CRYSTALS

(**Cry**ptographic **S**ui**t**e for **A**lgebraic **L**attice**s**)

CCA KEM: Kyber

Digital Signature: Dilithium

Roberto Avanzi – ARM
Joppe Bos – NXP
Leo Ducas – CWI
Eike Kiltz – RUB
Tancrede Lepoint – SRI
Vadim Lyubashevsky – IBM Research
John Schanck – U. Waterloo
Peter Schwabe – U. Radboud
Gregor Seiler – IBM Research
Damien Stehle – ENS Lyon

www.pq-crystals.org

Lattice Cryptography

Easy Problem

$$\begin{pmatrix} & & \\ & \mathbf{A} & \\ & & \end{pmatrix} \begin{bmatrix} \\ \mathbf{y} \\ \\ \end{bmatrix} = \begin{bmatrix} \mathbf{z} \\ \\ \\ \end{bmatrix} \bmod p$$

Given (**A**,**z**), find **y**

Easy! Just invert **A** and multiply by **z**

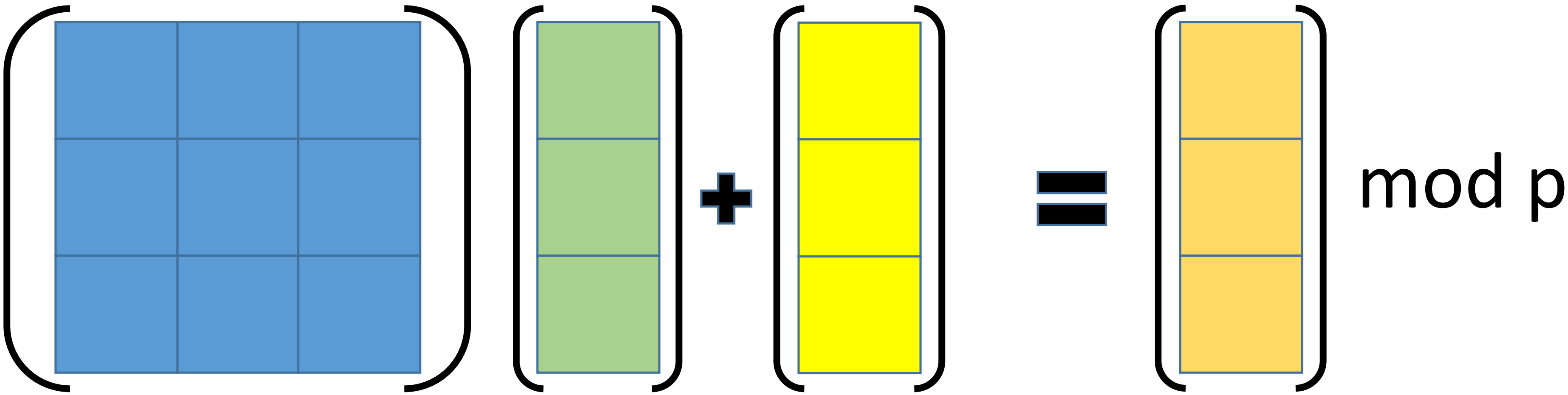
Hard Problem

$$\begin{pmatrix} \\ \mathbf{A} \\ \\ \end{pmatrix} \underbrace{\begin{bmatrix} \mathbf{y} \\ \\ \end{bmatrix} + \begin{bmatrix} \mathbf{e} \\ \\ \end{bmatrix}}_{\text{Small coefficients to enforce uniqueness}} = \begin{bmatrix} \mathbf{z} \\ \\ \end{bmatrix} \bmod p$$

Given (**A**,**z**), find (**y**,**e**)

Seems hard (would have many positive non-cryptographic applications if it were easy).

