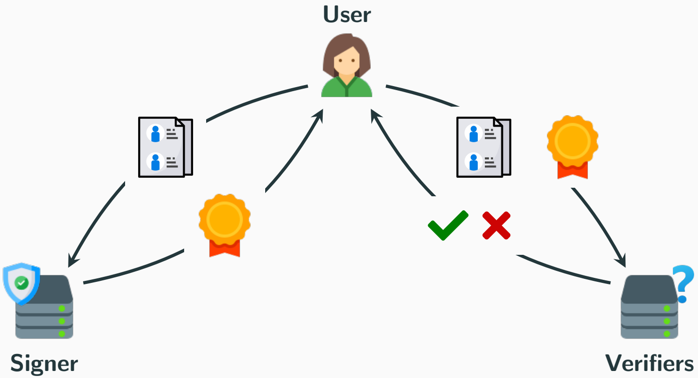
Worst-Case Lattice Sampler with Truncated Gadgets and Applications

December 09th, 2025

Corentin Jeudy[1], Olivier Sanders[1]

[1] Orange, Applied Crypto Group

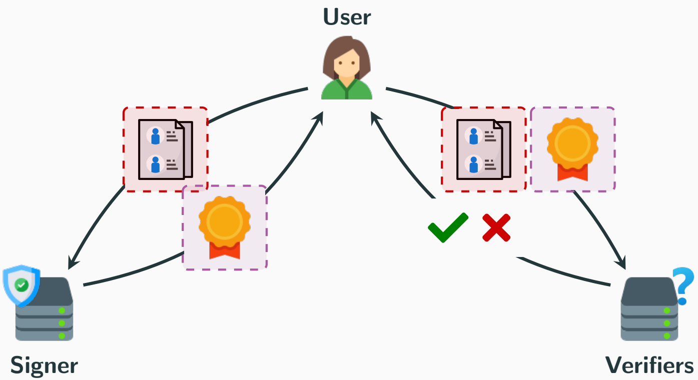
Digital Signatures



Digital Signatures

Private Data Leak

Tracking

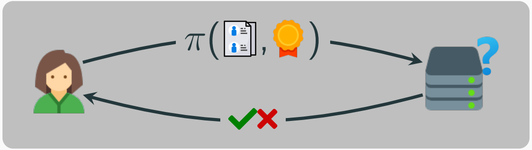


No control over the disclosed information: Verifiers (and attacker) learn everything

Traceable accross different authentications: Same signature allows tracing

Privacy from Zero-Knowledge Proofs

? How is **privacy** usually obtained? **Zero-Knowledge Proof** of **Signature** & **Message**



Proof of $\mathbf{x}$ s.t. $\mathbf{g}^{\mathbf{x}} = \mathbf{h}$	Proof of $\mathbf{x}$ s.t. $\mathbf{A}\mathbf{x} = \mathbf{u} \wedge \|\mathbf{x}\| \leq B$	Proof of $\mathbf{x}$ s.t. $\mathcal{H}(\mathbf{x}) = \mathbf{h}$

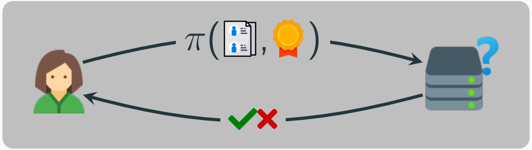
Algebraic → **Generic**

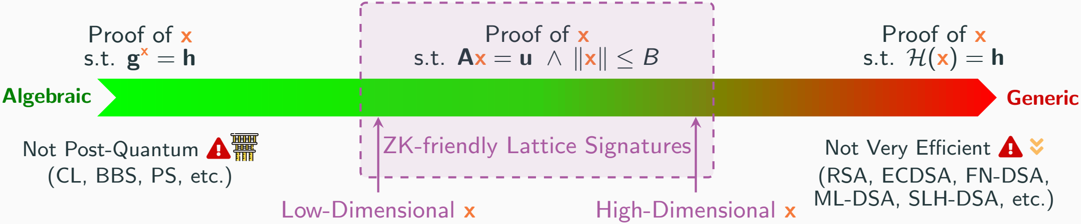
Not Post-Quantum ⚠
(CL, BBS, PS, etc.)

Not Very Efficient ⚠
(RSA, ECDSA, FN-DSA,
ML-DSA, SLH-DSA, etc.)

Privacy from Zero-Knowledge Proofs

? How is **privacy** usually obtained? **Zero-Knowledge Proof** of **Signature** & **Message**





Proof of $\mathbf{x}$
s.t. $\mathbf{g}^{\mathbf{x}} = \mathbf{h}$

Proof of $\mathbf{x}$
s.t. $\mathbf{A}\mathbf{x} = \mathbf{u} \ \wedge\ \|\mathbf{x}\| \leq B$

Proof of $\mathbf{x}$
s.t. $\mathcal{H}(\mathbf{x}) = \mathbf{h}$

Algebraic

Generic

Not Post-Quantum ⚠
(CL, BBS, PS, etc.)

