ETSI TR 103 823 V1.1.2 (2021-10)

CYBER;
Quantum-Safe Public-Key Encryption and Key Encapsulation

Reference

DTR/CYBER-QSC-0017rev

Keywords

algorithm, cybersecurity

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

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

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Cyber Security (CYBER).

Modal verbs terminology

In the present document "**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.

1 Scope

The present document provides technical descriptions of the Public-Key Encryption (PKE) and Key Encapsulation Mechanisms (KEMs) submitted to the National Institute for Standards and Technology (NIST) for the third round of their Post-Quantum Cryptography (PQC) standardization process.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

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 FIPS 197: "Advanced Encryption Standard (AES)".

[i.2] NIST FIPS 180-4: "Secure Hash Standard".

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

[i.4] NIST IR 8105: "Report on Post-Quantum Cryptography".

[i.5] NIST FIPS 186-4: "Digital Signature Standard (DSS)".

[i.6] NIST SP-56A: "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography".

[i.7] NIST SP-56B: "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography".

[i.8] NIST: "Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process", December 2016.

NOTE: Available at https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[i.9] NIST Post-Quantum Cryptography Standardization: "Round 1 Submissions".

NOTE: Available at https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[i.10] NIST IR 8240: "Status Report on the First Round of the NIST Post-Quantum Standardization Process".

[i.11] NIST Post-Quantum Cryptography Standardization: "Round 2 Submissions".

NOTE: Available at https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions.

[i.12] NIST IR 8309: "Status Report on the Second Round of the NIST Post-Quantum Standardization Process".

[i.13] NIST Post-Quantum Cryptography Standardization: "Round 3 Submissions".

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

[i.14] ETSI TR 103 616: "CYBER; Quantum-Safe Signatures".

[i.15] E. Fujisaki and T. Okamoto: "Secure integration of asymmetric and symmetric encryption schemes", CRYPTO, 1999.

[i.16] D. Hofheinz, K. Hövelmanns and E. Kiltz: "A modular analysis of the Fujisaki-Okamoto transformation", TCC, 2017.

[i.17] N. Drucker, Shay Gueron and D. Kostic: "QC-MDPC decoders with several shades of gray", PQCrypto, 2020.

[i.18] R. Canto Torres and N. Sendrier: "Analysis of information set decoding for a sub-linear error weight", PQCrypto, 2016.

[i.19] N. Bindel, M. Hamburg, K. Hövelmanns, A. Hülsing, and E. Persichetti: "Tighter proofs of CCA security in the quantum random oracle model", TCC, 2019.

[i.20] M.R. Albrecht, V. Gheorghiu, E.W. Postlethwaite and J.M. Schanck: "Estimating quantum speedups for lattice sieves", Cryptology ePrint Archive, Report 2019/1161, 2019.

[i.21] E. Prange: "The use of information sets in decoding cyclic codes", IRE Transactions on Information Theory 8.5 (1962): 5-9.

[i.22] P.J. Lee and E.F. Brickell: "An observation on the security of McEliece's public-key cryptosystem", EUROCRYPT, 1988.

[i.23] J. Stern: "A method for finding codewords of small weight", International Colloquium on Coding Theory and Applications. Springer, Berlin, Heidelberg, 1988.

[i.24] A. May, A. Meurer and E. Thomae: "Decoding random linear codes in O˜(2^0.054n)", ASIACRYPT, 2011.

[i.25] A. Becker, A. Joux, A. May and A. Meurer: "Decoding random binary linear codes in 2^(n/20): How 1 + 1 = 0 improves information set decoding", EUROCRYPT, 2012.

[i.26] N. Sendrier: "Decoding one out of many", PQCrypto, 2011.

[i.27] D.J. Bernstein: "Grover vs. McEliece", PQCrypto, 2010.

[i.28] G. Kachigar and J.-P. Tillich: "Quantum information set decoding algorithms", PQCrypto, 2017.

