Array of objects	O	

8.6. Session creation request message

Session creation request message is sent from the source KM to the corresponding QKDN controller. A session can be created to facilitate the key supply between the cryptographic applications and the KMs at both sides.

Table 6 shows parameters of Session creation request message.

Table 6 – Parameters of Session creation request message

Parameter	Description	Data type	M/O	Remarks
Application source ID	ID of the source cryptographic application (i.e., the application which connects to the source KM to receive KSA-keys)	string	M	
Application destination ID	ID of the destination cryptographic application (i.e., the application with which the source cryptographic application requests to communicate)	string	M	
Number of keys	Number of KSA-keys requested	integer	O	A default value is applied if omitted. This parameter can be used as “maximum number of KSA-keys requested during one session”.
Extension	Array of extension parameters	Array of objects	O	

8.7. Response to session creation request message

Response to session creation request message is sent from the corresponding QKDN controller to the source KM. After receiving a session creation request, the QKDN controller generates a session ID and notifies the destination KM with the session ID to create a session. After receiving the session creation result, the QKDN controller responds to the source KM with the session ID.

Table 7 shows parameters of Response to session creation request message.

Table 7 – Parameters of Response to session creation request message

Parameter	Description	Data type	M/O	Remarks
Session ID	ID of the session created for key supply	string	M	
Response	Result of the creation of the session	string	M	Success or failure reason
Destination KM ID	ID of the destination KM	string	M	
Extension	Array of extension parameters	Array of objects	O	

8.8. Session creation notification message

Session creation notification message is sent from the corresponding QKDN controller to the destination KM. The destination KM can notify the destination cryptographic application with the received session ID for a session creation.

Table 8 shows parameters of Session creation notification message.

Table 8 – Parameters of Session creation notification message

Parameter	Description	Data type	M/O	Remarks
Application source ID	ID of the source cryptographic application (i.e., the application which connects to the source KM to receive KSA-keys)	string	M	
Application destination ID	ID of the destination cryptographic application (i.e., the application with which the source cryptographic application requests to communicate)	string	M	
Session ID	ID of the session created for key supply	string	M	
Source KM ID	ID of the source KM	string	M	
Number of keys	Number of KSA-keys requested	integer	O	A default value is applied if omitted. This parameter can be used as “maximum number of KSA-keys requested during one session”.
Extension	Array of extension parameters	Array of objects	O	

8.9. Response to session creation notification message

Response to session creation notification message is sent from the destination KM to the corresponding QKDN controller. The destination KM responds to the corresponding QKDN controller with the session creation result.

Table 9 shows parameters of Response to session creation notification message.

Table 9 – Parameters of Response to session creation notification message

Parameter	Description	Data type	M/O	Remarks
Session ID	ID of the session created for key supply	string	M	
Response	Result of the creation of the session	string	M	Success or failure reason
Extension	Array of extension parameters	Array of objects	O	

8.10. Key reservation request

Table 10 shows parameters of Key reservation request.

Table 10 – Parameters of Key reservation request message

Parameter	Description	Data type	M/O	Remarks
Source KMA ID	ID of KMA that is the source in the entire key relay route	String	M	
Destination KMA ID	ID of KMA that is the destination in the entire key relay route	String	M	
Application destination ID	ID of the destination cryptographic application (i.e., the application with which the source cryptographic application requests to communicate)	String	O	
Extension	Array of extension parameters	Array of objects	O	

8.11. Response to key reservation request

Table 11 shows parameters of Response to key reservation request message.

Table 11 – Parameters of Response to key reservation request message

Parameter	Description	Data type	M/O	Remarks
Source KMA ID	ID of KMA that is the source in the entire key relay route	string	O	
Destination KMA ID	ID of KMA that is the destination in the entire key relay route	string	O	
Application destination ID	ID of the destination cryptographic application (i.e., the application with	string	O	

	which the source cryptographic application requests to communicate)			
Key IDs	IDs of KMA-keys reserved	string	M	
Response	Result of key reservation request	integer	M	
Extension	Array of extension parameters	Array of objects	O	

8.12. Key allocation request

Table 12 shows parameters of Key allocation request.

