ETSI TS 103 744 V1.1.1 (2020-12)

CYBER;
Quantum-safe Hybrid Key Exchanges

Reference

DTS/CYBER-QSC-0015

Keywords

key exchange, quantum safe cryptography

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 Technical Specification (TS) has been produced by ETSI Technical Committee Cyber Security (CYBER).

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.

Introduction

Hybrid Key Exchanges are constructions that combine a traditional key exchange, such as elliptic curve Diffie Hellman [1], with a quantum-safe key exchange such as Supersingular Isogeny Key Establishment (SIKE) [i.17], into a single key exchange. Hybrid key exchanges are a migration technique to move to quantum-safe technology in advance of establishing full security assurance in the underlying post-quantum cryptographic scheme.

1 Scope

The present document specifies several methods for deriving cryptographic keys from multiple shared secrets. The shared secrets are established using existing classical key agreement schemes, like elliptic curve Diffie-Hellman (ECDH) in NIST SP800-56Ar3 [1], and new quantum-safe key encapsulation mechanisms (KEMs).

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] NIST SP800-56Ar3: "Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography".

NOTE: Available at https://nvlpubs.nist.gov/nistpubs/SpecialPublications/nist.sp.800-56Ar3.pdf.

[2] IETF RFC 2104: "HMAC: Keyed-Hashing for Message Authentication".

NOTE: Available at https://tools.ietf.org/html/rfc2104.

[3] IETF RFC 5869: "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)".

NOTE: Available at https://tools.ietf.org/html/rfc5869.

[4] FIPS PUB 180-4: "Secure Hash Standard (SHS)".

NOTE: Available at https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.180-4.pdf.

[5] FIPS PUB 202: "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions".

NOTE: Available at https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.202.pdf.

[6] FIPS PUB 186-4: "Digital Signature Standard (DSS)".

NOTE: Available at https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.186-4.pdf.

[7] IETF RFC 5639: "Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation".

NOTE: Available at https://tools.ietf.org/html/rfc5639.

[8] IETF RFC 7748: "Elliptic Curves for Security".

NOTE: Available at https://tools.ietf.org/html/rfc7748.

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] NIST Post Quantum Round 2 Submission: "BIKE: Bit Flipping Key Encapsulation", Round 3 Submission, 22 October 2020.

NOTE: Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[i.2] NIST Post Quantum Round 3 Submission: "Classic McEliece: conservative code-based cryptography", 10 October 2020.

NOTE: Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[i.3] NIST Post Quantum Round 3 Submission: "CRYSTALS-Kyber", Version 3.0, 1 October 2020.

NOTE: Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[i.4] NIST Post Quantum Round 3 Submission: "FrodoKEM Learning With Errors Key Encapsulation", 30 September 2020.

NOTE: Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[i.5] NIST Post Quantum Round 3 Submission: "Hamming Quasi-Cyclic (HQC)", 1 October 2020.

NOTE: Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[i.6] NIST Post Quantum Round 3 Submission: "NTRU", 30 September 2020.

NOTE: Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[i.7] NIST Post Quantum Round 3 Submission: "NTRU Prime: round 3", 7 October 2020.

NOTE: Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[i.8] NIST Post Quantum Round 3 Submission: "SABER: Mod-LWR based KEM (Round 3 Submission)", Accessed 26 October 2020.

NOTE: Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[i.9] NIST Post Quantum Round 3 Submission: "Supersingular Isogeny Key Encapsulation", 1 October 2020.

NOTE: Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[i.10] Y. Dodis, R. Gennaro, J. Håstad, H. Krawczyk, and T. Rabin: "Randomness Extraction and Key derivation Using the CBC, Cascade, and HMAC Modes", Crypto 04, LNCS 3152, pp. 494-510. Springer Verlag, 2004.

[i.11] F. Giacon, F. Heuer, B. Poettering: "KEM Combiners", Public-Key Cryptography - PKC 2018, LNCS 10769.

NOTE: Available at https://eprint.iacr.org/2018/024.pdf.

