Falcon

Thomas Prest

Thales Communications & Security

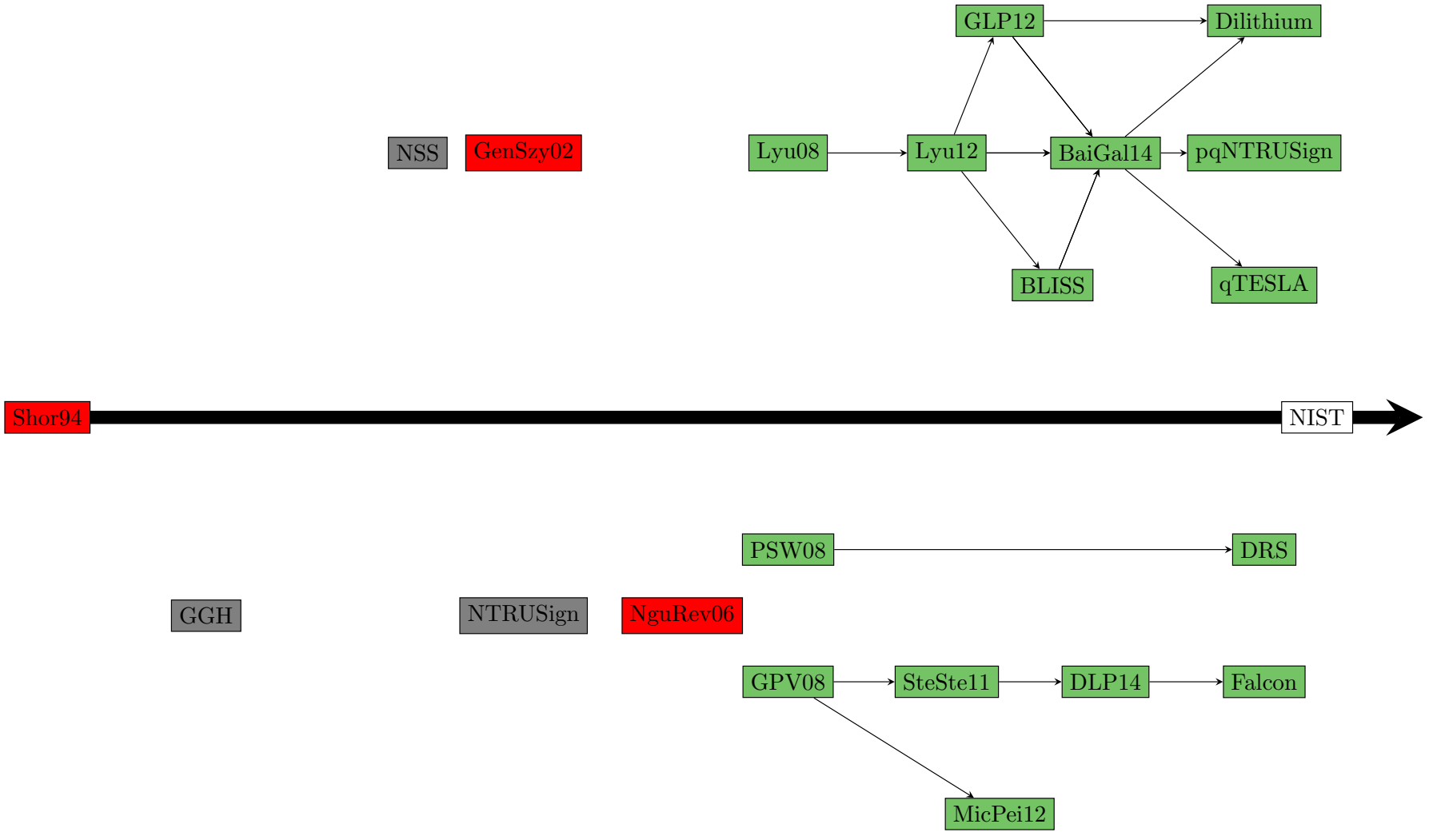
Lattice-based cryptography

Lattice-based cryptography in a nutshell [dPL17]:

> Every lattice-based cryptographic construction relies on the fact that when given a matrix $\mathbf{A}$ and a vector $\mathbf{y}$ over some ring R (such as $\mathbb{Z}_q$ or $\mathbb{Z}_q[X]/(X^d+1)$ with the usual addition and multiplication operations), it is hard to recover a vector $\mathbf{x}$ with small coefficients such that
>
> $$\mathbf{Ax} = \mathbf{y}.$$

Nice! Building signature schemes based on this principle should be easy, right?

A (non-exhaustive) timeline of lattice-based signature schemes



The early hash-and-sign schemes

Early proposals: GGH [GGH97] and NTRUSign [HHGP+03].

Alice	Bob
SK: $\mathbf{B} \in \mathbb{Z}_q^{m \times m}$ short s.t. $\mathbf{BA}^t = \mathbf{0}$	PK: $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$
Sign Compute $\mathbf{c}$ s.t. $\mathbf{cA} = H(\mathsf{m})$ $\mathbf{v} \leftarrow \lfloor \mathbf{cB}^{-1} \rceil \mathbf{B}$ $\mathbf{s} \leftarrow \mathbf{c} - \mathbf{v}$	
$\xrightarrow{\mathbf{s}}$	**Verify** Accept iff - $H(\mathsf{m}) = \mathbf{sA}$ - $\|\mathbf{s}\|$ is short

The early hash-and-sign schemes

Early proposals: GGH [GGH97] and NTRUSign [HHGP+03].

Alice	Bob
SK: $\mathbf{B} \in \mathbb{Z}_q^{m \times m}$ short s.t. $\mathbf{BA}^t = \mathbf{0}$	PK: $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

Sign

Compute $\mathbf{c}$ s.t. $\mathbf{cA} = H(\mathsf{m})$

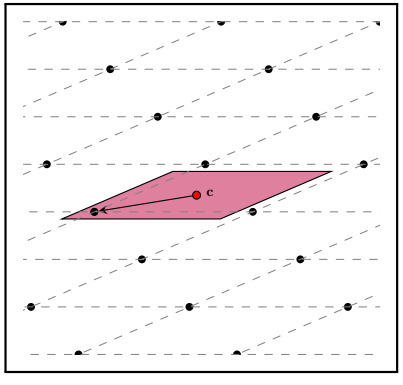
$\boxed{\mathbf{v} \leftarrow \lfloor \mathbf{cB}^{-1} \rceil \mathbf{B}}$

$\mathbf{s} \leftarrow \mathbf{c} - \mathbf{v}$

$\mathbf{s} \longrightarrow$

Verify

Accept iff
- $H(\mathsf{m}) = \mathbf{sA}$
- $\|\mathbf{s}\|$ is short



Outputting $\mathbf{v} \leftarrow \lfloor \mathbf{cB}^{-1} \rceil \mathbf{B}$ leaks the private key! [NR06]

Step I - Provably secure hash-and-sign over lattices

Theoretical framework formalized in [GPV08].