Hard Problem



Given (**A**,**z**), find (**y**,**e**)

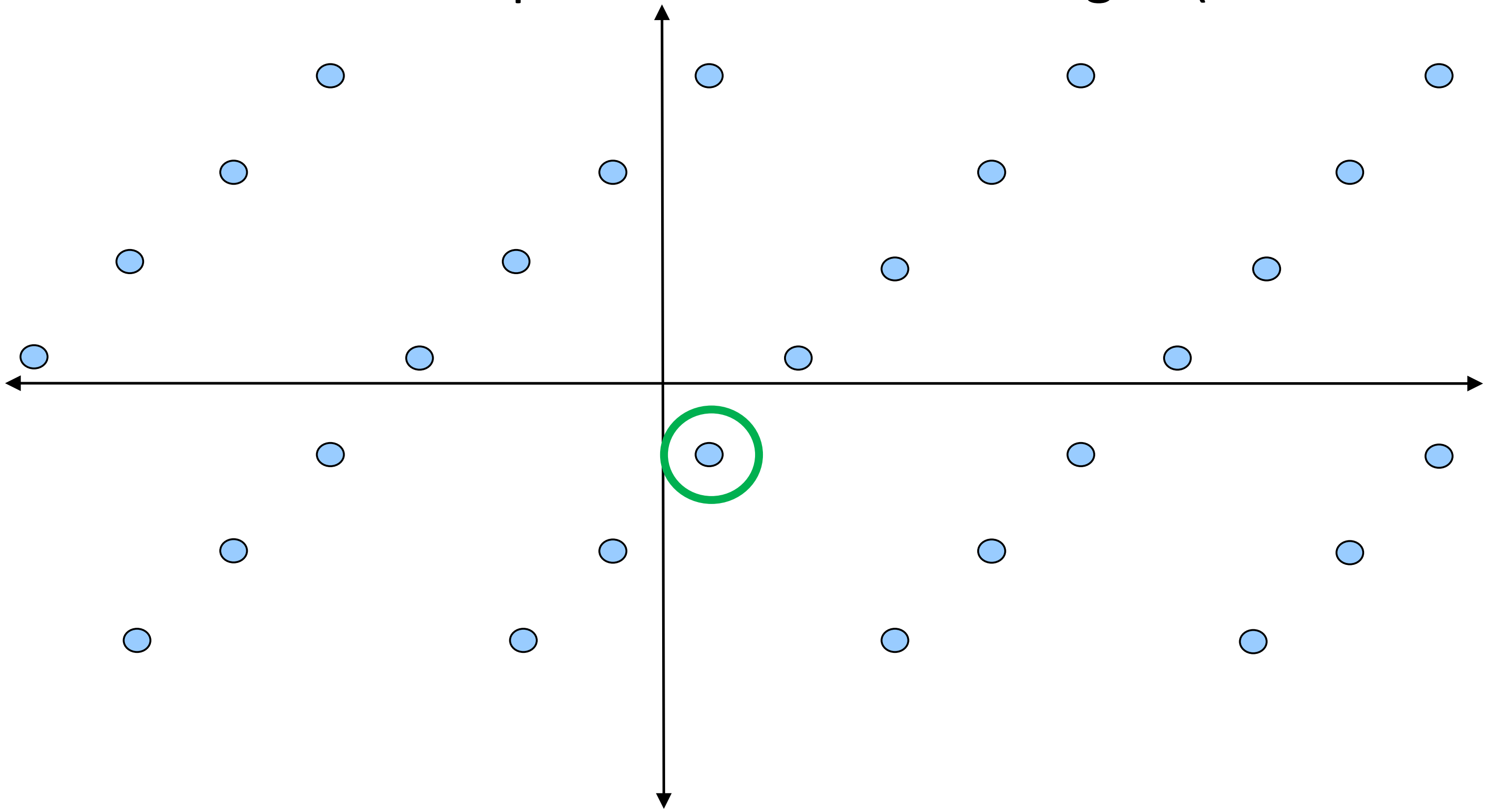
Seems hard.

Even when **A** is over $Z_p[X]/(f(X))$ for certain f(X).

Why is this "Lattice" Crypto?

All solutions $\binom{y}{e}$ to **A**y+e=**z** mod p form a "shifted" lattice.

