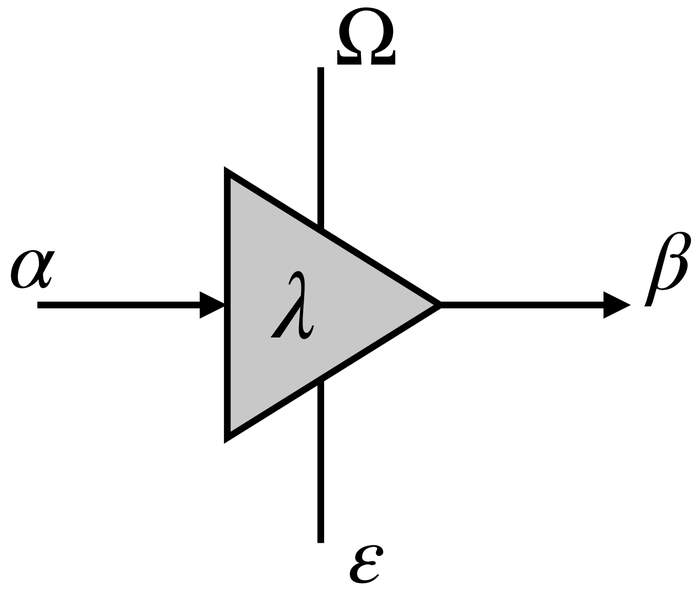


Somewhat Gentle Introduction to PQC

Jiří Pavlů and Tomáš Rosa (*presenting*)
Cryptology and Biometrics Competence Centre, Raiffeisenbank, Prague
Faculty of Mathematics and Physics, Charles University, Prague

extended math version

The **Two Flavors** of Quantum-Resistant Mechanisms

- **Cryptographic protocols based on quantum mechanics laws**
  - Quantum Key Distribution (QKD), for instance
  - unconditionally secure, provided everything in the whole scheme is
  - speed versus distance limits
  - cloud limits or even impossibility
  - not every classical scheme has its practical quantum variant, e.g. signatures
  - security authorities NSA, BSI, NCSC, ANSSI stay highly reserved at this moment
- **Classical algorithms for classical computing platforms**
  - post-quantum cryptographic suites
  - recommended widespread approach and our main topic here

The Algorithmic Approach of PQC

Traditional cryptosystems		Purpose	PQC Replacements	
Integer factorization	RSA	Encryption	Crystals-Kyber (ML-KEM, FIPS 203)	Learning with errors
Discrete logarithm	ElGamal			
	DH			
Elliptic curve discrete logarithm	ECDH			
Integer factorization	RSA	Signature	Crystals-Dilithium (ML-DSA, FIPS 204)	Learning with errors Short integer solution
Discrete logarithm	DSA		Falcon (FN-DSA, FIPS 206)*	
Elliptic curve discrete logatithm	ECDSA		SPHINCS+ (SLH-DSA, FIPS 205)	Hash inversion

*) FIPS 206 draft ~~is~~ was "... planned for late 2024."

Retroactive Cryptanalysis - Mosca's Inequality

$$X + Y < Z$$

- *X* is the relative time the information asset must remain secure
- *Y* is the relative migration time to a safe algorithm
- *Z* is the relative time left until effective cryptanalytic tool for the former method is available

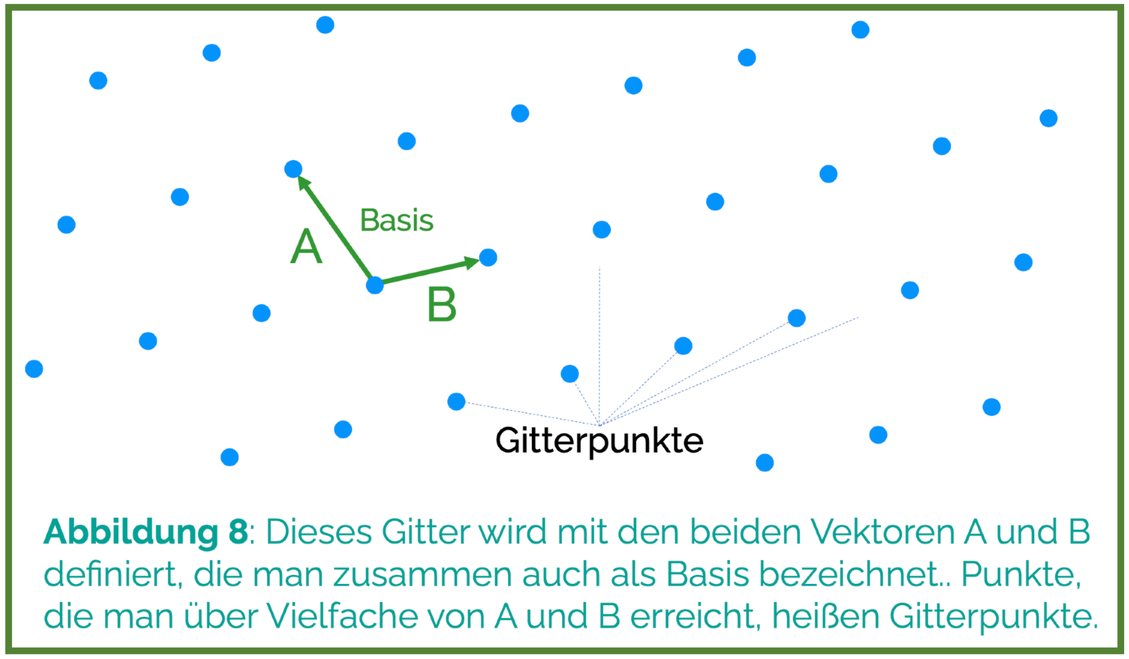


Abbildung 8: Dieses Gitter wird mit den beiden Vektoren A und B definiert, die man zusammen auch als Basis bezeichnet.. Punkte, die man über Vielfache von A und B erreicht, heißen Gitterpunkte.

Usual illustration of a discrete subgroup of a real Euclidean vector space (*called lattice*), featuring the good and the bad basis.

Note this is just an illustration, as especially in 2D we have the Gauss reduction algorithm for finding a good basis from a bad one.

