

ICS 35.030

**Festlegung eines Quantencomputer-resistenten
Verschlüsselungsverfahrens; Text Englisch**

Definition of a quantum computer resistant encryption scheme; Text in English

Définition d'un système de cryptage résistant aux ordinateurs quantiques; Texte en anglais

Gesamtumfang 36 Seiten

Dieses Dokument wurde durch die im Vorwort genannten Verfasser erarbeitet und verabschiedet.

Contents

	Page
Foreword	5
Introduction	6
1 Scope	7
2 Normative references	7
3 Terms and definitions.....	7
4 Symbols and abbreviations.....	8
5 Specification of KINDI-KEM — A post-quantum key encapsulation mechanism.....	11
5.1 General	11
5.2 Specification of KINDI-KEM — CPA-secure encryption.....	12
5.2.1 Description of key generation.....	12
5.2.2 Description of the encryption procedure.....	16
5.2.3 Description of the decryption procedure.....	20
5.3 Specification of KINDI-KEM – CCA-secure key encapsulation mechanism	23
5.3.1 Description of key generation.....	23
5.3.2 Description of the encapsulation procedure.....	23
5.3.3 Description of the decapsulation procedure.....	24
5.4 Number-theoretic transform (NTT).....	24
5.5 Optimizations	25
5.5.1 General	25
5.5.2 Polynomial multiplication.....	25
5.5.3 Advanced Vector Extensions (AVX)	25
5.5.4 Modular reductions	26
5.5.5 Encoding and decoding.....	26
5.5.6 Secret key.....	26
5.6 Parameters and instantiations	26
5.6.1 Choice of parameter sets.....	26
5.6.2 SHAKE instantiation	27
5.6.3 Advanced Encryption Standard (AES) instantiation.....	27
6 Implementation security	28
6.1 General	28
6.2 Learnings from attacks on classical cryptography	28
6.3 Specific threats to lattice-based cryptography	29
7 Requirements for cryptosystems.....	29
7.1 General	29
7.2 Target of evaluation (TOE).....	30
7.3 Requirements.....	31
7.3.1 General description of the TOE.....	31
7.3.2 Determination of the TOE.....	31
7.3.3 Product lifecycle (installation, delivery, decommissioning)	31
7.3.4 Architecture and design.....	31
7.3.5 Cryptography	31
7.3.6 Vulnerability management	31
7.3.7 Patch management	32
7.3.8 Change management.....	32

7.3.9 Incident management	32
7.3.10 Risk management	32
Annex A (informative) Security risk of quantum computing and counter measures	33
A.1 Security risk of quantum computing	33
A.1.1 Quantum computing differentiation from classical computing	33
A.1.2 Quantum computing impact on encryption schemes	34
A.2 Counter measures	34
A.2.1 General	34
A.2.2 Post-quantum cryptography	34
A.2.3 Quantum cryptography	34
A.2.4 Hybrid cryptography	35
A.3 Applications of post-quantum key encapsulation mechanisms	35
A.3.1 General	35
A.3.2 Key exchange and encryption	35
Bibliography	36

Tables

Table 1 — Key generation algorithm for CPA secure encryption	13
Table 2 — Subroutine to generate the NTT of the random matrix A	13
Table 3 — Subroutine to convert binary strings to integer representations	14
Table 4 — Subroutine to generate secret polynomials	15
Table 5 — Subroutine to convert polynomials into a binary string	16
Table 6 — Subroutine to convert an integer value into a binary string	16
Table 7 — Encryption routine	17
Table 8 — Auxiliary encryption subroutine	17
Table 9 — Subroutine to convert a bit string into a vector of polynomials	18
Table 10 — Subroutine to deterministically derive secrets from a random string	19
Table 11 — Subroutine to center the coefficients of polynomials around zero	19
Table 12 — Subroutine to compress a polynomial	20
Table 13 — Decryption routine	21
Table 14 — Subroutine to recover all secret values	21
Table 15 — Subroutine to decompress a polynomial	22
Table 16 — Subroutine to recover a binary polynomial from an input polynomial	22
Table 17 — Subroutine to convert centered coefficients into non-negative integers	22

Table 18 — Key generation algorithm for key encapsulation	23
Table 19 — Encapsulation algorithm	24
Table 20 — Decapsulation algorithm	24
Table 21 — Parameters.....	26
Table 22 — Sizes of keys and ciphertexts.....	27
Table A.1 — Performance differences between classical computers and quantum computers.....	33

Foreword

This DIN SPEC has been developed according to the PAS procedure. The development of a DIN SPEC according to the PAS procedure is carried out in DIN SPEC-consortiums and does not require the participation of all stakeholders.