ZK-friendly Lattice Signatures

Low-Dimensional $\mathbf{x}$

High-Dimensional $\mathbf{x}$

Not Very Efficient ⚠
(RSA, ECDSA, FN-DSA,
ML-DSA, SLH-DSA, etc.)

Gadget-Based Samplers

Micciancio-Peikert trapdoors [MP12][1]: Family of matrices $\mathbf{A}_t$ such that

$$\mathbf{A}_t = [\mathbf{A}' | t\mathbf{G} - \mathbf{A}'\mathbf{R}] \text{ and } \mathbf{A}' = [\mathbf{I}|\mathbf{A}]$$

verifies $\mathbf{A}_{\mathsf{T}}\mathbf{L} = t\mathbf{G} \bmod q$, with $\mathbf{L} = \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix}$

with $\mathbf{G} = [b^0\mathbf{I}| \ldots |b^{k-1}\mathbf{I}]$, and $k = \log_b q$
(base-b decomposition)

🔑 $\mathbf{R}$ 🔑 $\mathbf{B} = \mathbf{A}'\mathbf{R}$
🏷 t

Naive Approach: Compute $\mathbf{z}$ so that $t\mathbf{G}\mathbf{z} = \mathbf{u} \bmod q$, and return $\mathbf{L}\mathbf{z}$ as preimage of $\mathbf{u}$

⊗ Collecting many preimages will leak $\mathbf{R}$...

📖 Gaussian distribution on $\mathbf{z}$ and add Gaussian mask $\mathbf{p}$: preimages $\mathbf{v} = \mathbf{p} + \mathbf{L}\mathbf{z} = \begin{bmatrix} \mathbf{p}_1 + \mathbf{R}\mathbf{z} \\ \mathbf{p}_2 + \mathbf{z} \end{bmatrix}$

(and syndrome correction so that $t\mathbf{G}\mathbf{z} = \mathbf{w} = \mathbf{u} - \mathbf{A}_t\mathbf{p}$)

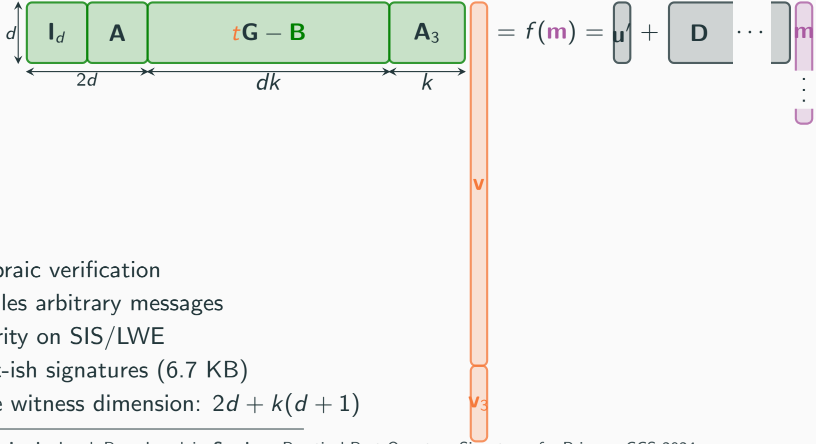
---

[1] Micciancio, Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. Eurocrypt 2012

ZK-Friendly Signature from Gadget Sampler

Signature scheme from [AGJ$^+$24][2]:

🔑 : $\mathbf{R}$ 🔑 : $\mathbf{B} = [\mathbf{I}_d | \mathbf{A}]\mathbf{R}$ 🏅 : $t, \mathbf{v}, \mathbf{v}_3$ 📄 : $\mathbf{m}$ PP : $(\mathbf{A}, \mathbf{A}_3, \mathbf{D}, \mathbf{u}', \mathbf{G} = [b^0\mathbf{I} | \ldots | b^{k-1}\mathbf{I}])$



$$\begin{bmatrix} \mathbf{I}_d & \mathbf{A} & t\mathbf{G} - \mathbf{B} & \mathbf{A}_3 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{v}_3 \end{bmatrix} = f(\mathbf{m}) = \mathbf{u}' + \mathbf{D} \cdots \mathbf{m}$$

- ✓ Algebraic verification
- ✓ Handles arbitrary messages
- ✓ Security on SIS/LWE
- ✓ Short-ish signatures (6.7 KB)
- ✗ Large witness dimension: $2d + k(d+1)$

---

[2] Argo, Güneysu, **Jeudy**, Land, Roux-Langlois, **Sanders**. Practical Post-Quantum Signatures for Privacy. CCS 2024

Reduce Dimension with Approximate Trapdoor

Reduce gadget dimension with "approximate trapdoors" [CGM19][3] with truncation.