We want to find the point closest to the origin (BDD Problem).



Brief History

- Lattices studied algorithmically at least since 1982 (LLL)
- Algebraic lattices since at least 1996 (NTRU)
- Lattices over $Z_p[X]/(X^n+1)$ since at least 2008 (SWIFFT)
- Last 10 years – one of the hottest area in cryptography. Lots of attention and some interesting algorithms discovered
  - But … **0** attacks against lattice crypto based on (Module-) SIS / LWE
  - Parameters were increased (around 50%) due to conservative considerations of “sieving” attacks requiring exponential space

CRYSTALS Math

Operations

Only two main operations needed (and both are very fast):

1. Evaluations of SHAKE256 (can use another XOF too)
2. Operations in the polynomial ring $R = Z_p[X]/(X^{256}+1)$

   prime $p = 2^{13} - 2^9 + 1$ (for Kyber)

   prime $p = 2^{23} - 2^{13} + 1$ (for Dilithium)

Very easy to adjust security because the same hardware/software can be reused

Ring Choice Rationale

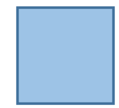
- 256-dimensional rings are “just right”
  - Large enough to efficiently encrypt 256-bit keys
  - Allow for a large enough challenge space for signatures
  - Allow for enough “granularity” to get the security we want

- $\mathbf{Z}_p[X]/(X^n+1)$, for a prime p, has been the most widely-used ring in the literature
  - Very easy to use and the most efficient one
  - Has a lot of properties that are useful in more advanced constructions

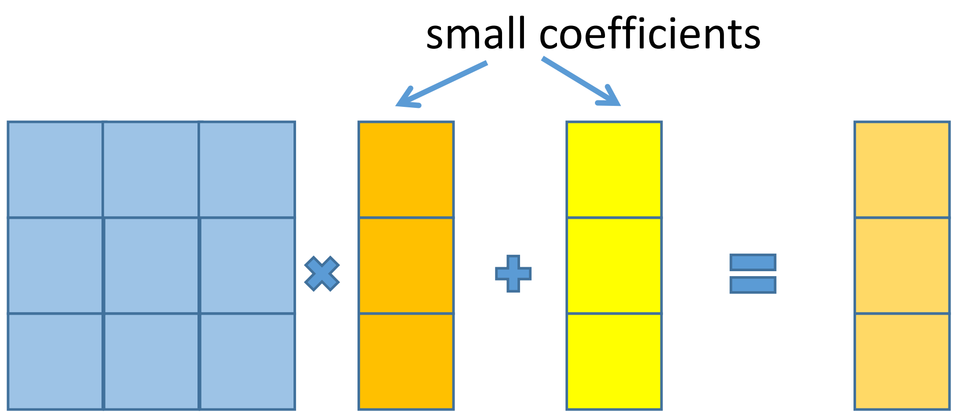
Operations

Basic Computational Domain:

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

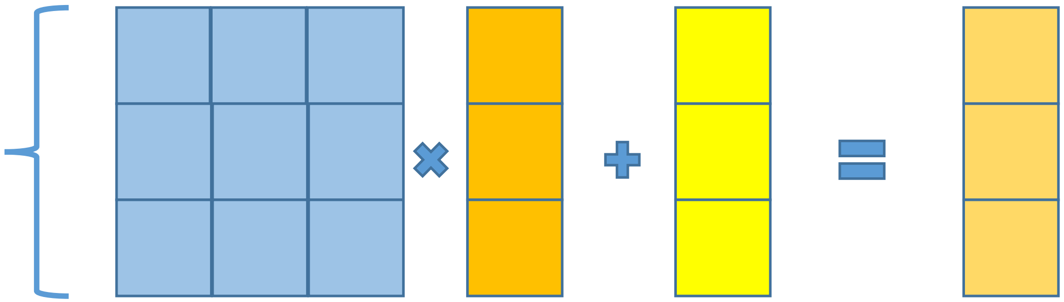


Operations used in the schemes:

