$\mathbb{Z}$alcon: An Alternative FPA-free NTRU Sampler for Falcon

Pierre-Alain Fouque[1], François Gérard[2], Mélissa Rossi[3], Yang Yu[4]

[1]Rennes Univ, Inria and IRISA

[2]University of Luxembourg

[3]ANSSI

[4]Tsinghua University

NIST 3rd PQC Standardization Conference

Overview

We present a variant of Falcon, called $\mathbb{Z}$alcon

- does not use floats
- simpler and comparably efficient
- allows a provable masking

Overview

We present a variant of Falcon, called $\mathbb{Z}$alcon

- does not use floats
- simpler and comparably efficient
- allows a provable masking

$\mathbb{Z}$alcon vs. Mitaka[1] (the concurrent work presented 1 hour ago)

- some high-level ideas are shared $\Rightarrow$ the same efficiency & compactness
- different samplers $\Rightarrow$ Mitaka needs floats, $\mathbb{Z}$alcon does not
- Mitaka and $\mathbb{Z}$alcon can be masked similarly

[1] Mitaka: A Simpler, Parallelizable, Maskable Variant of Falcon. Mehdi Tibouchi, Thomas Espitau, Akira Takahashi, Alexandre Wallet. NIST 3rd PQC Standardization Conference.

Background

Falcon

Falcon is a round 3 finalist for NIST PQC signatures

Falcon

Falcon is a round 3 finalist for NIST PQC signatures

It follows the GPV hash-and-sign framework[2]

- signing $\Leftrightarrow$ sampling a lattice Gaussian

[2] Trapdoors for Hard Lattices and New Cryptographic Constructions. Craig Gentry, Chris Peikert, Vinod Vaikuntanathan. STOC 2008.

Falcon

Falcon is a round 3 finalist for NIST PQC signatures

It follows the GPV hash-and-sign framework[2]

- signing $\Leftrightarrow$ sampling a lattice Gaussian

Two key ingredients

- optimal NTRU trapdoor[3] $\Rightarrow$ **compactness**
- fast Fourier sampler[4] $\Rightarrow$ **efficiency**

---

[2] Trapdoors for Hard Lattices and New Cryptographic Constructions. Craig Gentry, Chris Peikert, Vinod Vaikuntanathan. STOC 2008.

[3] Efficient Identity-based Encryption over NTRU Lattices. Léo Ducas, Vadim Lyubashevsky, Thomas Prest. Asiacrypt 2014.

[4] Fast Fourier Orthogonalization. Léo Ducas, Thomas Prest. ISSAC 2016.

NTRU

Let $f, g \in \mathbb{Z}[x]/\phi(x)$. The NTRU lattice defined by $h = f \cdot g^{-1} \bmod q$ is

$$\mathcal{L}_{NTRU} = \{(u, v) \in \mathcal{R}_n^2 : u = vh \bmod q\}.$$

- In Falcon, $\phi(x) = x^n + 1$ with $n = 2^\ell$

NTRU

Let $f, g \in \mathbb{Z}[x]/\phi(x)$. The NTRU lattice defined by $h = f \cdot g^{-1} \bmod q$ is

$$\mathcal{L}_{NTRU} = \{(u, v) \in \mathcal{R}_n^2 \, : \, u = vh \bmod q\}.$$

- In Falcon, $\phi(x) = x^n + 1$ with $n = 2^\ell$

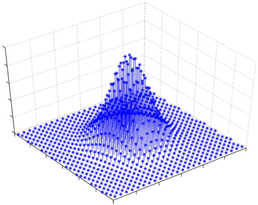
The trapdoor basis $\mathbf{B}_{f,g} = \begin{pmatrix} f & F \\ g & G \end{pmatrix}$ in Falcon

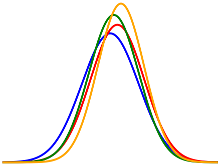
- f, g, F, G are short
- $\|(f, g)\| \approx 1.17\sqrt{q}$ to minimize the Gram-Schmidt norm $\|\mathbf{B}_{f,g}\|_{GS}$