[i.29] M. Naehrig and J. Renes: "Dual isogenies and their application to public-key compression for isogeny-based cryptography", ASIACRYPT, 2019.

[i.30] G. Pereira, J. Doliskani and D. Jao: "x-only point addition formula and faster torsion basis generation in compressed SIKE", Cryptology ePrint Archive, Report 2020/431, 2020.

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

weight: number of non-zero components of a vector or the number of non-zero coefficients of a polynomial

3.2 Symbols

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

$\boldsymbol{M}$	Bold upper-case letters denote matrices (over some ring or field)
$\boldsymbol{M}^T$	The transpose of the matrix $\boldsymbol{M}$
$\boldsymbol{I}_k$	The $k \times k$ identity matrix
$\boldsymbol{v}$	Bold lower-case letters denote vectors (over some ring or field)
$\boldsymbol{v}^T$	The transpose of the vector $\boldsymbol{v}$
$\langle \mathbf{a}, \mathbf{b} \rangle$	The inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ (defined over some common ring)
0_k	The all-zero vector consisting of k entries
$x \coloneqq y$	x is assigned the value of y
$x = y$	The values of x and y are equal
$x \neq y$	The values of x and y are not equal
$x \parallel y$	The concatenation of x and y
$\oplus$	Bitwise exclusive or
$\perp$	Failure
$\lceil x \rfloor$	The value of x when rounded to the nearest integer, with ties broken by rounding up
$\lfloor x \rceil_{q \to p}$	Modulus switching of x from modulus q to modulus p
$G(x)$	A cryptographic hash function
$H(x)$	A cryptographic hash function
$wt(f)$	The weight of the polynomial f
$\mathbb{F}$	A finite field
$\mathbb{F}_q$	A finite field modulo q
$\mathbb{Z}$	The ring of integers
$\mathbb{Z}_q$	The ring of integers modulo q
R	A ring of polynomials
R_q	A ring of polynomials modulo q
$R_q^{k \times k}$	The set of $k \times k$ matrices with coefficients in R_q
R_q^k	The set of $1 \times k$ matrices with coefficients in R_q
B_η	Centered binomial distribution of width η
χ	Probability distribution over $\mathbb{Z}$

3.3 Abbreviations

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

AES	Advanced Encryption Standard
BIKE	Bit Flipping Key Exchange
BKZ	Blockwise Korkine-Zolotarev
CCA	Chosen-Ciphertext Attack
CPA	Chosen-Plaintext Attack
DEM	Data Encapsulation Mechanism
HQC	Hamming Quasi-Cyclic
KEM	Key Encapsulation Mechanism
KDF	Key Derivation Function
LWE	Learning With Errors
LWR	Learning With Rounding
MLWE	Module Learning With Errors
MLWR	Module Learning With Rounding
NIST	National Institute of Standards and Technology
NTT	Number Theoretic Transform
OW-CPA	One-Wayness against Chosen-Plaintext Attack
PKE	Public-Key Encryption
PQC	Post-Quantum Cryptography
PRF	Pseudorandom Function
QROM	Quantum Random Oracle Model
RLWR	Ring Learning With Rounding
ROM	Random Oracle Model

SHA	Secure Hash Algorithm
SIDH	Supersingular Isogeny Diffie-Hellman
SIKE	Supersingular Isogeny Key Encapsulation
SVP	Shortest Vector Problem
XOF	Extendable Output Function

4 Introduction

The National Institute of Standards and Technology (NIST), an agency of the U.S. Department of Commerce, is responsible for producing cryptographic standards for the protection of sensitive U.S. Federal Government information. NIST standards, such as the Advanced Encryption Standard (AES) [i.1] and Secure Hash Algorithm (SHA) standards [i.2] [i.3], are used globally in many different protocols and products.