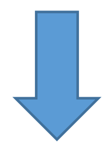


Modular Security

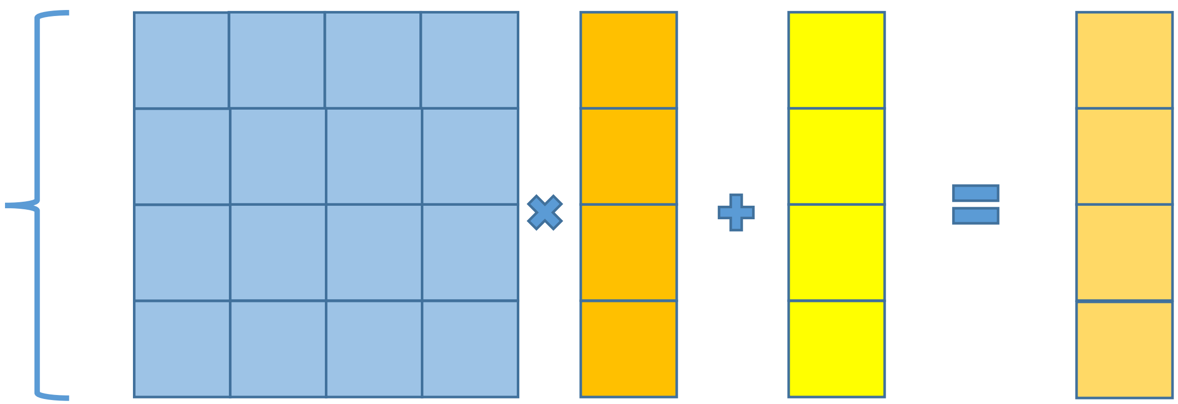
768-dim



to increase the security margin



1024-dim

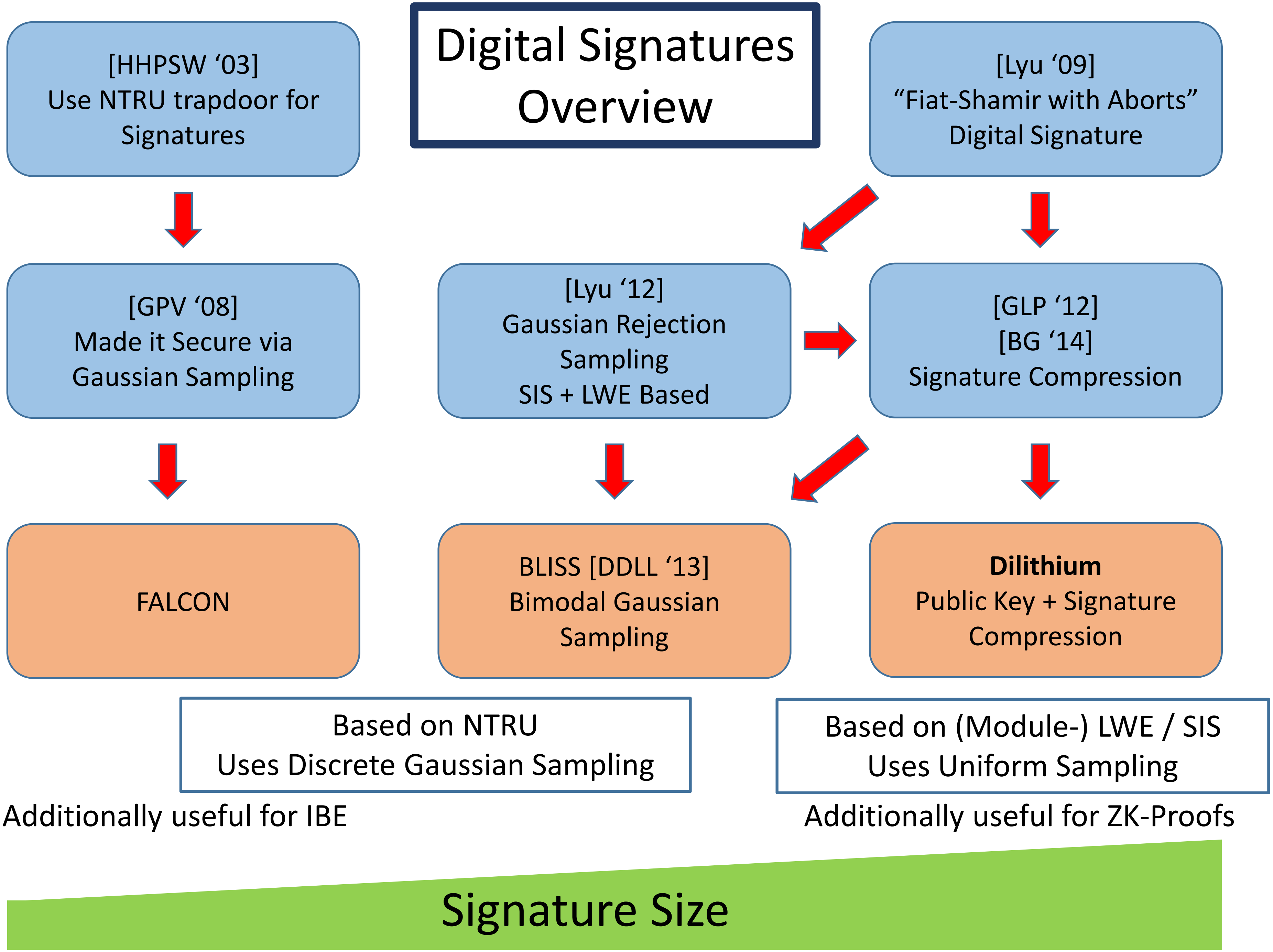


Just do more of the same operation

