# CRYSTALS:

## (Cryptographic Suite for Algebraic Lattices)

# Dilithium


Leo Ducas    Eike Kiltz    Tancrede Lepoint    Vadim Lyubashevsky

Peter Schwabe    Gregor Seiler    Damien Stehle

CWI Centrum Wiskunde & Informatica

Ruhr Universität Bochum RUB

Google

Radboud University

IBM

ENS DE LYON


www.pq-crystals.org/dilithium

# Dilithium

Lattice-based digital signature

Based on Generalized (a.k.a Module)-LWE / SIS problems

For all security levels, only need two main operations:

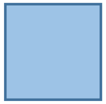
1. SHAKE (or any other XOF)
2. Operations in the polynomial ring

$$R = Z_p[X]/(X^{256}+1) \text{ for prime } p = 2^{23} - 2^{13} + 1$$

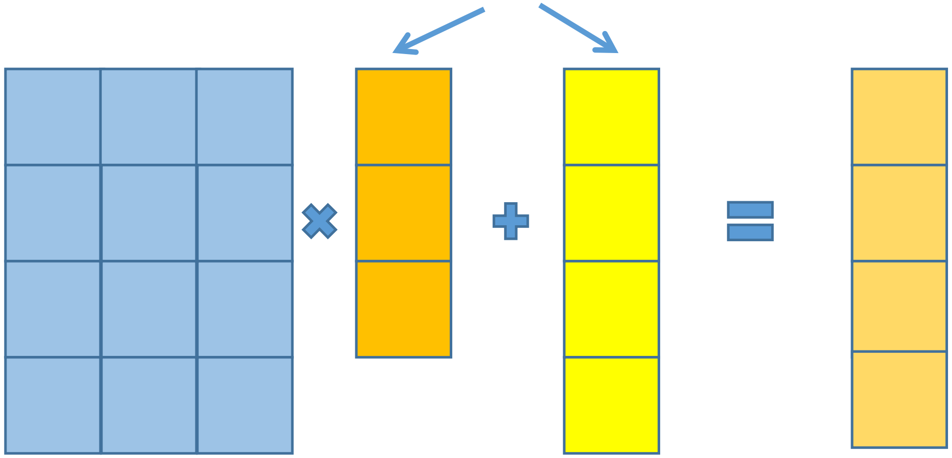
# Dilithium Operations

Basic Computational Domain:

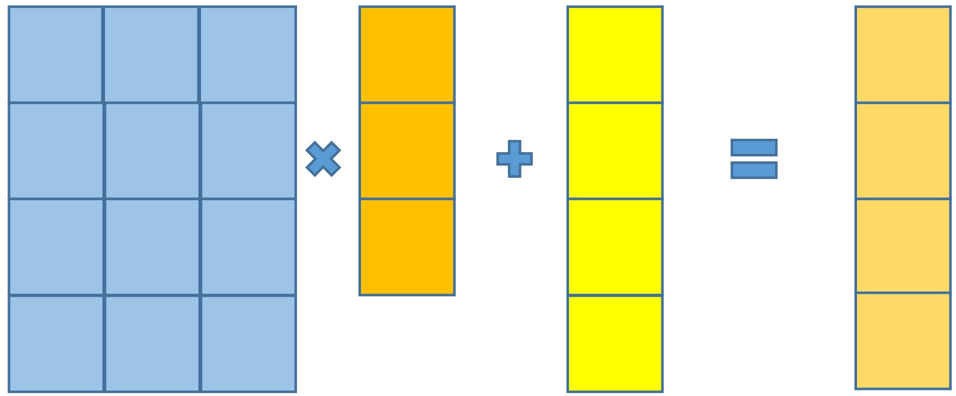
Polynomial ring $Z_p[x]/(x^{256}+1)$



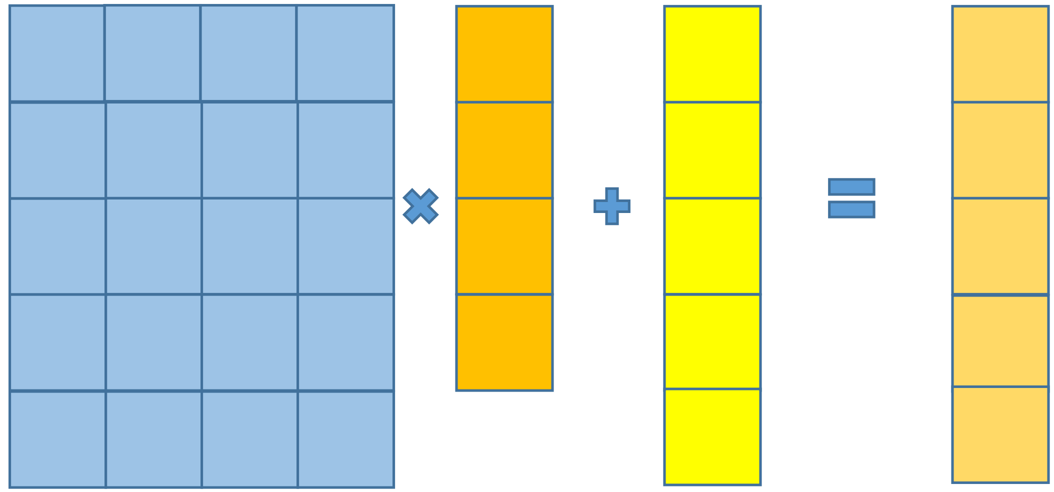
small coefficients



# Modular Security



to increase the security margin, do more of the same operation



# Dilithium Features

- Very simple to implement – all sampling is uniform

- It's fast (for all operations) and has the 2nd-smallest pk+sig size (after FALCON)