— Schmeh, K.: *Post-Quanten-Kryptografie verständlich erklärt*, 2022

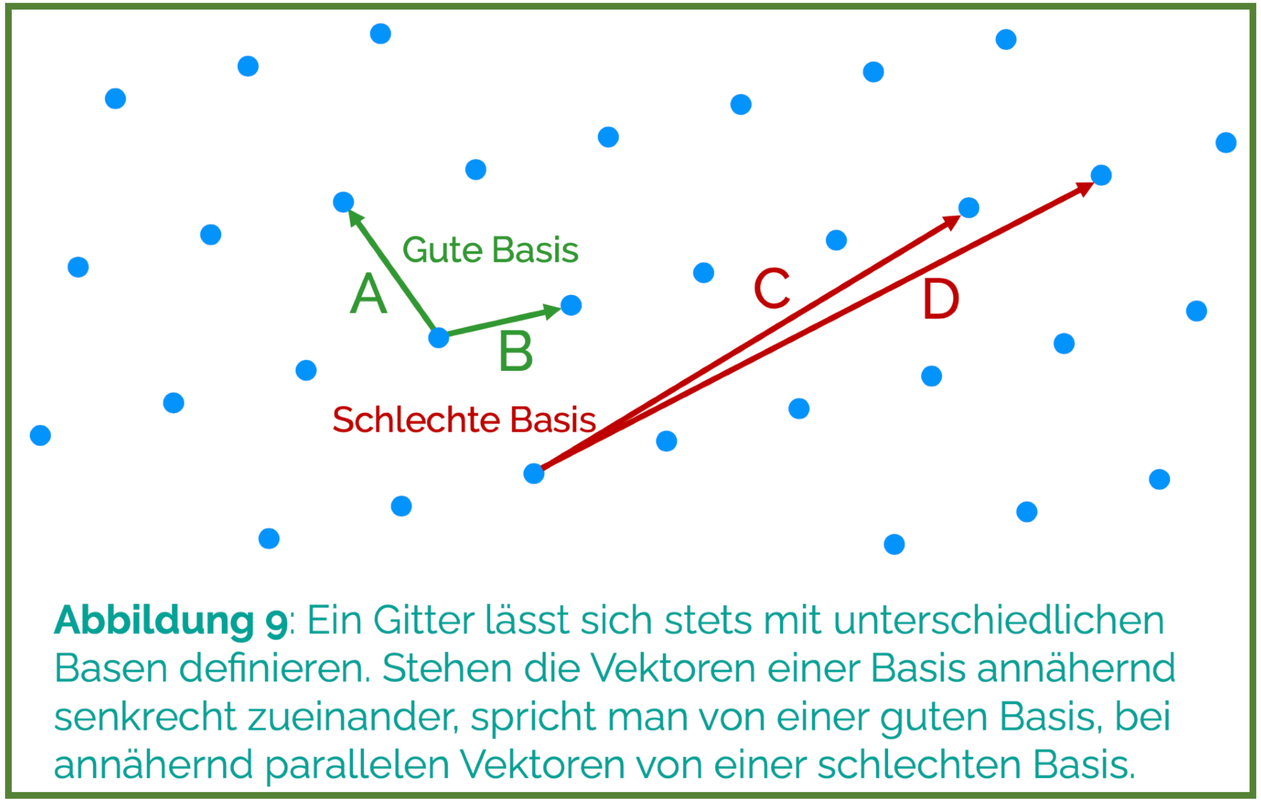


Abbildung 9: Ein Gitter lässt sich stets mit unterschiedlichen Basen definieren. Stehen die Vektoren einer Basis annähernd senkrecht zueinander, spricht man von einer guten Basis, bei annähernd parallelen Vektoren von einer schlechten Basis.