CRYSTALS-Dilithium

Leo Ducas – CWI
Eike Kiltz – RUB
Tancrede Lepoint – SRI

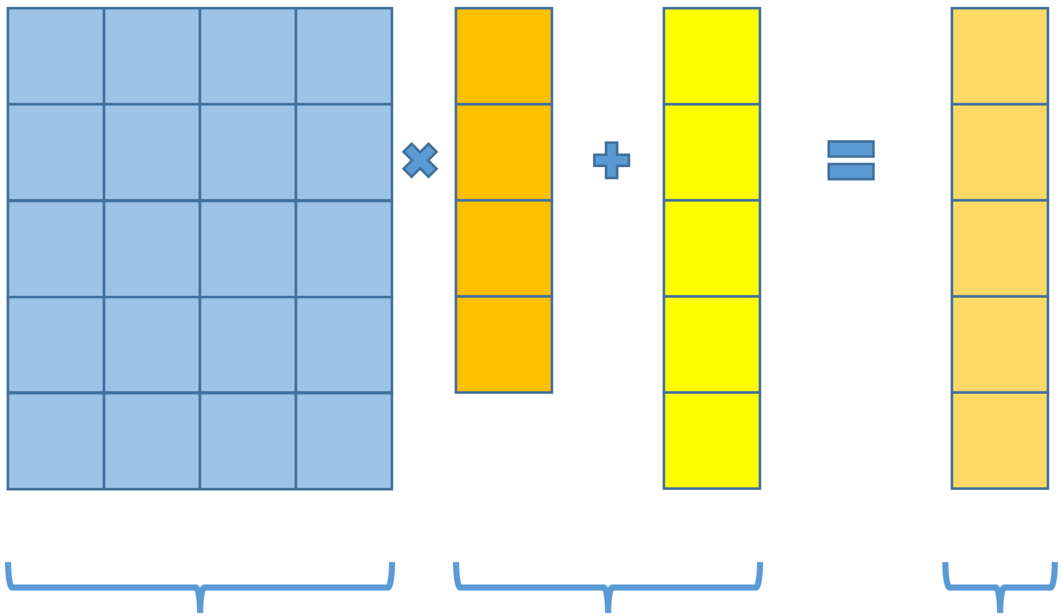
Vadim Lyubashevsky – IBM Research
Peter Schwabe – U. Radboud
Gregor Seiler – IBM Research
Damien Stehle – ENS Lyon



