Worst-Case Lattice Sampler with Truncated Gadgets and Applications

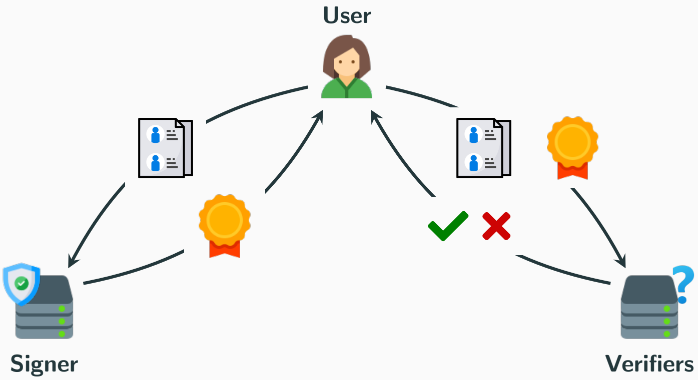
March 19th, 2025

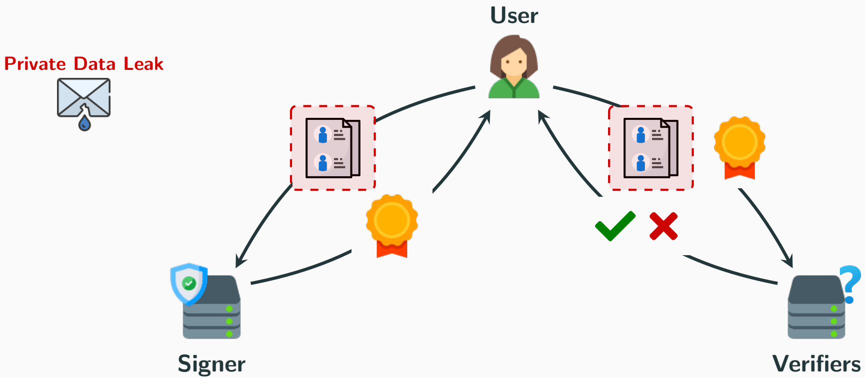
Corentin Jeudy

Orange, Applied Crypto Group

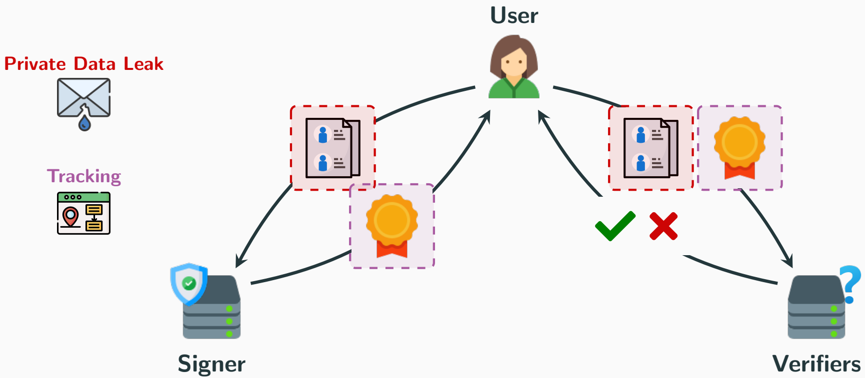
Joint work with Olivier Sanders

Digital Signatures





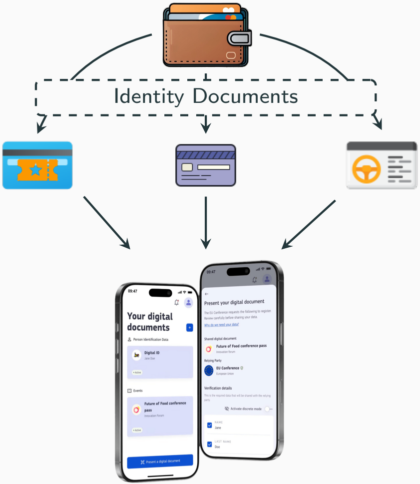
No control over the disclosed information: Verifiers (and attacker) learn everything



No control over the disclosed information: Verifiers (and attacker) learn everything
Traceable accross different authentications: Same signature allows tracing

European Digital Identity (EUDI) Wallet initiative

"a safe, reliable, and private means of digital identification for everyone in Europe."