Alice	Bob
SK: $\mathbf{B} \in \mathbb{Z}_q^{m \times m}$ short s.t. $\mathbf{BA}^t = \mathbf{0}$	PK: $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$
Sign Compute $\mathbf{c}$ s.t. $\mathbf{cA} = H(\mathsf{m})$ Sample $\mathbf{v} \in \mathbb{Z}^m \mathbf{B}$ close to $\mathbf{c}$ $\mathbf{s} \leftarrow \mathbf{c} - \mathbf{v}$	
$\xrightarrow{\mathbf{s}}$	**Verify** Accept iff - $H(\mathsf{m}) = \mathbf{sA}$ - $\|\mathbf{s}\|$ is short

ⓘ $\mathbf{v}$ is securely sampled using a *trapdoor sampler*.

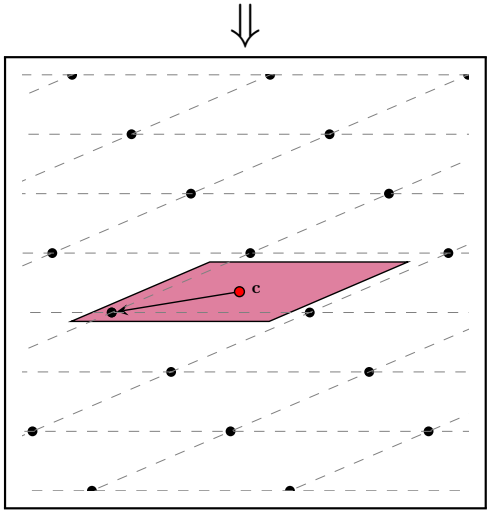
Trapdoor samplers: compute $\mathbf{z} \in \mathbb{Z}^n$ such that $\|(\mathbf{z} - \mathbf{t})\mathbf{B}\|$ is small

Approach 1

Algorithm 1 Round-off

Require: $\mathbf{B}$

1: **for** $j = n, \ldots, 1$ **do**
2: $\quad z_j \leftarrow \lfloor t_j \rceil$
3: **return z**

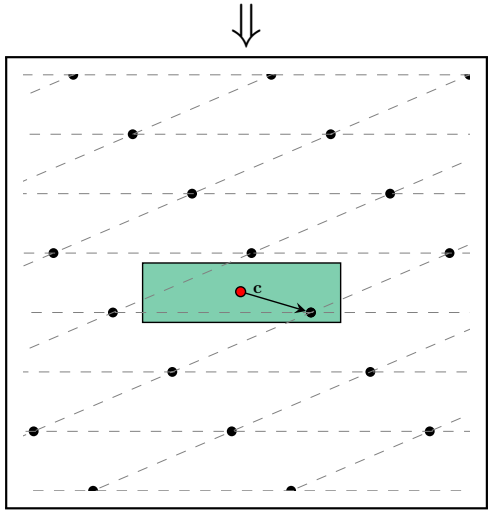


Approach 2

Algorithm 2 Nearest plane

Require: $\mathbf{B} = \mathbf{L} \cdot \tilde{\mathbf{B}}$

1: **for** $j = n, \ldots, 1$ **do**
2: $\quad \bar{t}_j \leftarrow t_j + \sum_{i>j}(t_i - z_i)L_{ij}$
3: $\quad z_j \leftarrow \lfloor \bar{t}_j \rceil$
4: **return z**



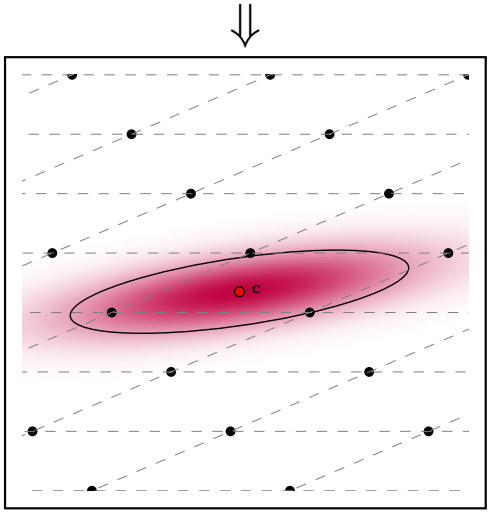
Trapdoor samplers: compute $\mathbf{z} \in \mathbb{Z}^n$ such that $\|(\mathbf{z} - \mathbf{t})\mathbf{B}\|$ is small

Approach 1

Algorithm 3 Rand. round-off

Require: $\mathbf{B}$

1: **for** $j = n, \ldots, 1$ **do**
2: $\quad z_j \leftarrow \lfloor t_j \rceil_\sigma$
3: **return z**

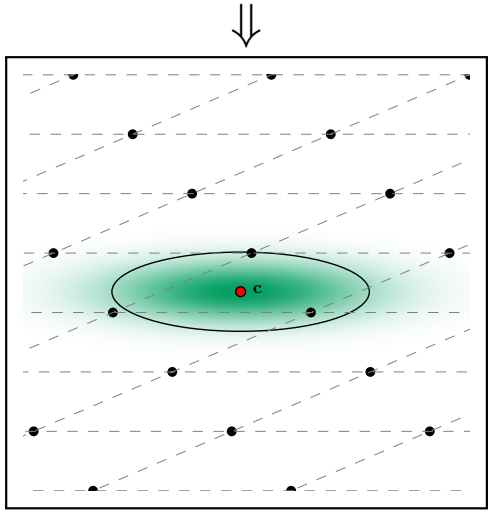


Approach 2

Algorithm 4 Rand. nearest plane

Require: $\mathbf{B} = \mathbf{L} \cdot \tilde{\mathbf{B}}$

1: **for** $j = n, \ldots, 1$ **do**
2: $\quad \bar{t}_j \leftarrow t_j + \sum_{i>j}(t_i - z_i)L_{ij}$
3: $\quad z_j \leftarrow \lfloor \bar{t}_j \rceil_\sigma$
4: **return z**



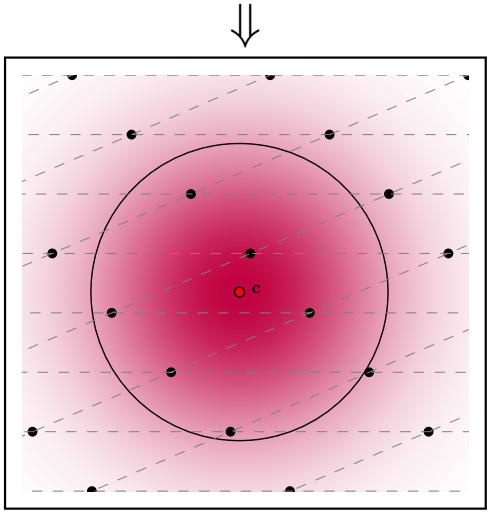
Trapdoor samplers: compute $\mathbf{z} \in \mathbb{Z}^n$ such that $\|(\mathbf{z} - \mathbf{t})\mathbf{B}\|$ is small

Approach 1

Algorithm 5 Peikert's sampler

Require: $\mathbf{B}$

1: $\mathbf{x} \leftarrow \mathbf{C} \cdot \lfloor \mathbf{0} \rceil_\sigma$
2: **for** $j = n, \ldots, 1$ **do**
3: $\quad z_j \leftarrow \lfloor t_j - x_j \rceil_\sigma$
4: **return** $\mathbf{z}$