Digital Signatures Overview
[HHPSW '03]
Use NTRU trapdoor for Signatures
[Lyu '09]
"Fiat-Shamir with Aborts" Digital Signature
[GPV '08]
Made it Secure via Gaussian Sampling
[Lyu '12]
Gaussian Rejection Sampling
SIS + LWE Based
[GLP '12]
[BG '14]
Signature Compression
FALCON
BLISS [DDLL '13]
Bimodal Gaussian Sampling
Dilithium
Public Key + Signature Compression
Based on NTRU
Uses Discrete Gaussian Sampling
Based on (Module-) LWE / SIS
Uses Uniform Sampling
Additionally useful for IBE
Additionally useful for ZK-Proofs
Signature Size

Signatures with Uniform Sampling

[Lyu '09] → … → [BG '14]



Public key
$\mathbf{A}$:=XOF(seed)
uniform mod p

Secret key
$\mathbf{s_1}$, $\mathbf{s_2}$ with
coefficients in {-5,…,5}

Public key
$\mathbf{t} := \mathbf{A}\mathbf{s_1} + \mathbf{s_2}$

Signatures with Uniform Sampling

[Lyu '09] → … → [BG '14]

$$\mathbf{A}\mathbf{s_1}+\mathbf{s_2}=\mathbf{t}$$

Sign(μ)

$\mathbf{y} \leftarrow$ Coefficients in $[-\gamma, \gamma]$

$c := H(\mathrm{high}(\mathbf{A}\mathbf{y}), \mu)$

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

Needed for security

if $|\mathbf{z}| > \gamma - \beta$ or $|\mathrm{low}(\mathbf{A}\mathbf{y} - c\mathbf{s_2})| > \gamma - \beta$

restart

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

Verify($\mathbf{z}$, c, μ)

Check that $|\mathbf{z}| \le \gamma - \beta$

and

$c = H(\mathrm{high}(\underbrace{\mathbf{A}\mathbf{z} - c\mathbf{t}}_{\mathbf{A}\mathbf{y} - c\mathbf{s_2}}), \mu)$

Signatures with Uniform Sampling

[Lyu '09] → … → [BG '14]

$$\mathbf{A}\mathbf{s}_1+\mathbf{s}_2=\mathbf{t}$$

Sign(μ)

$\mathbf{y} \leftarrow$ Coefficients in $[-\gamma, \gamma]$

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

$\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$

Needed for correctness

If $|\mathbf{z}| > \gamma - \beta$ or $|\text{low}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2)| > \gamma - \beta$

restart

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

Verify($\mathbf{z}$, c, μ)

Check that $|\mathbf{z}| \leq \gamma - \beta$

and

$c=H(\text{high}(\underbrace{\mathbf{A}\mathbf{z} - c\mathbf{t}}_{\mathbf{A}\mathbf{y} - c\mathbf{s}_2}), \mu)$

max($|c\mathbf{s}_2|$)

Because $|\text{low}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2)| \leq \gamma - \beta$, high$(\mathbf{A}\mathbf{y} - c\mathbf{s}_2)$ = high$(\mathbf{A}\mathbf{y})$

Chopping off Low-Order PK bits

$$\mathbf{A}\mathbf{s}_1+\mathbf{s}_2=\mathbf{t}_0+b\mathbf{t}_1$$

Sign(μ)

$\mathbf{y} \leftarrow$ Coefficients in $[-\gamma, \gamma]$

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

$\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$

If $|\mathbf{z}| > \gamma - \beta$ or $|\text{low}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2)| > \gamma - \beta$

restart

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

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

Check that $|\mathbf{z}| \leq \gamma - \beta$

and

$$c = H(\text{high}(\underbrace{\mathbf{A}\mathbf{z} - cb\mathbf{t}_1}_{\mathbf{A}\mathbf{y} - c\mathbf{s}_2 + c\mathbf{t}_0}), \mu)$$

$$\boxed{\text{Want high}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2) = \text{high}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2 + c\mathbf{t}_0)}$$

