Rejection-free Framework of Zero-Knowledge Proof based on (Hint-)Module Learning with Errors

Antoine Douteau, Adeline Roux-Langlois

CNRS, GREYC, Caen

Indocrypt 2025

December 14-17th

GREYC
Laboratoire de recherche en sciences du numérique

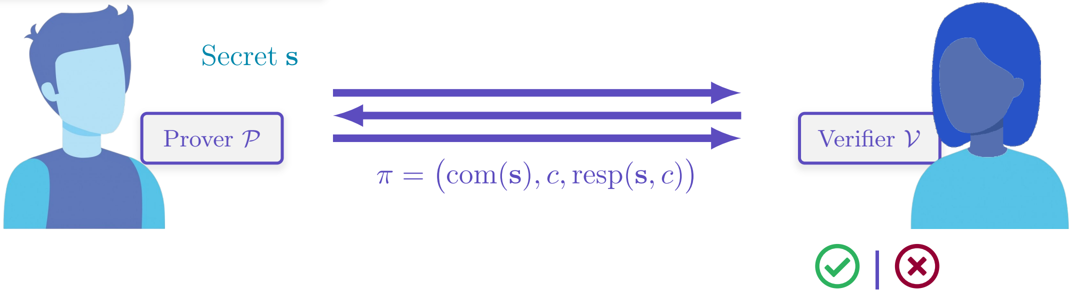
cnrs

RÉGION NORMANDIE

Proving without revealing

Zero-Knowledge

The context is public



Completeness

$\mathcal{V}$ accepts π if $\mathbf{s}$ is valid

(Know.) Soundness

If $\mathcal{V}$ accepts π then $\mathbf{s}$ is valid

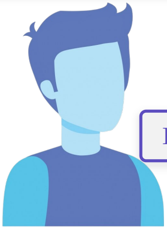
Zero-Knowledge

$\mathcal{V}$ does not gain info. with π

Proving without revealing

Commit-and-prove

The context is public

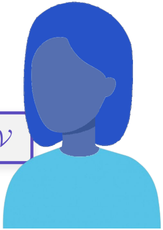


Secret $\mathbf{s}$

Prover $\mathcal{P}$

$$\pi = (\mathrm{com}(\mathbf{s}), c, \mathrm{resp}(\mathbf{s}, c)) + (\mathrm{additional\_material})$$

Verifier $\mathcal{V}$



✓ | ✗ $\wedge$ ✓ | ✗

Completeness

$\mathcal{V}$ accepts π if $\mathbf{s}$ is valid

(Know.) Soundness

If $\mathcal{V}$ accepts π then $\mathbf{s}$ is valid

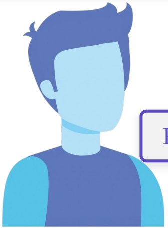
Zero-Knowledge

$\mathcal{V}$ does not gain info. with π

An idea of a lattice based ZK proof

Knowledge of an Ajtai opening

$\mathbf{A}$



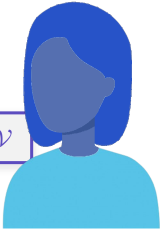
Smalls $\mathbf{s}$, $\mathbf{y}$

Prover $\mathcal{P}$

$\mathbf{t} = \mathbf{A}\mathbf{s}, \mathbf{w} = \mathbf{A}\mathbf{y}$ →

← small c

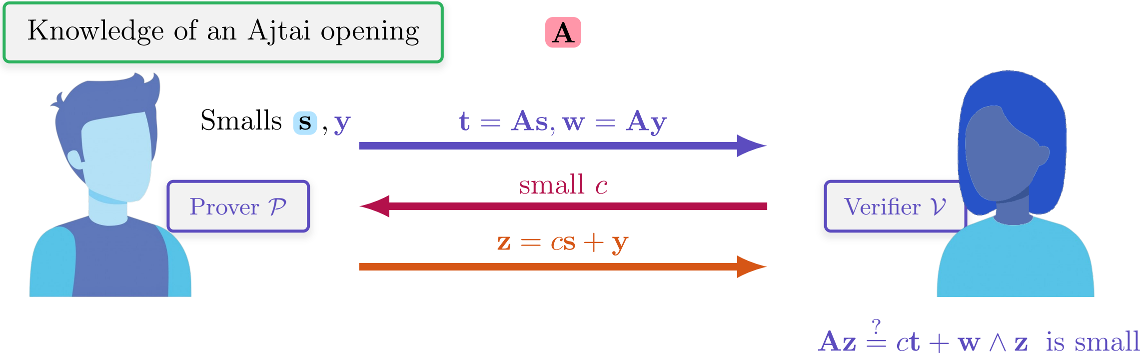
$\mathbf{z} = c\mathbf{s} + \mathbf{y}$ →



Verifier $\mathcal{V}$

$\mathbf{A}\mathbf{z} \stackrel{?}{=} c\mathbf{t} + \mathbf{w} \wedge \mathbf{z}$ is small

An idea of a lattice based ZK proof



The protocol is not zero-knowledge

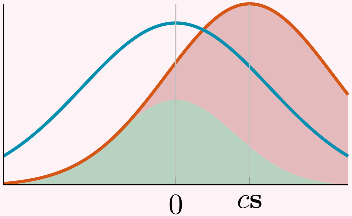
If c is small then it will more likely leak information about cs and therefore s.

