Rejection-free Framework of Zero-Knowledge Proof based on lattices

Antoine Douteau, Adeline Roux-Langlois
CNRS, GREYC, Caen

Séminaire Cryptis

April 21, 2026

GREYC
Laboratoire de recherche en sciences du numérique

cnrs

RÉGION NORMANDIE

Public-key Cryptography

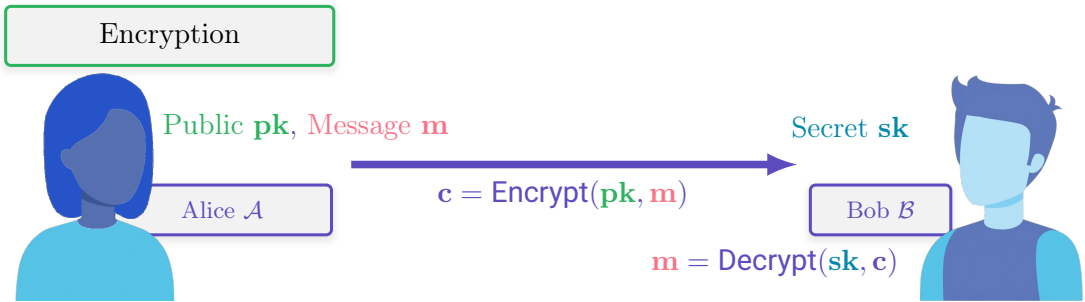
Lattice-based Cryptography

Lattice-based Signatures

Zero-Knowledge Proof without Aborts

Commit-and-prove Proof without Aborts

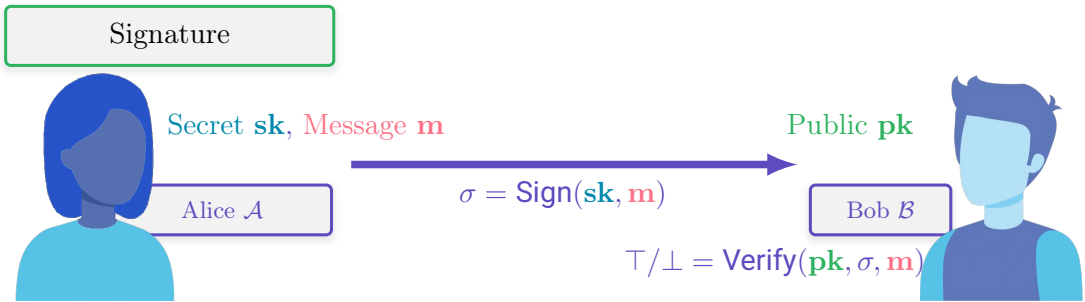
Public-key cryptography



Encryption

$\mathcal{A}$ transmits a confidential message to $\mathcal{B}$.

Public-key cryptography



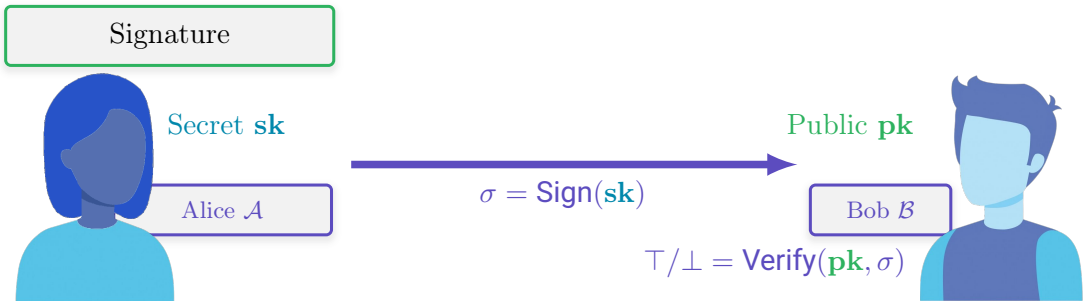
Encryption

$\mathcal{A}$ transmits a confidential message to $\mathcal{B}$.

Signature

Proves that the message originates from $\mathcal{A}$.

Public-key cryptography



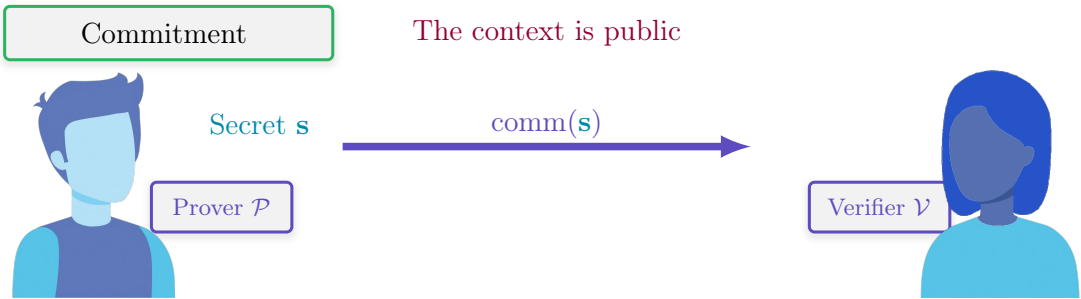
Encryption

$\mathcal{A}$ transmits a confidential message to $\mathcal{B}$.