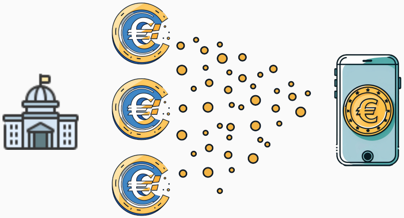
Emphasis on

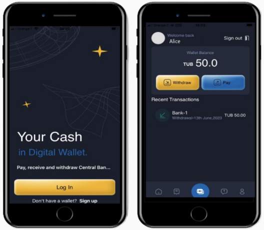
- **Anonymity**
- **Unlinkability**
- **Selective disclosure**

Anonymous Credentials

Digital Euro initiative (ECB) & **Project Tourbillon** (BIS & SNB)

"would not identify you or track your payments [...] for cash-like privacy"





Emphasis on

- ✓ **(Payer) Anonymity**
- ✓ **Unlinkability**
- ✓ **Scalability**

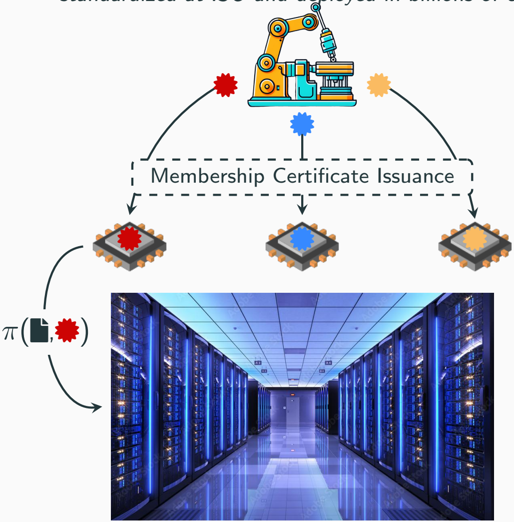
Anonymous Credentials

Blind Signature

eCash

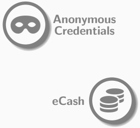
Group Attestation with **Built-in Revocation Mechanisms**

"standardized at ISO and deployed in billions of chips (TPM, Intel)"



Emphasis on

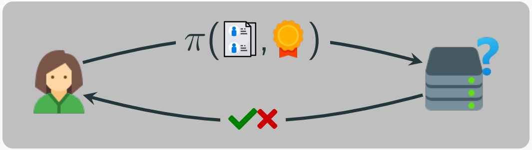
- **Anonymity**
- **Unlinkability**
- **Revocability**



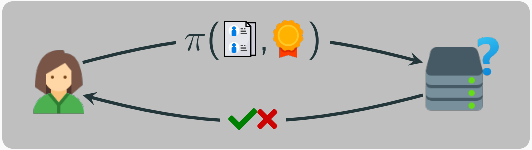
EPID
(group signatures)

Privacy from Zero-Knowledge Proofs

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



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



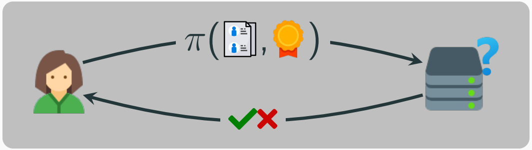
Proof of x s.t. $\mathbf{g}^{x} = \mathbf{h}$

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

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

Algebraic → **Generic**

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



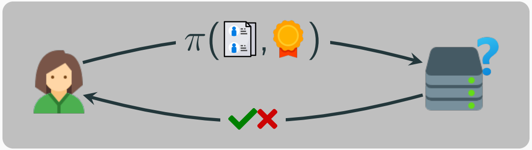
Proof of x s.t. $\mathbf{g}^{\mathsf{x}} = \mathbf{h}$

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

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

ECDSA/RSA

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



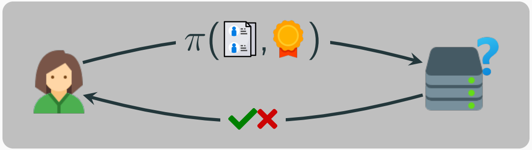
Proof of $\mathbf{x}$ s.t. $\mathbf{g}^{\mathbf{x}} = \mathbf{h}$	Proof of $\mathbf{x}$ s.t. $\mathbf{A}\mathbf{x} = \mathbf{u} \wedge \|\mathbf{x}\| \leq B$	Proof of $\mathbf{x}$ s.t. $\mathcal{H}(\mathbf{x}) = \mathbf{h}$