Transmitting without revealing

Rejection Sampling [Lyu09]:

GOAL: Artificially controlling output distribution of the responses $\mathrm{resp}(\mathbf{s}, c)$.

1. **Prover** aborts and restarts the entire protocol when $\mathrm{resp}(\mathbf{s}, c)$ leaks info about $\mathbf{s}$
2. **Verifier** sees a publicly known distribution from its point of view.



Accept
Reject and **resample,**
Scaled distribution.

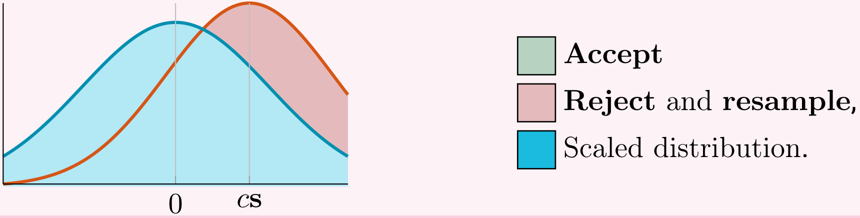
[Lyu09] Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures.; Lyubashevsky, 2009

Transmitting without revealing

Rejection Sampling [Lyu09]:

GOAL: Artificially controlling output distribution of the responses $\text{resp}(\mathbf{s}, c)$.

1. **Prover** aborts and restarts the entire protocol when $\text{resp}(\mathbf{s}, c)$ leaks info about $\mathbf{s}$
2. **Verifier** sees a publicly known distribution from its point of view.



[Lyu09] Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures.; Lyubashevsky, 2009

Transmitting without revealing

Rejection Sampling [Lyu09]:

GOAL: Artificially controlling output distribution of the responses $\mathrm{resp}(\mathbf{s}, c)$.

1. **Prover** aborts and restarts the entire protocol when $\mathrm{resp}(\mathbf{s}, c)$ leaks info about $\mathbf{s}$
2. **Verifier** sees a publicly known distribution from its point of view.

Problems

1. Non constant time as the prover can't predict the rejection,
2. Must involve costly masking to protect it from side-channel attacks.

[Lyu09] Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures.; Lyubashevsky, 2009

Contribution

Commit-and-prove

Zero-Knowledge

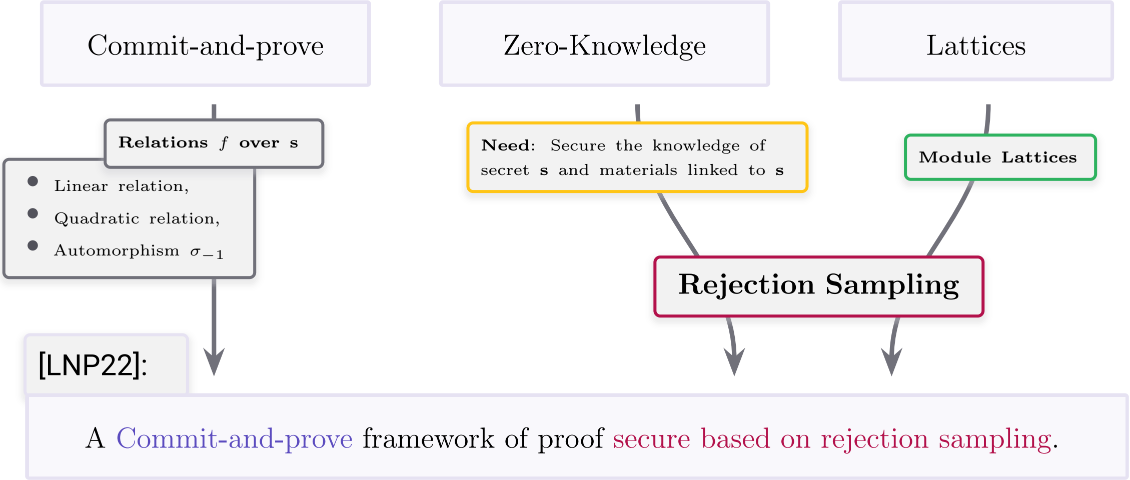
Lattices

[LNP22]:

A Commit-and-prove framework of proof secure based on rejection sampling.