Table 12 – Parameters of Key allocation request message

Parameter	Description	Data type	M/O	Remarks
Source KMA ID	ID of KMA that is the source in the entire key relay route	string	M	
Destination KMA ID	ID of KMA that is the destination in the entire key relay route	string	M	
Application destination ID	ID of the destination cryptographic application (i.e., the application with which the source cryptographic application requests to communicate)	string	O	
Key IDs	IDs of KMA-keys reserved	Array of objects	M	
Extension	Array of extension parameters	Array of objects	O	

8.13. Response to key allocation request

Table 13 shows parameters of Response to key allocation request message.

Table 13 – Parameters of Response to key allocation request message

Parameter	Description	Data type	M/O	Remarks
Source KMA ID	ID of KMA that is the source in the entire key relay route	string	O	
Destination KMA ID	ID of KMA that is the destination in the entire key relay route	string	O	
Application destination ID	ID of the destination cryptographic application (i.e., the application with which the source cryptographic application requests to communicate)	string	O	
Key IDs	IDs of KMA-keys reserved	Array of objects	O	
Response	Result of key allocation request	integer	M	

Extension	Array of extension parameters	Array of objects	O	

9. Security considerations

Control and management information is transferred through Ck reference point. Security requirements and measures to protect it are specified in [ITU-T X.1712].

Appendix I

Protocol implementation using TCP

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

This appendix describes a protocol implementation for messages and parameters using TCP which are described in clause 8.
NOTE 1 - Some of the parameters are mapped to a part of control information of the protocol instead of being mapped to a field in the data payload.

The KM can connect to the QKDN controller using TCP protocol [b-IETF RFC 9293]. The corresponding message format over TCP is as follows.

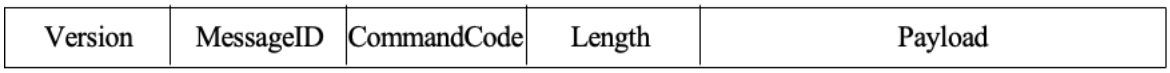

Figure I.1 – Message format over TCP

Version: the current version of the protocol format adopted, 2 bytes;

MessageID: the unique identifier of each message, 4 bytes;

CommandCode: a unique code that denotes different Command/Response messages transferred at the Ck interface, 2 bytes;

Length: the length of the message payload, 2 bytes;

Payload: the message parameters of a specific Command/Response message, JSON data format [b-IETF RFC 8259].

NOTE 2 – TLS protocol [b-IETF RFC 5246] can be implemented with TCP protocol for enhanced security.

At the connection establishment, mutual authentication between the KM and the QKDN controller shall be performed. After the mutual authentication, a Command/Response message can be transferred at the Ck interface for key relay request.

NOTE 3 – When applying TLS protocol, the KM can verify the validity of a certificate the QKDN controller possesses and confirm the ID of the QKDN controller it is connecting to, based on the certificate. Similarly, the QKDN controller can verify the validity of a certificate the KM possesses and confirm the ID of the connecting KM based on the certificate.

Table I.1 shows a list of CommandCode vs. Command/Response message name.

Table I.1 – CommandCode vs. Command/Response message name

CommandCode	Command/Response message name
0x1401	Key relay next hop request
0x4102	Response to key relay next hop request
0x1403	Key relay request notification
0x4104	Key relay request
0x1405	Response to key relay request

0x1406	Session creation request
0x4107	Response to session creation request
0x4108	Session creation notification
0x1409	Response to session creation notification
0x410A	Key reservation request
0x140B	Response to key reservation request
0x410C	Key allocation request
0x140D	Response to key allocation request

The first two digits “14” in a CommandCode indicates that the corresponding message is sent from the KM to the QKDN controller, and “41” indicates that the corresponding message is sent from the QKDN controller to the KM.

Appendix II

Protocol implementation using gRPC

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

This appendix describes a protocol implementation for messages and parameters using gRPC which are described in clause 8.

II.1 Mapping of signalling messages to gRPC messages

gRPC (Remote Procedure Call) is a cross-platform open source high performance remote RPC framework. It is currently being developed under the Cloud Native Computing Foundation (CNCF) under the Linux Foundation. It uses HTTP 2.0 and supports multiple programming languages [b-CNCF gRPC].

