INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2022-2024

SG11-TD454/GEN

STUDY GROUP 11

Original: English

Question(s): 2/11 Geneva, 10-19 May 2023

TD

Source: Editors

Title: Output – draft baseline text of draft Recommendation ITU-T Q.QKDN_Ak: Protocols for Ak interface for QKDN" (Geneva, 10-19 May 2023)

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: Kaoru KENYOSHI
NICT
Japan
Tel: +81 50 3566 5852
E-mail: kaoru.kenyoshi@nict.go.jp

Contact: Mariko Honda
NICT
Japan
E-mail: mariko.honda@ntt-at.co.jp

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 Q.QKDN_Ak: Protocols for Ak interface for QKDN (Geneva, 10-19 May 2023).

Summary

This is the output of draft Recommendation ITU-T Q.QKDN_Ak: Protocols for Ak interface for QKDN, based on the discussion results on C127R1 and C198 with modifications at the Q2/11 meeting (Geneva, 10-19 May 2023).

Draft Recommendation ITU-T Q.QKDN_Ak

Protocols for Ak interface for QKDN

Summary

Recommendation ITU-T Q.QKDN_Ak specifies protocols for Ak interface in quantum key distribution networks (QKDN).

Keywords

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

Table of Contents

Draft Recommendation ITU-T Q.QKDN_Ak

Protocols for Ak interface for QKDN

1. Scope

This Recommendation specifies protocols at Ak interface for QKDN especially the following areas.

- signalling procedures for Ak interface for QKDN;
- signalling messages and parameters for Ak 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 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*.

[ITU-T Y.3803] Recommendation ITU-T Y.3803 (2020), *Quantum key distribution networks - Key management.*

[ITU-T Q.QKDN_profr] draft Recommendation Q.QKDN_profr, *Quantum key distribution networks - Protocol framework.*

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 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.

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

3.1.4 **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.

3.1.5 **key supply agent-key (KSA-key)** [ITU-T Y.3803]: Key data stored and processed in a key supply agent (KSA), and securely shared between a KSA and a matching KSA.

3.1.6 **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.

3.1.7 **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.

3.1.8 **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).

3.1.9 **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.

3.1.10 **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).

~~This Recommendation uses the following terms defined elsewhere:~~

3.2. Terms defined in this Recommendation

None.

4. Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

HTTPS	HyperText Transfer Protocol Secure
ID	Identifier
KM	Key manager
KSA	Key Supply Agent
QKD	Quantum Key Distribution
QKDN	QKD Network
TCP	Transmission Control Protocol

5. Conventions

None.

6. Ak interface

Ak is a reference point connecting a cryptographic application and a key supply function in a KSA. It is responsible for sending key requests from the cryptographic application to the KSA, performing authentication between the cryptographic application and the KSA, and supplying keys from the KSA to the cryptographic application.

7. Signalling procedure

The following two modes are specified for the key request and key supply at the Ak interface.

1) Key supply upon request mode: Both KMs at the source and destination sides initiate key supplies after receiving key requests from the corresponding cryptographic applications.
2) Proactive key supply mode: The KM at the source side initiates a key supply upon request, and then instructs the KM at the destination side to make a proactive key supply.

NOTE - The proactive key supply mode can be adopted in scenarios where the cryptographic applications at both sides are restricted to have no direct communication before they have KSA-keys.

Examples of signalling procedure of key request, key relay, and key supply in QKDN are described in the Appendix I of [ITU-T Q.QKDN_profr]. The protocol suites applied for the signalling are specified in clause 8 of [ITU-T Q.QKDN_profr].

7.1. Signalling procedure for key supply upon request mode

7.1.1. Key request at the source side

When cryptographic application needs keys for encryption, the cryptographic application sends a key request to the KM, and the KM supplies keys to the cryptographic application. If the KM doesn't have enough number of keys in key storage, the KM initiates key generation or key relay to share the necessary number of keys, and supply them to the cryptographic applications when the key generation or key relay is completed.

Figure 1 shows signalling procedures for key request at the Ak interface at the source side.