This document has been developed and adopted by the initiator(s) and authors named below:

- QuantiCor Security GmbH: Dr. Rachid El Bansarkhani;
- KPMG AG Wirtschaftsprüfungsgesellschaft: Hans-Peter Fischer, Andreas Schwab;
- Protiviti GmbH: Dr. Michael Riecker;
- Technische Universität Darmstadt: Dr. Juliane Krämer;
- TÜV NORD CERT GmbH: Matthias Springer.

At present, there are no standards covering this topic in the body of German Standards.

DIN SPECs are not part of the body of German Standards.

A draft of this DIN SPEC has not been published.

Despite great efforts to ensure the accuracy, reliability and precision of technical and non-technical information, the DIN SPEC-consortium cannot give any explicit or implicit assurance or warranty in respect of the accuracy of the document. Users of this document are hereby made aware that the consortium cannot be held liable for any damage or loss. The application of this DIN SPEC does not release users from the responsibility for their own actions and is applied at their own risk.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. DIN shall not be held responsible for identifying any or all such patent rights.

Provision of this document free of charge as a PDF via the Beuth WebShop has been financed in advance.

For current information on this document, please go to DIN's website (www.din.de) and search for the document number in question.

Introduction

Quantum computers can break cryptographic schemes that are based on the discrete logarithm problem or the RSA assumption. This poses a major threat to IT infrastructures and systems not transitioning to secure alternatives such as cryptosystems based on well-studied lattice problems that are conjectured to withstand quantum computer attacks (see Annex A for additional information).

Lattices as mathematical objects have been studied by early mathematicians due to their extremely rich combinatorial structure appearing in many areas of mathematics. But they have also been extensively utilized in cryptography to build powerful cryptosystems, where the security stems from the worst-case hardness of well-studied lattice problems. Beside the NTRU assumption the main computational assumptions exploited to build lattice-based cryptosystems are the hardness of the problems LWE/SIS , Ring-LWE/Ring-SIS and $\text{Module-LWE/Module-SIS}$. The efficiency of cryptosystems increases by imposing more structure. Thus, structured instantiations of the respective schemes represent attractive targets for use in practice. The decision problems are widely employed to build lattice-based encryption schemes, where the public key and ciphertexts can essentially be represented as LWE instances or their structured variants. CPA-security is thus obtained almost for free.

KINDI-KEM is a lattice-based key encapsulation mechanism proposed by El Bansarkhani [1]. Key encapsulation mechanisms allow parties, for instance, to securely establish a common secret key during the execution of the TLS protocol in order to protect the communication channel (see Annex A for additional information). KINDI-KEM is based on Module-LWE and employs trapdoors in order to recover the secret and error terms used to conceal sensitive data. It is based on the encryption scheme LARA [2] and the ALWE -based construction [3]. The CPA-secure encryption scheme embeds the message into the error term and further encrypts a random string (similar to a KEM) in the secret term, from which a key can be derived for a symmetric key cipher. By use of the trapdoor, the secret and error terms are recovered during decryption and are thus available for inspection. In many encryption schemes, this is indeed not possible. Since the message is embedded in the error term, small expansion factors can be realized at competitive parameters, i.e., more keys or data can be encrypted per ciphertext bit (e.g. for sign-then-encrypt scenarios). Furthermore, it can easily be transformed to ensure CCA security.

1 Scope

This document defines a quantum computer resistant encryption and key encapsulation mechanism (KINDI-KEM) with security based on the hardness of Module-LWE. The underlying encryption scheme hides arbitrary data in the error term and further encrypts a random string in the secret term of Module LWE instances which is exploited for the key encapsulation mechanism. For this purpose, the necessary functionality and relevant security features are defined. Furthermore, parameter sets are defined that specify different security and efficiency levels. The security level comprises the bit security of the procedure and the efficiency level comprises the sizes of the keys and key texts.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

FIPS PUB 197, *Advanced Encryption Standard (AES)*¹

FIPS PUB 202, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*²

SP 800-38A, *Recommendation for Block Cipher Modes of Operation: Methods and Techniques*³

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

DIN and DKE maintain terminological databases for use in standardization at the following addresses:

- DIN-TERMinologieportal: available at <https://www.din.de/go/din-term>
- DKE-IEV: available at <http://www.dke.de/DKE-IEV>

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>

3.1

Quantum computing

automated data processing (computing) through the systematic utilization of quantum effects

Note 1 to entry: In the broadest sense, the term quantum computing includes hardware (e.g. quantum computers), software (e.g. quantum algorithms) and the data processing processes themselves (e.g. quantum simulations).