In April 2016 NIST announced [i.4] their intention to augment their existing portfolio of public-key cryptography standards [i.5], [i.6], [i.7] by developing new standards for post-quantum cryptography. In December 2016 they initiated a competition-like process with a call for proposals [i.8] for digital signatures, Public-Key Encryption (PKE) schemes, and Key Encapsulation Mechanisms (KEMs), that will remain secure even in the presence of a cryptographically relevant quantum computer. The goal of the process is to perform several rounds of public evaluation over a three to five-year period, and select one or more acceptable algorithms for standardization based on that evaluation.

NIST's deadline for submissions was November 2017. They received 69 candidates that met the minimum acceptance criteria and submission requirements: 20 digital signature schemes, and 49 PKE schemes and KEMs. Five submissions were quickly broken and formally withdrawn from the process by their designers. This left a total of 64 first round candidates [i.9]. In January 2019 NIST announced [i.10] that 26 candidate algorithms would progress to the second round of evaluation: nine digital signature schemes, and 17 PKE schemes and KEMs [i.11].

In July 2020 NIST announced [i.12] that 15 candidate algorithms would progress to the third round of evaluation. These were split into seven finalists and eight alternate candidates. NIST described the finalists as the algorithms they consider to be the most promising for the majority of use cases, and the most likely to be ready for standardization after the end of the third round. The seven finalists include three digital signature schemes, and four PKE schemes and KEMs. The alternate candidates were described as having potential for future standardization, but most likely after another round of evaluation. The eight alternate candidates include three digital signature schemes, and five PKE schemes and KEMs.

The purpose of the present document is to give concise descriptions of the nine PKE schemes and KEMs remaining in the third round of NIST's standardization process. ETSI TR 103 616 [i.14] provides similar descriptions of the six remaining digital signature schemes.

The four PKE and KEM finalists are:

- **Classic McEliece** (see clause 6.1)
- **KYBER** (see clause 6.2)
- **NTRU** (see clause 6.3)
- **SABER** (see clause 6.4)

The five PKE and KEM alternate candidates are:

- **Bit Flipping Key Exchange (BIKE)** (see clause 7.1)
- **FrodoKEM** (see clause 7.2)
- **Hamming Quasi-Cyclic (HQC)** (see clause 7.3)
- **NTRU Prime** (see clause 7.4)
- **Supersingular Isogeny Key Exchange (SIKE)** (see clause 7.5)

Each of these schemes has a different profile in terms of security properties and performance characteristics, so it is expected that some of these schemes will be more suited to specific deployment scenarios than others.

The descriptions provided in the present document are not intended to be substitutes for the detailed specifications submitted to NIST. Instead, the emphasis is on clear mathematical descriptions that facilitate easy comparison of the different schemes. Implementation details, such as how to encode polynomials as bit strings, have been omitted wherever possible. As such, some of the descriptions differ from the submissions in terms of level of abstraction, use of notation, and choice of variable names. It is expected that details of some of the schemes, such as specific parameter choices, will change during the third round of evaluation, so for consistency the descriptions are based on the official submission packages provided to NIST at the beginning of the third round [i.13].

5 Background

5.1 Terminology

A PKE scheme consists of a triple of algorithms:

- **Key Generation (PKE.KeyGen).** Returns a new public and private key pair.
- **Encryption (PKE.Enc).** Takes a public key and plaintext as input and returns a ciphertext.
- **Decryption (PKE.Dec).** Takes a private key and ciphertext as input and returns a plaintext.

NOTE 1: Some of the PKE schemes described in the present document use randomized encryption where the same public key and plaintext correspond to many different possible ciphertexts. In these schemes the randomness is derived from an additional input to the encryption process.

NOTE 2: Some of the PKE schemes described in the present document can have decryption failures where the plaintext returned by the decryption process does not match the original plaintext used in encryption. Decryption is assumed to always return a plaintext.

PKE schemes are usually unsuitable for bulk data encryption. Consequently, they are often converted into KEMs where one party encapsulates a session key for another party using the second party's public key. The session key, or a value derived from that key, is subsequently used by both parties to perform bulk data encryption using a (symmetric) Data Encapsulation Mechanism (DEM) such as AES. This approach is often referred to as the KEM/DEM paradigm.