Signature

Proves that the message originates from $\mathcal{A}$.

Signature via Zero-Knowledge proof



Binding

Hard to cheat to another $\mathbf{s}'$ committed.

Hiding

Hide the underlying secret $\mathbf{s}$ committed.

Signature via Zero-Knowledge proof

Zero-Knowledge proof

The context is public

Prover $\mathcal{P}$

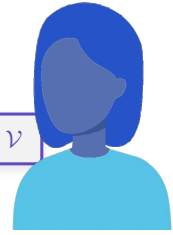
Secret $\mathbf{s}$

$\mathrm{comm}(\mathbf{s})$

a random challenge c

$\mathrm{resp}(\mathbf{s}, c)$

$\pi = \big(\mathrm{comm}(\mathbf{s}), c, \mathrm{resp}(\mathbf{s}, c)\big)$



Verifier $\mathcal{V}$

Completeness

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

(Know.) Soundness

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

Zero-Knowledge

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

Advanced Signature via Zero-Knowledge proof

Commit-and-prove

The context is public

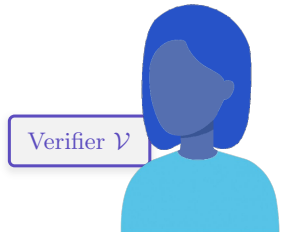
Secret $\mathbf{s}$

Prover $\mathcal{P}$

$\mathrm{comm}(\mathbf{s})$

a random challenge c

$\mathrm{resp}(\mathbf{s}, c)$



Verifier $\mathcal{V}$

$\pi = \big(\mathrm{comm}(\mathbf{s}), c, \mathrm{resp}(\mathbf{s}, c)\big)$

$+$ (additional_material)

Completeness

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

(Know.) Soundness

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

Zero-Knowledge

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

Public-key Cryptography

Lattice-based Cryptography

Lattice-based Signatures

Zero-Knowledge Proof without Aborts

Commit-and-prove Proof without Aborts

Post-quantum cryptography: Lattices

Security is guaranteed through a polynomial reduction to a problem.

$$\text{Security of a cryptosystem} \geq \text{Cryptographic assumption}$$

Since Shor's algorithm 1994, previous cryptographic assumptions are "broken" i.e. solvable in polynomial time with quantum algorithms.

Post-quantum cryptography: resistance against both attacks.

- Multivariate polynomials, isogenies, error-correcting code, ...,
- After NIST competition: 3/5 standards are based on **lattices**,
- Due to its algebraic structure, lattices can be used for advanced primitives.

Standard lattice assumptions

Learning With Errors (LWE) by Regev, 2005, 2009:

One of the problem that gave rise to lattice-based cryptography.

- Solve the linear equation system with some error/noise:

$$
\begin{aligned}
s_1 &+ 14s_2 &+ 20s_3 &+ 15s_4 &+ 15s_5 &\approx 9 &\bmod 29\\
14s_1 &+ 5s_2 &+ 4s_3 &+ 15s_4 &+ 2s_5 &\approx 20 &\bmod 29\\
20s_1 &+ 5s_2 &+ 1s_3 &+ 21s_4 &+ 28s_5 &\approx 19 &\bmod 29\\
19s_1 &+ 20s_2 &+ 1s_3 &+ 21s_4 &+ 15s_5 &\approx 13 &\bmod 29\\
5s_1 &+ 21s_2 &+ 18s_3 &+ 4s_4 &+ 27s_5 &\approx 1 &\bmod 29\\
5s_1 &+ 16s_2 &+ 5s_3 &+ 20s_4 &+ 4s_5 &\approx 12 &\bmod 29
\end{aligned}
$$

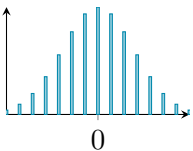
Standard lattice assumptions

Learning With Errors (LWE) by Regev, 2005, 2009:

Environment: $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$ for a prime q, $m \geq n$,

Public parameters: $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

Private information: $\mathbf{s} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$ or $\mathcal{D}^n_{\mathbb{Z},\alpha q}$