[i.12] N. Bindel, J. Brendel, M. Fischlin, B. Goncalves, D. Stebila: "Hybrid Key Encapsulation Mechanisms and Authenticated Key Exchange", IACR eprint 2018-903.

NOTE: Available at https://eprint.iacr.org/2018/903.pdf.

[i.13] N. Bindel, U. Herath, M. McKague, D. Stebila: "Transitioning to a Quantum-Resistant Public Key Infrastructure", Post-Quantum Cryptography, 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands Proceedings. pp. 384-405. Springer International Publishing, Cham (2017).

[i.14] Simon, D. R.: "On the power of quantum computation", SFCS 94 Proceedings of the 35th Annual Symposium on Foundations of Computer Science, November 1994, Pages 116-123.

NOTE: Available at https://doi.org/10.1109/SFCS.1994.365701.

[i.15] Shor, P.W.: "Algorithms for quantum computation: discrete logarithms and factoring", SFCS 94: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, November 1994, Pages 124-134.

NOTE: Available at https://dl.acm.org/doi/abs/10.1109/SFCS.1994.365700.

[i.16] NIST CAVP SP 800-56A ECC CDH Primitive Test Vectors.

NOTE: Available at https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Algorithm-Validation-Program/documents/components/ecccdhtestvectors.zip.

[i.17] SIKE Round 2 Known Answers Tests (KATs).

NOTE: Available at https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/submissions/SIKE-Round2.zip.

[i.18] Campagna, M., Petcher, A.: "Security of Hybrid Key Encapsulation", IACR eprint 2020-1364.

NOTE: Available at https://eprint.iacr.org/2020/1364.pdf.

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

asymmetric cryptography: cryptographic system that utilizes a pair of keys, a private key known only to one entity, and a public key which can be openly distributed without loss of security

big-endian: octet ordering that signifies "big-end", or most significant octet value is stored to the left, or at the lowest storage location

EXAMPLE: The decimal value 108591, which is 0x0001A82F as a hex encoded 32-bit integer, is encoded as a length 4 octet string as 0001A82F.

cryptographic hash function: function that maps a bit string of arbitrary length to a fixed length bit string (*message digest* or *digest* for short)

NOTE: Hash functions are designed to satisfy the following properties:

1) (One-way) It is computationally infeasible to find any input that maps to any pre-specified output.

2) (Collision resistant) It is computationally infeasible to find any two distinct inputs that map to the same output.

cryptographic key: binary string used as a secret by a cryptographic algorithm

EXAMPLE: AES-256 requires a random 256-bit string as a secret key.

entity: person, device or system that is executing the steps of one of the processes defined or referenced in the present document

key agreement scheme: key-establishment procedure in which the resultant secret keying material is a function of contributions of the entities participating, such that no entity can predetermine that value of the secret keying material independently of the other entities' contributions

key derivation: process to derive key material from one or more shared secrets

key encapsulation mechanism: method to secure the establishment of a cryptographic key for transmission using public key cryptography

key establishment/exchange method: cryptographic procedure by which cryptographic keys are established between two parties

label: octet string that specifies a separation of use for the application or instance of the key derivation or exchange

message digest/digest: fixed-length output of a cryptographic hash function over a variable length input

octet string: ordered sequence of octets/bytes consisting of 8-bits each

private key: key in an asymmetric cryptographic scheme that is kept secret

public key: key in an asymmetric cryptographic scheme that can be made public without loss of security

public key cryptography: See asymmetric cryptography.

random oracle: theoretical black box that responds to every unique query with a uniformly random selection from the set of possible responses, with repeated queries receiving the same response

security level: measure of the strength of a cryptographic algorithm. If 2^n operations are required to break the cryptographic algorithm/scheme/method, then the security level is *n*. Sometimes also referred to as *bit-strength*

shared secret: secret value that has been computed using a key-establishment scheme