A KEM consists of a triple of algorithms:

- **Key Generation (KEM.KeyGen).** Returns a new public and private key pair.
- **Encapsulation (KEM.Enc).** Takes a public key as input and returns a randomly selected session key and a ciphertext that is an encapsulation of the session key.
- **Decapsulation (KEM.Dec).** Takes as input a private key and a ciphertext and returns a session key.

NOTE 3: Some of the KEM schemes described in the present document can have decapsulation failures where the session key returned by the decapsulation process does not match the encapsulated session key.

In practice, PKE schemes and KEMs usually involve two parties: a sender and a recipient. The sender encrypts data or encapsulates a key for the recipient, using the recipient's public key.

5.2 Families of post-quantum algorithms

There are five prominent families of post-quantum algorithms:

- **Code-based schemes.** The security of code-based schemes depends on the difficulty of decoding vectors to find the closest codeword or the shortest error vector. Code-based schemes generally fall into two categories: McEliece-style schemes, which use error correcting codes that can be efficiently decoded given some private information; and noisy ElGamal-style schemes, which use random linear codes. Code-based cryptography lends itself more naturally to the construction of PKE schemes and KEMs than to digital signature algorithms.

- **Lattice-based schemes.** The security of lattice-based schemes depends on the difficulty of finding vectors in a lattice that are relatively short, or relatively close to some target vector. Lattice-based schemes generally fall into two categories: NTRU-style schemes, which use lattices that have been specifically constructed to contain private short vectors; and Learning With Errors (LWE) or Learning With Rounding (LWR) style schemes, which use particular classes of random lattices. Lattice-based cryptography can be used to construct PKE schemes, KEMs, and digital signature algorithms. In many cases lattice-based schemes admit worst-case to average-case security reductions, though these reductions are often not relevant to proposed parameter sets; see Annex A for more information.

- **Multivariate schemes.** The security of multivariate schemes depends on the difficulty of solving systems of quadratic or higher degree multivariate polynomials. Multivariate cryptography lends itself more naturally to the construction of digital signature algorithms than to PKE and KEM schemes.

- **Isogeny-based schemes.** The security of isogeny-based schemes depends on the difficulty of recovering a secret isogeny between a pair of elliptic curves. Isogeny-based cryptography seems to lend itself more naturally to the construction of PKE and KEM schemes than to digital signatures, though there has been some recent progress in this area.

- **Symmetric schemes.** The security of such schemes depends on the security of symmetric cryptographic primitives such as hash functions and block ciphers. Symmetric cryptography only lends itself to the construction of digital signature algorithms. Examples include hash-based signatures, such as SPHINCS+, and the PICNIC digital signature scheme.

Different post-quantum schemes utilize different algebraic structures. In code- and lattice-based cryptography, the introduction of more structure can lead to improved computational performance and reduced bandwidth requirements. However, there is a risk that additional structure could introduce new, more efficient attack possibilities. For example, the most efficient lattice-based schemes, which utilize rings of polynomials, have the most algebraic structure, but because it is unclear how to exploit this additional structure, security costings usually assume that it offers an attacker no extra advantage. Understanding whether additional algebraic structure introduces new attack possibilities, for both code- and lattice-based cryptography, remains an important research topic.

5.3 Security categories

NIST have provided guidance on the evaluation criteria they intend to apply to candidate submissions [i.8]. As part of this guidance, they have defined the following security categories in terms of the (classical or quantum) resources required to attack different NIST-approved symmetric primitives:

- **Category 1.** Resources equivalent to or greater than key recovery for AES-128.

- **Category 2.** Resources equivalent to or greater than collision search for SHA3-256.

- **Category 3.** Resources equivalent to or greater than key recovery for AES-192.

- **Category 4.** Resources equivalent to or greater than collision search for SHA3-384.

- **Category 5.** Resources equivalent to or greater than key recovery for AES-256.