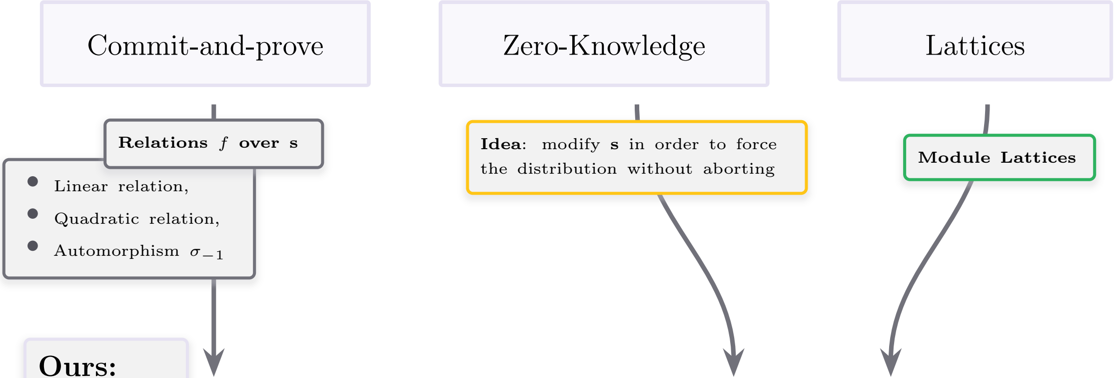
[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

Contribution



[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

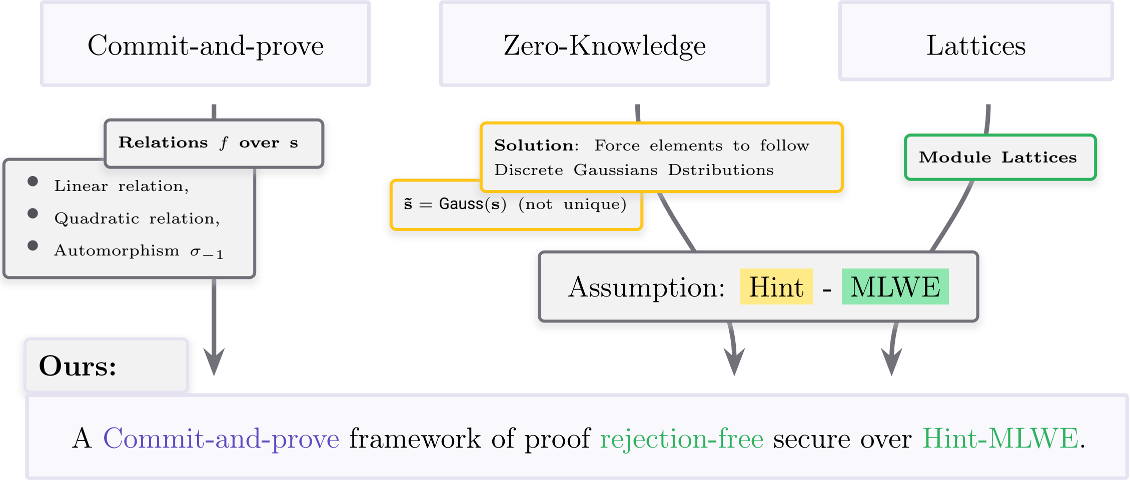
Contribution



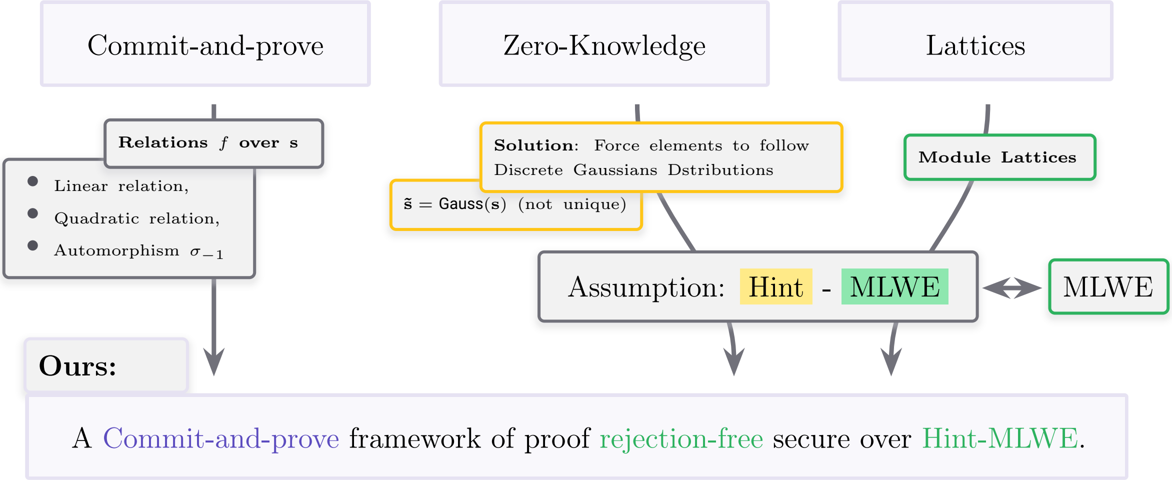
Ours:

A Commit-and-prove framework of proof rejection-free secure over Hint-MLWE.