Algebraic → **Generic**

Classical Groups ⚠ | ECDSA/RSA ⚠

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. $\mathcal{H}(\mathbf{x}) = \mathbf{h}$

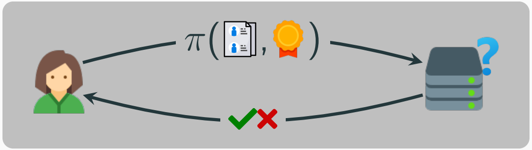
Algebraic → **Generic**

Classical Groups

ECDSA/RSA

Falcon, Dilithium

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



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

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

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

Algebraic → **Generic**

Classical Groups

ZK-friendly Lattice Signatures

ECDSA/RSA

Falcon, Dilithium

Low-Dimensional x

High-Dimensional x

Outline

Applications

3. Applications for Privacy
   Sizes & Timings

A New Sampler

2. Worst-Case Truncated
   Sampler via Projection

Motivation

1. ZK-Friendly Signatures
   from Gadget Samplers

Zero-Knowledge-Friendly Signatures from Gadget Samplers

$\mathsf{ISIS}_{m,d,q,\beta}$

Given $(\mathbf{A}, \mathbf{u}) \hookleftarrow U(R_q^{d \times m+1})$, find $\mathbf{x} \in R^m$ such that $\mathbf{A}\mathbf{x} = \mathbf{u} \bmod q$, $\|\mathbf{x}\| \leq \beta$.
When $\mathbf{u} = \mathbf{0}$, we ask $\mathbf{x} \neq \mathbf{0}$.

Decision: Distinguish $\mathbf{A}\mathbf{x} \bmod q$ for a random short $\mathbf{x}$ from a random $\mathbf{u}$ $\longrightarrow$ **LWE**.

Lattice Assumption and Trapdoors

$\mathsf{ISIS}_{m,d,q,\beta}$

Given $(\mathbf{A}, \mathbf{u}) \hookleftarrow U(R_q^{d\times m+1})$, find $\mathbf{x} \in R^m$ such that $\mathbf{A}\mathbf{x} = \mathbf{u} \bmod q$, $\|\mathbf{x}\| \leq \beta$.
When $\mathbf{u} = \mathbf{0}$, we ask $\mathbf{x} \neq \mathbf{0}$.

Decision: Distinguish $\mathbf{A}\mathbf{x} \bmod q$ for a random short $\mathbf{x}$ from a random $\mathbf{u}$ $\longrightarrow$ **LWE**.

ISIS is hard unless we know a trapdoor $\mathbf{R}$ on $\mathbf{A}$.

- Ability to invert $f_{\mathbf{A}} : \mathbf{x} \mapsto \mathbf{A}\mathbf{x} \bmod q$ over bounded domain
  - Ability to randomize preimage finding without leaking $\mathbf{R}$ → **Preimage Sampling**
    - Design secure signatures [GPV08][1]: Find short $\mathbf{x}$ such that $\mathbf{A}\mathbf{x} = \mathcal{H}(\mathbf{m}) \bmod q$

[1] Gentry, Peikert, Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. STOC 2008.

$\mathsf{ISIS}_{m,d,q,\beta}$

Given $(\mathbf{A}, \mathbf{u}) \hookleftarrow U(R_q^{d\times m+1})$, find $\mathbf{x} \in R^m$ such that $\mathbf{A}\mathbf{x} = \mathbf{u} \bmod q$, $\|\mathbf{x}\| \leq \beta$.
When $\mathbf{u} = \mathbf{0}$, we ask $\mathbf{x} \neq \mathbf{0}$.

Decision: Distinguish $\mathbf{A}\mathbf{x} \bmod q$ for a random short $\mathbf{x}$ from a random $\mathbf{u}$ $\longrightarrow$ **LWE**.

ISIS is hard unless we know a trapdoor $\mathbf{R}$ on $\mathbf{A}$.