NIST recommended that submissions include parameter sets that meet the requirements for categories 1, 2 and/or 3, as they believe that these categories will provide sufficient security for the foreseeable future. However, to demonstrate flexibility, and to protect against future cryptanalytic breakthroughs, NIST also recommended that submissions include at least one parameter set that provides a substantially higher level of security. Submitters were asked to include justifications for the security categories claimed for their proposed parameter sets.

5.4 Security properties

The two main security goals that are relevant for PKE schemes and KEMs are referred to as indistinguishability under chosen-plaintext, and indistinguishability under chosen-ciphertext, where the latter provides a stronger notion of security than the former. Both security goals are usually modelled as games:

- **Chosen-Plaintext Attack (CPA) security for PKE.** The attacker selects two plaintexts and is given the corresponding ciphertext for one of them. The attacker's goal is to determine which of the plaintexts was encrypted. The scheme is CPA-secure if the attacker cannot do significantly better than guessing.

- **Chosen-Ciphertext Attack (CCA) security for PKE.** The attacker selects two plaintexts and is given the corresponding ciphertext for one of them. The attacker's goal is to determine which of the plaintexts was encrypted. The attacker is allowed to request the decryption of ciphertexts of their choice, except for the challenge ciphertext. The scheme is CCA-secure if the attacker cannot do significantly better than guessing, even with access to the decryption oracle.

- **Chosen-Plaintext Attack (CPA) security for KEMs.** The attacker is given a ciphertext and either a session key that is encapsulated by that ciphertext, or a uniformly random key. The attacker's goal is to determine whether they have been given the session key, or a random key. The scheme is CPA-secure if the attacker cannot do significantly better than guessing.

- **Chosen-Ciphertext Attack (CCA) security for KEMs.** The attacker is given a ciphertext and either a session key that is encapsulated by that ciphertext, or a uniformly random key. The attacker's goal is to determine whether they have been given the session key, or a random key. The attacker is allowed to request the decapsulation of ciphertexts of their choice, except for the challenge ciphertext. The scheme is CCA-secure if the attacker cannot do significantly better than guessing, even with access to the decryption oracle.

Expanded definitions of these properties and additional definitions are given in Annex B.

There are standard techniques available for converting a CPA-secure PKE scheme into a CCA-secure KEM. The most common approach is to use a variant of the Fujisaki-Okamoto transform [i.15]. Broadly speaking, this usually involves deriving the randomness required for encryption (or encapsulation) from the value to be encrypted (or encapsulated); note that this includes the randomness required for the sender to generate an ephemeral key pair. This allows the recipient to attempt to reconstruct the received ciphertext, and check that the protocol has been followed as expected.

As mentioned above, CCA security is stronger than CPA security: a recipient's public key that is used for encryption or encapsulation in a CPA-secure scheme can only be safely used once, or the security of the scheme could be compromised, but a recipient's public key that is used for encryption or encapsulation in a CCA-secure scheme can be safely reused. If an active adversary is able to reuse a recipient's public key in a CPA-secure scheme, they can send messages that consist of erroneous ciphertexts that will reveal information about the recipient's private key.

NIST have stated that they intend to standardize at least one CCA-secure PKE scheme or KEM for general use, and that they will consider standardizing a CPA-secure PKE scheme or KEM for applications where keys are never reused [i.8]. NIST have not mandated that submissions include proofs of CPA or CCA security, but they will consider proofs where they are made available.

5.5 Finalists and alternate candidates at a glance

Table 1 contains a summary of each of the NIST public-key encryption and key encapsulation finalists.