Note $\mathbf{G}_L = [b^0\mathbf{I}_d|\dots|b^{\ell-1}\mathbf{I}_d]$, $\mathbf{G}_H = [b^{\ell}\mathbf{I}_d|\dots|b^{k-1}\mathbf{I}_d]$. Now: $\mathbf{A}_t = [\mathbf{A}'|t\mathbf{G}_H - \mathbf{A}'\mathbf{R}]$, with $\mathbf{A}' = [\mathbf{I}_d|\mathbf{A}]$.
Sampling $\mathbf{v}'$ s.t. $\mathbf{A}_t\mathbf{v}' + \mathbf{e} = \mathbf{u}$ with $\mathbf{e}$ small is sufficient.

$$\mathbf{A}_t\mathbf{v}' + \mathbf{e} = \mathbf{u} \iff [\mathbf{I}_d|\mathbf{A}|t\mathbf{G}_H - \mathbf{A}'\mathbf{R}]\underbrace{\left(\mathbf{v}' + \begin{bmatrix}\mathbf{e}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}\right)}_{\text{exact preimage } \mathbf{v}} = \mathbf{u}$$

---

[3] Chen, Genise, Mukherjee. Approximate trapdoors for lattices and smaller hash-and-sign signatures. Asiacrypt 2019.

Reduce Dimension with Approximate Trapdoor

Reduce gadget dimension with “approximate trapdoors” [CGM19][3] with truncation.

Note $\mathbf{G}_L = [b^0\mathbf{I}_d|\ldots|b^{\ell-1}\mathbf{I}_d]$, $\mathbf{G}_H = [b^{\ell}\mathbf{I}_d|\ldots|b^{k-1}\mathbf{I}_d]$. Now: $\mathbf{A}_t = [\mathbf{A}'|t\mathbf{G}_H - \mathbf{A}'\mathbf{R}]$, with $\mathbf{A}' = [\mathbf{I}_d|\mathbf{A}]$. Sampling $\mathbf{v}'$ s.t. $\mathbf{A}_t\mathbf{v}' + \mathbf{e} = \mathbf{u}$ with $\mathbf{e}$ small is sufficient.

$$\mathbf{A}_t\mathbf{v}' + \mathbf{e} = \mathbf{u} \quad\Longleftrightarrow\quad [\mathbf{I}_d|\mathbf{A}|t\mathbf{G}_H - \mathbf{A}'\mathbf{R}]\underbrace{\left(\mathbf{v}' + \begin{bmatrix}\mathbf{e}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}\right)}_{\text{exact preimage } \mathbf{v}} = \mathbf{u}$$

Naive Approach: Compute $\mathbf{z} = (\mathbf{z}_L, \mathbf{z}_H)$ so that $t(\mathbf{G}_L\mathbf{z}_L + \mathbf{G}_H\mathbf{z}_H) = \mathbf{u} \bmod q$, and return $\mathbf{v}' = \mathbf{L}\mathbf{z}_H$ as an approximate preimage of $\mathbf{u}$. The error is $\mathbf{e} = t\mathbf{G}_L\mathbf{z}_L$.

---

[3] Chen, Genise, Mukherjee. Approximate trapdoors for lattices and smaller hash-and-sign signatures. Asiacrypt 2019.

Reduce Dimension with Approximate Trapdoor

📖 Reduce gadget dimension with "approximate trapdoors" [CGM19][3] with truncation.
Note $\mathbf{G}_L = [b^0\mathbf{I}_d|\dots|b^{\ell-1}\mathbf{I}_d]$, $\mathbf{G}_H = [b^{\ell}\mathbf{I}_d|\dots|b^{k-1}\mathbf{I}_d]$. Now: $\mathbf{A}_t = [\mathbf{A}'|t\mathbf{G}_H - \mathbf{A}'\mathbf{R}]$, with $\mathbf{A}' = [\mathbf{I}_d|\mathbf{A}]$.
Sampling $\mathbf{v}'$ s.t. $\mathbf{A}_t\mathbf{v}' + \mathbf{e} = \mathbf{u}$ with $\mathbf{e}$ small is sufficient.

$$\mathbf{A}_t\mathbf{v}' + \mathbf{e} = \mathbf{u} \quad \Longleftrightarrow \quad [\mathbf{I}_d|\mathbf{A}|t\mathbf{G}_H - \mathbf{A}'\mathbf{R}]\underbrace{\left(\mathbf{v}' + \begin{bmatrix}\mathbf{e}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}\right)}_{\text{exact preimage } \mathbf{v}} = \mathbf{u}$$

Naive Approach: Compute $\mathbf{z} = (\mathbf{z}_L, \mathbf{z}_H)$ so that $t(\mathbf{G}_L\mathbf{z}_L + \mathbf{G}_H\mathbf{z}_H) = \mathbf{u} \bmod q$, and return $\mathbf{v}' = \mathbf{L}\mathbf{z}_H$ as an approximate preimage of $\mathbf{u}$. The error is $\mathbf{e} = t\mathbf{G}_L\mathbf{z}_L$.