3.2 Symbols

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

A || B	The concatenation of binary strings A followed by B
∅	A zero-length octet string
$[x]_n$	An integer value *x* expressed as an *n*-bit integer
$\lceil q \rceil$	The least integer value *x* greater than or equal to *q*
len(A)	The number of octets in an octet string *A*
hash( )	A cryptographic hash function
digest_len	The length in octets of a hash function's digest
C	A ciphertext value created by a KEM
d	A private key for elliptic curve cryptography
k	A cryptographic secret or key
P	A public key for an asymmetric cryptographic scheme
psk	A pre-shared key
Q	A public key for elliptic curve cryptography
sk	A private key for an asymmetric cryptographic scheme

3.3 Abbreviations

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

AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CDH	Cofactor Diffie-Hellman
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
HKDF	HMAC-based Key Derivation Function
HMAC	Hash-based Message Authentication Code

IND-CCA	INDistinguishability under Chosen-Ciphertext Attacks
IND-CPA	INDistinguishability under Chosen-Plaintext Attacks
KDF	Key Derivation Function
KEM	Key Encapsulation Mechanism
LNCS	Lecture Notes in Computer Science
MA	Message from entity A
MB	Message from entity B
NIST	National Institute of Standards and Technology
OW-CPA	One-Way Chosen-Plaintext Attack
PRF	PseudoRandom Function
QA	A public-key from entity A
QB	A public-key from entity B
QKD	Quantum Key Distribution
RSA	Rivest, Shamir and Adelman
SIKE	Supersingular Isogeny Key Encapsulation
SP	Special Publication
SSH	Secure Shell
TLS	Transport Layer Security

4 Purpose of quantum-safe hybrid key exchanges

4.1 Status of quantum-safe key exchange protocols

NIST has initiated a process of analysing and standardizing one or more new quantum-safe key encapsulation mechanisms suitable to replace classical key exchanges. At the time of the present document, there are 9 round 3 post-quantum KEMs still under consideration [i.1] to [i.9].

The present document addresses the following cases:

1) One or more key exchange method establishes a shared secret from which randomness extraction is necessary.

2) One or more key exchange method incorporates a hash-based key derivation function prior to use within the hybrid method defined in the present document.

The quantum-safe hybrid key exchanges specified in the present document ensure that the derived key is at least as secure as the maximum security of the key exchange method. The resulting hybrid scheme will remain secure if one of the key exchange methods remains secure.

Quantum Key Distribution (QKD) provides an alternative method of establishing a shared secret between two entities using quantum mechanics. The scope of the present document is limited to elliptic curve Diffie-Hellman and quantum-safe key encapsulation mechanisms.

5 Architecture for quantum-safe hybrid key exchange

5.1 Functional entities

There are two entities defined for quantum-safe hybrid key exchange, an Initiator *A* that initiates a key exchange mechanism, and a Responder *B* who responds to the request. The entities communicate over a network medium.

EXAMPLE: Examples of such mediums are: ethernet, wireless and cellular networks.

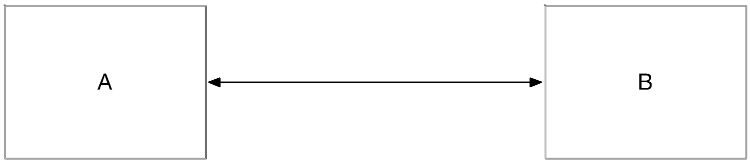

Figure 1: Communicating entities *A* and *B*

5.2 Information relationships (reference points)

The network media over which the Initiator and Responder communicate will have a packet formatting scheme that allows the encoding and transmission of octet (byte) strings. The Initiator and Responder will exchange messages, where each message is an octet string that can span multiple packets. *MA* denotes a message from *A* to *B*, and *MB* denotes a message sent from *B* to *A*.

A initiates a hybrid key exchange by the transmission of a message to *B*. *B* responds to this message. The exchange between the entities can consist of a single message in each direction or multiple rounds of messages.