Gaussian sampler of Falcon

Falcon uses a ring-efficient variant of Klein sampler

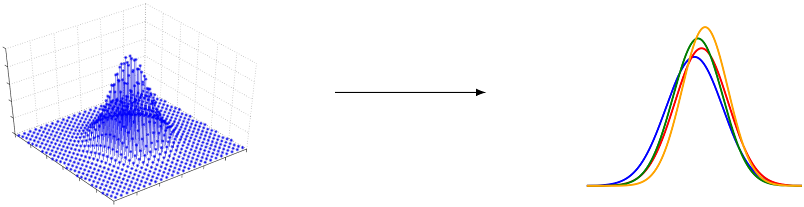
- exploits the tower of rings structure
- reduces the high-dimensional Gaussian to 1-dimensional Gaussians





Falcon uses a ring-efficient variant of Klein sampler

- exploits the tower of rings structure
- reduces the high-dimensional Gaussian to 1-dimensional Gaussians



With precomputed Falcon tree, the sampler is efficient

Drawbacks of Falcon sampler

There are still some issues w.r.t. Falcon sampler. . .

- heavily uses FPA (Gram-Schmidt orthogonalization)
- inherently sequential and reliant on special rings
- involved integer Gaussians have secret-dependent std. dev. and the secure implementation leads to efficiency loss[5]
- too complicated to mask

[5] Isochronous Gaussian Sampling: From Inception to Implementation. James Howe, Thomas Prest, Thomas Ricosset, Mélissa Rossi. PQCrypto 2020.

Drawbacks of Falcon sampler

There are still some issues w.r.t. Falcon sampler...

- heavily uses FPA (Gram-Schmidt orthogonalization)
- inherently sequential and reliant on special rings
- involved integer Gaussians have secret-dependent std. dev. and the secure implementation leads to efficiency loss[5]
- too complicated to mask

Let's resolve them!

[5] Isochronous Gaussian Sampling: From Inception to Implementation. James Howe, Thomas Prest, Thomas Ricosset, Mélissa Rossi. PQCrypto 2020.

$\mathbb{Z}$alcon

A first attempt

Klein sampler = randomized Babai's **nearest plane** algorithm

A first attempt

Klein sampler = randomized Babai's **nearest plane** algorithm

$$\Downarrow$$

Peikert sampler = randomized Babai's **round-off** algorithm

A first attempt

Klein sampler = randomized Babai's **nearest plane** algorithm

$$\Downarrow$$

Peikert sampler = randomized Babai's **round-off** algorithm

- offline: sample a pertubation **p** of covariance $\mathbf{\Sigma}_p = s^2 I - \mathbf{B}\mathbf{B}^t$

$\mathbf{\Sigma}_p$

A first attempt

Klein sampler = randomized Babai's **nearest plane** algorithm

$$\Downarrow$$

Peikert sampler = randomized Babai's **round-off** algorithm

- offline: sample a pertubation $\mathbf{p}$ of covariance $\mathbf{\Sigma}_p = s^2 I - \mathbf{B}\mathbf{B}^t$
- online: sample $D_{\mathcal{L}, r\sqrt{\mathbf{\Sigma}}, \mathbf{c}-\mathbf{p}} = \mathbf{B} \cdot D_{\mathbb{Z}^n, r, \mathbf{c}''}$ with $\mathbf{\Sigma} = \mathbf{B}\mathbf{B}^t$

$\mathbf{\Sigma}_p \quad + \quad \mathbf{B}\mathbf{B}^t$

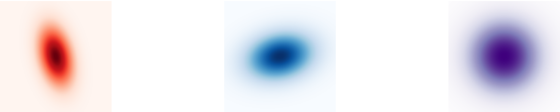
A first attempt

Klein sampler = randomized Babai's **nearest plane** algorithm

$$\Downarrow$$

Peikert sampler = randomized Babai's **round-off** algorithm

- offline: sample a pertubation $\mathbf{p}$ of covariance $\boldsymbol{\Sigma}_p = s^2 I - \mathbf{B}\mathbf{B}^t$
- online: sample $D_{\mathcal{L}, r\sqrt{\boldsymbol{\Sigma}}, \mathbf{c}-\mathbf{p}} = \mathbf{B} \cdot D_{\mathbb{Z}^n, r, \mathbf{c}''}$ with $\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}^t$



$$\boldsymbol{\Sigma}_p \quad + \quad \mathbf{B}\mathbf{B}^t \quad = \quad s^2 I$$

A first attempt

Peikert sampler resolves previous issues

- can be FPA-free[a]
- online sampling is parallelizable; compatible with arbitrary rings
- base samplings are independent of the secret
- simpler and supporting efficient masking

[a] Integral Matrix Gram Root and Lattice Gaussian Sampling without Floats. Léo Ducas, Steven Galbraith, Thomas Prest, Yang Yu. Eurocrypt 2020.