? Can we handle the **convolution as before** with the **additional error** $\mathbf{e}$?

---

[3] Chen, Genise, Mukherjee. Approximate trapdoors for lattices and smaller hash-and-sign signatures. Asiacrypt 2019.

What About Security?

✖ To prove $\mathbf{v}$ does not leak $\mathbf{R}$, [CGM19] must be able to simulate $\mathbf{e}$ (as it depends on $\mathbf{p}$). Requires knowing the distribution of $\mathbf{e}$, which causes two problems:

❶ Distribution of $\mathbf{e}$ difficult when $\mathbf{u}$ is arbitrary/adversarially chosen

❷ Distribution of $\mathbf{e}$ depends on tag t, which must stay hidden (for security proof)

What About Security?

⊗ To prove $\mathbf{v}$ does not leak $\mathbf{R}$, [CGM19] must be able to simulate $\mathbf{e}$ (as it depends on $\mathbf{p}$). Requires knowing the distribution of $\mathbf{e}$, which causes two problems:

❶ Distribution of $\mathbf{e}$ difficult when $\mathbf{u}$ is arbitrary/adversarially chosen

❷ Distribution of $\mathbf{e}$ depends on tag t, which must stay hidden (for security proof)

≈ Proposed solution requires $\mathbf{u} = f(\mathbf{m})$ to be a *consistent, random, reprogrammable* function of $\mathbf{m}$. That is... a **random oracle**.

✓ Fine for hash-and-sign standard signatures,

✗ Not for ZK-friendly signatures, where $f(\mathbf{m})$ is algebraic (e.g. $f(\mathbf{m}) = \mathbf{u}' + \mathbf{Dm}$).

What About Security?

✗ To prove $\mathbf{v}$ does not leak $\mathbf{R}$, [CGM19] must be able to simulate $\mathbf{e}$ (as it depends on $\mathbf{p}$). Requires knowing the distribution of $\mathbf{e}$, which causes two problems:

❶ Distribution of $\mathbf{e}$ difficult when $\mathbf{u}$ is arbitrary/adversarially chosen

❷ Distribution of $\mathbf{e}$ depends on tag t, which must stay hidden (for security proof)

≈ Proposed solution requires $\mathbf{u} = f(\mathbf{m})$ to be a *consistent, random, reprogrammable* function of $\mathbf{m}$. That is... a **random oracle**.

- ✓ Fine for hash-and-sign standard signatures,
- ✗ Not for ZK-friendly signatures, where $f(\mathbf{m})$ is algebraic (e.g. $f(\mathbf{m}) = \mathbf{u}' + \mathbf{Dm}$).

[CGM19] **not applicable to the main use-cases** of gadget samplers ($\mathbf{u}$ arbitrary)

Back To Square One

Use the perturbation to hide (some of) the error using convolution. Split $\mathbf{R}$ into $(\mathbf{R}_1, \mathbf{R}_2)$ so that $[\mathbf{I}_d|\mathbf{A}]\mathbf{R} = \mathbf{R}_1 + \mathbf{A}\mathbf{R}_2$. The unperturbed preimage is

$$\mathbf{v} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{I}_{d(k-\ell)} \end{bmatrix} \mathbf{z}_H + \begin{bmatrix} t\mathbf{G}_L\mathbf{z}_L \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

Back To Square One

Use the perturbation to hide (some of) the error using convolution. Split $\mathbf{R}$ into $(\mathbf{R}_1, \mathbf{R}_2)$ so that $[\mathbf{I}_d|\mathbf{A}]\mathbf{R} = \mathbf{R}_1 + \mathbf{A}\mathbf{R}_2$. The unperturbed preimage is

$$\mathbf{v} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{I}_{d(k-\ell)} \end{bmatrix} \mathbf{z}_H + \begin{bmatrix} t\mathbf{G}_L\mathbf{z}_L \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \boxed{\begin{bmatrix} t\mathbf{G}_L & \mathbf{R}_1 \\ \mathbf{0} & \mathbf{R}_2 \\ \mathbf{0} & \mathbf{I}_{d(k-\ell)} \end{bmatrix}} \begin{bmatrix} \mathbf{z}_L \\ \mathbf{z}_H \end{bmatrix}$$

- ☒ $\mathbf{G}_L$ large compared to $\mathbf{R}_i \implies$ needs large perturbation
- ☒ Matrix not full rank when $\ell > 1 \implies$ complex lattice smoothing analysis