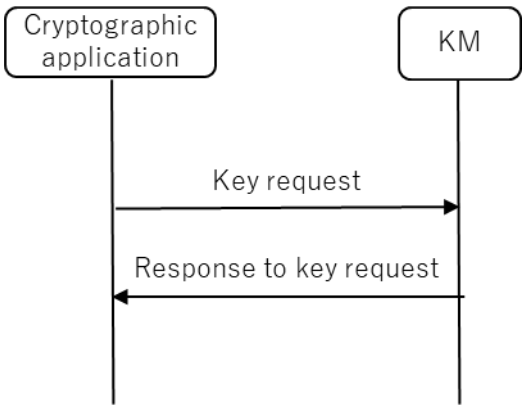

Figure 1 – Signalling procedures for key request at the Ak interface at the source side

7.1.2. Key request with ID at the destination side

The destination cryptographic application requests the key to the KM which has the connection with itself. The destination cryptographic application sends a request with the key identifier (ID) which is received from the source cryptographic application in order to specify the key.

Figure 2 shows signalling procedures for key request with ID at the Ak interface at the destination side.

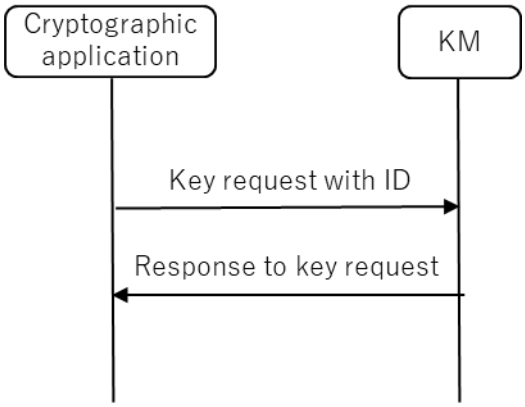

Figure 2 – Signalling procedures for key request with ID at the Ak interface at the destination side

7.2. Signalling procedure for proactive key supply mode

7.2.1. Session creation at the source side

When cryptographic application needs keys for encryption, the cryptographic application first sends a session creation request to the KM at the source side. Then the source KM notifies the KM at the destination side to create a session and responses with a session ID to the source cryptographic application when the session is successfully created. Based on the created session, the source cryptographic application can request keys to the source KM.

Figure 3 shows signalling procedures for session creation at the Ak interface at the source side.

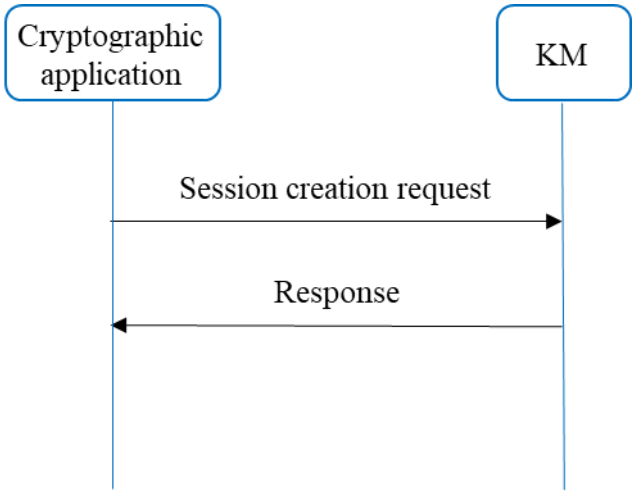

Figure 3 – Signalling procedures for session creation at the Ak interface at the source side

7.2.2. Session creation at the destination side

The destination cryptographic application receives a session creation notification from the KM which has the connection with itself. The destination KM sends the notification with the session ID which is received from the source KM in order to specify the session.

Figure 4 shows signalling procedures for session creation at the Ak interface at the destination side.

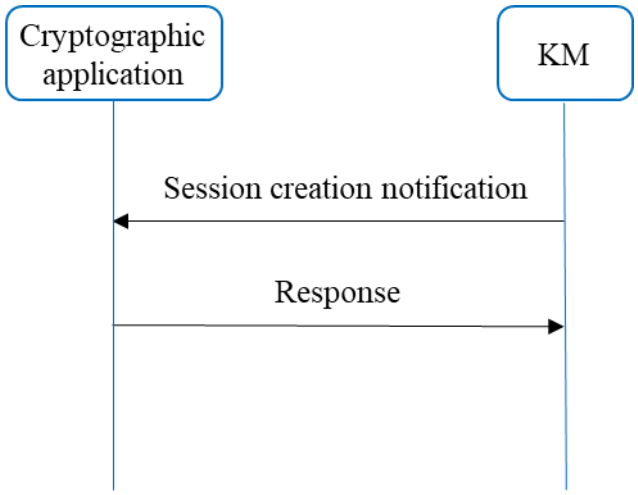

Figure 4 – Signalling procedures for session creation at the Ak interface at the destination side

7.2.3. Key request with session ID at the source side

With a created session, the KM at the source side supplies KSA-keys upon key request from the source cryptographic application.