Lattice-Based Encryption - Geometric Illustration

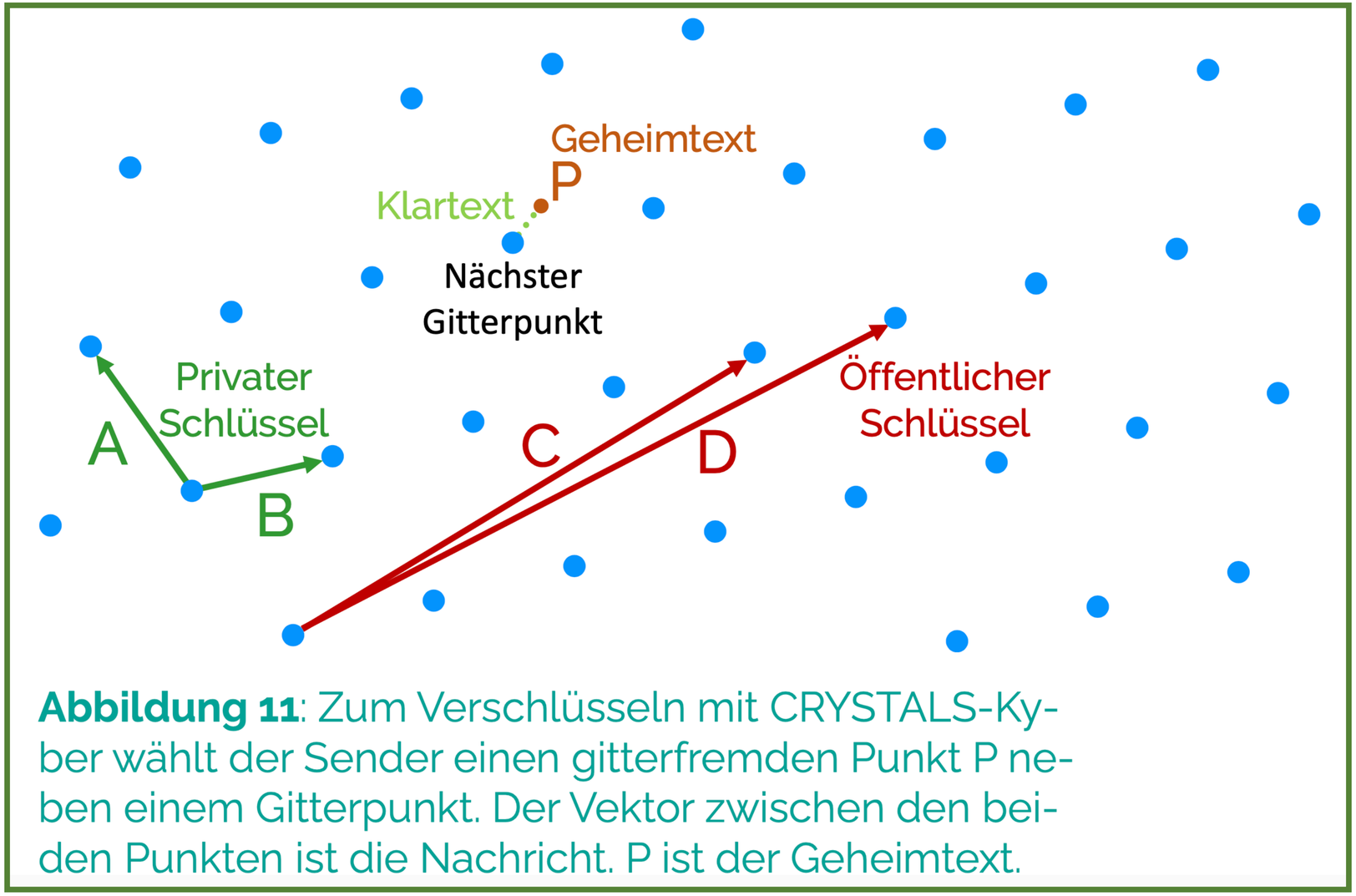


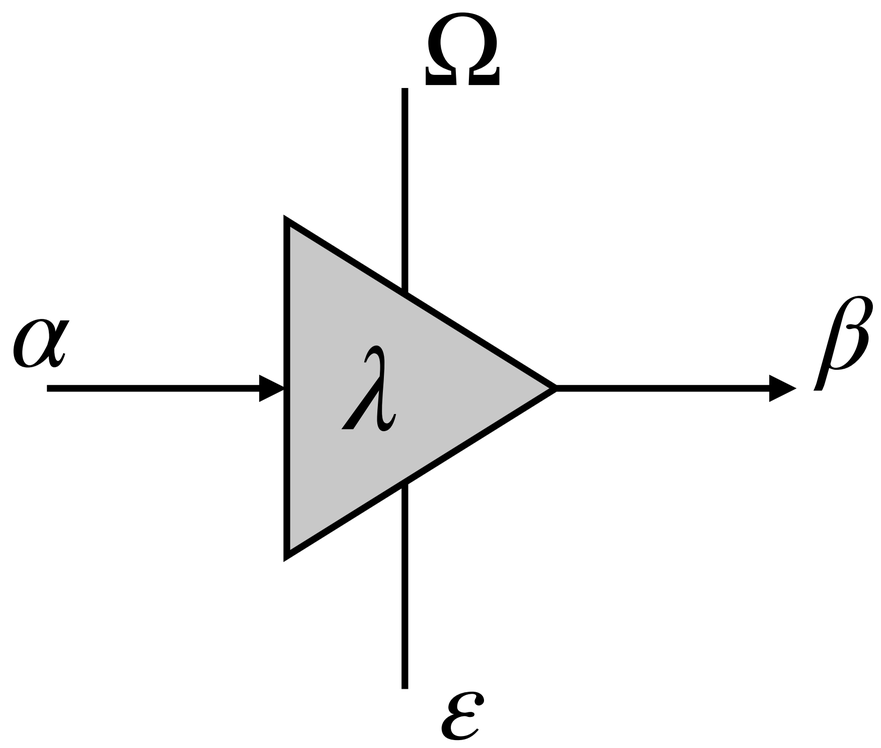
Abbildung 11: Zum Verschlüsseln mit CRYSTALS-Kyber wählt der Sender einen gitterfremden Punkt P neben einem Gitterpunkt. Der Vektor zwischen den beiden Punkten ist die Nachricht. P ist der Geheimtext.

— Schmeh, K.: *Post-Quanten-Kryptografie verständlich erklärt*, 2022

There is just one slight caveat in there...

… this is NOT how ML-KEM is constructed

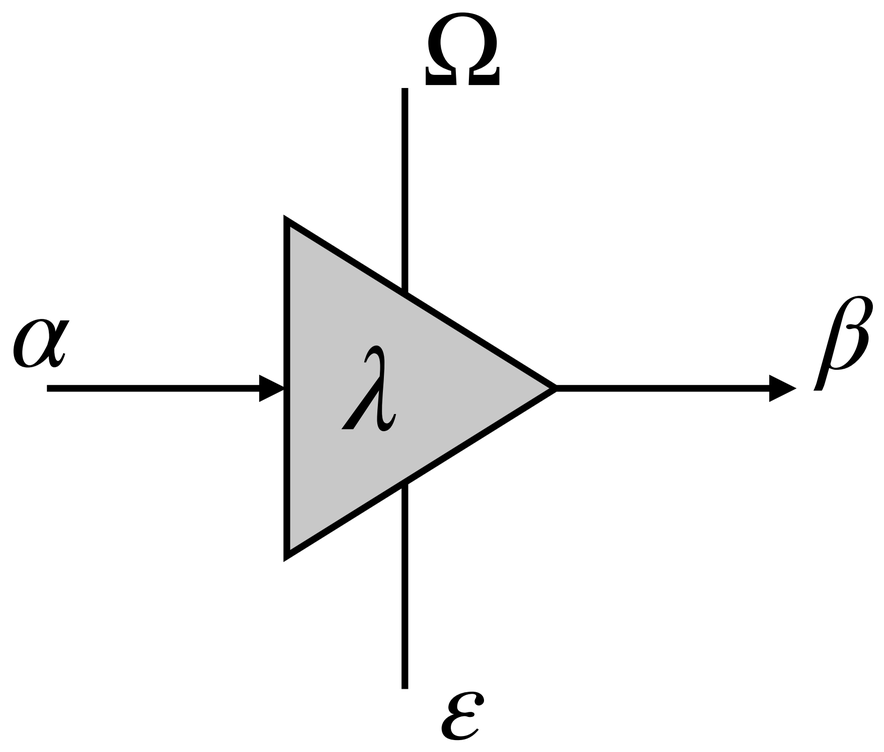
LWE Gate - General Definition



$$\Omega \times \alpha + \varepsilon = \beta$$

Standard-LWE λ_0	$\Omega \in \mathbb{F}_q^{n \times m} = \mathbb{Z}_q^{n \times m}$ $\alpha \in \mathbb{F}_q^m$ $\beta, \varepsilon \in \mathbb{F}_q^n$
Ring-LWE λ_ρ	$\Omega \in R_q = \mathbb{Z}_q[x] \big/ \langle x^n + 1 \rangle$ $\alpha \in R_q$ $\beta, \varepsilon \in R_q$
Module-LWE λ_μ	$\Omega \in R_q^{n \times m}, R_q$ *see above* $\alpha \in R_q^m$ $\beta, \varepsilon \in R_q^n$