Back To Square One

Use the perturbation to hide (some of) the error using convolution. Split $\mathbf{R}$ into $(\mathbf{R}_1, \mathbf{R}_2)$ so that $[\mathbf{I}_d|\mathbf{A}]\mathbf{R} = \mathbf{R}_1 + \mathbf{A}\mathbf{R}_2$. The unperturbed preimage is

$$\mathbf{v} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{I}_{d(k-\ell)} \end{bmatrix} \mathbf{z}_H + \begin{bmatrix} t\mathbf{G}_L\mathbf{z}_L \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \boxed{\begin{bmatrix} t\mathbf{G}_L & \mathbf{R}_1 \\ \mathbf{0} & \mathbf{R}_2 \\ \mathbf{0} & \mathbf{I}_{d(k-\ell)} \end{bmatrix}} \begin{bmatrix} \mathbf{z}_L \\ \mathbf{z}_H \end{bmatrix}$$

factor

Public Part

($\mathbf{K}$-projection)

$$\underbrace{\boxed{\begin{bmatrix} \mathbf{G}_L & & \\ & \mathbf{I}_d & \\ & & \mathbf{I}_{d(k-\ell)} \end{bmatrix}}}_{\mathbf{K}} \cdot \underbrace{\boxed{\begin{bmatrix} t\mathbf{I}_d & \mathbf{0} & \mathbf{R}_1 \\ \mathbf{0} & t\mathbf{I}_{d(\ell-1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{d(k-\ell)} \end{bmatrix}}}_{\mathbf{L}}$$

Private Part

($\mathbf{L}$ full rank)

Perturb $\mathbf{Lz}$ and project with $\mathbf{K}$ afterwards.

Tailor the Perturbation: Elliptic Gaussians

We need to compensate the covariance $s_z^2 \mathbf{L}\mathbf{L}^T$

$$\mathbf{L}\mathbf{L}^T = \begin{bmatrix} t^2\mathbf{I}_d + \mathbf{R}_1\mathbf{R}_1^T & \mathbf{0} & \mathbf{R}_1\mathbf{R}_2^T & \mathbf{R}_1 \\ \mathbf{0} & t^2\mathbf{I}_{d(\ell-1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_2\mathbf{R}_1^T & \mathbf{0} & \mathbf{R}_2\mathbf{R}_2^T & \mathbf{R}_2 \\ \mathbf{R}_1^T & \mathbf{0} & \mathbf{R}_2^T & \mathbf{I}_{d(k-\ell)} \end{bmatrix}$$

Tailor the Perturbation: Elliptic Gaussians

We need to compensate the covariance $s_z^2\mathbf{L}\mathbf{L}^T$

$$\mathbf{L}\mathbf{L}^T = \begin{bmatrix} t^2\mathbf{I}_d + \mathbf{R}_1\mathbf{R}_1^T & \mathbf{0} & \mathbf{R}_1\mathbf{R}_2^T & \mathbf{R}_1 \\ \mathbf{0} & t^2\mathbf{I}_{d(\ell-1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_2\mathbf{R}_1^T & \mathbf{0} & \mathbf{R}_2\mathbf{R}_2^T & \mathbf{R}_2 \\ \mathbf{R}_1^T & \mathbf{0} & \mathbf{R}_2^T & \mathbf{I}_{d(k-\ell)} \end{bmatrix}$$

We aim for $\mathbf{S} = \mathrm{diag}(s_1^2, s_2^2, s_3^2, s_4^2)$. We expect to need

$$s_1 = O(s_z(t + \|\mathbf{R}_1\|_2)), \quad s_2 = O(s_z t), \quad s_3 = O(s_z\|\mathbf{R}_2\|_2) \quad \text{and} \quad s_4 = O(s_z).$$

Tailor the Perturbation: Elliptic Gaussians

We need to compensate the covariance $s_z^2\mathbf{L}\mathbf{L}^T$

$$\mathbf{L}\mathbf{L}^T = \begin{bmatrix} t^2\mathbf{I}_d + \mathbf{R}_1\mathbf{R}_1^T & \mathbf{0} & \mathbf{R}_1\mathbf{R}_2^T & \mathbf{R}_1 \\ \mathbf{0} & t^2\mathbf{I}_{d(\ell-1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_2\mathbf{R}_1^T & \mathbf{0} & \mathbf{R}_2\mathbf{R}_2^T & \mathbf{R}_2 \\ \mathbf{R}_1^T & \mathbf{0} & \mathbf{R}_2^T & \mathbf{I}_{d(k-\ell)} \end{bmatrix}$$

We aim for $\mathbf{S} = \mathrm{diag}(s_1^2, s_2^2, s_3^2, s_4^2)$. We expect to need

$$s_1 = O(s_z(t + \|\mathbf{R}_1\|_2)), \quad s_2 = O(s_z t), \quad s_3 = O(s_z\|\mathbf{R}_2\|_2) \quad \text{and} \quad s_4 = O(s_z).$$

✓ We get $s_1 = \alpha\sqrt{t^2 + 3\|\mathbf{R}_1\|_2^2}$, $s_2 = \alpha t$, $s_3 = \alpha\sqrt{3}\|\mathbf{R}_2\|_2$ and $s_4 = \alpha\sqrt{3}$ are sufficient, with $\alpha = s_z^2/\sqrt{s_z^2 - \eta_\varepsilon(\mathbb{Z}^{dk})^2} \approx s_z$.