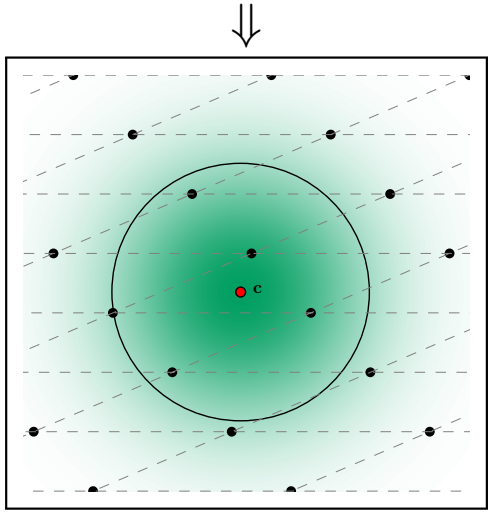


Approach 2

Algorithm 6 Klein's sampler

Require: $\mathbf{B} = \mathbf{L} \cdot \tilde{\mathbf{B}}$

1: **for** $j = n, \ldots, 1$ **do**
2: $\quad \bar{t}_j \leftarrow t_j + \sum_{i>j} (t_i - z_i) L_{ij}$
3: $\quad z_j \leftarrow \lfloor \bar{t}_j \rceil_{\sigma / \|\tilde{\mathbf{b}}_j\|}$
4: **return** $\mathbf{z}$



Step II - GPV framework + NTRU lattices

Instantiation of the GPV framework over NTRU lattices [SS11, DLP14]

Simply take $\mathbf{A} = \left[\begin{array}{c|c} 1 & h \end{array}\right]$ and $\mathbf{B} = \left[\begin{array}{c|c} g & -f \\ \hline G & -F \end{array}\right]$, where:

$$\begin{array}{ccc} fG - gF & = & q \mod (x^n + 1) \\ h & = & gf^{-1} \mod (q, x^n + 1) \end{array} \quad (1)$$

Shortcomings:

1. Cumbersome key generation (slow, requires a lot of memory)
2. Signature generation is either:
   - with Klein's sampler, secure but slow: $O(n^2)$
   - with Peikert's sampler, less secure but fast: $O(n \log n)$
3. Use of floating-point arithmetic (FPA) $\Rightarrow$ which precision?
4. Parameters may be improved (?)

The rest of this talk: addressing these shortcomings.
The techniques also apply to the IBE of [DLP14].

I - The issue of the key generation

Key generation at a high-level:

1. generate small $f, g \in \mathbb{Z}[x]/(x^n + 1)$
2. solve the NTRU equation, i.e. find $F, G \in \mathbb{Z}[x]/(x^n + 1)$ such that

$$fG - gF = 1 \bmod (x^n + 1) \tag{2}$$

3. do simple stuff

Existing methods for step 2 were very cumbersome in time (∼1 second), memory (∼3 Mbytes) and implementation efforts (depends on who implements it).
Can we do better?

I - Exploiting the tower of rings structure

We have the following tower of rings:

$$\mathbb{Z} \subseteq \mathbb{Z}[x]/(x^2+1) \subseteq \cdots \subseteq \mathbb{Z}[x]/(x^{n/2}+1) \subseteq \mathbb{Z}[x]/(x^n+1)$$

and the field norm allows to "navigate" along this tower!

Let $\mathcal{Q}_n = \mathbb{Q}[x]/(x^n+1)$. The field norm N is defined by:

$$\begin{array}{rccc} \mathsf{N} & : & \mathcal{Q}_n & \to & \mathcal{Q}_{n/2} \\ & & f & \to & f f^\times \end{array} \tag{3}$$

where $f^\times$ denotes the Galois conjugate of f for the field extension $\mathcal{Q}_n/\mathcal{Q}_{n/2}$...
Or more simply in our case, $f^\times(x) = f(-x)$.

Fun fact: if we have this relationship over $\mathbb{Z}[x]/(x^{n/2}+1)$:

$$\mathsf{N}(f)G' - \mathsf{N}(g)F' = 1 \tag{4}$$

for some F', G', then we have this relationship over $\mathbb{Z}[x]/(x^n+1)$:

$$f(f^\times G') - g(g^\times F') = 1 \tag{5}$$

I - Outline of the new key generation algorithm

$$\begin{array}{ccc} \mathbb{Z}[x]/(x^n+1) & \ni & f, g \\ \cup\!\!\!\!\!\!\backslash \\ \mathbb{Z}[x]/(x^{n/2}+1) \\ \cup\!\!\!\!\!\!\backslash \\ \mathbb{Z}[x]/(x^{n/4}+1) \\ \cup\!\!\!\!\!\!\backslash \\ \vdots \\ \cup\!\!\!\!\!\!\backslash \\ \mathbb{Z} \end{array} \tag{6}$$

I - Outline of the new key generation algorithm

$$
\begin{array}{ccc}
\mathbb{Z}[x]/(x^n+1) & \ni & f, g \\
\cup\!\!\!\!\!\!\backslash & & \downarrow \\
\mathbb{Z}[x]/(x^{n/2}+1) & \ni & \mathsf{N}(f), \mathsf{N}(g) \\
\cup\!\!\!\!\!\!\backslash & & \\
\mathbb{Z}[x]/(x^{n/4}+1) & & \\
\cup\!\!\!\!\!\!\backslash & & \\
\vdots & & \\
\cup\!\!\!\!\!\!\backslash & & \\
\mathbb{Z} & &
\end{array}
\tag{6}
$$

I - Outline of the new key generation algorithm

$$
\begin{array}{ccc}
\mathbb{Z}[x]/(x^n+1) & \ni & f, g \\
\cup\!\!\!\!\!\!\backslash & & \downarrow \\
\mathbb{Z}[x]/(x^{n/2}+1) & \ni & \mathsf{N}(f), \mathsf{N}(g) \\
\cup\!\!\!\!\!\!\backslash & & \downarrow \\
\mathbb{Z}[x]/(x^{n/4}+1) & \ni & \mathsf{N}^2(f), \mathsf{N}^2(g) \\
\cup\!\!\!\!\!\!\backslash & & \\
\vdots & & \\
\cup\!\!\!\!\!\!\backslash & & \\
\mathbb{Z} & &
\end{array}
\tag{6}
$$

I - Outline of the new key generation algorithm

$$
\begin{array}{ccc}
\mathbb{Z}[x]/(x^n+1) & \ni & f, g \\
\cup\!\!\!\!\!\!\backslash & & \downarrow \\
\mathbb{Z}[x]/(x^{n/2}+1) & \ni & \mathsf{N}(f), \mathsf{N}(g) \\
\cup\!\!\!\!\!\!\backslash & & \downarrow \\
\mathbb{Z}[x]/(x^{n/4}+1) & \ni & \mathsf{N}^2(f), \mathsf{N}^2(g) \\
\cup\!\!\!\!\!\!\backslash & & \downarrow \\
\vdots & \vdots & \vdots \\
\cup\!\!\!\!\!\!\backslash & & \\
\mathbb{Z} & &
\end{array}
\tag{6}
$$