LWE Gate - Security Arguments



$$\Omega \times \alpha + \varepsilon = \beta$$

Standard-LWE λ_0	β indistinguishable from $u \leftarrow \left[\mathbb{Z}_q^n\right]$ in particular, $\beta \mapsto \alpha$ is hard
Ring-LWE λ_ρ	β indistinguishable from $u \leftarrow \left[R_q\right]$ in particular, $\beta \mapsto \alpha$ is hard
Module-LWE λ_μ	β indistinguishable from $u \leftarrow \left[R_q^n\right]$ in particular, $\beta \mapsto \alpha$ is hard

Daring to be Like Diffie-Hellman...

$$\Omega \times \alpha + \varepsilon \qquad \text{versus} \qquad g^a \mod p$$

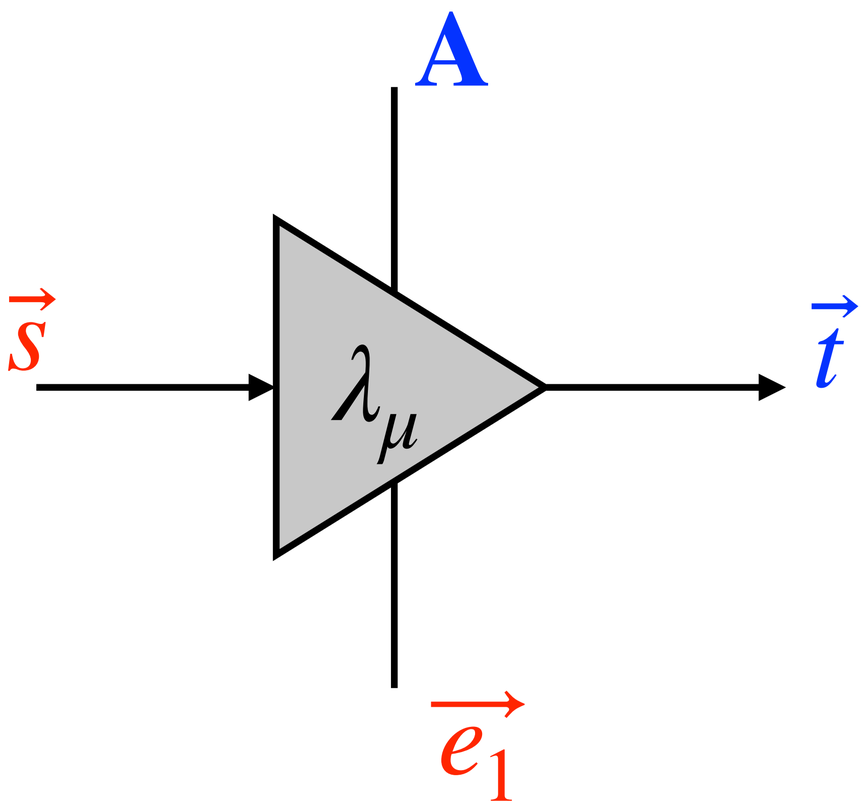
Adjoint instead of Abelian Group

$$\left\langle \mathbf{A}\vec{v}, \overrightarrow{w} \right\rangle = \left\langle \vec{v}, \mathbf{A}^T\overrightarrow{w} \right\rangle$$

versus

$$\left(g^a\right)^b \mod p = \left(g^b\right)^a \mod p$$

Module-LWE Encryption Scheme
setup phase



$\overrightarrow{e_1} \leftarrow [\beta_2]^m$

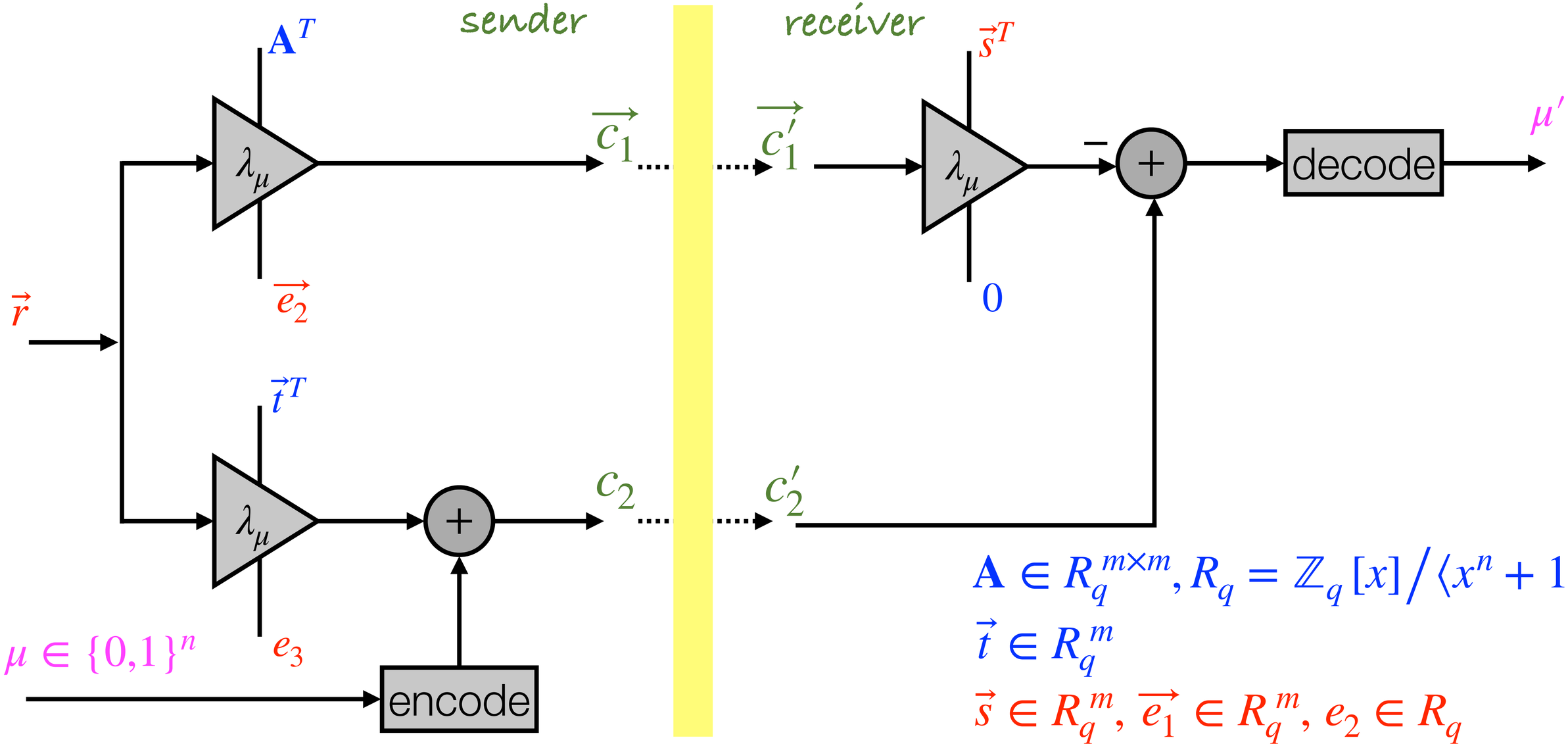
sk: $\vec{s} \leftarrow [\beta_1]^m$

pk: $\mathbf{A} \leftarrow R_q^{m \times m},\ R_q = \mathbb{Z}_q[x] / \langle x^n + 1 \rangle$