Can be adapted to general tags $\mathbf{T}$ *(invertible* $d \times d$ *matrices). Relevant quantity becomes* $\|\mathbf{T}\|_2$ *in the expressions of the* s_i.

Our Truncated Sampler

We then take $\mathbf{A}_t = [\mathbf{A}'|t\mathbf{G}_H - \mathbf{A}'\mathbf{R}]$ and

$$\mathbf{S} = \begin{bmatrix} s_1^2\mathbf{I}_d & & & \\ & s_2^2\mathbf{I}_{d(\ell-1)} & & \\ & & s_3^2\mathbf{I}_d & \\ & & & s_4^2\mathbf{I}_{d(k-\ell)} \end{bmatrix}$$

- $\mathbf{p} \hookleftarrow \mathcal{D}_{\mathbb{Z}^{d(k+1)},\sqrt{\mathbf{S}_p}}$ $\qquad \mathbf{S}_p = \mathbf{S} - s_z^2\mathbf{L}\mathbf{L}^T$
- $\mathbf{w} \leftarrow t^{-1}(\mathbf{u} - \mathbf{A}_t\mathbf{K}\mathbf{p}) \bmod q$
- $\mathbf{z} \hookleftarrow \mathcal{D}_{\mathcal{L}_q^{\mathbf{w}}(\mathbf{G}),s_z}$ $\qquad$ verifies $\mathbf{G}\mathbf{z} = \mathbf{w} \bmod q$
- $\mathbf{v}' \leftarrow \mathbf{p} + \mathbf{L}\mathbf{z}$
- Output $\mathbf{v} = \mathbf{K}\mathbf{v}'$ $\qquad$ verifies $\mathbf{A}_t\mathbf{v} = \mathbf{u} \bmod q$

Truncated Sampler

Can be adapted to general tags $\mathbf{T}$ *(invertible* $d \times d$ *matrices).*

Our Truncated Sampler

We then take $\mathbf{A}_t = [\mathbf{A}'|t\mathbf{G}_H - \mathbf{A}'\mathbf{R}]$ and

$$\mathbf{S} = \begin{bmatrix} s_1^2\mathbf{I}_d & & & \\ & s_2^2\mathbf{I}_{d(\ell-1)} & & \\ & & s_3^2\mathbf{I}_d & \\ & & & s_4^2\mathbf{I}_{d(k-\ell)} \end{bmatrix}$$

Truncated Sampler

- $\mathbf{p} \hookleftarrow \mathcal{D}_{\mathbb{Z}^{d(k+1)},\sqrt{\mathbf{S}_p}}$ $\qquad \mathbf{S}_p = \mathbf{S} - s_z^2\mathbf{L}\mathbf{L}^T$
- $\mathbf{w} \leftarrow t^{-1}(\mathbf{u} - \mathbf{A}_t\mathbf{K}\mathbf{p}) \bmod q$
- $\mathbf{z} \hookleftarrow \mathcal{D}_{\mathcal{L}_q^{\mathbf{w}}(\mathbf{G}),s_z}$ $\qquad$ verifies $\mathbf{G}\mathbf{z} = \mathbf{w} \bmod q$
- $\mathbf{v}' \leftarrow \mathbf{p} + \mathbf{L}\mathbf{z}$
- Output $\mathbf{v} = \mathbf{K}\mathbf{v}'$ $\qquad$ verifies $\mathbf{A}_t\mathbf{v} = \mathbf{u} \bmod q$

Let us zoom in on the **perturbation sampler**

Can be adapted to general tags $\mathbf{T}$ *(invertible* $d \times d$ *matrices).*

Online-Offline Perturbation Sampling Material

Perturbation sampling is the most time-consuming. Let's optimize with precomputations.

$$\mathbf{S}_p = \begin{bmatrix} s_1^2\mathbf{I} - s_z^2(tt^*\mathbf{I}_d + \mathbf{R}_1\mathbf{R}_1^*) & \mathbf{0} & -s_z^2\mathbf{R}_1\mathbf{R}_2^* & -s_z^2\mathbf{R}_1 \\ \mathbf{0} & s_2^2\mathbf{I} - s_z^2 tt^*\mathbf{I}_{d(\ell-1)} & \mathbf{0} & \mathbf{0} \\ -s_z^2\mathbf{R}_2\mathbf{R}_1^* & \mathbf{0} & s_3^2\mathbf{I} - s_z^2\mathbf{R}_2\mathbf{R}_2^* & -s_z^2\mathbf{R}_2 \\ -s_z^2\mathbf{R}_1^* & \mathbf{0} & -s_z^2\mathbf{R}_2^* & s_4^2\mathbf{I} - s_z^2\mathbf{I}_{d(k-\ell)} \end{bmatrix}$$