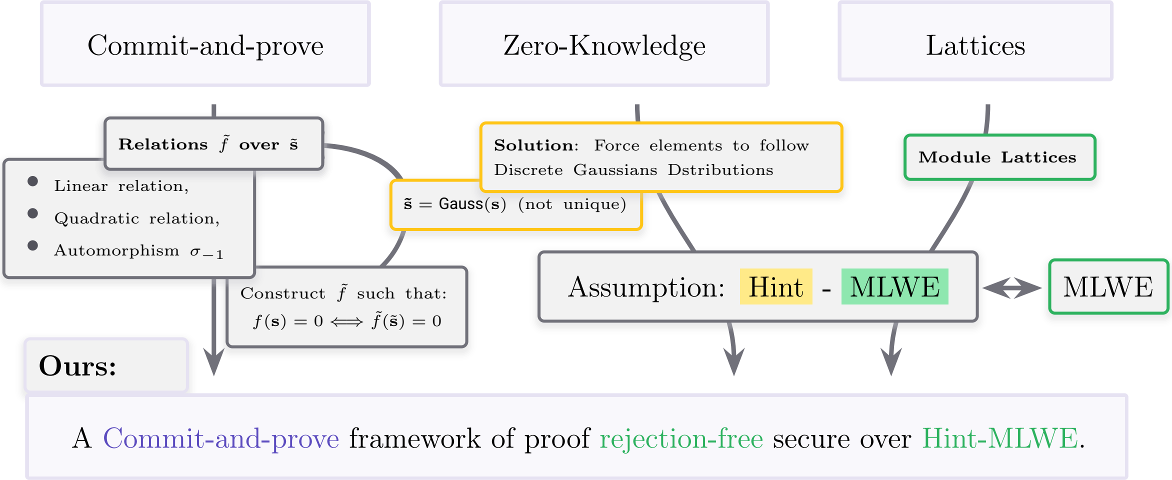
Contribution



Contribution



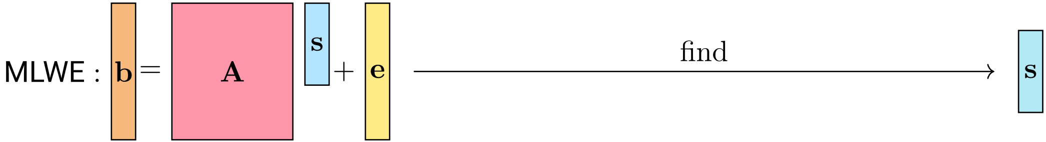
Contribution



Lattice assumption

Module Learning With Errors (MLWE) [Reg05,LS15]:

Consider $\mathcal{R} = \mathbb{Z}[X]/\langle X^d + 1 \rangle$ for d a power of two, $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ with q an integer.



$\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}^{m \times n})$

$\mathbf{b} \in \mathcal{R}^m$

Public matrices and vectors

$\mathbf{s} \leftarrow \mathcal{D}_{\mathcal{R}^n, \sigma_{\mathbf{s}}}$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}^m, \sigma_{\mathbf{e}}}$

Discrete Gaussians

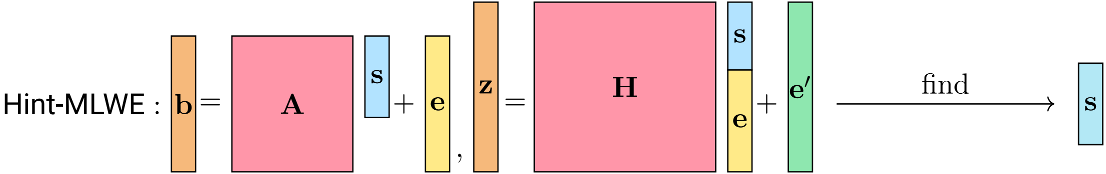
[Reg05] On lattices, Learning With Errors, random linear codes, and cryptography; Regev, 2005

[LS15] Worst-Case to Average-Case Reductions for Module Lattices; Langlois and Stehlé, 2015

Lattice assumption

Module Learning With Errors (MLWE) with hints about the secret/error: Hint-MLWE [KLSS23]

Consider $\mathcal{R} = \mathbb{Z}[X]/\langle X^d + 1\rangle$ for d a power of two, $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ with q an integer.