pk: $\vec{t} = \mathbf{A}\vec{s} + \overrightarrow{e_1}$

we set $m = n$, for the general LWE gate

Module-LWE Encryption Scheme
encryption/decryption of *n*-bit messages



$\mathbf{A} \in R_q^{m \times m}, R_q = \mathbb{Z}_q[x] / \langle x^n + 1 \rangle$

$\vec{t} \in R_q^m$

$\vec{s} \in R_q^m, \overrightarrow{e_1} \in R_q^m, e_2 \in R_q$

FIPS 203

Federal Information Processing Standards Publication

Module-Lattice-Based Key-Encapsulation Mechanism Standard

Category: Computer Security **Subcategory: Cryptography**

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
https://doi.org/10.6028/NIST.FIPS.203

Published August 13, 2024

- Fujisaki-Okamato extension to convert IND-CPA scheme to CCA2 secure one
- Number Theoretic Transform for faster ring operations
- Mandatory and recommended security checks
- Key and ciphertext data length optimizations
- Precise definition of the three parametric ML-KEM schemes based on M-LWE
  - ***M**odule **L**attice refers to lattices corresponding to certain R-modules*

[https://doi.org/10.6028/NIST.FIPS.203]

Table 2. Approved parameter sets for ML-KEM

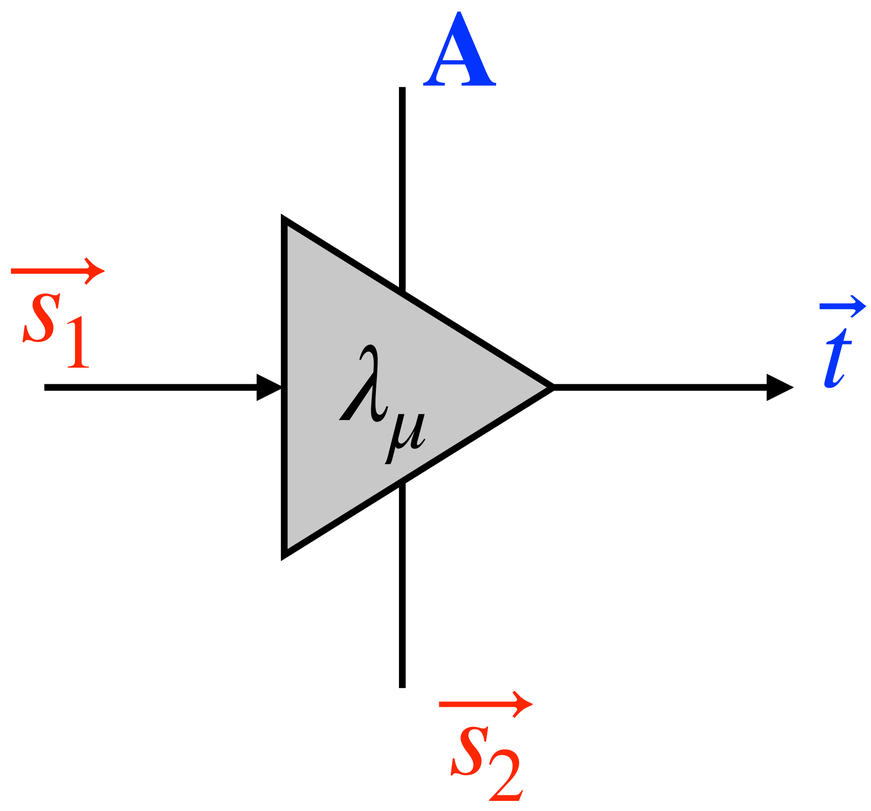
	n	q	k	η_1	η_2	d_u	d_v	required RBG strength (bits)
ML-KEM-512	256	3329	2	3	2	10	4	128
ML-KEM-768	256	3329	3	2	2	10	4	192
ML-KEM-1024	256	3329	4	2	2	11	5	256

Table 3. Sizes (in bytes) of keys and ciphertexts of ML-KEM

	encapsulation key	decapsulation key	ciphertext	shared secret key
ML-KEM-512	800	1632	768	32
ML-KEM-768	1184	2400	1088	32
ML-KEM-1024	1568	3168	1568	32

n ~ polynomial degree, k ~ module vector dimension

Module-LWE/SIS Signature Scheme
setup phase



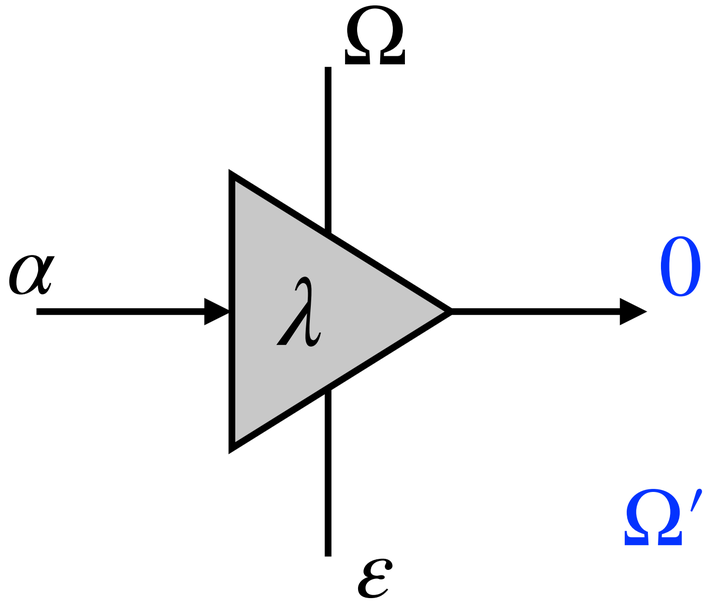
sk: $\vec{s_1} \leftarrow [\beta_1]^l, \vec{s_2} \leftarrow [\beta_1]^k$

pk: $\mathbf{A} \leftarrow R_q^{k \times l}, \; R_q = \mathbb{Z}_q[x] / \langle x^n + 1 \rangle$