Figure 5 shows signalling procedure for key request with session ID at the Ak interface at the source side.

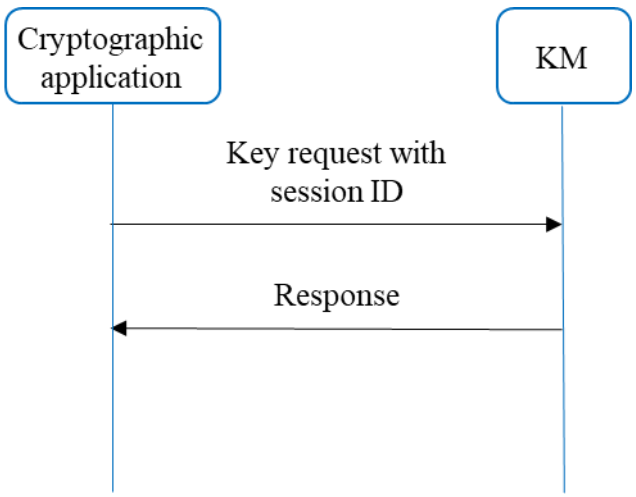

Figure 5 – Signalling procedures for key request with session ID at the Ak interface at the source side

7.2.4. Proactive key supply at the destination side

The KM at the destination side proactively supplies KSA-keys to the destination cryptographic application which has the connection with it. This scheme is applicable when the key request from

the source cryptographic application is received by the source KM, which then instructs the destination KM to make a proactive key supply.

Figure 6 shows signalling procedures for proactive key supply at the Ak interface at the destination side.

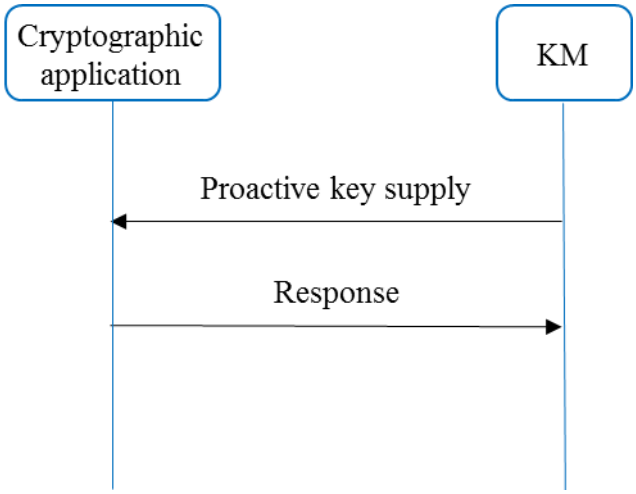

Figure 6 – Signalling procedures for proactive key supply at the Ak interface at the destination side

8. Signalling messages and parameters

This clause specifies messages and their parameters for the Ak 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.

NOTE – The messages and parameters defined in this clause are independent of a specific protocol. Different protocols can have different implementations of these messages and parameters. The recommended protocol implementations are described in Annex A and B. 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. Messages and parameters for key supply upon request mode

8.1.1. Key request message

Key request message is sent from the cryptographic application to the KM at the source side for requesting of keys.

Table 1 shows parameters of Key request message.

Table 1 – Parameters of Key request message

Parameter	Description	Data type	M/O	Remarks
Application source ID	ID of the source cryptographic application (i.e., the application which sends this message)	string	O	
Application destination ID	ID of the destination cryptographic application (i.e., the application with which the source cryptographic	string	M	

	application requests to communicate)			
Application name	Name of the cryptographic application	string	O	
Number of KSA-keys	Number of KSA-keys requested	integer	O	A default value is applied if omitted.
Size of KSA-key	Length of each KSA-key requested	integer	O	A default value is applied if omitted.
Extension	Array of extension parameters	Array of objects	O	

8.1.2. Key request with ID message

When receiving the KSA-key, the source cryptographic application sends the corresponding key ID to the destination cryptographic application. The destination cryptographic application sends a request to the destination KM with the key ID. Then the destination cryptographic application receives the key which has been shared between the source and destination KMs.

Table 2 shows parameters of Key request with ID message.