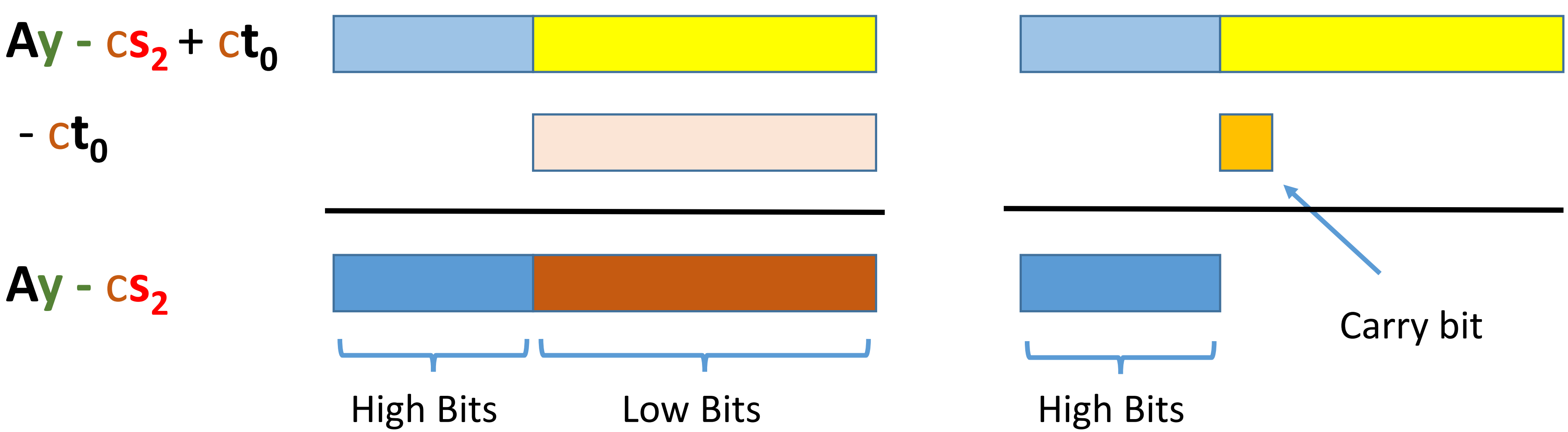
The Carry Hint Vector

Want $\mathrm{high}(\mathbf{A}y - c\mathbf{s}_2) = \mathrm{high}(\mathbf{A}y - c\mathbf{s}_2 + c\mathbf{t}_0)$

The signer knows $\mathbf{A}y - c\mathbf{s}_2 + c\mathbf{t}_0$ and $c\mathbf{t}_0$

The verifier knows $\mathbf{A}y - c\mathbf{s}_2 + c\mathbf{t}_0$

Each 23-bit coefficient



Dilithium
(high-level overview)

$$\mathbf{A}\mathbf{s}_1+\mathbf{s}_2=\mathbf{t}_0+b\mathbf{t}_1$$

Sign(μ)

$\mathbf{y}$ ← Coefficients in [-γ, γ]

c := H(high($\mathbf{A}\mathbf{y}$), μ)

$\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$

If $|\mathbf{z}| > \gamma - \beta$ or $|\mathrm{low}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2)| > \gamma - \beta$

restart

Create carry bit hint vector h

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

Verify($\mathbf{z}$, c, μ)

Check that $|\mathbf{z}| \le \gamma - \beta$

and

$$c = H(\underbrace{\mathrm{high}(h \text{ “+” } \mathbf{A}\mathbf{z} - cb\mathbf{t}_1)}_{\mathrm{high}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2)}, \mu)$$

Hint h
- adds 100 – 200 bytes to the signature
- Saves ≈ 2KB in the public key

Parameters for CRYSTALS-Dilithium

(> 128-bit quantum security)