A first attempt

Peikert sampler resolves previous issues

- can be FPA-free[a]
- online sampling is parallelizable; compatible with arbitrary rings
- base samplings are independent of the secret
- simpler and supporting efficient masking

[a] Integral Matrix Gram Root and Lattice Gaussian Sampling without Floats. Léo Ducas, Steven Galbraith, Thomas Prest, Yang Yu. Eurocrypt 2020.

But security loss is significant

- The Gaussian quality achieved by Peikert $= s_1(\mathbf{B}_{f,g}) \cdot \eta_\epsilon(\mathbb{Z}^n)$
  that by Klein $= \|\mathbf{B}_{f,g}\|_{GS} \cdot \eta_\epsilon(\mathbb{Z}^n)$

A first attempt

Peikert sampler resolves previous issues

- can be FPA-free[a]
- online sampling is parallelizable; compatible with arbitrary rings
- base samplings are independent of the secret
- simpler and supporting efficient masking

[a] Integral Matrix Gram Root and Lattice Gaussian Sampling without Floats. Léo Ducas, Steven Galbraith, Thomas Prest, Yang Yu. Eurocrypt 2020.

But security loss is significant

- The Gaussian quality achieved by Peikert $= s_1(\mathbf{B}_{f,g}) \cdot \eta_\epsilon(\mathbb{Z}^n)$ that by Klein $= \|\mathbf{B}_{f,g}\|_{GS} \cdot \eta_\epsilon(\mathbb{Z}^n)$
- $s_1(\mathbf{B}_{f,g}) = O\left(n^{\frac{1}{4}}\sqrt{\log n}\right) \cdot \sqrt{q} \qquad \|\mathbf{B}_{f,g}\|_{GS} = O(1) \cdot \sqrt{q}$

A first attempt

Peikert sampler resolves previous issues

- can be FPA-free[a]
- online sampling is parallelizable; compatible with arbitrary rings
- base samplings are independent of the secret
- simpler and supporting efficient masking

[a] Integral Matrix Gram Root and Lattice Gaussian Sampling without Floats. Léo Ducas, Steven Galbraith, Thomas Prest, Yang Yu. Eurocrypt 2020.

But security loss is significant

- The Gaussian quality achieved by Peikert $= s_1(\mathbf{B}_{f,g}) \cdot \eta_\epsilon(\mathbb{Z}^n)$
  that by Klein $= \|\mathbf{B}_{f,g}\|_{GS} \cdot \eta_\epsilon(\mathbb{Z}^n)$
- $s_1(\mathbf{B}_{f,g}) = O\left(n^{\frac{1}{4}}\sqrt{\log n}\right) \cdot \sqrt{q}$ $\quad \|\mathbf{B}_{f,g}\|_{GS} = O(1) \cdot \sqrt{q}$
- bit security loss (quantum core SVP):
  $108 \to$ **52** for $n = 512$ $\quad 252 \to$ **130** for $n = 1024$

Our new sampler

Peikert sampler

- offline: sample a pertubation $\mathbf{p}$ of covariance $\boldsymbol{\Sigma}_p = s^2 I - \mathbf{B}\mathbf{B}^t$
- online: sample $D_{\mathcal{L},r\sqrt{\boldsymbol{\Sigma}},\mathbf{c}-\mathbf{p}} = \mathbf{B} \cdot D_{\mathbb{Z}^n,r,\mathbf{c}''}$ with $\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}^t$

Our new sampler

To enhance security, we work with Gram-Schmidt basis $\mathbf{B}^*$ instead of $\mathbf{B}$

- offline: sample a pertubation $\mathbf{p}$ of covariance $\mathbf{\Sigma}_p = s^2 I - \mathbf{B}^* \mathbf{B}^{*t}$
- online: sample $D_{\mathcal{L}, r\sqrt{\mathbf{\Sigma}}, \mathbf{c}-\mathbf{p}} = \mathbf{B}^* \cdot D_{\mathcal{L}(\mathbf{U}), r, \mathbf{c}''}$ with $\mathbf{\Sigma} = \mathbf{B}^* \mathbf{B}^{*t}$