Table 2 – Parameters of Key request with ID message

Parameter	Description	Data type	M/O	Remarks
Application source ID	ID of the source cryptographic application	string	M	
Application destination ID	ID of the destination cryptographic application (i.e., the application which sends this message)	string	O	
Application name	Name of the cryptographic application	string	O	
~~KSA key ID~~	~~ID of the KSA key requested~~	~~string~~	~~M~~	~~This ID is notified from the source cryptographic application.~~
~~KSA k~~Key IDs	IDs of the KSA-keys requested	array of objects	M	These IDs are notified from the source cryptographic application
~~KSA k~~Key ID	ID of the KSA-key requested	string	M	
Key extension	Extensions to key file	object	O	
Extension	Array of extension parameters	Array of objects	O	

8.1.3. Response to key request message

Response to key request message is sent from the KM to the cryptographic application in response to the key request or the key request with ID from the cryptographic application. The KM supplies the requested KSA-keys to the cryptographic application. There is no difference between the source side and the destination side for the response to key request.

Table 3 shows parameters of Response to key request message.

Table 3 – Parameters of Response to key request message

Parameter		Description	Data type	M/O	Remarks
Keys		Key file consists of key data and metadata.	Array of objects	M	
	~~KSA k~~Key	KSA-key data provided for the request	string	M	
	~~KSA k~~Key ID	ID of the KSA-key provided	string	M	
	Key extension	Extensions to key file	object	O	Hash value etc.
Response		Result of key supply	string	M	Success or failure reason
Extension		Array of extension parameters	Array of objects	O	

8.2. Messages and parameters for proactive key supply mode

8.2.1. Session creation request message

Session creation request message is sent from the cryptographic application to the KM at the source side. A session will be created to facilitate the key supply between the cryptographic applications and the KMs at both sides.

Table 4 shows parameters of Session creation request message.

Table 4 – 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 sends this message)	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	
Application name	Name of the source cryptographic application	string	O	
Maximum number	Maximum number of KSA-keys requested during one session	integer	O	
Extension	Array of extension parameters	Array of objects	O	

8.2.2. Response to session creation request message

Response to session creation request message is sent from the KM to the cryptographic application at the source side. After receiving a session creation request, the source KM notifies the KM at the destination side to create a session and responses with a session ID to the source cryptographic application when the session is successfully created.

Table 5 shows parameters of Response to session creation request message.

Table 5 – Parameters of Response to session creation request message

Parameter	Description	Data type	M/O	Remarks
Session ID	ID of the session created	string	M	
Response	Result of the creation of the session	string	M	Success, failure reason, or status table of key supply.
Source KM ID	ID of the source KM	string	O	
Extension	Array of extension parameters	Array of objects	O	

8.2.3. Session creation notification message

Session creation notification message is sent from the KM to the cryptographic application at the destination side. The destination KM proactively sends the session ID to the destination cryptographic application, and notifies it with the ID of the source cryptographic application that requests to communicate with it.

Table 6 shows parameters of Session creation notification message.

Table 6 – 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 receives this message)	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	
Application name	Name of the source cryptographic application	string	O	
Session ID	ID of the session created	string	M	
Maximum number	Maximum number of KSA-keys requested during one session	integer	O	
Extension	Array of extension parameters	Array of objects	O	

8.2.4. Response to session creation notification message

Response to session creation notification message is sent from the cryptographic application to the KM in response to the session creation notification at the destination side. The destination cryptographic application notifies the result of the creation of the session to the destination KM.

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

Table 7 – Parameters of Response to session creation notification message

Parameter	Description	Data type	M/O	Remarks
Session ID	ID of the session created	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	

8.2.5. Key request with session ID message

With a created session, the cryptographic application sends a key request with session ID message to the KM at the source side. Then the source KM supplies the requested KSA-keys to the source cryptographic application during the session.

Table 8 shows parameters of Key request with session ID message.

Table 8 – Parameters of Key request with session ID message

Parameter	Description	Data type	M/O	Remarks
Session ID	ID of the session created	string	M	
Number of KSA-keys	Number of KSA-keys requested	integer	O	A default value is applied if omitted.
Size of KSA-key	Length of each KSA-key requested	integer	O	A default value is applied if omitted.
Extension	Array of extension parameters	Array of objects	O	

8.2.6. Response to key request with session ID message

Response to key request with session ID message is sent from the KM to the cryptographic application at the source side. The source KM supplies the requested KSA-keys to the source cryptographic application during the created session.

Table 9 shows parameters of Response to key request with session ID message.