Table 1: Summary of finalists

Scheme	Family	Type	Structure	Categories					Security		Comments
				1	2	3	4	5	CPA	CCA	
Classic McEliece	Codes	McEliece	None	Y		Y		Y	Y	Y	NOTE 1
KYBER	Lattice	LWE	Module	Y		Y		Y	Y	Y	NOTE 2
NTRU	Lattice	NTRU	Ring	Y		Y			Y	Y	NOTE 3
SABER	Lattice	LWR	Module	Y		Y		Y	Y	Y	
NOTE 1: Classic McEliece is a merger of the second round Classic McEliece and NTS-KEM submissions.											
NOTE 2: The KYBER submission states that only the CCA version is to be used in practice.											
NOTE 3: NTRU is a merger of the first round NTRUEncrypt and NTRU-HRSS-KEM submissions.											

Table 2 contains a summary of each of the NIST public-key encryption and key encapsulation alternate candidates.

Table 2: Summary of alternate candidates

Scheme	Family	Type	Structure	Categories					Security		Comments
				1	2	3	4	5	CPA	CCA	
BIKE	Codes	McEliece	Ring	Y		Y		Y	Y		NOTE 1
FrodoKEM	Lattice	LWE	None	Y		Y		Y	Y	Y	
HQC	Lattice	Random	Ring	Y		Y		Y	Y	Y	
NTRU Prime	Lattice	NTRU	Field	Y	Y	Y	Y	Y	Y	Y	NOTE 2
SIKE	Other	Isogeny	None	Y	Y	Y		Y	Y	Y	
NOTE 1: The BIKE submission does not formally claim that the proposed parameters are CCA-secure. NOTE 2: The NTRU Prime submission states that only the CCA version is to be used in practice.											

6 Finalists

6.1 Classic McEliece

6.1.1 Overview

Classic McEliece is a merger of the Classic McEliece and NTS-KEM submissions from the second round of the NIST standardization process. Classic McEliece consists of a CCA-secure KEM built from a OW-CPA-secure PKE scheme using a variant of the Fujisaki-Okamoto transform from [i.16]. The security of Classic McEliece is based on the difficulty of the syndrome decoding problem for general binary linear codes.

A binary Goppa code is defined by a monic irreducible polynomial $g(X) \in \mathbb{F}_{2^m}[X]$ of degree t, and a sequence of n distinct elements $(\alpha_1, \ldots, \alpha_n)$ where $\alpha_i \in \mathbb{F}_{2^m}$. These define a parity-check matrix $\widetilde{\boldsymbol{H}} \in \mathbb{F}_2^{t \times n}$ by setting the (i, j)-th entry of $\widetilde{\boldsymbol{H}}$ to be $\alpha_j^{i-1}/g(\alpha_j)$. The matrix $\widetilde{\boldsymbol{H}}$ is associated with a parity-check matrix $\widehat{\boldsymbol{H}} \in \mathbb{F}_2^{mt \times n}$ to define a binary linear code $\mathcal{C} = \{ \boldsymbol{c} \in \mathbb{F}_2^n \mid \widehat{\boldsymbol{H}}\boldsymbol{c}^T = 0\}$ of length n and dimension $k = n - mt$ with an efficient algorithm for decoding up to t errors.

Given a public general parity-check matrix $\boldsymbol{H} \in \mathbb{F}_2^{(n-k) \times n}$ for the code, it is believed to be computationally hard to recover the private Goppa parity-check matrix $\widehat{\boldsymbol{H}} \in \mathbb{F}_2^{(n-k) \times n}$ that allows for fast decoding. It is also believed that without the private parity-check matrix, there are no decoding algorithms that are more efficient than generic information set decoding.

Classic McEliece is defined as a CCA-secure KEM only, as the underlying PKE scheme is a building block that is not intended as a separate submission to the NIST standardization process.

6.1.2 Parameters

The main parameters for Classic McEliece are:

- n, the code length;
- t, the error-correction capability;
- m, the degree of the field $\mathbb{F}_{2^m}$; and
- $k = n - mt$, the code dimension.

6.1.3 Auxiliary primitives

Classic McEliece makes use of two auxiliary, symmetric primitives:

- H, a 256-bit cryptographic hash function; and
- KDF, a key derivation function.

The submission describes how to use SHAKE-256 to instantiate these primitives.