Our new sampler

To enhance security, we work with Gram-Schmidt basis $\mathbf{B}^*$ instead of $\mathbf{B}$

- offline: sample a pertubation $\mathbf{p}$ of covariance $\boldsymbol{\Sigma}_p = s^2 I - \mathbf{B}^* \mathbf{B}^{*t}$
- online: sample $D_{\mathcal{L}, r\sqrt{\boldsymbol{\Sigma}}, \mathbf{c}-\mathbf{p}} = \mathbf{B}^* \cdot D_{\mathcal{L}(\mathbf{U}), r, \mathbf{c}''}$ with $\boldsymbol{\Sigma} = \mathbf{B}^* \mathbf{B}^{*t}$

$$\mathbf{B}_{f,g} = \begin{pmatrix} f & F \\ g & G \end{pmatrix} = \begin{pmatrix} f & F^* = -\frac{q\bar{g}}{f\bar{f}+g\bar{g}} \\ g & G^* = \frac{q\bar{f}}{f\bar{f}+g\bar{g}} \end{pmatrix} \begin{pmatrix} 1 & u \\ & 1 \end{pmatrix} = \mathbf{B}^*_{f,g} \mathbf{U}$$

Our new sampler

To enhance security, we work with Gram-Schmidt basis $\mathbf{B}^*$ instead of $\mathbf{B}$

- offline: sample a pertubation $\mathbf{p}$ of covariance $\mathbf{\Sigma}_p = s^2 I - \mathbf{B}^* \mathbf{B}^{*t}$
- online: sample $D_{\mathcal{L}, r\sqrt{\mathbf{\Sigma}}, \mathbf{c}-\mathbf{p}} = \mathbf{B}^* \cdot D_{\mathcal{L}(\mathbf{U}), r, \mathbf{c}''}$ with $\mathbf{\Sigma} = \mathbf{B}^* \mathbf{B}^{*t}$

$$\mathbf{B}_{f,g} = \begin{pmatrix} f & F \\ g & G \end{pmatrix} = \begin{pmatrix} f & F^* = -\frac{q\bar{g}}{f\bar{f}+g\bar{g}} \\ g & G^* = \frac{q\bar{f}}{f\bar{f}+g\bar{g}} \end{pmatrix} \begin{pmatrix} 1 & u \\ & 1 \end{pmatrix} = \mathbf{B}^*_{f,g} \mathbf{U}$$

- $D_{\mathcal{L}(\mathbf{U}), r, \mathbf{c}''}$ is still easy and highly parallelizable
- $s_1(\mathbf{B}_{f,g}) = O\left(n^{\frac{1}{4}} \sqrt{\log n}\right) \cdot \sqrt{q} \quad \Rightarrow \quad s_1(\mathbf{B}^*_{f,g}) = O\left(n^{\frac{1}{8}} \log^{\frac{1}{4}} n\right) \cdot \sqrt{q}$
- security (quantum core SVP):
  $108 \rightarrow 52 \rightarrow$ **79** for $n = 512$
  $252 \rightarrow 130 \rightarrow$ **185** for $n = 1024$

Our new sampler

To avoid FPA, we further replace $\mathbf{B}^*$ with an integral approximate $\widetilde{\mathbf{B}^*}$

Our new sampler

To avoid FPA, we further replace $\mathbf{B}^*$ with an integral approximate $\widetilde{\mathbf{B}^*}$

- $u \;\Rightarrow\; \tilde{u} = \frac{\lfloor p \cdot u \rceil}{p}$ for some $p \in \mathbb{Z}$

Our new sampler

To avoid FPA, we further replace $\mathbf{B}^*$ with an integral approximate $\widetilde{\mathbf{B}^*}$

- $u \;\Rightarrow\; \tilde{u} = \frac{\lfloor p \cdot u \rceil}{p}$ for some $p \in \mathbb{Z}$

All intermediate values are **integral** too

- $\widetilde{\mathbf{B}^*} = \mathbf{B} \begin{pmatrix} 1 & -\tilde{u} \\ & 1 \end{pmatrix} \in \frac{1}{p}\mathcal{R}^{2\times 2}$
- $\widetilde{\mathbf{B}^*}^{-1} = \begin{pmatrix} 1 & \tilde{u} \\ & 1 \end{pmatrix} \mathbf{B}^{-1} \in \frac{1}{pq}\mathcal{R}^{2\times 2}$