I - Outline of the new key generation algorithm

$$
\begin{array}{ccc}
\mathbb{Z}[x]/(x^n+1) & \ni & f, g \\
\cup\!\!\upharpoonleft & & \downarrow \\
\mathbb{Z}[x]/(x^{n/2}+1) & \ni & \mathsf{N}(f), \mathsf{N}(g) \\
\cup\!\!\upharpoonleft & & \downarrow \\
\mathbb{Z}[x]/(x^{n/4}+1) & \ni & \mathsf{N}^2(f), \mathsf{N}^2(g) \\
\cup\!\!\upharpoonleft & & \downarrow \\
\vdots & \vdots & \vdots \\
\cup\!\!\upharpoonleft & & \downarrow \\
\mathbb{Z} & \ni & \mathsf{N}^\ell(f), \mathsf{N}^\ell(g)
\end{array}
\tag{6}
$$

I - Outline of the new key generation algorithm

$$
\begin{array}{ccccc}
\mathbb{Z}[x]/(x^n+1) & \ni & f, g & & \\
\cup\!\!\upharpoonleft & & \downarrow & & \\
\mathbb{Z}[x]/(x^{n/2}+1) & \ni & \mathsf{N}(f), \mathsf{N}(g) & & \\
\cup\!\!\upharpoonleft & & \downarrow & & \\
\mathbb{Z}[x]/(x^{n/4}+1) & \ni & \mathsf{N}^2(f), \mathsf{N}^2(g) & & \\
\cup\!\!\upharpoonleft & & \downarrow & & \\
\vdots & \vdots & \vdots & & \\
\cup\!\!\upharpoonleft & & \downarrow & & \\
\mathbb{Z} & \ni & \mathsf{N}^\ell(f), \mathsf{N}^\ell(g) & \rightarrow & F^{[\ell]}, G^{[\ell]}
\end{array}
\tag{6}
$$

I - Outline of the new key generation algorithm

$$
\begin{array}{ccccc}
\mathbb{Z}[x]/(x^n+1) & \ni & f, g & & \\
\cup\!\!\upharpoonleft & & \downarrow & & \\
\mathbb{Z}[x]/(x^{n/2}+1) & \ni & \mathsf{N}(f), \mathsf{N}(g) & & \\
\cup\!\!\upharpoonleft & & \downarrow & & \\
\mathbb{Z}[x]/(x^{n/4}+1) & \ni & \mathsf{N}^2(f), \mathsf{N}^2(g) & & \\
\cup\!\!\upharpoonleft & & \downarrow & & \\
\vdots & \vdots & \vdots & & \vdots \\
\cup\!\!\upharpoonleft & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathsf{N}^\ell(f), \mathsf{N}^\ell(g) & \rightarrow & F^{[\ell]}, G^{[\ell]}
\end{array}
\tag{6}
$$

I - Outline of the new key generation algorithm

$$
\begin{array}{ccccc}
\mathbb{Z}[x]/(x^n+1) & \ni & f, g & & \\
\cup\!\!\upharpoonleft & & \downarrow & & \\
\mathbb{Z}[x]/(x^{n/2}+1) & \ni & \mathsf{N}(f), \mathsf{N}(g) & & \\
\cup\!\!\upharpoonleft & & \downarrow & & \\
\mathbb{Z}[x]/(x^{n/4}+1) & \ni & \mathsf{N}^2(f), \mathsf{N}^2(g) & \rightarrow & F^{[2]}, G^{[2]} \\
\cup\!\!\upharpoonleft & & \downarrow & & \uparrow \\
\vdots & \vdots & \vdots & & \vdots \\
\cup\!\!\upharpoonleft & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathsf{N}^\ell(f), \mathsf{N}^\ell(g) & \rightarrow & F^{[\ell]}, G^{[\ell]}
\end{array}
\tag{6}
$$

I - Outline of the new key generation algorithm

$$
\begin{array}{ccccc}
\mathbb{Z}[x]/(x^n+1) & \ni & f, g & & \\
\cup\!\!\upharpoonleft & & \downarrow & & \\
\mathbb{Z}[x]/(x^{n/2}+1) & \ni & \mathsf{N}(f), \mathsf{N}(g) & \rightarrow & F^{[1]}, G^{[1]} \\
\cup\!\!\upharpoonleft & & \downarrow & & \uparrow \\
\mathbb{Z}[x]/(x^{n/4}+1) & \ni & \mathsf{N}^2(f), \mathsf{N}^2(g) & \rightarrow & F^{[2]}, G^{[2]} \\
\cup\!\!\upharpoonleft & & \downarrow & & \uparrow \\
\vdots & \vdots & \vdots & & \vdots \\
\cup\!\!\upharpoonleft & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathsf{N}^\ell(f), \mathsf{N}^\ell(g) & \rightarrow & F^{[\ell]}, G^{[\ell]}
\end{array}
\tag{6}
$$

I - Outline of the new key generation algorithm

$$
\begin{array}{ccccc}
\mathbb{Z}[x]/(x^n+1) & \ni & f, g & \to & F, G \\
\cup\!\!\!\downarrow & & \downarrow & & \uparrow \\
\mathbb{Z}[x]/(x^{n/2}+1) & \ni & \mathsf{N}(f), \mathsf{N}(g) & \to & F^{[1]}, G^{[1]} \\
\cup\!\!\!\downarrow & & \downarrow & & \uparrow \\
\mathbb{Z}[x]/(x^{n/4}+1) & \ni & \mathsf{N}^2(f), \mathsf{N}^2(g) & \to & F^{[2]}, G^{[2]} \\
\cup\!\!\!\downarrow & & \downarrow & & \uparrow \\
\vdots & \vdots & \vdots & & \vdots \\
\cup\!\!\!\downarrow & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathsf{N}^\ell(f), \mathsf{N}^\ell(g) & \to & F^{[\ell]}, G^{[\ell]}
\end{array}
\tag{6}
$$

I - Outline of the new key generation algorithm

$$
\begin{array}{ccccc}
\mathbb{Z}[x]/(x^n+1) & \ni & f, g & \to & F, G \\
\cup\!\!\upharpoonright & & \downarrow & & \uparrow \\
\mathbb{Z}[x]/(x^{n/2}+1) & \ni & \mathsf{N}(f), \mathsf{N}(g) & \to & F^{[1]}, G^{[1]} \\
\cup\!\!\upharpoonright & & \downarrow & & \uparrow \\
\mathbb{Z}[x]/(x^{n/4}+1) & \ni & \mathsf{N}^2(f), \mathsf{N}^2(g) & \to & F^{[2]}, G^{[2]} \\
\cup\!\!\upharpoonright & & \downarrow & & \uparrow \\
\vdots & \vdots & \vdots & & \vdots \\
\cup\!\!\upharpoonright & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathsf{N}^\ell(f), \mathsf{N}^\ell(g) & \to & F^{[\ell]}, G^{[\ell]}
\end{array}
\tag{6}
$$

At each lower level:

- The coefficients grow (in bitsize) by a factor 2...
- ... but the number of coefficients is divided by 2.

We gain in practice:

- a factor 100 in memory consumption ($\Rightarrow$ 30KBytes)
- a factor 10 in time

Extends techniques of "overstretched NTRU" [ABD16, KF17], but constructively!