Online-Offline Perturbation Sampling Material

Perturbation sampling is the most time-consuming. Let's optimize with precomputations.

$$\mathbf{S}_p = \begin{bmatrix} s_1^2\mathbf{I} - s_z^2(tt^*\mathbf{I}_d + \mathbf{R}_1\mathbf{R}_1^*) & \mathbf{0} & -s_z^2\mathbf{R}_1\mathbf{R}_2^* & -s_z^2\mathbf{R}_1 \\ \mathbf{0} & s_2^2\mathbf{I} - s_z^2 tt^*\mathbf{I}_{d(\ell-1)} & \mathbf{0} & \mathbf{0} \\ -s_z^2\mathbf{R}_2\mathbf{R}_1^* & \mathbf{0} & s_3^2\mathbf{I} - s_z^2\mathbf{R}_2\mathbf{R}_2^* & -s_z^2\mathbf{R}_2 \\ -s_z^2\mathbf{R}_1^* & \mathbf{0} & -s_z^2\mathbf{R}_2^* & s_4^2\mathbf{I} - s_z^2\mathbf{I}_{d(k-\ell)} \end{bmatrix}$$

① Part in s_2^2 can be independently sampled (no precomputation needed)

Online-Offline Perturbation Sampling Material

Perturbation sampling is the most time-consuming. Let's optimize with precomputations.

$$\mathbf{S}_p = \begin{bmatrix} s_1^2 \mathbf{I} - s_z^2 (tt^* \mathbf{I}_d + \mathbf{R}_1 \mathbf{R}_1^*) & \mathbf{0} & -s_z^2 \mathbf{R}_1 \mathbf{R}_2^* & -s_z^2 \mathbf{R}_1 \\ \mathbf{0} & s_2^2 \mathbf{I} - s_z^2 tt^* \mathbf{I}_{d(\ell-1)} & \mathbf{0} & \mathbf{0} \\ -s_z^2 \mathbf{R}_2 \mathbf{R}_1^* & \mathbf{0} & s_3^2 \mathbf{I} - s_z^2 \mathbf{R}_2 \mathbf{R}_2^* & -s_z^2 \mathbf{R}_2 \\ -s_z^2 \mathbf{R}_1^* & \mathbf{0} & -s_z^2 \mathbf{R}_2^* & s_4^2 \mathbf{I} - s_z^2 \mathbf{I}_{d(k-\ell)} \end{bmatrix}$$

① Part in s_2^2 can be independently sampled (no precomputation needed)

② Part in s_3^2 and s_4^2 independent of t. Sampling material precomputed at key generation

Online-Offline Perturbation Sampling Material

Perturbation sampling is the most time-consuming. Let's optimize with precomputations.

$$\mathbf{S}_p = \begin{bmatrix} s_1^2\mathbf{I} - s_z^2(tt^*\mathbf{I}_d + \mathbf{R}_1\mathbf{R}_1^*) & \mathbf{0} & -s_z^2\mathbf{R}_1\mathbf{R}_2^* & -s_z^2\mathbf{R}_1 \\ \mathbf{0} & s_2^2\mathbf{I} - s_z^2 tt^*\mathbf{I}_{d(\ell-1)} & \mathbf{0} & \mathbf{0} \\ -s_z^2\mathbf{R}_2\mathbf{R}_1^* & \mathbf{0} & s_3^2\mathbf{I} - s_z^2\mathbf{R}_2\mathbf{R}_2^* & -s_z^2\mathbf{R}_2 \\ -s_z^2\mathbf{R}_1^* & \mathbf{0} & -s_z^2\mathbf{R}_2^* & s_4^2\mathbf{I} - s_z^2\mathbf{I}_{d(k-\ell)} \end{bmatrix}$$

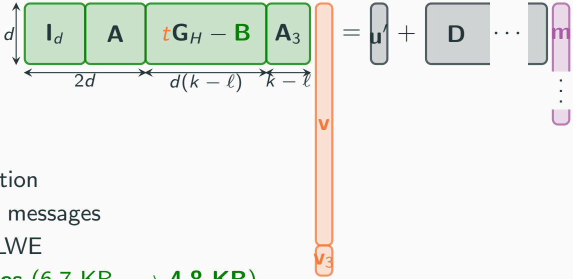
① Part in s_2^2 can be independently sampled (no precomputation needed)

② Part in s_3^2 and s_4^2 independent of t. Sampling material precomputed at key generation

③ Part in s_1^2 depends on t. Schur complements must be *computed online*. But only d dimensions out of $d(k+1)$