- Ability to invert $f_{\mathbf{A}} : \mathbf{x} \mapsto \mathbf{A}\mathbf{x} \bmod q$ over bounded domain
  - Ability to randomize preimage finding without leaking $\mathbf{R}$ → **Preimage Sampling**

Gadget-based samplers [MP12][1] are well suited for signatures without ROM

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

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

$$\mathbf{A_T} = [\mathbf{A}'|\mathbf{TG} - \mathbf{A}'\mathbf{R}] \text{ and } \mathbf{A}' = [\mathbf{I}|\mathbf{A}]$$

verifies $\mathbf{A_T}\mathbf{L} = \mathbf{TG} \bmod q$, with $\mathbf{L} = \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix}$

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

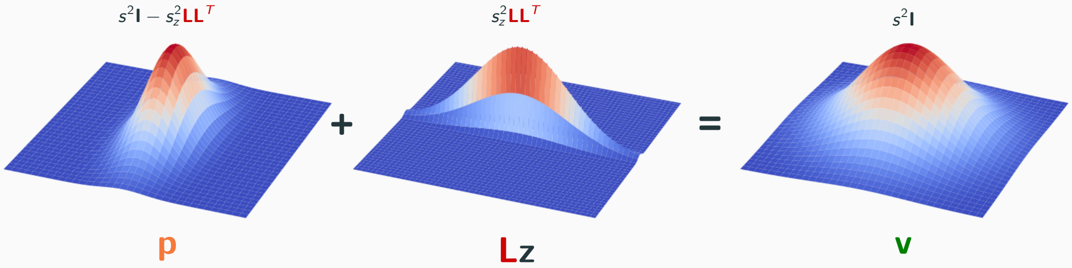
$\mathbf{R}$ $\mathbf{B} = \mathbf{A}'\mathbf{R}$
$\mathbf{T}$ $(= t\mathbf{I})$

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

- Collecting many preimages will leak $\mathbf{R}$...
- Distribution on $\mathbf{z}$ and add mask $\mathbf{p}$: preimages $\mathbf{v} = \mathbf{p} + \mathbf{Lz} = \begin{bmatrix} \mathbf{p}_1 + \mathbf{Rz} \\ \mathbf{p}_2 + \mathbf{z} \end{bmatrix}$

  (and syndrome correction so that $\mathbf{TGz} = \mathbf{w} = \mathbf{u} - \mathbf{A_T}\mathbf{p}$)

Compensate statistical leakage by adapting covariance of **p** [MP12]. Only for **z** and **p** Gaussian

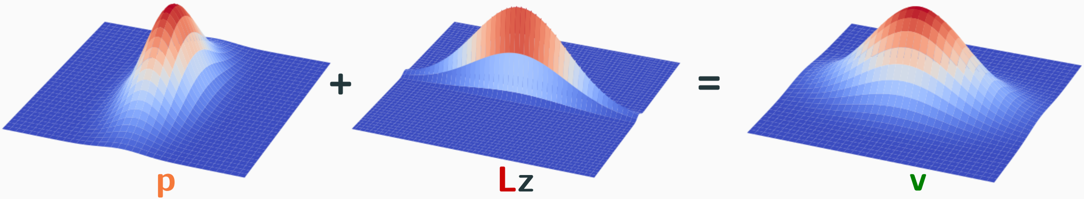


Quality: $s \approx s_z\sqrt{1+\|\mathbf{R}\|_2^2}$ with $s_z \approx \eta_\varepsilon(\mathcal{L}_q^\perp(\mathbf{G}))$.

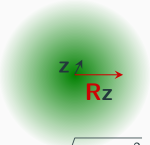
How to Choose the Mask? Elliptical Convolution

Use **elliptical Gaussians** instead of spherical

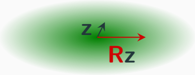
$\begin{bmatrix} s_1^2\mathbf{I} & \\ & s_2^2\mathbf{I} \end{bmatrix} - s_z^2\mathbf{L}\mathbf{L}^T$ $\quad$ $s_z^2\mathbf{L}\mathbf{L}^T$ $\quad$ $\begin{bmatrix} s_1^2\mathbf{I} & \\ & s_2^2\mathbf{I} \end{bmatrix}$



Spherical Sampling	Elliptical Sampling



$$\mathbf{v} = \mathbf{p} + \begin{bmatrix} \mathbf{Rz} \\ \mathbf{z} \end{bmatrix}$$