$\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}^{m\times n})$, $\mathbf{H} \in \mathcal{R}^{(n+m)\times(n+m)}$, $\mathbf{s} \leftarrow \mathcal{D}_{\mathcal{R}^n,\sigma_{\mathbf{s}}}$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}^m,\sigma_{\mathbf{e}}}$, $\mathbf{e}' \leftarrow \mathcal{D}_{\mathcal{R}^{(n+m)},\sigma_{\mathbf{e}'}}$

$\mathbf{b} \in \mathcal{R}^m$, $\mathbf{z} \in \mathcal{R}^{n+m}$

Discrete Gaussians

Public matrices and vectors

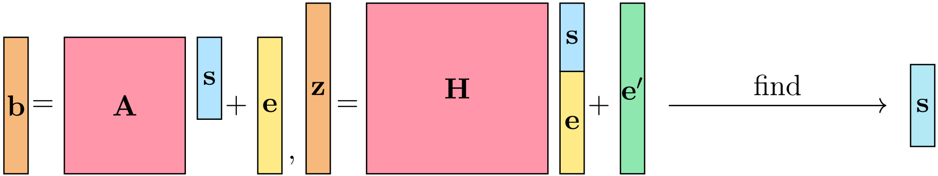
[KLSS23] Toward Practical Lattice-Based Proof of Knowledge from Hint-MLWE; Kim et al., 2023

Lattice assumption

Module Learning With Errors (MLWE) with hints about the secret/error: Hint-MLWE [KLSS23]

Consider $\mathcal{R} = \mathbb{Z}[X]/\langle X^d + 1\rangle$ for d a power of two, $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ with q an integer.

Hint-MLWE :



$\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}^{m\times n})$, $\mathbf{H} \in \mathcal{R}^{(n+m)\times(n+m)}$, $\mathbf{s} \leftarrow \mathcal{D}_{\mathcal{R}^n,\sigma_{\mathbf{s}}}$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}^m,\sigma_{\mathbf{e}}}$, $\mathbf{e}' \leftarrow \mathcal{D}_{\mathcal{R}^{(n+m)},\sigma_{\mathbf{e}'}}$

$\mathbf{b} \in \mathcal{R}^m$, $\mathbf{z} \in \mathcal{R}^{n+m}$

Discrete Gaussians

Public matrices and vectors

MLWE $\Longrightarrow$ Hint-MLWE [KLSS23], when secrets and errors follow Discrete Gaussians.

[KLSS23] Toward Practical Lattice-Based Proof of Knowledge from Hint-MLWE; Kim et al., 2023

Our modification of the framework

In ours: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- Use $\tilde{\mathbf{z}}$ as hint in Hint-MLWE ensuring that it is built on a Gaussian element.

In [LNP22]: $\mathbf{z} := c\,\mathbf{s} + \mathbf{y}$

- Use rejection sampling to ensure zero-knowledge.

[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

Our modification of the framework

In ours: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- Use $\tilde{\mathbf{z}}$ as hint in Hint-MLWE ensuring that it is built on a Gaussian element.
- Use the Gaussianised $\tilde{\mathbf{s}} = \mathsf{Gauss}(\mathbf{s})$ (such that $\mathbf{G}_b\tilde{\mathbf{s}} = \mathbf{s}$) using [HSS24].

In [LNP22]: $\mathbf{z} := c\,\mathbf{s} + \mathbf{y}$

- Use rejection sampling to ensure zero-knowledge.

[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the framework

In ours: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- Use $\tilde{\mathbf{z}}$ as hint in Hint-MLWE ensuring that it is built on a Gaussian element.
- Use the Gaussianised $\tilde{\mathbf{s}} = \mathsf{Gauss}(\mathbf{s})$ (such that $\mathbf{G}_b\tilde{\mathbf{s}} = \mathbf{s}$) using [HSS24].

$$\mathbf{s} := \begin{bmatrix} 1 & b & \cdots & b^k & & & & \mathbf{0} \\ & & & & \ddots & & & \\ & & \mathbf{0} & & & 1 & b & \cdots & b^k \end{bmatrix} \tilde{\mathbf{s}} \text{ with } \tilde{\mathbf{s}} = \begin{bmatrix} \tilde{\mathbf{s}}_0 \\ \vdots \\ \tilde{\mathbf{s}}_k \end{bmatrix}$$

$$=: \mathbf{G}_b \text{ with } k = \lfloor \log_b q \rfloor - 1$$

[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the framework

In ours: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- Use $\tilde{\mathbf{z}}$ as hint in Hint-MLWE ensuring that it is built on a Gaussian element.
- Use the Gaussianised $\tilde{\mathbf{s}} = \mathsf{Gauss}(\mathbf{s})$ (such that $\mathbf{G}_b\tilde{\mathbf{s}} = \mathbf{s}$) using [HSS24].