Signature in the Standard Model

🔑 : $\mathbf{R}_1, \mathbf{R}_2$ 🔑 : $\mathbf{B} = \mathbf{R}_1 + \mathbf{A}\mathbf{R}_2$ 🏅 : $t, \mathbf{v}, \mathbf{v}_3$ 📄 : $\mathbf{m}$ PP : $(\mathbf{A}, \mathbf{A}_3, \mathbf{D}, \mathbf{u}', \mathbf{G}_H = [b^\ell \mathbf{I} | \ldots | b^{k-1}\mathbf{I}])$



- ✓ Algebraic verification
- ✓ Handles arbitrary messages
- ✓ Security on SIS/LWE
- ✓ **Shorter signatures** (6.7 KB $\longrightarrow$ **4.8 KB**)
- ✓ **Smaller witness dimension:** $2d + k(d+1) \quad \longrightarrow \quad 2d + (k-\ell)(d+1)$

Signature in the Standard Model: Performance

For $k = 5$:

	$\lvert$pk$\rvert$	$\lvert$sig$\rvert$	Sec. (Core-SVP)
$\ell = 0$	47.5 KB	6.7 KB	126
$\ell = 1$	38.0 KB	5.9 KB	123
$\ell = 2$	28.5 KB	4.8 KB	121

Signature in the Standard Model: Performance

For $k = 5$:

	$\lvert\text{pk}\rvert$	$\lvert\text{sig}\rvert$	Sec. (Core-SVP)
$\ell = 0$	47.5 KB	6.7 KB	126
$\ell = 1$	38.0 KB	5.9 KB	123
$\ell = 2$	28.5 KB	4.8 KB	121

Procedure	Average Time ($\ell = 0$)	Average Time ($\ell = 2$)
`SamplePerturb`	52.0 ms	80.2 ms
`SampleGadget`	1.8 ms	1.8 ms
`SamplePre`	56.5 ms	83.9 ms
`Sign`	56.9 ms	84.3 ms
`Verify`	1.1 ms	0.7 ms

Small overhead due to online covariance computations

Timings from proof-of-concept implementation for comparison purposes. Absolute timings can be vastly improved with an optimized implementation

Applications for Privacy

Example improvements in **group signatures** [LNPS21][4] [LNP22][5], **anonymous credentials** [AGJ$^+$24][6], **blind signatures** [JS25][7]

	Original Size	**Ours**
Group Signature (gsig)	86.8 KB	**75.7 KB**
Anonymous Credentials (show)	60.8 KB	**54.0 KB**
Blind Signature (bsig)	41.1 KB	**36.3 KB**

(Full comparison in the paper (2024/1952), with different values of ℓ)

---

[4] Lyubashevsky, Nguyen, Plançon, Seiler. Shorter Lattice-Based Group Signatures via “Almost Free” Encryption and Other Optimizations. Asiacrypt 2021

[5] Lyubashevsky, Nguyen, Plançon. Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler and More General. Crypto 2022

[6] Argo, Güneysu, **Jeudy**, Land, Roux-Langlois, **Sanders**. Practical Post-Quantum Signatures for Privacy. CCS 2024

[7] **Jeudy**, **Sanders**. Improved Lattice Blind Signatures from Recycled Entropy. Crypto 2025

Wrapping Up

✓ **Preimage Sampler** with **Truncated Gadgets** in the **worst case**

- Unlocks truncated gadgets in their main applications
- Same structure: drop-in replacement to full gadget sampler [MP12]
- Reduced dimension: immediate improvement in many privacy-driven applications

? **Perspectives**

- More efficient perturbation sampler?
- Optimized implementation (dedicated backend, parallelization, parameter selection)

Wrapping Up

✓ **Preimage Sampler** with **Truncated Gadgets** in the **worst case**

- Unlocks truncated gadgets in their main applications
- Same structure: drop-in replacement to full gadget sampler [MP12]
- Reduced dimension: immediate improvement in many privacy-driven applications

? **Perspectives**

- More efficient perturbation sampler?
- Optimized implementation (dedicated backend, parallelization, parameter selection)

Thank You!

References i

- S. Argo, T. Güneysu, C. Jeudy, G. Land, A. Roux-Langlois, and O. Sanders.
  Practical Post-Quantum Signatures for Privacy.
  In CCS, 2024.
- Y. Chen, N. Genise, and P. Mukherjee.
  Approximate Trapdoors for Lattices and Smaller Hash-and-Sign Signatures.
  In ASIACRYPT, 2019.
- C. Jeudy and O. Sanders.
  Improved Lattice Blind Signatures from Recycled Entropy.
  In CRYPTO, 2025.
- V. Lyubashevsky, N. K. Nguyen, and M. Plançon.
  Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General.
  CRYPTO, 2022.

References ii

V. Lyubashevsky, N. K. Nguyen, M. Plançon, and G. Seiler.
Shorter Lattice-Based Group Signatures via "Almost Free" Encryption and Other Optimizations.
In ASIACRYPT, 2021.

D. Micciancio and C. Peikert.
Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller.
In EUROCRYPT, 2012.