$s \approx s_z\sqrt{1 + \|\mathbf{R}\|_2^2}$

$s_1 \approx \sqrt{2}s_z\|\mathbf{R}\|_2, \quad s_2 \approx \sqrt{2}s_z$

The Original Gadget Sampler

We let $s_z \approx \eta_\varepsilon(\mathcal{L}_q^\perp(\mathbf{G}))$, $s_1 \approx \sqrt{2}s_z\|\mathbf{R}\|_2$, $s_2 \approx \sqrt{2}s_z$ and define

$$\mathbf{S}_p = \begin{bmatrix} s_1^2\mathbf{I}_{2d} & \\ & s_2^2\mathbf{I}_{dk} \end{bmatrix} - s_z^2 \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_{dk} \end{bmatrix} \begin{bmatrix} \mathbf{R}^T & \mathbf{I}_{dk} \end{bmatrix}$$

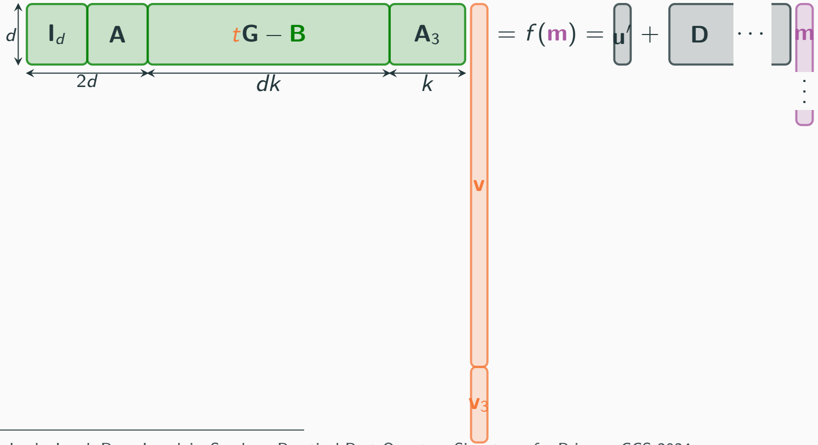
The sampler finds a preimage of $\mathbf{u}$ for $\mathbf{A_T} = [\mathbf{A}'|\mathbf{TG} - \mathbf{A}'\mathbf{R}]$

- $\mathbf{p} \hookleftarrow \mathcal{D}_{\mathbb{Z}^{d(2+k)},\sqrt{\mathbf{S}_p}}$
- $\mathbf{w} \leftarrow \mathbf{T}^{-1}(\mathbf{u} - \mathbf{A_T}\mathbf{p}) \bmod q$
- $\mathbf{z} \hookleftarrow \mathcal{D}_{\mathcal{L}_q^{\mathbf{w}}(\mathbf{G}),s_z}$ — verifies $\mathbf{Gz} = \mathbf{w} \bmod q$
- $\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} \leftarrow \mathbf{p} + \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \mathbf{z}$ — $= \mathbf{p} + \mathbf{Lz}$
- Output $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2)$ — verifies $\mathbf{A_T}\mathbf{v} = \mathbf{u} \bmod q$

MP Sampler

Signature scheme from [AGJ⁺24][2]:

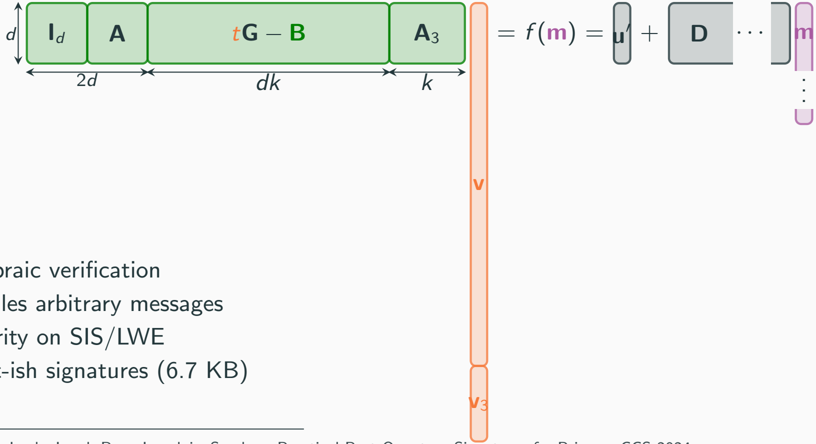
: $\mathbf{R}$: $\mathbf{B} = [\mathbf{I}_d|\mathbf{A}]\mathbf{R}$: $t, \mathbf{v}, \mathbf{v}_3$: $\mathbf{m}$ PP : $(\mathbf{A}, \mathbf{A}_3, \mathbf{D}, \mathbf{u}', \mathbf{G} = [b^0\mathbf{I}| \dots |b^{k-1}\mathbf{I}])$