1 [Access on 2020-11-19]. Available at: <https://csrc.nist.gov/publications/detail/fips/197/final>

2 [Access on 2020-11-19]. Available at: <https://csrc.nist.gov/publications/detail/fips/202/final>

3 [Access on 2020-11-19]. Available at: <https://csrc.nist.gov/publications/detail/sp/800-38a/final>

3.2

Key encapsulation mechanism

certain type of public key encryption scheme that is developed to securely transport keys for symmetric key primitives

4 Symbols and abbreviations

The following abbreviations are used in this document:

AES	Advanced Encryption Standard;
ALWE	Augmented learning with errors;
AVX	Advanced Vector Extensions;
CCA	Chosen ciphertext attack;
CPA	Chosen plaintext attack;
EOF	Extendable output function;
FO	Fujisaki-Okamoto;
IIoT	Industrial internet of things;
IoT	Internet of things;
KEM	Key encapsulation mechanism;
LWE	Learning with errors;
NTRU	Public key cryptosystem;
NTT	Number-theoretic transform;
NTT^{-1}	Inverse NTT;
PQ	Post quantum;
RSA	Public key cryptosystem;
SHA3	Secure hash algorithm 3;
SIS	Short integer solution;
SVP	Shortest Vector Problem;
TLS	Transport layer security;
TOE	Target of evaluation;
VPN	Virtual private network;
XOF	Extendable output function.

The following symbols are used in this document:

A	represents a matrix of polynomials;
a	is a polynomial;
a_i	is a polynomial for any integer i ;
a_i	is a positive integer for any non-negative integer i ;
b	is a bit string;

$\mathbf{b}_i$	is a polynomial for any integer i ;
b_i	is the i -th byte for any integer i ;
bit_i	is a bit for any integer i ;
bst	is a bit string;
$\mathbf{c}$	is a vector of polynomials;
c	is a positive integer;
$\mathbf{c}_i$	is a polynomial for any integer i ;
c_i	is a bit string for any positive integer i ;
$cipher_i$	is a bit string for any integer i ;
$coins$	is a bit string;
$counter$	is a non-negative integer;
$ctxt$	is a bit string;
d	is a non-negative integer;
$diff$	is a bit value;
$\mathbf{e}$	is a vector of polynomials;
$\mathbf{e}_i$	is a polynomial for any integer i ;
$\mathbf{f}_i$	is a polynomial for any integer i ;
$\mathbf{g}$	is a vector of polynomials;
$\mathbf{g}_i$	is a polynomial for any integer i ;
g_1	is an integer;
g_2	is a non-negative integer;
i	is an integer;
j	is an integer;
K	is a bit string;
k	is a positive integer;
m_i	is a positive integer for any integer i ;
msg	is a bit string;
n	is a positive integer;
pk	is the public key represented as a bit string;
$pkstrg$	is a bit string;
psd	is a bit string;
p	is a positive integer;
q	is a positive integer;
R	is the ring $\mathbb{Z}[X]/(X^n + 1)$;
R_q	is the ring $\mathbb{Z}_q[X]/(X^n + 1)$;
$\mathbf{r}$	is a vector of polynomials;
$\mathbf{r}_i$	is a polynomial for any integer i ;

r	is a non-negative integer;
rn	is a rational number;
S	is a set;
$\mathbf{s}$	is a vector of polynomials;
sd	is a bit string;
$secSeed$	is a bit string;
$\mathbf{s}_i$	is a polynomial for any integer i ;
sk	is the secret key represented as a bit string;
ssd	is a bit string;
$skstrg$	is a bit string;
$\mathbf{t}$	is a polynomial;
t	is a positive integer;
$\mathbf{u}$	is a vector of polynomials;
$\mathbf{u}_i$	is a polynomial for any integer i ;
$\mathbf{v}$	is a polynomial;
X	is a variable;
γ	is an integer;
λ	is a positive integer;
μ	is a bit string;
ω	is an integer;
$\mathbf{A}^T$	represents the transpose of the matrix $\mathbf{A}$;
$\overline{\mathbf{A}}$	denotes the NTT representation of matrix $\mathbf{A}$;
$\mathbf{A}[i][j]$	denotes the polynomial of the matrix $\mathbf{A}$ at the i -th row and j -th column;
$\overline{\mathbf{A}}[i][j]$	denotes the polynomial of the matrix $\overline{\mathbf{A}}$ at the i -th row and j -th column;
$\mathbf{a}[k]$	is the k -th coefficient of the polynomial $\mathbf{a}$, where $k \geq 0$;
b'	is a bit string;
$\overline{\mathbf{b}}_i$	is the NTT representation of the polynomial $\mathbf{b}_i$ for any integer i ;
$\overline{\mathbf{b}}_i[j]$	is the j -th coefficient of the polynomial $\overline{\mathbf{b}}_i$, where $j \geq 0$ and $i \geq 0$;
$\mathbf{c}'$	is a vector of polynomials;
$\overline{\mathbf{c}}$	is the NTT representation of $\mathbf{c}$;
$\hat{\mathbf{c}}_1$	is a polynomial;
$\hat{c}_1$	is a bit string;
$\mathbf{e}_i[j]$	is the j -th coefficient of the polynomial $\mathbf{e}_i$, where $j \geq 0$ and $i \geq 0$;
$\overline{\mathbf{e}}$	is the NTT representation of $\mathbf{e} + \mathbf{g}$;
$\mathbf{f}_i[j]$	is the j -th coefficient of the polynomial $\mathbf{f}_i$, where $j \geq 0$ and $i \geq 0$;
$\overline{\mathbf{f}}_i$	is the NTT representation of the polynomial $\mathbf{f}_i$ for any integer i ;