II - Fast Fourier Sampling [DP16]

Klein's sampler interprets $\mathcal{Q}_n = \mathbb{Q}[x]/(x^n + 1)$ as a $\mathbb{Q}$-linear space of dimension n:

$$\mathbf{B} = \left[\begin{array}{c|c} g & -f \\ \hline G & -F \end{array}\right] \in \mathbb{Z}[x]/(x^n+1)^{2\times 2} \qquad \mapsto \qquad \left[\begin{array}{c|c} \mathcal{C}(g) & -\mathcal{C}(f) \\ \hline \mathcal{C}(G) & -\mathcal{C}(F) \end{array}\right] \in \mathbb{Z}^{2n\times 2n}$$

$\Rightarrow$ completely ignores the rich algebraic structure of $\mathcal{Q}_n$!

II - Fast Fourier Sampling [DP16]

Klein's sampler interprets $\mathcal{Q}_n = \mathbb{Q}[x]/(x^n + 1)$ as a $\mathbb{Q}$-linear space of dimension n:

$$\mathbf{B} = \left[\begin{array}{c|c} g & -f \\ \hline G & -F \end{array}\right] \in \mathbb{Z}[x]/(x^n+1)^{2\times 2} \qquad \mapsto \qquad \left[\begin{array}{c|c} \mathcal{C}(g) & -\mathcal{C}(f) \\ \hline \mathcal{C}(G) & -\mathcal{C}(F) \end{array}\right] \in \mathbb{Z}^{2n\times 2n}$$

$\Rightarrow$ completely ignores the rich algebraic structure of $\mathcal{Q}_n$!

Splitting polynomials between their odd and even coefficients yields this chain of space isomorphisms:

$$\mathbb{Q}^n \cong (\mathcal{Q}_2)^{n/2} \cong \ldots \cong (\mathcal{Q}_{n/2})^2 \cong \mathcal{Q}_n \tag{7}$$

We will take advantage of this to devise a *recursive* variant of Klein's sampler.

We reformulate the problem that our signature algorithm solves. Given:

- a challenge $t_0, t_1 \in \mathcal{Q}_n$,
- the secret basis $\mathbf{B} \in \mathbb{Z}[x]/(x^n+1)^{2\times 2}$ (and its GSO),

sample $z_0, z_1 \in \mathbb{Z}[x]/(x^n+1)$ such that $(z_0, z_1) \cdot \mathbf{B}$ is close to $(t_0, t_1) \cdot \mathbf{B}$.

Can we sample z_1 so that $(0, z_1) \cdot \mathbf{B}$ is close to $(0, t_1) \cdot \mathbf{B}$, then adaptively sample z_0?

- OK, just a generalization of Klein's sampler over $\mathcal{Q}_n$ instead of $\mathbb{Q}$.

We reformulate the problem that our signature algorithm solves. Given:

- a challenge $t_0, t_1 \in \mathcal{Q}_n$,
- the secret basis $\mathbf{B} \in \mathbb{Z}[x]/(x^n+1)^{2\times 2}$ (and its GSO),

sample $z_0, z_1 \in \mathbb{Z}[x]/(x^n+1)$ such that $(z_0, z_1)\cdot\mathbf{B}$ is close to $(t_0, t_1)\cdot\mathbf{B}$.

Can we sample z_1 so that $(0, z_1)\cdot\mathbf{B}$ is close to $(0, t_1)\cdot\mathbf{B}$, then adaptively sample z_0?

- OK, just a generalization of Klein's sampler over $\mathcal{Q}_n$ instead of $\mathbb{Q}$.

Problem: sampling z_1 boils down to making $z_1 g$ close to $t_1 g$ for a given $g \in \mathcal{Q}_n$.
How to do that optimally without completely breaking the structure?

- Break $\mathcal{Q}_n$ into $\mathcal{Q}_{n/2}^2$!
- By splitting in odds/even coefficients, z_1, t_1 can be seen as elements of $\mathcal{Q}_{n/2}^2$.
- Similarly, g can be seen as an element of $\mathcal{Q}_{n/2}^{2\times 2}$ (because it actually is an endomorphism).

We reformulate the problem that our signature algorithm solves. Given:

- a challenge $t_0, t_1 \in \mathcal{Q}_n$,
- the secret basis $\mathbf{B} \in \mathbb{Z}[x]/(x^n+1)^{2\times 2}$ (and its GSO),

sample $z_0, z_1 \in \mathbb{Z}[x]/(x^n+1)$ such that $(z_0, z_1)\cdot\mathbf{B}$ is close to $(t_0, t_1)\cdot\mathbf{B}$.

Can we sample z_1 so that $(0, z_1)\cdot\mathbf{B}$ is close to $(0, t_1)\cdot\mathbf{B}$, then adaptively sample z_0?

- OK, just a generalization of Klein's sampler over $\mathcal{Q}_n$ instead of $\mathbb{Q}$.

Problem: sampling z_1 boils down to making $z_1 g$ close to $t_1 g$ for a given $g \in \mathcal{Q}_n$.
How to do that optimally without completely breaking the structure?

- Break $\mathcal{Q}_n$ into $\mathcal{Q}_{n/2}^2$!
- By splitting in odds/even coefficients, z_1, t_1 can be seen as elements of $\mathcal{Q}_{n/2}^2$.
- Similarly, g can be seen as an element of $\mathcal{Q}_{n/2}^{2\times 2}$ (because it actually is an endomorphism).

Situation now identical to the beginning, but over a smaller subfield $\Rightarrow$ recursion!
We can find vectors as close as Klein's sampler would, but in time $O(n \log n)$.

Security proofs involving distributions

- **The standard approach:** using the statistical distance Δ.
  - Take a hard problem relying on some ideal distribution $\mathcal{Q}$,
  - Replace $\mathcal{Q}$ by a "real-life" distribution $\mathcal{P}$,
  - If $\Delta(\mathcal{P}, \mathcal{Q})$ is small enough, we win: the problem is still hard.
- **Lattice-based cryptography:** often relevant to replace SD by *Rényi divergence*.
  - More aggressive parameters [LSS14, LPSS14, BLL+15, BGM+16, Pre17, HLS17]
  - KEMs distributions [ADPS16, BCD+16]
  - Reduction between LWE problems [AD17]

Example. We consider a cryptographic scheme doing q queries to a distribution $\mathcal{D}_i$ ($i \in \{0, 1\}$), we note ε_i the probability of an event.

- With the statistical distance:

$$\varepsilon_0 \geq \varepsilon_1 - q\Delta(\mathcal{D}_1, \mathcal{D}_0) \qquad \boxed{\Delta \leq 2^{-\lambda} \Rightarrow \text{we win}}$$

- With the Rényi divergence:

$$\varepsilon_0 \geq \varepsilon_1^{\frac{a}{a-1}} / R_a(\mathcal{D}_1 \| \mathcal{D}_0)^q \qquad \boxed{\log R_a \leq 1/q \Rightarrow \text{we win}}$$

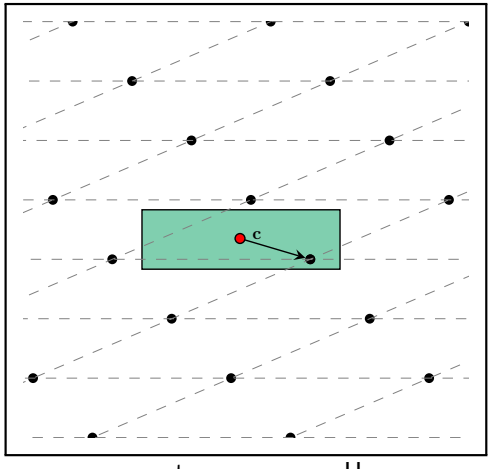
III - Improving the required precision with the Rényi divergence

What is the proper way to evaluate the required precision of the FPA operations?