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

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 : $(\mathbf{A}, \mathbf{A}_3, \mathbf{D}, \mathbf{u}', \mathbf{G} = [b^0\mathbf{I}| \ldots |b^{k-1}\mathbf{I}])$

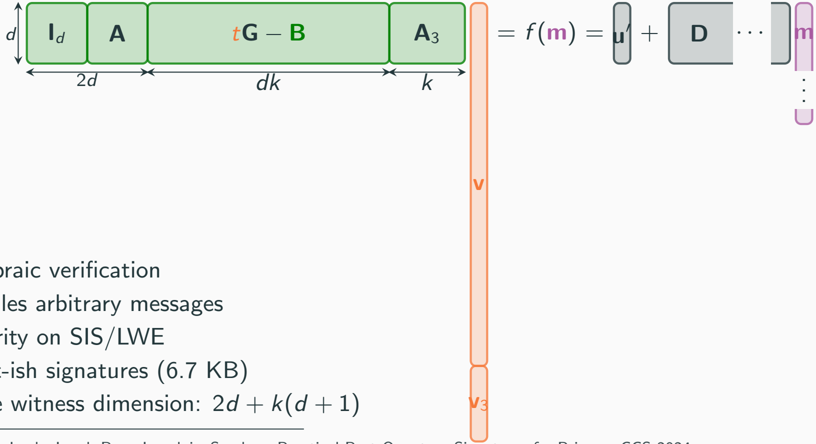


- Algebraic verification
- Handles arbitrary messages
- Security on SIS/LWE
- Short-ish signatures (6.7 KB)

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

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 : $(\mathbf{A}, \mathbf{A}_3, \mathbf{D}, \mathbf{u}', \mathbf{G} = [b^0\mathbf{I}|\ldots|b^{k-1}\mathbf{I}])$



- ✓ 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

Worst-Case Sampler with Truncated Gadgets via Projection

Reduce Dimension with Approximate Trapdoor

Reduce gadget dimension with "approximate trapdoors" [CGM19][3]: Sampling $\mathbf{v}'$ s.t. $\mathbf{A}'_{\mathsf{T}}\mathbf{v}' + \mathbf{e} = \mathbf{u}$ with $\mathbf{e}$ small is sufficient.

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

Reduce gadget dimension with "approximate trapdoors" [CGM19][3]: Sampling $\mathbf{v}'$ s.t. $\mathbf{A}'_{\mathbf{T}}\mathbf{v}' + \mathbf{e} = \mathbf{u}$ with $\mathbf{e}$ small is sufficient.

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: $\mathbf{A}'_{\mathbf{T}} = [\mathbf{A}'|\mathbf{T}\mathbf{G}_H - \mathbf{A}'\mathbf{R}]$, with $\mathbf{A}' = [\mathbf{I}_d|\mathbf{A}]$.

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

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

Reduce gadget dimension with "approximate trapdoors" [CGM19][3]: Sampling $\mathbf{v}'$ s.t. $\mathbf{A}'_{\mathbf{T}}\mathbf{v}' + \mathbf{e} = \mathbf{u}$ with $\mathbf{e}$ small is sufficient.

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: $\mathbf{A}'_{\mathbf{T}} = [\mathbf{A}'|\mathbf{T}\mathbf{G}_H - \mathbf{A}'\mathbf{R}]$, with $\mathbf{A}' = [\mathbf{I}_d|\mathbf{A}]$.

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

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

[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]: Sampling $\mathbf{v}'$ s.t. $\mathbf{A}'_{\mathbf{T}}\mathbf{v}' + \mathbf{e} = \mathbf{u}$ with $\mathbf{e}$ small is sufficient.

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: $\mathbf{A}'_{\mathbf{T}} = [\mathbf{A}'|\mathbf{T}\mathbf{G}_H - \mathbf{A}'\mathbf{R}]$, with $\mathbf{A}' = [\mathbf{I}_d|\mathbf{A}]$.