	5 x 4 matrices	6 x 5 matrices
Public Key	≈ 1.5 KB	≈ 1.8 KB
Signature	≈ 2.7 KB	≈ 3.4 KB

Public key generation / verification: > 10,000 per second

Signing : > 3,000 per second

Security Reductions

Signature Scheme Security

(Proof framework for Fiat-Shamir Schemes in the ROM)

Math Problem $\leq$ Hybrid 1 $\leq$ Real Signature Scheme

Math Problem

1. A_3 gets math problem
2. A_3 solves math problem

Non-Interactive and no hash H

Hybrid 1

1. A_2 gets the public key and access to hash H
2. A_2 forges a signature

Non-Interactive

Real Signature Scheme

1. A_1 gets the public key and access to hash H
2. A_1 asks signature queries
3. A_1 forges a signature

Interactive

- Reduction in the QROM [Unr '17]
- Tight reduction in the QROM if the signing is deterministic [KLS '18]

Dilithium Security

Non-tight in the ROM

Input: random $\mathbf{A}$, $\mathbf{t}$
Output: short $\mathbf{s}_1$, $\mathbf{s}_2$ and c such that
$$\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 - \mathbf{t}c = 0$$

(Module)-SIS + (Module)-LWE

$\leq$

Tight in the QROM

Input: random $\mathbf{A}$, $\mathbf{t}$, and an XOF H
Output: short $\mathbf{s}_1$, $\mathbf{s}_2$, c and μ such that
$$H(\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 - \mathbf{t}c, \mu) = c$$

Hybrid 1 (Self-Target SIS)

Non-tight reduction in the ROM using rewinding

The Same as Schnorr Signatures

Non-tight in the ROM		Tight in the QROM
Input: random $\mathbf{A}$, $\mathbf{t}$ Output: short $\mathbf{s}_1$, $\mathbf{s}_2$ and c such that $\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 - \mathbf{t}c = 0$	$\leq$	Input: random $\mathbf{A}$, $\mathbf{t}$, and an XOF H Output: short $\mathbf{s}_1$, $\mathbf{s}_2$, c and μ such that $H(\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 - \mathbf{t}c, \mu) = c$
(Module)-SIS + (Module)-LWE		Hybrid 1 (Self-Target SIS)

Input: random g,h Output: x, c such that $g^x h^c = 1$	$\leq$	Input: random g, h, and an XOF H Output: x, c and μ such that $H(g^x h^c, \mu) = c$
Discrete Log		"Self-Target Discrete Log"

Is the “Self-Target” Assumption Worrisome in the QROM?

- We believe not

- No example where using “rewinding” in the proof left the scheme vulnerable to a quantum attacker

- Analogous to computationally-binding classical commitments not having a proof in the QROM
  (and there is no NIST competition for post-quantum commitments)

Base on (Module-)LWE in the QROM?

Dilithium

	recommended	high
Public Key	≈ 1.5 KB	≈ 1.8 KB
Signature	≈ 2.7 KB	≈ 3.4 KB

“Katz-Wang” Tight Dilithium [AFLT ‘12, ABB+ ‘15, Unr ‘17, KLS ‘18]

		recommended	high
Public Key	**5X larger**	≈ 7.7 KB	≈ 9.6 KB
Signature	**2X larger**	≈ 5.7 KB	≈ 7.1 KB

Also significantly (> **10X**) slower

Basis for Our Parameter Settings

LWE parameters (i.e. secret key recovery)

used the recently en vogue sieving analysis

SIS parameters (i.e. message forgery) –

the same analysis + improved (potential) algorithm for ℓ_∞ - SIS

Possible Trade-Offs
(open to community suggestions)

- Smaller secret key coefficients e.g. {-5,…,5} → {-1,0,1}
  - Signatures will be smaller
  - Makes combinatorial hybrid attacks more likely and gets further away from WC-AC parameters

- Module-LWE → Module-LWR
  - (Maybe) a slight reduction in the key size
  - Probably nothing goes wrong with security if sk coefficients are large enough

- Increase the rejection probability
  - Slower, but smaller signatures