Table 9 – Parameters of Response to key request with session ID message

Parameter		Description	Data type	M/O	Remarks
Session ID		ID of the session created	string	M	
Keys		Key file consists of key data and metadata.	Array of objects	M	
	~~KSA k~~Key	KSA-key data provided for the request	string	M	
	~~KSA k~~Key ID	ID of the KSA-key provided	string	M	
	Key extension	Extensions to key file	object	O	Hash value etc.
Response		Result of key supply	string	M	Success or failure

				reason
Extension	Array of extension parameters	Array of objects	O	

8.2.7. Proactive key supply message

Proactive key supply message is sent from the KM to the cryptographic application at the destination side. The destination KM proactively supplies the KSA-key to the destination cryptographic application during the created session.

Table 10 shows parameters of Proactive key supply message.

Table 10 – Parameters of Proactive key supply message

Parameter		Description	Data type	M/O	Remarks
Session ID		ID of the session created	string	M	
Keys		Key file consists of key data and metadata.	Array of objects	M	
	~~KSA k~~Key	KSA-key data supplied	string	M	
	~~KSA k~~Key ID	ID of the KSA-key supplied	string	M	
	Key extension	Extensions to key file	object	O	Hash value etc.
Extension		Array of extension parameters	Array of objects	O	

Editor's note: The name of parameters of Keys and Key should be considered in future meeting.

8.2.8. Response to proactive key supply message

Response to proactive key supply message is sent from the cryptographic application to the KM in response to the proactive key supply at the destination side. The destination cryptographic application notifies the result of the receipt of the KSA-key to the destination KM.

Table 11 shows parameters of Response to proactive key supply message.

Table 11 – Parameters of Response to proactive key supply message

Parameter	Description	Data type	M/O	Remarks
Session ID	ID of the session created	string	M	
KSA-key ID	ID of the KSA-key received	string	M	
Response	Result of the receipt of the KSA-key	string	M	Success or failure reason
Extension	Array of extension parameters	Array of objects	O	

9. Security considerations

Key data and associated metadata are transferred through Ak reference point. Security requirements and measures to protect them are specified in [ITU-T X.1712].

Annex A

Protocol implementation ~~for~~using TCP

(This annex forms an integral part of this Recommendation.)

This annex describes a protocol implementation for messages and parameters ~~for~~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 cryptographic application can connect to the KM using TCP protocol [b-IETF RFC 9293~~793~~]. The corresponding message format over TCP is as follows.

Version	MessageID	CommandCode	Length	Payload

Figure A.1 – Message format over TCP

Version: the current version of the message 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 Ak 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.

~~*Editor's note - Detail implementation of TLS protocol needs to be addressed.*~~

At the connection establishment, mutual authentication between the cryptographic application and the KM is performed. After the mutual authentication, a Command/Response message can be transferred at the Ak interface for key request and key supply.

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

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

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

CommandCode	Command/Response message name
0x01	Key request
0x02	Key request with ID
0x03	Response to key request
0x04	Session creation request
0x05	Response to session creation request

0x06	Session creation notification
0x07	Response to session creation notification
0x08	Key request with session ID
0x09	Response to key request with session ID
0x0A	Proactive key supply
0x0B	Response to proactive key supply

Annex B

Protocol implementation for key supply upon request mode using HTTPS

(This annex forms an integral part of this Recommendation.)

The signalling ~~procedure~~messages and parameters for key supply upon request mode specified in clause ~~7.1~~8.1 can be implemented ~~as the REST-based key delivery procedure~~ using HTTPS according to the protocol and data format of REST-based key delivery API specified in [b-ETSI GS QKD 014]. ~~To illustrate how the signalling procedure is implemented as the REST-based key delivery procedure, t~~This annex describes the mapping of the messages and parameters ~~for that mode~~ specified in clause 8.1 to the corresponding data format specified in [b-ETSI GS QKD 014].

NOTE – In this implementation, the cryptographic application and the KM correspond to the SAE (Secure Application Entity) and the KME (Key Management Entity) defined in [b-ETSI GS QKD 014] respectively.

B.1 Key request message

In this implementation ~~T~~the Key request message specified in clause 8.1.1 ~~is implemented as~~ corresponds to the HTTPS request ~~phase~~ of the HTTPS transaction performed as the 'Get Key' method specified in [b-ETSI GS QKD 014]. Table B.1 shows the mapping of the Key request message to the 'Get Key' method.