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

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

? 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?

? Reduced dimension, but what about security?

What About Security?

? Reduced dimension, but what about security? **Well, it's complicated.**

✗ 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:

1. Distribution of $\mathbf{e}$ difficult when $\mathbf{u}$ is arbitrary/adversarially chosen
2. Distribution of $\mathbf{e}$ depends on tag $\mathbf{T}$, which must stay hidden

What About Security?

? Reduced dimension, but what about security? **Well, it's complicated.**

✗ 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:

1. Distribution of $\mathbf{e}$ difficult when $\mathbf{u}$ is arbitrary/adversarially chosen
2. Distribution of $\mathbf{e}$ depends on tag $\mathbf{T}$, which must stay hidden

≈ 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?

? Reduced dimension, but what about security? **Well, it's complicated.**

✗ 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:

1. Distribution of $\mathbf{e}$ difficult when $\mathbf{u}$ is arbitrary/adversarially chosen
2. Distribution of $\mathbf{e}$ depends on tag $\mathbf{T}$, which must stay hidden

≈ 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. $f(\mathbf{m}) = \mathbf{u}' + \mathbf{D}\mathbf{m}$).

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

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

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

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

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

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

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

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

factor

Public Part ($\mathbf{K}$-projection) · **Private Part** ($\mathbf{L}$ full rank)

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

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

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

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

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

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

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

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

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

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

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

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

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

Our Truncated Sampler

We then take $\mathbf{A_T} = [\mathbf{A}'|\mathbf{T}\mathbf{G}_H - \mathbf{A}'\mathbf{R}]$ 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

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

We then take $\mathbf{A_T} = [\mathbf{A}'|\mathbf{TG}_H - \mathbf{A}'\mathbf{R}]$ 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

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

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

Perturbation sampling represents the vast majority of the computation time. Let's optimize with precomputations. Take $\mathbf{T} = t\mathbf{I}_d$ with t invertible modulo q.

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

Perturbation sampling represents the vast majority of the computation time. Let's optimize with precomputations. Take $\mathbf{T} = t\mathbf{I}_d$ with t invertible modulo q.

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

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

Perturbation sampling represents the vast majority of the computation time. Let's optimize with precomputations. Take $\mathbf{T} = t\mathbf{I}_d$ with t invertible modulo q.

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

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. Precomputation done at key generation

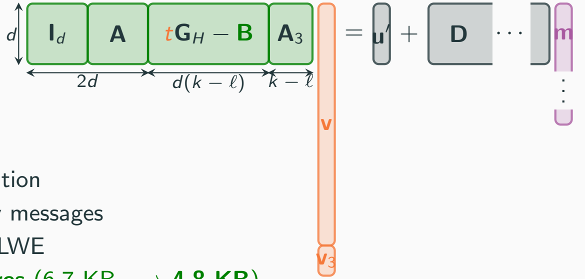
Perturbation sampling represents the vast majority of the computation time. Let's optimize with precomputations. Take $\mathbf{T} = t\mathbf{I}_d$ with t invertible modulo q.

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

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

Applications:
(More) Practical Post-Quantum Privacy

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



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

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 $k = 5$:

	$\|pk\|$	$\|sig\|$	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

Example improvements in **group signatures** [LNPS21][4] [LNP22][5], **anonymous credentials** [AGJ$^+$24][6], **blind signatures** [JS24][7]

	Improvement	Final Size
Group Signature	15.7 %	$\lvert \mathsf{gsig} \rvert = 75.7$ KB
Anonymous Credentials	11.2 %	$\lvert \mathsf{show} \rvert = 54.0$ KB
Blind Signature	11.8 %	$\lvert \mathsf{bsig} \rvert = 36.3$ 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. ePrint 2024/1289

Conclusion and Directions

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)

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!

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. Gentry, C. Peikert, and V. Vaikuntanathan.
Trapdoors for Hard Lattices and New Cryptographic Constructions.
In STOC, 2008.

C. Jeudy and O. Sanders.
Improved Lattice Blind Signatures from Recycled Entropy.
IACR Cryptol. ePrint Arch., page 1289, 2024.

V. Lyubashevsky, N. K. Nguyen, and M. Plançon.
Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General.
CRYPTO, 2022.

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.