Comparison with other samplers

	quality	FPA
Klein (Falcon)	$\|\mathbf{B}\|_{GS} = O(\sqrt{q})$	Yes
Peikert	$s_1(\mathbf{B}) = O\left(n^{\frac{1}{4}}\sqrt{\log n}\sqrt{q}\right)$	No
Hybrid[6] (Mitaka)	$s_1(\mathbf{B}^*) = O\left(n^{\frac{1}{8}}\log^{\frac{1}{4}} n\sqrt{q}\right)$	Yes
Ours ($\mathbb{Z}$alcon)	$s_1(\widetilde{\mathbf{B}^*}) = O\left(n^{\frac{1}{8}}\log^{\frac{1}{4}} n\sqrt{q}\right)$	No

[6] Gaussian Sampling in Lattice-Based Cryptography. Thomas Prest. PhD thesis, ENS Paris, 2015.

Comparison with other samplers

	quality	FPA
Klein (Falcon)	$\|\mathbf{B}\|_{GS} = O(\sqrt{q})$	Yes
Peikert	$s_1(\mathbf{B}) = O\left(n^{\frac{1}{4}}\sqrt{\log n}\sqrt{q}\right)$	No
Hybrid[6] (Mitaka)	$s_1(\mathbf{B}^*) = O\left(n^{\frac{1}{8}}\log^{\frac{1}{4}} n\sqrt{q}\right)$	Yes
Ours ($\mathbb{Z}$alcon)	$s_1(\widetilde{\mathbf{B}^*}) = O\left(n^{\frac{1}{8}}\log^{\frac{1}{4}} n\sqrt{q}\right)$	No

- Hybrid: Klein over $\mathcal{R}$ with Peikert as subroutine
- Ours: Peikert sampler with a smaller covariance

[6] Gaussian Sampling in Lattice-Based Cryptography. Thomas Prest. PhD thesis, ENS Paris, 2015.

Improved Key Generation

The security not only relies on **Sampler** but also on **Trapdoor**

Improved Key Generation

The security not only relies on **Sampler** but also on **Trapdoor**

To enhance security, we further use a refined key generation

- $s_1(\mathbf{B}^*_{f,g}) \;\Rightarrow\; \min\{s_1(\mathbf{B}^*_{f,\sigma_i(g)})\}$ where $\sigma_i : x \mapsto x^{2i+1}$

Improved Key Generation

The security not only relies on **Sampler** but also on **Trapdoor**

To enhance security, we further use a refined key generation

- $s_1(\mathbf{B}^*_{f,g}) \ \Rightarrow\ \min\{s_1(\mathbf{B}^*_{f,\sigma_i(g)})\}$ where $\sigma_i : x \mapsto x^{2i+1}$
- $\sigma_{f,g}/\sqrt{\frac{q}{2n}} : 1.17 \ \Rightarrow\ 1.36\ /\ 1.47$ for $n = 512\ /\ 1024$

Improved Key Generation

The security not only relies on **Sampler** but also on **Trapdoor**

To enhance security, we further use a refined key generation

- $s_1(\mathbf{B}^*_{f,g}) \;\Rightarrow\; \min\{s_1(\mathbf{B}^*_{f,\sigma_i(g)})\}$ where $\sigma_i : x \mapsto x^{2i+1}$
- $\sigma_{f,g}/\sqrt{\frac{q}{2n}}$: $1.17 \;\Rightarrow\; 1.36$ / 1.47 for $n = 512$ / 1024
- security (quantum core SVP):
  $108 \to 52 \to 79 \to$ **83** for $n = 512$
  $252 \to 130 \to 185 \to$ **192** for $n = 1024$

Improved Key Generation

The security not only relies on **Sampler** but also on **Trapdoor**

To enhance security, we further use a refined key generation

- $s_1(\mathbf{B}^*_{f,g}) \;\Rightarrow\; \min\{s_1(\mathbf{B}^*_{f,\sigma_i(g)})\}$ where $\sigma_i : x \mapsto x^{2i+1}$
- $\sigma_{f,g}/\sqrt{\frac{q}{2n}} : 1.17 \;\Rightarrow\; 1.36$ / 1.47 for $n = 512$ / 1024
- security (quantum core SVP):
  $108 \to 52 \to 79 \to$ **83** for $n = 512$
  $252 \to 130 \to 185 \to$ **192** for $n = 1024$