Take a prime integer $q = b^2 + 1$, e.g. we take $17 = 4^2 + 1$.
Let $\mathbf{s} = \begin{bmatrix} 1 & 7 & -1 \end{bmatrix}$ decomposed in base 4 as $\hat{\mathbf{s}} = \begin{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} & \begin{bmatrix} 3 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 4 \end{bmatrix} \end{bmatrix}$ (L to R).

[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the framework

In ours: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- Use $\tilde{\mathbf{z}}$ as hint in Hint-MLWE ensuring that it is built on a Gaussian element.
- Use the Gaussianised $\tilde{\mathbf{s}} = \mathsf{Gauss}(\mathbf{s})$ (such that $\mathbf{G}_b\tilde{\mathbf{s}} = \mathbf{s}$) using [HSS24].

Take a prime integer $q = b^2 + 1$, e.g. we take $17 = 4^2 + 1$.
Let $\mathbf{s} = \begin{bmatrix} 1 & 7 & -1 \end{bmatrix}$ decomposed in base 4 as $\hat{\mathbf{s}} = \begin{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} & \begin{bmatrix} 3 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 4 \end{bmatrix} \end{bmatrix}$ (L to R).
We add a Gaussian error sampled from $\{\mathbf{x} \mid \mathbf{G}_b\mathbf{x} = \mathbf{0}\}$ centred in $\mathbf{0}$.

- Here, consider the Gaussian sampled element $\hat{\mathbf{e}} = \begin{bmatrix} \begin{bmatrix} 5 & 3 \end{bmatrix} & \begin{bmatrix} 1 & 4 \end{bmatrix} & \begin{bmatrix} -4 & 7 \end{bmatrix} \end{bmatrix}$.

[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the framework

In ours: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- Use $\tilde{\mathbf{z}}$ as hint in Hint-MLWE ensuring that it is built on a Gaussian element.
- Use the Gaussianised $\tilde{\mathbf{s}} = \mathsf{Gauss}(\mathbf{s})$ (such that $\mathbf{G}_b\tilde{\mathbf{s}} = \mathbf{s}$) using [HSS24].

Take a prime integer $q = b^2 + 1$, e.g. we take $17 = 4^2 + 1$.

Let $\mathbf{s} = \begin{bmatrix} 1 & 7 & -1 \end{bmatrix}$ decomposed in base 4 as $\hat{\mathbf{s}} = \begin{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} & \begin{bmatrix} 3 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 4 \end{bmatrix} \end{bmatrix}$ (L to R).

We add a Gaussian error sampled from $\{\mathbf{x} \mid \mathbf{G}_b\mathbf{x} = \mathbf{0}\}$ centred in $\mathbf{0}$.

- Here, consider the Gaussian sampled element $\hat{\mathbf{e}} = \begin{bmatrix} \begin{bmatrix} 5 & 3 \end{bmatrix} & \begin{bmatrix} 1 & 4 \end{bmatrix} & \begin{bmatrix} -4 & 7 \end{bmatrix} \end{bmatrix}$.

We finally compute: $\tilde{\mathbf{s}} = \hat{\mathbf{s}} + \hat{\mathbf{e}} = \begin{bmatrix} \begin{bmatrix} 6 & 3 \end{bmatrix} & \begin{bmatrix} 4 & 5 \end{bmatrix} & \begin{bmatrix} -4 & 11 \end{bmatrix} \end{bmatrix}$, Gaussian centred in $\hat{\mathbf{s}}$.

[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the framework

In ours: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- Use $\tilde{\mathbf{z}}$ as hint in Hint-MLWE ensuring that it is built on a Gaussian element.
- Use the Gaussianised $\tilde{\mathbf{s}} = \mathsf{Gauss}(\mathbf{s})$ (such that $\mathbf{G}_b\tilde{\mathbf{s}} = \mathbf{s}$) using [HSS24].

Let $q = 17$ and $b = 4$.

$$\begin{bmatrix} 1 \\ 7 \\ -1 \end{bmatrix} := \begin{bmatrix} 1 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 6 \\ 3 \end{bmatrix} \\ \begin{bmatrix} 4 \\ 5 \end{bmatrix} \\ \begin{bmatrix} -4 \\ 11 \end{bmatrix} \end{bmatrix} \mod 17$$

[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the framework

In ours: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- Use $\tilde{\mathbf{z}}$ as hint in Hint-MLWE ensuring that it is built on a Gaussian element.
- Use the Gaussianised $\tilde{\mathbf{s}} = \mathsf{Gauss}(\mathbf{s})$ (such that $\mathbf{G}_b\tilde{\mathbf{s}} = \mathbf{s}$) using [HSS24].

$$\mathbf{s} := \sum_{0\le i<\lfloor \log_b q\rfloor} b^i\,\hat{\mathbf{s}}_i + \sum_{0\le i<\lfloor \log_b q\rfloor} b^i\,\hat{\mathbf{e}}_i = \sum_{0\le i<\lfloor \log_b q\rfloor} b^i\,\tilde{\mathbf{s}}_i$$