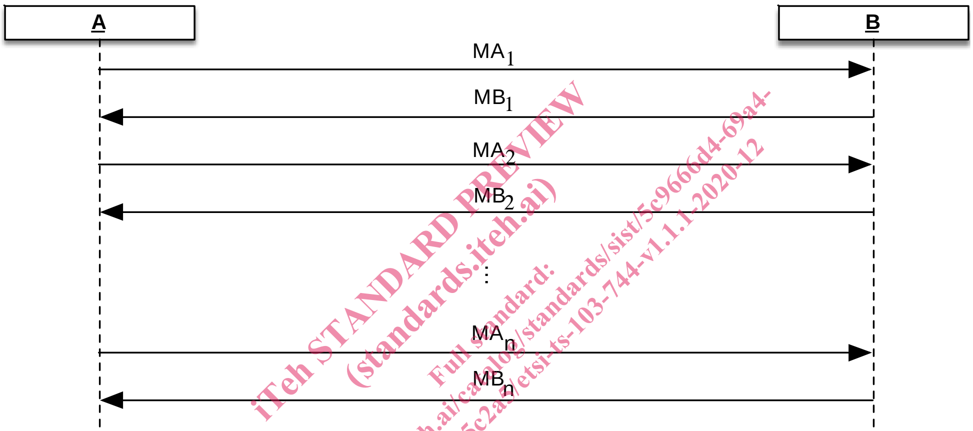

Figure 2: Messages exchanged between entities *A* and *B*

The transcript of the key exchange is the list of all messages exchanged between *A* and *B*, in the sequence order they were sent:

$$transcript = (MA_1, MB_1, MA_2, MB_2, \ldots MA_n, MB_n)$$

6 Introductory information

6.1 Introduction

Quantum-safe hybrid key exchange mechanisms combine a classic key exchange method like ECDH and a quantum-safe key-encapsulation mechanism (KEM). The hybrid exchange mechanisms specified in the present document use two or more shared secrets to derive cryptographic key material using a key derivation function. The key derivation functions for the hybrid key exchanges specified in the present document provide both the key expansion property and random extraction as per Crypto 04, LNCS 3152 [i.10].

6.2 Notation

6.2.1 Radix

The prefix "0x" indicates hexadecimal numbers.

6.2.2 Conventions

The assignment operator "=", as used in several programming languages:

<variable> = <expression>

means that <*variable*> assumes the value that <*expression*> had before the assignment took place. For instance:

$$x = x + y + 3$$

means:

(new value of x) becomes (old value of x) + (old value of y) + 3.

6.2.3 Bit/Byte ordering

All data variables are represented with the most significant bit (or byte) on the left-hand side and the least significant bit (or byte) on the right-hand side. Where a variable is broken down into a number of sub-strings, the left most (most significant) sub-string is numbered 0, the next most significant is numbered 1 and so on through to the least significant.

EXAMPLE: An n-bit MESSAGE is subdivided into 64-bit substrings M_0, M_1, …, M_i so if the message is:

0x0123456789ABCDEFFEDCBA98765432 1086545381AB594FC28786404C50A37…

then:

M_0 = 0x0123456789ABCDEF
M_1 = 0xFEDCBA9876543210
M_2 = 0x86545381AB594FC2
M_3 = 0x8786404C50A37…

6.2.4 Integer encoding

Integers are represented in the bit/byte ordering defined in clause 6.2.3. The most significant bit (or byte) on the left-hand side and the least significant bit (or byte) on the right-hand side.

EXAMPLE: a 32-bit integer of the value I = 37 is encoded as:

I = 0x00000025

NOTE: This is big-endian or network byte ordering.

7 Cryptographic primitives

7.1 Hash functions (hash)

A hash function maps an arbitrary length bit string *(input)* to a fixed length (*digest_len*) octet string output *(digest)*:

digest = hash(input)

Approved hash functions for the purpose of the present document shall be limited to those in the following list:

- SHA-256, SHA-384, SHA-512, SHA-512/256 as defined in FIPS PUB 180-4 [4].
- SHA3-256, SHA3-384, SHA3-512 as defined in FIPS PUB 202 [5].