Mitaka uses similar but more comprehensive techniques

- gain around 15 bits of security with more randomness and time

Implementation

Integer Gaussian sampling

$\mathbb{Z}$alcon needs two types of integer Gaussian samplers

- arbitrary center: $D_{\mathbb{Z},r,c}$ with $c \in \frac{1}{Q}\mathbb{Z}$ (online)
- large width: $D_{\mathbb{Z},Lr}$ (offline)

Integer Gaussian sampling

$\mathbb{Z}$alcon needs two types of integer Gaussian samplers

- arbitrary center: $D_{\mathbb{Z},r,c}$ with $c \in \frac{1}{Q}\mathbb{Z}$ (online)
- large width: $D_{\mathbb{Z},Lr}$ (offline)

We follow Micciancio-Walter approach[7]

- fully over integers
- offline / online

[7] Gaussian Sampling over the Integers: Efficient, Generic, Constant-time. Daniele Micciancio, Michael Walter. Crypto 2017.

Preliminary results

Caveat: the implementation is still ongoing

Preliminary results

Caveat: the implementation is still ongoing

Online sampling seems encouraging

- base sampler for arbitrary center samplings is implemented via CDT
- storage for tables: $33 \times 15 \times 82 = 40590$ bits
- unoptimized result on i7-1065G7 CPU @ 1.30GHz for $n = 512$: ≈ 400 online samplings per seconds

Preliminary results

Caveat: the implementation is still ongoing

Online sampling seems encouraging

- base sampler for arbitrary center samplings is implemented via CDT
- storage for tables: $33 \times 15 \times 82 = 40590$ bits
- unoptimized result on i7-1065G7 CPU @ 1.30GHz for $n = 512$:
  ≈ 400 online samplings per seconds

Offline sampling is costly

- it requires $\approx 2^{15}$ calls of $D_{\mathbb{Z},Lr}$ and $L = 2^{35}$
- but all these samplings are identical and secret-independent

Masking

Masking

Our sampler can be masked with standard techniques.

- It is possible to only mask the online phase $\rightarrow$ more efficient as the main randomness generation can be made offline.

Our building blocks:

- masked CDT [8]
- masked NTT multiplications (between 2 sensitive polys)

We provide a complete proof of masking in the ISW model.

[8] GALACTICS: Gaussian sampling for lattice-based constant-time implementation of cryptographic signatures, revisted. Gilles Barthe, Sonia Belaid, Thomas Espitau, Pierre-Alain Fouque, Mélissa Rossi, Mehdi Tibouchi. CCS 2019.
An Efficient and Provable Masked Implementation of qTESLA. François Gérard, Mélissa Rossi. CARDIS 2019.

Masking

Our sampler can be masked with standard techniques.

- It is possible to only mask the online phase $\rightarrow$ more efficient as the main randomness generation can be made offline.

Our building blocks:

- masked CDT [8]
- masked NTT multiplications (between 2 sensitive polys)

We provide a complete proof of masking in the ISW model.

Mitaka uses a different building block for the Gaussian generation: share-by-share based on Gaussian convolution.
This efficient gadget can be directly applied to $\mathbb{Z}$alcon.

[8] GALACTICS: Gaussian sampling for lattice-based constant-time implementation of cryptographic signatures, revisted. Gilles Barthe, Sonia Belaid, Thomas Espitau, Pierre-Alain Fouque, Mélissa Rossi, Mehdi Tibouchi. CCS 2019.
An Efficient and Provable Masked Implementation of qTESLA. François Gérard, Mélissa Rossi. CARDIS 2019.

Conclusion

We present $\mathbb{Z}$alcon, an FPA-free and simpler variant of Falcon

Conclusion

We present $\mathbb{Z}$alcon, an FPA-free and simpler variant of Falcon

We present one of the first provable maskings for lattice Gaussian sampling

Conclusion

We present $\mathbb{Z}$alcon, an FPA-free and simpler variant of Falcon

We present one of the first provable maskings for lattice Gaussian sampling

The implementation is still in progress...

	pk (bytes)	sig (bytes)	NIST security level
Falcon-512	897	666	1
$\mathbb{Z}$alcon-512	897	≈ 766	1^-
Dilithium-1^-	992	1843	1^-
Falcon-1024	1793	1280	5
$\mathbb{Z}$alcon-1024	1793	≈ 1526	3
Dilithium-3	1952	3293	3

Thank you!