- Uses NTT for multiplication – very fast and can be done in place to reduce stack size

- Lattices over $Z_p[X]/(X^n+1)$ used in concrete schemes since SWIFFT [LMPR '08]. Algebraic lattices since NTRU [HPS '96].
  - The algorithmic framework for cryptanalysis is stable since [S '87] and [AKS '01]. These techniques are being "squeezed out" right now.
  - Some parameter increase due to conservative considerations of "sieving" attacks requiring exponential space

# Parameters and Runtime

| **Quantum Security:** | **90** | **128** | **160** |
|---|---|---|---|
| pk size (bytes) | 1184 | 1472 | 1760 |
| sig size (bytes) | 2044 | 2701 | 3366 |
| key gen. cycles | 110K | 156K | 221K |
| verify cycles | 110K | 155K | 220K |
| sign cycles (median) | 315K | 440K | 465K |
| sign cycles with 64B sk | 345K | 475K | 496K |

* on an Intel Core-i7 6600U (Skylake) CPU using SHAKE as the XOF

Changes from round 1 submission:

- No changes in the design or parameter settings
- Included randomized signing mode in addition to deterministic
- Optimizations of the code (and fixed 1 implementation bug in Dec. 2017)

# Dilithium

=

LWE / SIS - Fiat-Shamir [L ‘09] + [L ‘12]

+

Signature Size Reduction [BG ‘14]

+

Public Key Reduction [DKL+ ‘18]

# Dilithium Algorithms

**KeyGen()**

$\mathbf{A} \leftarrow R^{5 \times 4}$; $\mathbf{s_1} \leftarrow [-5, 5]^4$ , $\mathbf{s_2} \leftarrow [-5, 5]^5$

$\mathbf{A}\mathbf{s_1}+\mathbf{s_2} = \mathbf{t} = \text{low}(\mathbf{t})+\text{high}(\mathbf{t})$

**SK:** $(\mathbf{s_1}, \mathbf{s_2})$, **PK:** $(\mathbf{A} \leftarrow R^{5 \times 4}$ , $\text{high}(\mathbf{t}))$

**Sign(μ)**

$\mathbf{y} \leftarrow [-\gamma, \gamma]^4$

$c := H(\text{high}(\mathbf{Ay}), \mu)$

$\mathbf{z} := \mathbf{y} + c\mathbf{s_1}$

Restart if $|\mathbf{z}| > \gamma - \beta$ or $|\text{low}(\mathbf{Ay} - c\mathbf{s_2})| > \gamma - \beta$

Create a small carry bit hint vector **h**

**Signature =** $(\mathbf{z}, c, \mathbf{h})$

Makes the distribution of **z** independent of $\mathbf{s_i}$

Carry bits caused by ignoring $c \cdot \text{low}(\mathbf{t})$

**Verify**$(\mathbf{z}, c, \mathbf{h}, \mu)$

Use **h** and $\mathbf{Az} - c \cdot \text{high}(\mathbf{t})$ to reconstruct $\text{high}(\mathbf{Az} - c\mathbf{t})$

Verify: $|\mathbf{z}| \leq \gamma - \beta$ and $c = H(\underbrace{\text{high}(\mathbf{Az} - c\mathbf{t})}_{= \text{high}(\mathbf{Ay})}, \mu)$

# Security Proof Reduction in the QROM

Tight reduction from:

1. LWE
2. ST-SIS: given random $\mathbf{A}$,$\mathbf{t}$, find
   $\mu$, short $c \neq 0$, $\mathbf{z}_i$ satisfying $H(\mathbf{A}\mathbf{z}_1+\mathbf{z}_2 - c\mathbf{t}, \mu)=c$

In the ROM, ST-SIS = SIS: (with the usual Schnorr-type security loss)

given random $\mathbf{A}$, $\mathbf{t}$, find short $c \neq 0$, $\mathbf{z}_i$ satisfying $\mathbf{A}\mathbf{z}_1+\mathbf{z}_2 - c\mathbf{t} = \mathbf{0}$

# Dilithium Security

1. In the QROM, *tightly* based on LWE and STSIS [Unr ’17, KLS ’18]
   - For a ring R with a bigger p, ST-SIS is vacuously hard, so the scheme is based on just LWE in the QROM. Dilithium-Q [KLS ‘18]