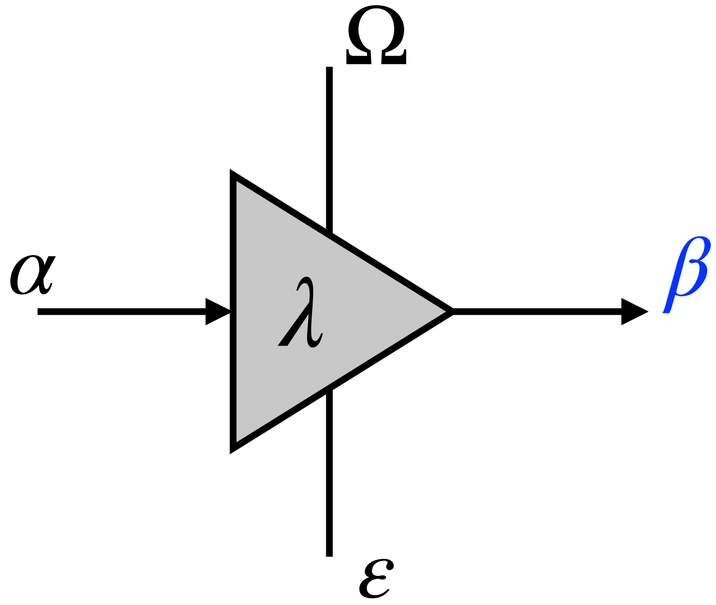
pk: $\vec{t} = \mathbf{A}\vec{s_1} + \vec{s_2}$

the noise vector $\vec{s_2}$ is a part of the secret private key; it governs Aborts in Fiat-Shamir later on

LWE Gate from the SIS Viewpoint



$$\Omega' = \left[\Omega \parallel \mathbf{I}\right]$$



$$\Omega' \times \begin{bmatrix} \alpha \\ \varepsilon \end{bmatrix} = 0$$

homogeneous case

$$\Omega' \times \begin{bmatrix} \alpha \\ \varepsilon \end{bmatrix} = \beta \neq 0$$

inhomogeneous case

The homogeneous and inhomogeneous problems are essentially equivalent for typical params.
[Peikert, https://ia.cr/2015/939]

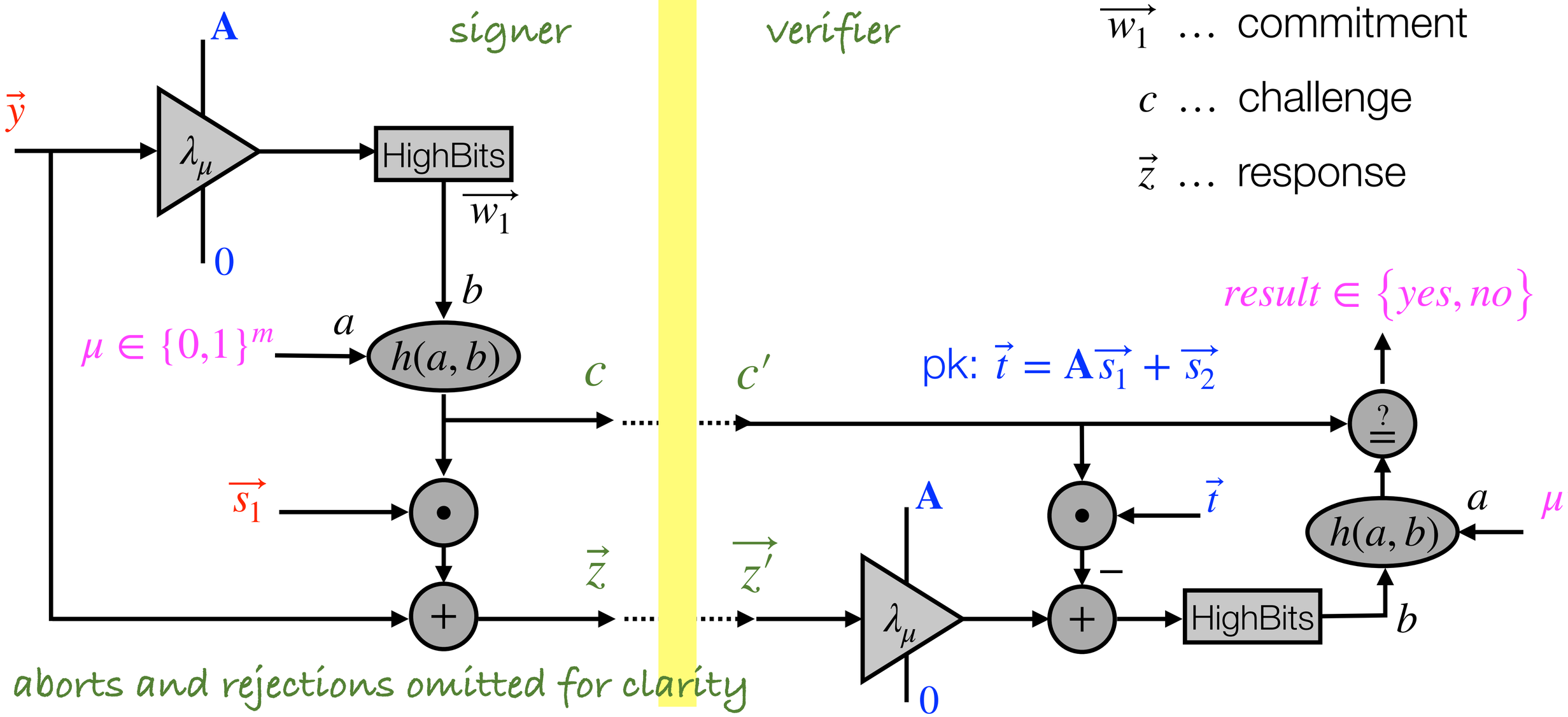
LWE or SIS - Heuristic Arguments

- Are we searching for **the particular solution** that we know it exists and that was used to setup the problem by opponent? *The noisy vector is primarily just an obstacle.*
  - we view the solution as a short **coordinate vector** for a lattice
  - we apply **Bounded-Distance-Decoding** to find the solution