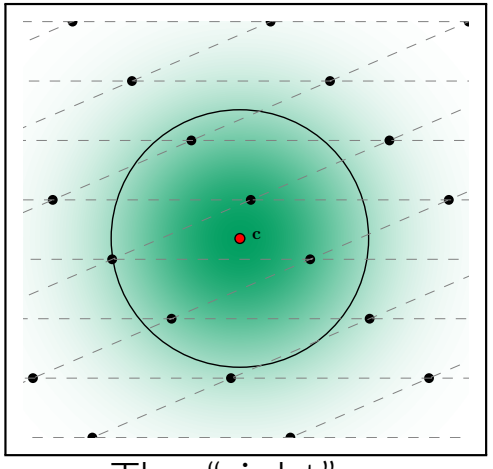
- Statistical distance analysis $\Rightarrow$ FPA operations require a precision of $\lambda + \text{polylog}(n, ...)$ bits.
- Rényi divergence analysis $\Rightarrow$ FPA operations require a precision of $\log_2(q_s)/2 + \text{polylog}(n, ...)$ bits, where q_s is the number of public queries

In NIST's CFP, $\log_2(q_s) \leq 64 \Rightarrow$ taking a precision of 53 bits is (provably!) sufficient.

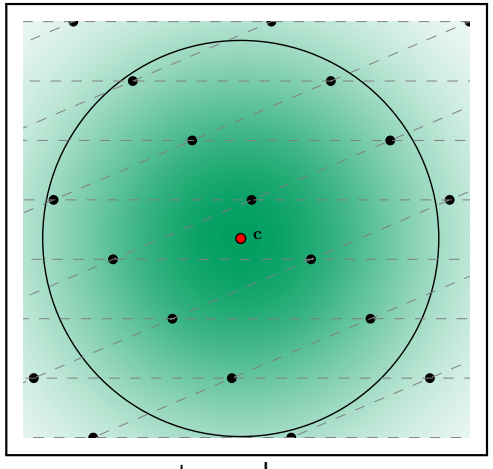
IV - Improving the standard deviation with the Rényi divergence



σ too small



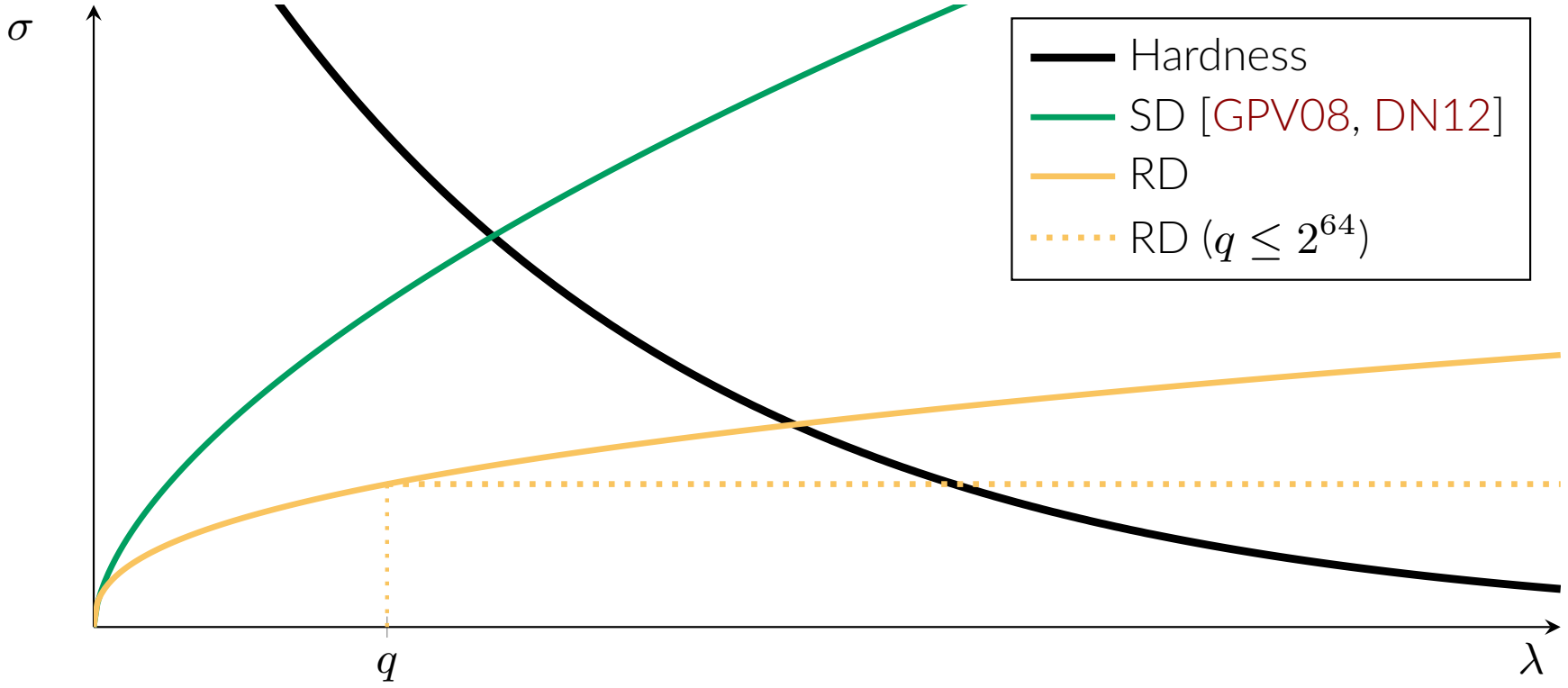
The "right" σ



σ too large

1. σ too large $\Rightarrow$ the trapdoor sampler is useless in a cryptographic context.
2. σ too small $\Rightarrow$ the trapdoor sampler does not behave like a perfect Gaussian.