Table B.1 – Mapping of Key request message to Get Key method

Parameter	M/O	Data type	Implementation in 'Get Key' method
Application source ID	O	string	None
Application destination ID	M	string	'{target_SAE_ID}' part of the Access URL
Application name	O	string	None
Number of KSA-keys	O	integer	The 'number' item in the Key request data format
Size of KSA-key	O	integer	The 'size' item in the Key request data format
Extension	O	array of objects	The 'extension_mandatory' or 'extension_optional' item in the Key request data format

B.2 Key request with ID message

In this implementation T~~t~~he Key request with ID message specified in clause 8.1.2 ~~is implemented as~~corresponds to the HTTPS request ~~phase~~ of the HTTPS transaction performed as the 'Get Key with ID' method specified in [b-ETSI GS QKD 014]. Table B.2 shows the mapping of the Key request with ID message to the 'Get Key with ID' method.

Table B.2 – Mapping of Key request with ID message to Get Key with ID method

Parameter	M/O	Data type	Implementation in 'Get Key with ID' method
Application source ID	M	string	'{initiator_SAE_ID}' part of the Access URL
Application destination ID	O	string	None
Application name	O	string	None

~~KSA k~~Key IDs	M	array of objects	The 'key_IDs' item in the Key IDs data format
~~KSA k~~Key ID	M	string	The 'key_ID' item in the Key IDs data format
Key extension	O	object	The 'key_ID_extension' item in the Key IDs data format
Extension	O	array of objects	The 'key_IDs_extension' item in the Key IDs data format

B.3 Response to key request message

In this implementation ~~T~~the Response to key request message specified in clause 8.1.3 ~~is implemented as~~corresponds to the HTTPS response ~~phase~~ of the HTTPS transaction performed as the 'Get Key' method or the 'Get Key with ID' method. Table B.3 shows the mapping of the Response to key request message to the 'Get Key' method or the 'Get Key with ID' method.

Table B.3 – Mapping of Response to key request message to Get Key/Get Key with ID method

Parameter	M/O	Data type	Implementation in 'Get Key' or 'Get Key with ID' method
~~Key~~Keys	M	array of objects	The 'Keys' item in the Key container data format
~~KSA k~~Key	M	string	The 'key' item in the Key container data format
~~KSA K~~key ID	M	string	The 'key_ID' item in the Key container data format
Key extension	O	object	The 'key_ID_extension' item in the Key container data format
Response	M	string	The status code of HTTPS transaction performed as 'Get Key' method or 'Get Key with ID' method
Extension	O	array of objects	The 'key_container_extension' item in the Key container data format

~~This annex describes a protocol implementation for messages and parameters for HTTPS which are described in clause 8.~~

~~NOTE – This protocol implementation is aligned with the protocol specified in [b-ETSI GS QKD 014].~~

~~B.1 Key request message~~

~~Table B.1 shows HTTPS profiles for Key request message.~~

~~Table B.1 – HTTPS profiles for Key request message~~

~~Parameter~~	~~Mapped to~~	~~Data type~~
~~Application source ID~~	~~Implicit (Source of HTTPS connection)~~	
~~Application destination ID~~	~~Part of the URL to which the source~~	~~string~~

	cryptographic application connects ('target_SAE_ID' part)	
Number of KSA keys	HTTPS request body ('number' in key request container)	JSON ('number': value)
Size of KSA key	HTTPS request body ('size' in key request container)	JSON ('size' : size)
Extension	HTTPS request body ('extension_mandatory' or 'extension_optional' in key request container)	JSON array

B.2 Key request with ID message

Table B.2 shows HTTPS profiles for Key request with ID message

Table B.2 – HTTPS profiles for Key request with ID message

Parameter	Mapped to	Data type
Application source ID	Part of the URL to which the source cryptographic application connects ('initiator SAE_ID' part)	string
Application destination ID	Implicit (Source of HTTPS connection)	
KSA key ID	HTTPS request body	JSON array (of 'key_ID':key_ID)
Extension	HTTPS request body	JSON array

B.3 Response to key request message

Table B.3 shows HTTPS profiles for Response to key request message

Table B.3 – HTTPS profiles for Response to key request message

Parameter		Mapped to	Data type
Key		HTTPS response body	Array of objects
	KSA key	HTTPS response body	JSON ('key': base64string)
	KSA key ID	HTTPS response body	JSON ('key_ID': uuid)
	Key extension	HTTPS response body	object
Response		HTTPS response header	String(integer)
Extension		HTTPS response body	object

Bibliography

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

[b-ETSI GS QKD 014] ~~Group Specification~~ ETSI GS QKD 014 (2019), *Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API.*

[b-IETF RFC 9293~~793~~] IETF RFC 9293~~793~~, ~~*TRANSMISSION CONTROL PROTOCOL*~~ *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.*

_____