- Or, are we searching for **“something like this” instead**, without any a priori hint anything like this was used to setup the problem by opponent? *The noisy vector is a natural part of the solution.*
  - we view the solution as a certain short **lattice vector directly**
  - we apply a sort of a **Short-Vector-Problem** to find the solution

Up to a scaling factor, the lattices mentioned for LWE and SIS are duals of each other.
[Peikert, https://ia.cr/2015/939]

Module-LWE/SIS Schnorr-Fiat-Shamir Signature Scheme
signature generation/verification



FIPS 204

Federal Information Processing Standards Publication

Module-Lattice-Based Digital Signature Standard

Category: Computer Security **Subcategory: Cryptography**

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
https://doi.org/10.6028/NIST.FIPS.204

Published August 13, 2024

- Fiat-Shamir with Aborts extension
- Rejection sampling to minimize private key leakage - transcript attack
- Number Theoretic Transform for faster ring operations
- Key and signature data length optimizations
- Precise definition of the three parametric ML-DSA schemes based on M-LWE and M-SIS
  - ***M****odule* ***L****attice refers to lattices corresponding to certain R-modules*

[https://doi.org/10.6028/NIST.FIPS.204]

Table 2. Sizes (in bytes) of keys and signatures of ML-DSA

	Private Key	Public Key	Signature Size
ML-DSA-44	2560	1312	2420
ML-DSA-65	4032	1952	3309
ML-DSA-87	4896	2592	4627

ML-DSA parameter sets

(see Sections 6.1 and 6.2 of this document)	Values assigned by each parameter set ML-DSA-44	ML-DSA-65	ML-DSA-87
q - modulus [see §6.1]	8380417	8380417	8380417
ζ - a 512th root of unity in $\mathbb{Z}_q$ [see §7.5]	1753	1753	1753
d - # of dropped bits from $\mathbf{t}$ [see §6.1]	13	13	13
τ - # of ± 1's in polynomial c [see §6.2]	39	49	60
λ - collision strength of $\tilde{c}$ [see §6.2]	128	192	256
γ_1 - coefficient range of $\mathbf{y}$ [see §6.2]	2^{17}	2^{19}	2^{19}
γ_2 - low-order rounding range [see §6.2]	$(q-1)/88$	$(q-1)/32$	$(q-1)/32$
(k, ℓ) - dimensions of $\mathbf{A}$ [see §6.1]	(4,4)	(6,5)	(8,7)
η - private key range [see §6.1]	2	4	2
$\beta = \tau \cdot \eta$ [see §6.2]	78	196	120
ω - max # of 1's in the hint $\mathbf{h}$ [see §6.2]	80	55	75
Challenge entropy $\log_2 \binom{256}{\tau} + \tau$ [see §6.2]	192	225	257
Repetitions (see explanation below)	4.25	5.1	3.85
Claimed security strength	Category 2	Category 3	Category 5

FALCON

- **Fa**st Fourier **L**attice-based **CO**mpact signatures over **N**TRU
- Its goal is to minimize $\left|PublicKey\right| + \left|Signature\right|$
  - FALCON-512: 897 + 666 bytes = 1563 bytes (secret key: 1281 bytes)
  - FALCON-1024: 1793 + 1280 bytes = 3073 bytes (secret key: 2305 bytes)
- Chosen for standardization in NIST PQC standardization process
  - FIPS 206: FN-DSA ~ FFT over NTRU-lattice based Digital Signature Algorithm
  - ETA: 2024H2 (well... perhaps later)

FALCON Signature Scheme in Nutshell

- Based on hash-and-sign Gentry, Peikert, and Vaikuntanathan (GPV) framework
- Employs:
  - NTRU lattices with compact polynomial-generated bases
  - Fast Fourier (trapdoor) sampling
- In spirit, we can see similar optimizations as with module-lattice schemes

GPV Key Setup

- Framework for lattice-based signatures
- Public key: full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $n < m$, generating lattice Λ
- Private key: short matrix $\mathbf{B} \in \mathbb{Z}_q^{m \times m}$, generating orthogonal q-ary lattice $\Lambda_q^{\perp}$
  - for any $\vec{x} \in \Lambda$ and $\vec{y} \in \Lambda_q^{\perp}$, we have $\left\langle \vec{x}, \vec{y} \right\rangle \equiv 0 \pmod{q}$
  - in particular and equivalently $\mathbf{B} \times \mathbf{A}^T \equiv \mathbf{0} \pmod{q}$

GPV Signature Definition, Verification, and Generation

- For a message μ, the signature is $\vec{s} \in \mathbb{Z}_q^m$, such that $\vec{s}\mathbf{A}^T = H(\mu)$ and $\vec{s}$ is short
- Signature verification: straightforward
- Signature generation:
  - find arbitrary $\overrightarrow{s_0}$, such that $\overrightarrow{s_0}\mathbf{A}^T = H(\mu)$
  - solve (approximate) Closest Vector Problem with respect to $\mathbf{B}$ to find $\vec{v} \in \Lambda_q^{\perp}$ close to $\overrightarrow{s_0}$
  - the signature is $\vec{s} = \overrightarrow{s_0} - \vec{v}$; note that this way $\|\overrightarrow{s_0} - \vec{v}\|$ shall be small
  - proof: $\vec{s}\mathbf{A}^T = \overrightarrow{s_0}\mathbf{A}^T - \vec{v}\mathbf{A}^T = \overrightarrow{s_0}\mathbf{A}^T - \vec{0} = \overrightarrow{s_0}\mathbf{A}^T = H(\mu)$, while $\vec{s}$ is short (cf. above)

Vulnerabilities we went through before and probably will go again

- Implementation faults, for instance:
  - faulty encryption/decryption
  - faulty signature generation/verification
- Computational faults
  - such as were RSA-CRT vulnerabilities
- Side channels
  - sensitive data leakage

NTT - Number Theoretic Transform

- Specialized discrete Fourier transform to speed up multiplication in certain rings of convolution polynomials
- Can be also interpreted as a sort of Chinese Remainder Theorem machinery
- Is a vital core of LWE based algorithms ML-KEM and ML-DSA
- Is a fruitful target of **fault and side channel attacks**

$$R_q := \mathbb{Z}_q[X]/(X^{256}+1) \qquad T_q := \bigoplus_{i=0}^{127} \mathbb{Z}_q[X]/\left(X^2 - \zeta^{2\mathsf{BitRev}_7(i)+1}\right)$$

$$\hat{f} := \left(f \bmod (X^2 - \zeta^{2\mathsf{BitRev}_7(0)+1}), \ldots, f \bmod (X^2 - \zeta^{2\mathsf{BitRev}_7(127)+1})\right)$$

$$f \times_{R_q} g = \mathsf{NTT}^{-1}(\hat{f} \times_{T_q} \hat{g})$$

— NIST FIPS 203: ML-KEM, August 13th, 2024

Floating Point FFT in FALCON (FN-DSA)