IV - Improving the standard deviation with the Rényi divergence



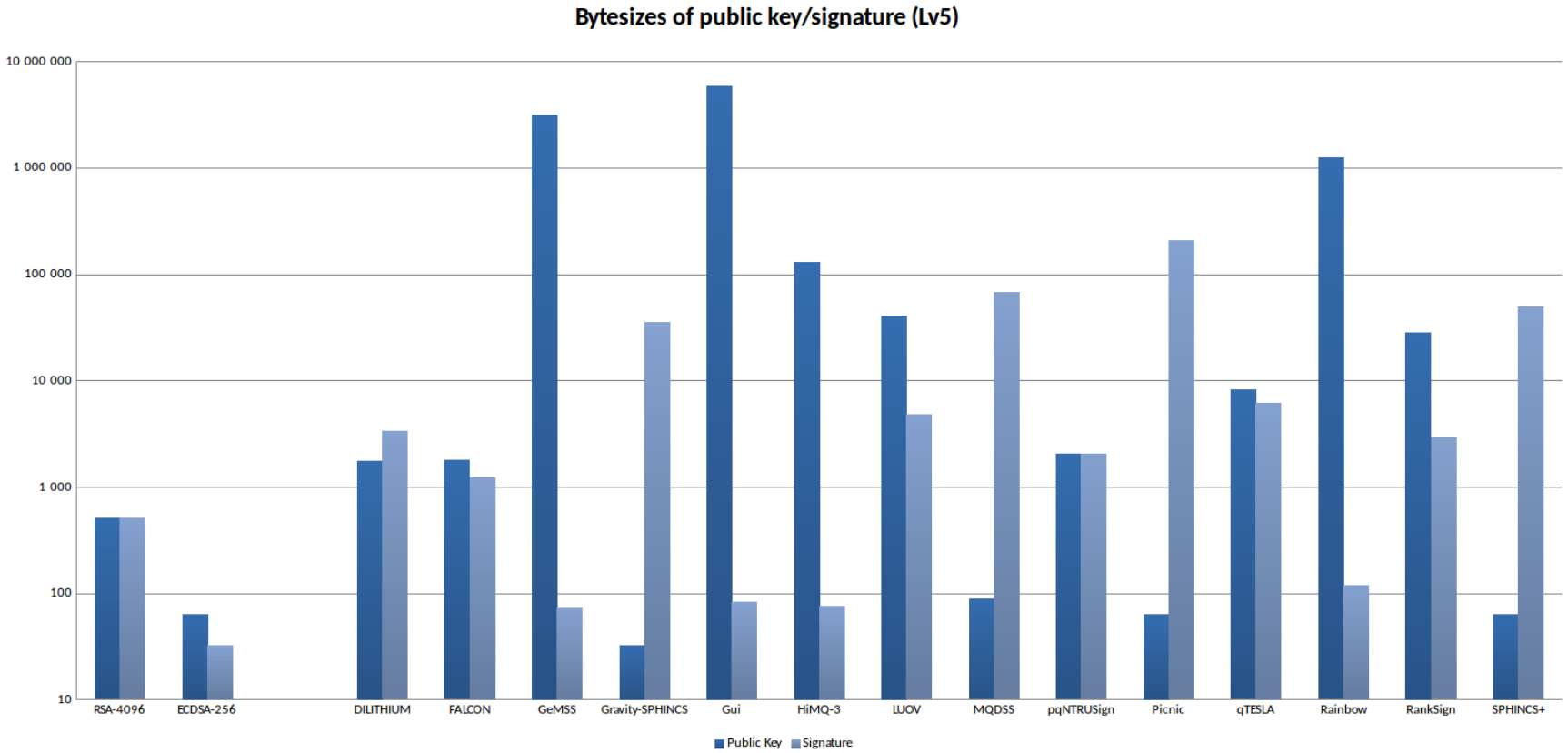
The adequate value for σ is at the intersection of the hardness curve (constraint 1) and the SD/RD curve (constraint 2).

- Rényi divergence-based analysis is much more efficient than if SD-based.
- Interesting fact: in practice, σ is not conditioned by λ but by q.

In practice, we gain about 30 bits of security (compared to the SD).

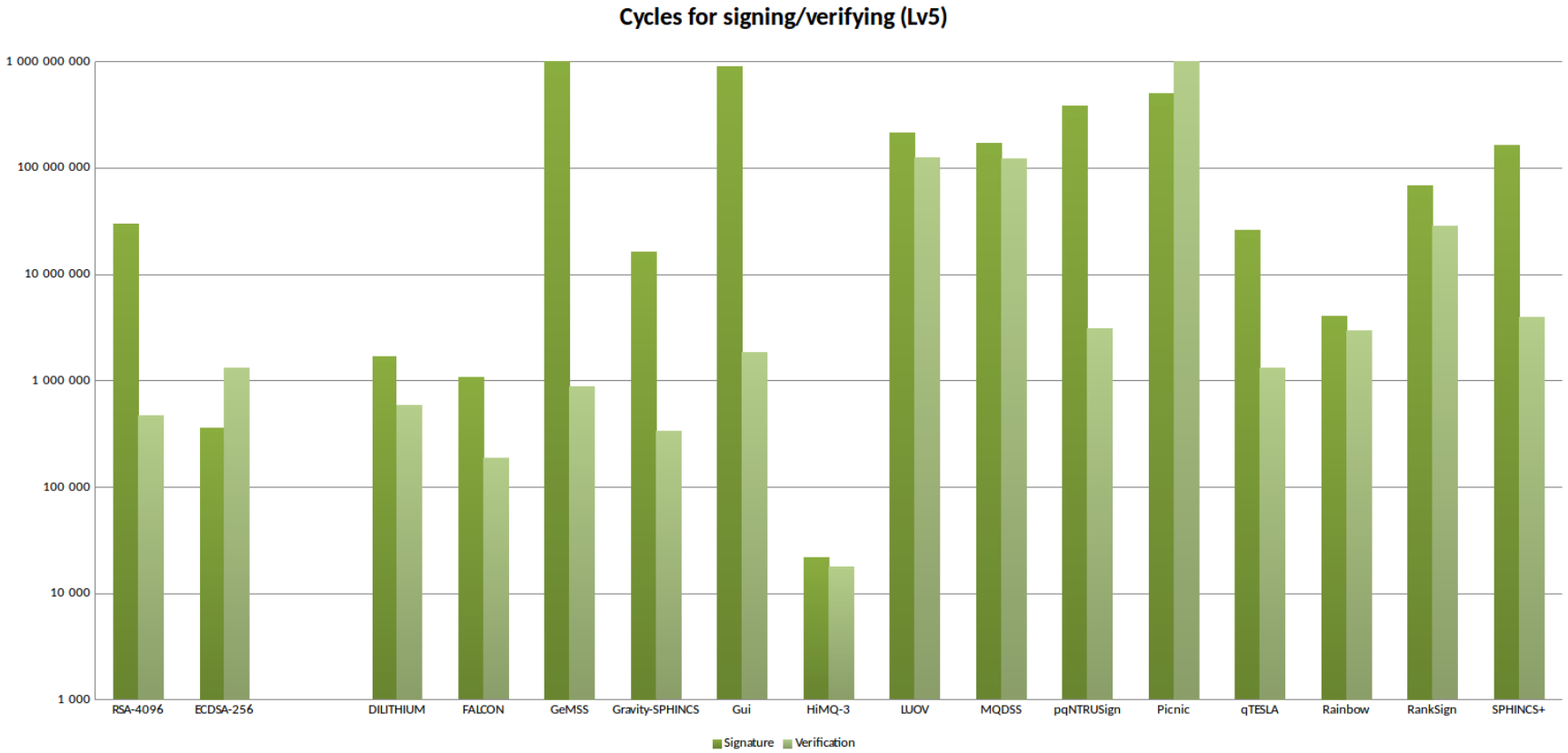
Falcon

The product of all these improvements is Falcon (joint work with Fouque, Hoffstein, Kirchner, Lyubashevsky, Pornin, Ricosset, Seiler, Whyte, Zhang).



Falcon

The product of all these improvements is Falcon (joint work with Fouque, Hoffstein, Kirchner, Lyubashevsky, Pornin, Ricosset, Seiler, Whyte, Zhang).