Base-b decomposition   Gaussian error over $\Lambda_q^{\perp}(\mathbf{G}_b)$

[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the framework

In ours: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- Use $\tilde{\mathbf{z}}$ as hint in Hint-MLWE ensuring that it is built on a Gaussian element.
- Use the Gaussianised $\tilde{\mathbf{s}} = \mathsf{Gauss}(\mathbf{s})$ (such that $\mathbf{G}_b\tilde{\mathbf{s}} = \mathbf{s}$) using [HSS24].
- With the naive construction: Use $\tilde{\mathbf{z}}$ to prove relations over $\tilde{\mathbf{s}}$.

In [LNP22]: $\mathbf{z} := c\,\mathbf{s} + \mathbf{y}$

- Use rejection sampling to ensure zero-knowledge.
- Use $\mathbf{z}$ to prove relations over $\mathbf{s}$.

[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

Our modification of the framework

In ours: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- Use $\tilde{\mathbf{z}}$ as hint in Hint-MLWE ensuring that it is built on a Gaussian element.
- Use the Gaussianised $\tilde{\mathbf{s}} = \mathsf{Gauss}(\mathbf{s})$ (such that $\mathbf{G}_b\tilde{\mathbf{s}} = \mathbf{s}$) using [HSS24].
- What we did: Use $\tilde{\mathbf{z}}$ to prove relations over $\mathbf{s}$.

In [LNP22]: $\mathbf{z} := c\,\mathbf{s} + \mathbf{y}$

- Use rejection sampling to ensure zero-knowledge.
- Use $\mathbf{z}$ to prove relations over $\mathbf{s}$.

[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

Extension to a commit-and-prove protocol

Black-box use to prove different relations

By just commiting $\mathbf{s}$, we consider public relations over $\mathbf{x} := [\mathbf{s}, \sigma_{-1}(\mathbf{s})]$.

$$\mathbf{x}^\top \mathbf{R}_2 \mathbf{x} + \mathbf{r}_1^\top \mathbf{x} + r = t$$

$$\mathbf{R}\mathbf{x} = \mathbf{t}$$

Extension to a commit-and-prove protocol

Black-box use to prove different relations

By just commiting $\mathbf{s}$, we consider public relations over $\mathbf{x} := [\mathbf{s}, \sigma_{-1}(\mathbf{s})]$.

$$\mathbf{x}^\top \mathbf{R}_2 \mathbf{x} + \mathbf{r}_1^\top \mathbf{x} + r = t$$

$$\mathbf{R}\mathbf{x} = \mathbf{t}$$

Some usecases and examples:

- Constant coeff of $\left(\mathbf{s}^\top \sigma_{-1}(\mathbf{s})\right)[0]$ is equal to $\langle \vec{s}, \vec{s} \rangle$ i.e. the constant coefficient of the quadratic relation for $\mathbf{R}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}$, $\mathbf{r}_1 = \mathbf{0}$ and $r = 0$.
- Knowledge of $\mathbf{A}\mathbf{s} = \mathbf{0}$ such that $0 < \|\mathbf{s}\| < \beta$ ($\mathbf{s}$ is a SIS solution),
- For the norm $\|\mathbf{s}\| < \beta \iff \beta^2 - \left(\mathbf{s}^\top \sigma(\mathbf{s})\right)[0] > 0$ + proof of no-wraparound.

Extension to a commit-and-prove protocol

Black-box use to prove different relations

By just commiting $\mathbf{s}$, we consider public relations over $\mathbf{x} := [\mathbf{s}, \sigma_{-1}(\mathbf{s})]$.

$$\mathbf{x}^\top \mathbf{R}_2 \mathbf{x} + \mathbf{r}_1^\top \mathbf{x} + r = t$$

$$\mathbf{R}\mathbf{x} = \mathbf{t}$$

As $\sigma(\tilde{\mathbf{s}}) = \widetilde{\sigma_{-1}(\mathbf{s})}$ then $\tilde{\mathbf{x}} := [\tilde{\mathbf{s}}, \sigma_{-1}(\tilde{\mathbf{s}})]$

Modifying relations over $\mathbf{s}$ to relation over $\tilde{\mathbf{s}}$: remind that $\mathbf{x} = \mathbf{G}_b \tilde{\mathbf{x}}$.

- In [LNP22]: use $\mathbf{z} = c\mathbf{s} + \mathbf{y}$ to prove $\mathbf{x}^\top \mathbf{R}_2 \mathbf{x} + \mathbf{r}_1^\top \mathbf{x} + r_0 = 0$.
- Exploit the same strategy $\tilde{\mathbf{z}} = c\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$ to prove $\tilde{\mathbf{x}}^\top \mathbf{R}_2 \tilde{\mathbf{x}} + \mathbf{r}_1^\top \tilde{\mathbf{x}} + r_0 = 0$.