- Automatic offloading of sensitive computation to a Floating Point Unit (FPU) naturally invokes side-channels that are uneasy to predict and prevent

4.1 Floating-Point

Signature generation, and also part of key pair generation, involve the use of complex numbers. These can be approximated with standard IEEE 754 floating-point numbers ("binary64" format, commonly known as "double precision"). Each such number is encoded over 64 bits, that split into the following elements:

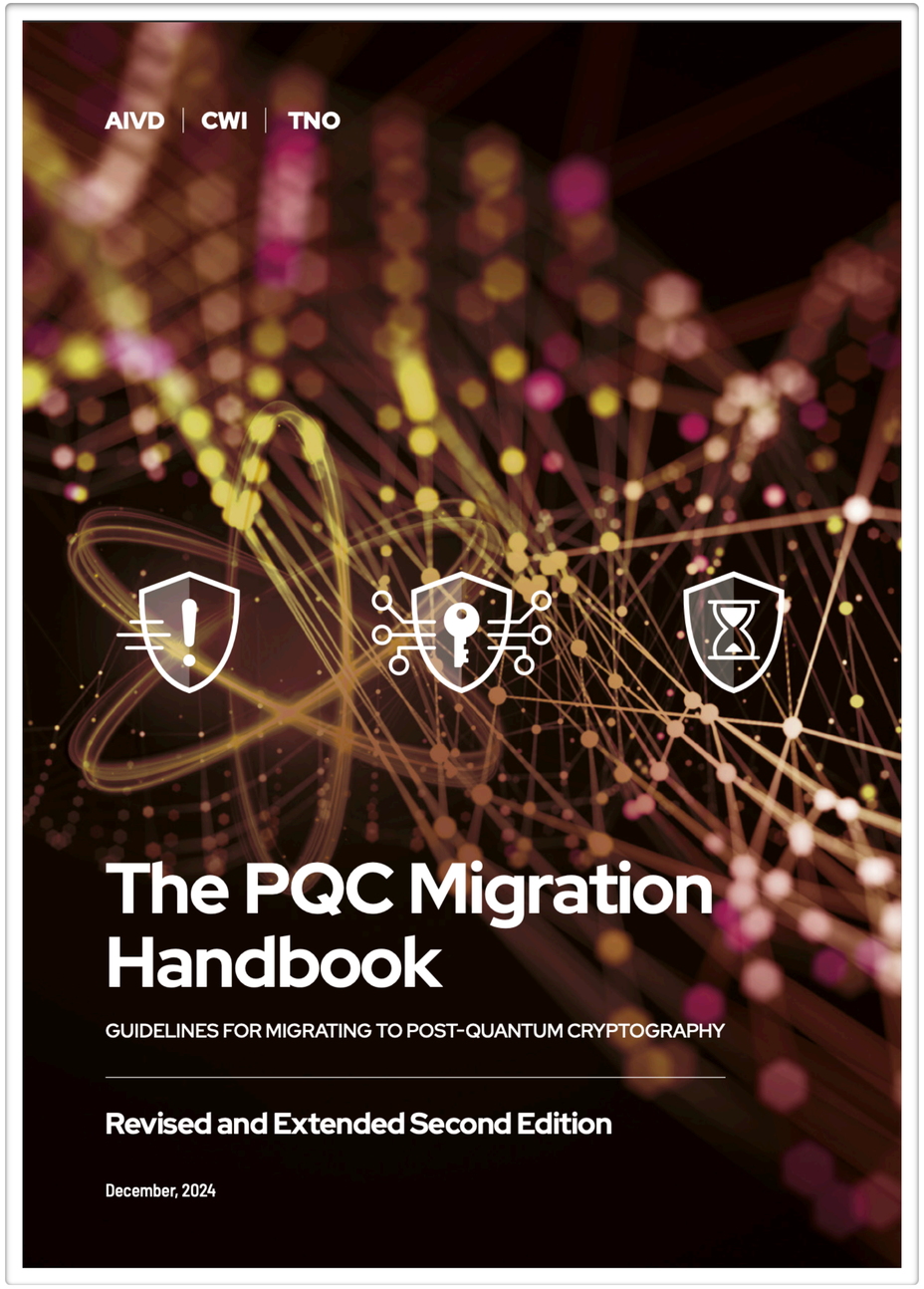
- a sign $s = \pm 1$ (1 bit);
- an exponent e in the -1022 to $+1023$ range (11 bits);
- a mantissa m such that $1 \leq m < 2$ (52 bits).

In general, the represented value is $sm2^e$. The mantissa is encoded as $2^{52}(m - 1)$; it has 53 bits of precision, but its top bit, of value 1 by definition, is omitted in the encoding.

[Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU, v1.2]

~~FALCON~~ FAILCON?

Well, that was the easy part, now comes...
the real IT



-- https://publications.tno.nl/publication/34643386/fXcPVHsX/TNO-2024-pqc-en.pdf

Tasty Low-Hanging Fruits

```
C:\Users\rflab>echo Q | openssl s_client -connect online.rb.cz:443 -servername online.rb.cz -brief -tls1_3
Connecting to 185.250.72.130
depth=2 C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
verify error:num=20:unable to get local issuer certificate
CONNECTION ESTABLISHED
Protocol version: TLSv1.3
Ciphersuite: TLS_AES_256_GCM_SHA384
Peer certificate: C=CZ, L=Praha, O=Raiffeisenbank a.s., CN=online.rb.cz
Hash used: SHA256
Signature type: rsa_pss_rsae_sha256
Verification error: unable to get local issuer certificate
Negotiated TLS1.3 group: X25519MLKEM768
DONE

C:\Users\rflab>
```

Thank you for your attention

Co-funded by the European Union
Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Cybersecurity Competence Centre. Neither the European Union nor the European Cybersecurity Competence Centre can be held responsible for them

Supported by ECCC
The project funded under Grant Agreement No. 101158662 is supported by the European Cybersecurity Competence Centre

History (year-month-day format)

- 2025-05-26, version 1