$\bar{f}_i[j]$	is the j -th coefficient of the polynomial $\bar{f}_i$, where $j \geq 0$ and $i \geq 0$;
$g_i[j]$	is the j -th coefficient of the polynomial g_i , where $j \geq 0$ and $i \geq 0$;
$\bar{p}$	is the NTT representation of p ;
$\bar{p}_i$	is the NTT representation of the polynomial p_i ;
$r_i[j]$	is the j -th coefficient of the polynomial r_i , where $j \geq 0$ and $i \geq 0$;
$\bar{r}$	is the NTT representation of r ;
s'	is a polynomial;
$\hat{s}_1$	is a polynomial;
v'	is a polynomial;
$\mathbb{Q}$	is the set of rationals;
$\mathbb{Q}[X]$	is the set of polynomials in X with rational coefficients;
$\mathbb{Z}$	is the set of integers;
$\mathbb{Z}_q$	is the set of integers modulo q ;
$\mathbb{Z}[X]$	is the set of polynomials in X with integer coefficients;
ℓ	is a positive integer;
$[p]$	is a polynomial with coefficients all equal to p ;
$\circ$	represents pointwise or coefficient-wise multiplication, e.g. $\bar{f}_3 = \bar{f}_1 \circ \bar{f}_2 = \begin{bmatrix} \bar{f}_1[0] \cdot \bar{f}_2[0] \\ \vdots \\ \bar{f}_1[n-1] \cdot \bar{f}_2[n-1] \end{bmatrix};$
$\oplus$	is the XOR-operation.

5 Specification of KINDI-KEM — A post-quantum key encapsulation mechanism

5.1 General

The inputs to and outputs of the algorithms key generation, encryption, decryption, encapsulation, decapsulation are byte arrays. Let bst be a bit string. Then, $bst[i]$ is the bit value at position i and $bst[i:j]$ denotes the substring starting at position i and ending at position j for $i, j \geq 0$. The expression $b||c$ denotes the concatenation of two bit strings or byte arrays b and c , where b is stored at lower addresses than c (e.g. $bst[0:j] = b, bst[j+1:k] = c$ for $0 < j < k$).

The ring R is defined as $\mathbb{Z}[X]/(X^n + 1)$ and R_q as $\mathbb{Z}_q[X]/(X^n + 1)$, where q (e.g. $q = 7681$) denotes the modulus and $n = 2^k$ the ring dimension for an integer $k > 1$ (e.g. $n = 256$). For a matrix $A \in R_q^{\ell \times \ell}$ (or $A \in R^{\ell \times \ell}$) or vector $r \in R^\ell$ of polynomials the elements A^T or r^T denote the respective transposes (e.g. $[b_1, \dots, b_\ell]^T = \begin{bmatrix} b_1 \\ \vdots \\ b_\ell \end{bmatrix}$).

For a matrix A , the entry at the i -th row and j -th column is denoted by $A[i][j]$. For example, $A[i][j] \in R_q$ is a polynomial for $A \in R_q^{\ell \times \ell}$. For a polynomial r_j the i -th coefficient is denoted by $r_j[i]$. Any polynomial r_j can be represented as $r_j[0] + r_j[1] \cdot X + r_j[2] \cdot X^2 + \dots + r_j[n-1] \cdot X^{n-1}$, where $r_j[0]$ is the constant coefficient. In implementations just the coefficients of a polynomial are stored.

For any $rn \in \mathbb{Q}$ the closest integer to rn is denoted as $[rn]$. Elements of $\mathbb{Z} + 0.5$ are always rounded up to the next greater integer. For any polynomial $r \in \mathbb{Q}[X]/(X^n + 1)$ the operation $[r]$ rounds each coefficient of r to