Table II.1-1 - Example mapping of signalling messages to gRPC messages

Signalling messages	gRPC message name
Key relay request notification	KeyRelayRequestNotification
Key relay request	KeyRelayRequest
Response to Key relay Request	KeyRelayResponse
Key reservation request	KeyReservationRequest
Response to key reservation request	KeyReservationResponse
Key allocation request	KeyAllocationRequest
Response to key allocation request	KeyAllocationResponse

II.2 Key relay request notification

Table Ⅱ.2-1 shows gRPC profiles for “Key relay request notification” message mapping example.

Table Ⅱ.2-1 – gRPC profiles for “Key relay request notification” message mapping example

Parameter	Mapped to	Data type
Source KMA ID	gRPC ‘start_km’	String
Destination KMA ID	gRPC ‘end_km’	String
Application destination ID	not mapped	
Extension	not mapped	

Ⅱ.3 Key relay request and response message

Table Ⅱ.3-1 shows gRPC profiles for “ key relay request” message mapping example. Table Ⅱ.3-2 shows an example of response message mapping.

Table Ⅱ.3-1 – gRPC profiles for “Key relay request” message mapping example

Parameter	Mapped to	Data type
Source KMA ID	gRPC ‘start_km’	String
Destination KMA ID	gRPC ‘end_km’	String
Transit KMA IDs	gRPC ‘kma_id’	String
Key relay request ID	gRPC ‘keyrelayrequest_id’	String
Extension	not mapped	

Table Ⅱ.3-2 – gRPC profiles for “Response to key relay request” message mapping example

Parameter	Mapped to	Data type
Response	gRPC ‘result_code’ ex) 0: OK, 1: NG	integer
Key relay request ID	gRPC ‘keyrelayrequest_id’	String
Extension	gRPC ‘error_message’	String

Ⅱ.4 Key reservation request and response message

Table Ⅱ.4-1 shows gRPC profiles for “Key reservation request” message mapping example. Table Ⅱ.4-2 shows an example of response message mapping.

Table Ⅱ.4-1 – gRPC profiles for “Key reservation request” message mapping example

Parameter	Mapped to	Data type
Source KMA ID	gRPC ‘start_km’	String
Destination KMA ID	gRPC ‘end_km’	String
Application destination ID	not mapped	
Extension	not mapped	

Table Ⅱ.4-2 – gRPC profiles for “Response to key reservation request” message mapping example

Parameter	Mapped to	Data type
Source KMA ID	gRPC ‘start_km’	String
Destination KMA ID	gRPC ‘end_km’	String
Application destination ID	not mapped	
Key IDs	gRPC ‘key_id’	String

Response	gRPC ‘result_code’ ex) 0: OK, 1: NG	integer
Extension	gRPC ‘error_message’	String

Ⅱ.5 Key allocation request and response message

Table Ⅱ.5-1 shows gRPC profiles for “Key allocation request” message mapping example. Table Ⅱ.5-2 shows an example of response message mapping.

Table Ⅱ.5-1 – gRPC profiles for “Key allocation request” message mapping example

Parameter	Mapped to	Data type
Source KMA ID	gRPC ‘start_km’	String
Destination KMA ID	gRPC ‘end_km’	String
Application destination ID	not mapped	
Key IDs	gRPC ‘key_id’	String
Extension	not mapped	

Table Ⅱ.5-2 – gRPC profiles for “Response to key allocation request” message mapping example

Parameter	Mapped to	Data type
Source KMA ID	gRPC ‘start_km’	String
Destination KMA ID	gRPC ‘end_km’	String
Application destination ID	not mapped	
Key IDs	gRPC ‘key_id’	String
Response	gRPC ‘result_code’ ex) 0: OK, 1: NG	Integer
Extension	gRPC ‘error_message’	String

Bibliography

[b-CNCF gRPC] https://grpc.io/docs/what-is-grpc/
https://www.cncf.io/projects/grpc/

[b-ETSI GR QKD 007] ETSI GR QKD 007 (2018), *Quantum Key Distribution (QKD); Vocabulary.*

[b-IETF RFC 9293] IETF RFC 9293, *Transmission Control Protocol (TCP).*

[b-IETF RFC 5246] IETF RFC 5246, *The Transport Layer Security (TLS) Protocol Version 1.2.*

[b-IETF RFC 8259] IETF RFC 8259, *The JavaScript Object Notation (JSON) Data Interchange Format.*

_____