2. In the ROM, based on LWE and SIS [L ‘09, L ’12]

3. In the QROM, based on the *special-sound* and *collapsing* properties of the underlying interactive protocol [DFMS ‘19].
   - Special soundness based on SIS [L ‘12, DKL+ ‘18]
   - It is conjectured in [DFMS ‘19] that the Dilithium protocol is collapsing

4. In the QROM, the collapsing property is (non-tightly) based on LWE. [LZ ‘19]

# Comparison to qTESLA

same "style" as Dilithium (i.e. uses [L '09]+[L '12]+[BG '14] as a starting point)

but … qTESLA had an incorrect security argument that bypassed the requirement for SIS to be hard

| | qTESLA Round2 128-bit | qTESLA Round2 128-bit | qTESLA Round2 160-bit | qTESLA Round1 128-bit | Dilithium 128-bit |
|---|---|---|---|---|---|
| pk size (bytes) | 800 | 2336 | 38432 | 2976 | 1472 |
| sig size (bytes) | 2432 | 2144 | 5664 | 2720 | 2701 |
| | completely broken [LS '19] (attack is faster than real signing) | relies on a version of SIS with much less security than Dilithium | security claims like Dilithium-Q [KLS '18] which is based on only LWE in the QROM<br><br>parameters for 160-bit Dilithium-Q:<br>pk: 9632<br>sig: 7098 | proof of a stronger claim was wrong, but may have the same security as Dilithium<br><br>instantiation of [BG '14] – no public key reduction | |

Can be made somewhat fast using ideas from e.g. [B '19]. Guess: ≈ 10X slower than Dilithium

# Dilithium and FALCON

If the goals are:

- Compactness
- Very easy implementation on all devices

Use Fiat-Shamir signatures with uniform sampling: **Dilithium**

If the goal is:

- Maximum Compactness

Use hash-and-sign signatures over NTRU lattices with Gaussian sampling: **FALCON**

| | Dilithium (90-bit) | FALCON (100-bit) | Dilithium (128-bit) | Dilithium (160-bit) | FALCON (256-bit) |
|---|---|---|---|---|---|
| pk size (bytes) | 1184 | 897 | 1472 | 1760 | 1793 |
| sig size (bytes) | 2044 | 652 | 2701 | 3366 | 1261 |

# Dilithium and FALCON

**Dilithium**

+ + Fast Verification
+ + Fast Signing
+ + Simple to implement everywhere – particularly important for low-power devices where generic signatures (e.g. SPHINCS) are too slow [KRSS ‘19]
+ Compact

**FALCON**

+ + Fast Verification
+ + Fast Signing (if Floating Point Unit is Present)
+ + Very compact
- Very delicate signing procedure – messing up the floating point precision can lead to leaking the secret key
- Emulating the FPU using integer arithmetic can lead to significant slow-downs

? How easy is it to mask?

## Both schemes serve a purpose

Techniques lead to practical ZK-based privacy primitives

Techniques lead to a practical IBE

# CRYSTALS:

(Cryptographic Suite for Algebraic Lattices)

# Dilithium

# Thank You

Centrum Wiskunde & Informatica

RUHR UNIVERSITÄT BOCHUM

Radboud University

ENS DE LYON

www.pq-crystals.org/dilithium

# Bibliography


- [AKS ‘01] Miklós Ajtai, Ravi Kumar, D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. STOC 2001
- [B ’19] Dan Bernstein. Comment on the PQC forum Jan. 19, 2019. https://groups.google.com/a/list.nist.gov/forum/#!topic/pqc-forum/VK9dROwgY0Y
- [BG ‘14] Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. CT-RSA ‘14
- [DFMS ‘19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. CRYPTO 2019
- [DKL+ ‘18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-Dilithium: A lattice-based digital signature scheme. CHES 2018
- [HPS ‘98] Jeffrey Hoffstein, Jill Phipher, Joseph Silverman. NTRU: A Ring-Based Public Key Cryptosystem. ANTS 1998
- [KLS ‘18] Eike Kiltz, Vadim Lyubashevsky, Christian Schaffner. A Concrete Treatment of Fiat-Shamir Signatures in the Quantum Random-Oracle Model. EUROCRYPT 2018
- [KRSS ‘19] Matthias J. Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4. Workshop Record of the Second PQC Standardization Conference
- [LMPR ’08] Vadim Lyubashevsky, Daniele Micciancio, Alon Rosen, Chris Peikert. SWIFFT: A Modest proposal for FFT hashing. FSE ‘08
- [LS ’19] Vadim Lyubashevsky and Peter Schwabe. Official qTESLA comment on the NIST mailing list. April 14, 2019.
- [LZ ‘19] Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. CRYPTO 2019
- [L ‘09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. ASIACRYPT 2009.
- [L ‘12] Vadim Lyubashevsky. Lattice signatures without trapdoors. EUROCRYPT 2012
- [S ’87] Claus-Peter Schnorr. A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms. Theor. Comp. Sci. 1987
- [Unr ’17] Dominique Unruh. Post-quantum security of Fiat-Shamir. ASIACRYPT 2017