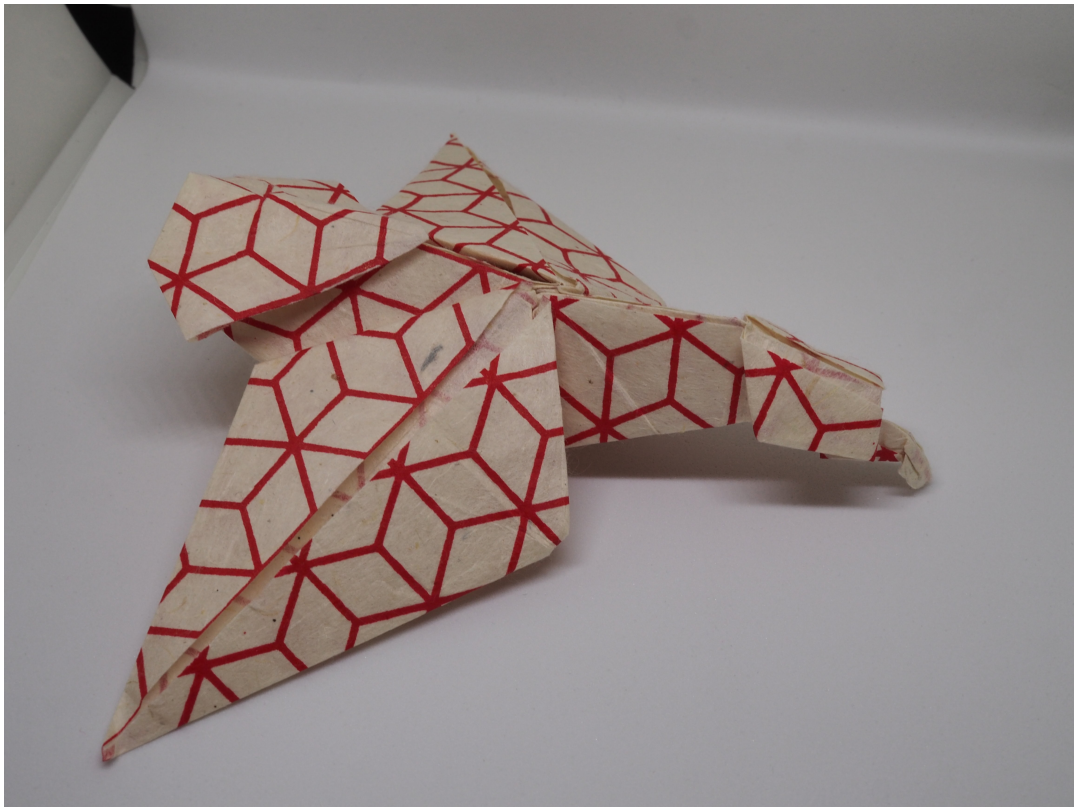
Falcon vs lattice-based Fiat-Shamir schemes

Fiat-Shamir schemes:

- Arguably simpler
- Avoid floating-point arithmetic
- Easy to protect against SCA ($\Rightarrow$ large signatures)
- Hard security proofs in the QROM

Falcon:

- Easy security proof in the QROM [BDF+11]
- Small public key and signatures. In addition:
  - Opt. mode 1: key recovery $\Rightarrow$ public key can be compressed to 40 bytes
  - Opt. mode 2: message recovery $\Rightarrow$ small message can be recovered from signature
- Easy to extend to advanced constructions (ABE, (H)IBE, etc.)
- Hard to protect against SCA
- Currently uses floating-point arithmetic



https://falcon-sign.info

Thanks!

Thanks to Fabrice Mouhartem for the Falcon origami!


Martin R. Albrecht, Shi Bai, and Léo Ducas.
A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes.
In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 153–178. Springer, Heidelberg, August 2016.


Martin R. Albrecht and Amit Deo.
Large modulus ring-LWE $\geq$ module-LWE.
In Takagi and Peyrin [TP17], pages 267–296.


Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe.
Post-quantum key exchange - A new hope.
In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 327–343, 2016.


Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila.
Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE.
In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 16*, pages 1006–1018. ACM Press, October 2016.


Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry.
Random oracles in a quantum world.

In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011.


Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen.
On the hardness of learning with rounding over small modulus.
In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 209–224. Springer, Heidelberg, January 2016.


Shi Bai, Adeline Langlois, Tancrède Lepoint, Damien Stehlé, and Ron Steinfeld.
Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance.
In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 3–24. Springer, Heidelberg, November / December 2015.


Léo Ducas, Vadim Lyubashevsky, and Thomas Prest.
Efficient identity-based encryption over NTRU lattices.
In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 22–41. Springer, Heidelberg, December 2014.


Léo Ducas and Phong Q. Nguyen.
Faster Gaussian lattice sampling using lazy floating-point arithmetic.
In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 415–432. Springer, Heidelberg, December 2012.


Léo Ducas and Thomas Prest.
Fast fourier orthogonalization.

In Sergei A. Abramov, Eugene V. Zima, and Xiao-Shan Gao, editors, *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2016, Waterloo, ON, Canada, July 19-22, 2016*, pages 191–198. ACM, 2016.


Rafaël del Pino and Vadim Lyubashevsky.
Amortization with fewer equations for proving knowledge of small secrets.
In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 365–394. Springer, Heidelberg, August 2017.


Oded Goldreich, Shafi Goldwasser, and Shai Halevi.
Public-key cryptosystems from lattice reduction problems.
In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 112–131. Springer, Heidelberg, August 1997.


Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan.
Trapdoors for hard lattices and new cryptographic constructions.
In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.


Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte.
NTRUSIGN: Digital signatures using the NTRU lattice.
In Marc Joye, editor, *CT-RSA 2003*, volume 2612 of *LNCS*, pages 122–140. Springer, Heidelberg, April 2003.


Andreas Hülsing, Tanja Lange, and Kit Smeets.
Rounded gaussians – fast and secure constant-time sampling for lattice-based crypto.
Cryptology ePrint Archive, Report 2017/1025, 2017.
https://eprint.iacr.org/2017/1025.


Paul Kirchner and Pierre-Alain Fouque.
Revisiting lattice attacks on overstretched NTRU parameters.
In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 3–26. Springer, Heidelberg, May 2017.


San Ling, Duong Hieu Phan, Damien Stehlé, and Ron Steinfeld.
Hardness of k-LWE and applications in traitor tracing.
In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 315–334. Springer, Heidelberg, August 2014.


Adeline Langlois, Damien Stehlé, and Ron Steinfeld.
GGHLite: More efficient multilinear maps from ideal lattices.
In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 239–256. Springer, Heidelberg, May 2014.


Phong Q. Nguyen and Oded Regev.
Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures.
In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 271–288. Springer, Heidelberg, May / June 2006.


Thomas Prest.
Sharper bounds in lattice-based cryptography using the Rényi divergence.
In Takagi and Peyrin [TP17], pages 347–374.


Damien Stehlé and Ron Steinfeld.
Making NTRU as secure as worst-case problems over ideal lattices.
In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 27–47. Springer, Heidelberg, May 2011.


Tsuyoshi Takagi and Thomas Peyrin, editors.
ASIACRYPT 2017, *Part I*, volume 10624 of *LNCS*. Springer, Heidelberg, December 2017.