[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

Extension to a commit-and-prove protocol

Black-box use to prove different relations

By just commiting $\mathbf{s}$, we consider public relations over $\mathbf{x} := [\mathbf{s}, \sigma_{-1}(\mathbf{s})]$.

$\mathbf{x}^\top \mathbf{R}_2 \mathbf{x} + \mathbf{r}_1^\top \mathbf{x} + r = t$

$\mathbf{R}\mathbf{x} = \mathbf{t}$

As $\sigma(\tilde{\mathbf{s}}) = \widetilde{\sigma_{-1}(\mathbf{s})}$ then $\tilde{\mathbf{x}} := [\tilde{\mathbf{s}}, \sigma_{-1}(\tilde{\mathbf{s}})]$

Modifying relations over $\mathbf{s}$ to relation over $\tilde{\mathbf{s}}$: remind that $\mathbf{x} = \mathbf{G}_b \tilde{\mathbf{x}}$.

- In [LNP22]: use $\mathbf{z} = c\mathbf{s} + \mathbf{y}$ to prove $\mathbf{x}^\top \mathbf{R}_2 \mathbf{x} + \mathbf{r}_1^\top \mathbf{x} + r_0 = 0$.
- Exploit the same strategy $\tilde{\mathbf{z}} = c\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$ to prove $\tilde{\mathbf{x}}^\top \mathbf{R}_2 \tilde{\mathbf{x}} + \mathbf{r}_1^\top \tilde{\mathbf{x}} + r_0 = 0$.
- Both the relation and $\mathbf{G}_b$ are public, we use $\tilde{\mathbf{z}}$ to prove $\mathbf{x}^\top \mathbf{R}_2 \mathbf{x} + \mathbf{r}_1^\top \mathbf{x} + r_0 = 0$ by proving the equivalent relation $\tilde{\mathbf{x}}^\top (\mathbf{G}_b^\top \mathbf{R}_2 \mathbf{G}_b) \tilde{\mathbf{x}} + (\mathbf{r}_1^\top \mathbf{G}_b) \tilde{\mathbf{x}} + r = 0$.

[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

Conclusion

- A rejection-free alternative to prove computable relations,
- Without restriction to norm-bounded secret **but** not extendable to norm relations,
- Implementation usable (as an addon) of the LaZer library from [LSS24]:

Protocol	Phase	$(p \approx 2^{60}, m = 12)$	$(p \approx 2^{80}, m = 120)$
[LNP22]	**Prove**	324.22	4462.29
Ours		**59.45**	**3961.24**
[LNP22]	**Verify**	18.54	1629.12
Ours		**41.74**	**3624.34**

Mean times (in million cycles), with p the moduli and m the size of the secret.

[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

[LSS24] The LaZer Library: Lattice-Based Zero Knowledge and Succinct Proofs for Quantum-Safe Privacy; Lyubashevsky, Seiler, and Steuer, 2024

Conclusion

- A rejection-free alternative to prove computable relations,
- Without restriction to norm-bounded secret **but** not extendable to norm relations,
- Implementation usable (as an addon) of the LaZer library from [LSS24]:

Protocol	Phase	$(p \approx 2^{60}, m = 12)$	$(p \approx 2^{80}, m = 120)$
[LNP22]	**Prove**	324.22	4462.29
Ours	**Prove**	**59.45**	**3961.24**
[LNP22]	**Verify**	18.54	1629.12
Ours	**Verify**	**41.74**	**3624.34**

Verifying phase at most $\times 3$

Mean times (in million cycles), with p the moduli and m the size of the secret.

[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

[LSS24] The LaZer Library: Lattice-Based Zero Knowledge and Succinct Proofs for Quantum-Safe Privacy; Lyubashevsky, Seiler, and Steuer, 2024

Conclusion

- A rejection-free alternative to prove computable relations,
- Without restriction to norm-bounded secret **but** not extendable to norm relations,
- Implementation usable (as an addon) of the LaZer library from [LSS24]:

Protocol	Phase	$(p \approx 2^{60}, m = 12)$	$(p \approx 2^{80}, m = 120)$
[LNP22]	**Prove**	324.22	4462.29
Ours		**59.45**	**3961.24**
[LNP22]	**Verify**	18.54	1629.12
Ours		**41.74**	**3624.34**

Proving phase
at most $\times \frac{3}{\text{rej. rate}}$

Mean times (in million cycles), with p the moduli and m the size of the secret.

[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

[LSS24] The LaZer Library: Lattice-Based Zero Knowledge and Succinct Proofs for Quantum-Safe Privacy; Lyubashevsky, Seiler, and Steuer, 2024