$\mathbf{e} \leftarrow \mathcal{D}^m_{\mathbb{Z},\alpha q}$

Input : $\mathbf{A}$, $\mathbf{A}\,\mathbf{s} + \mathbf{e}$

Find : $\mathbf{s}$

Standard lattice assumptions

Short Integer Solution (SIS) by Ajtai, 1996:

Environment: $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$ for a prime q, $m \geq n$,

Public parameters: $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n \times m})$ ($n \times m$)

Input : $\mathbf{A}$

Find : $\mathbf{s} \in \mathbb{Z}_q^m$ of small norm $\|\mathbf{s}\| \leq \beta$

such that: $\mathbf{A}\,\mathbf{s} = \mathbf{0}$

Standard lattice assumptions

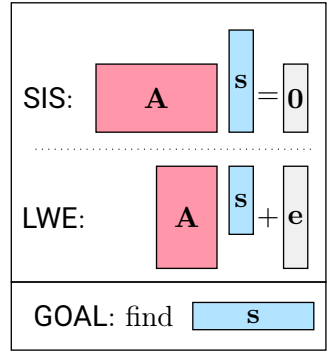
Reduction wtih fundamental lattice problems:

SIS can be seen as ApproxSVP over lattices of the form

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \mod q\}$$

LWE can be seen as ApproxCVP over lattices of the form

$$\Lambda_q(\mathbf{A}) = \{\mathbf{A}\mathbf{s} \mod q \text{ for } \mathbf{s} \in \mathbb{Z}^n\}$$



Public-key Cryptography

Lattice-based Cryptography

Lattice-based Signatures

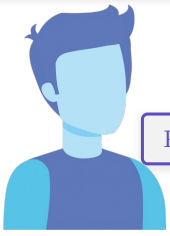
Zero-Knowledge Proof without Aborts

Commit-and-prove Proof without Aborts

Proving without revealing

Zero-Knowledge

The context is public



Secret $\mathbf{s}$

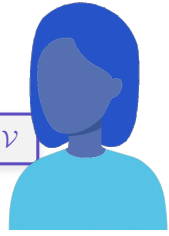
Prover $\mathcal{P}$

$\mathrm{comm}(\mathbf{s})$

a random challenge c

$\mathrm{resp}(\mathbf{s}, c)$

$\pi = \big(\mathrm{comm}(\mathbf{s}), c, \mathrm{resp}(\mathbf{s}, c)\big)$



Verifier $\mathcal{V}$

Completeness

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

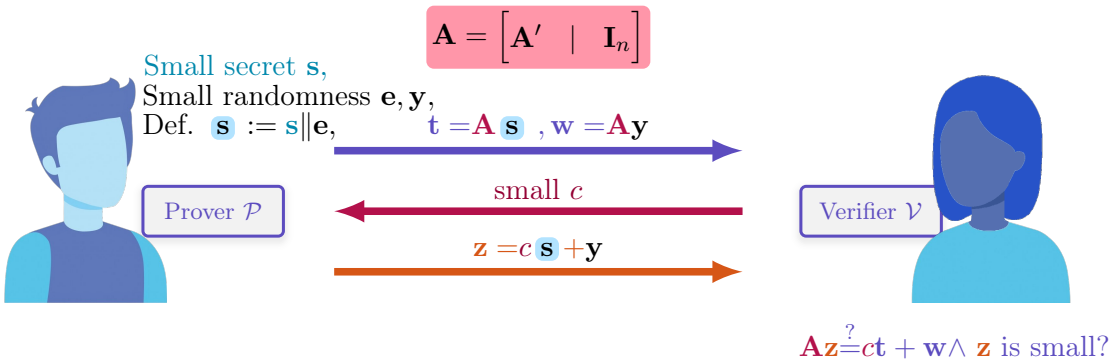
(Know.) Soundness

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 Zero-Knowledge proof

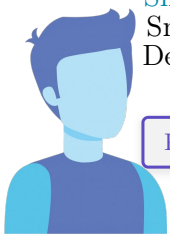


An idea of a lattice-based Zero-Knowledge proof

SIS: $\mathbf{A}$ $\mathbf{s}$ $= \mathbf{0}$

GOAL: find $\mathbf{s}$

$$\mathbf{A} = \left[\mathbf{A}' \mid \mathbf{I}_n\right]$$

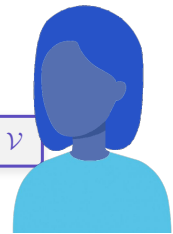


Small secret $\mathbf{s}$,
Small randomness $\mathbf{e}$,
Def. $\mathbf{s} := \mathbf{s}\|\mathbf{e}$,

$\mathbf{t} = \mathbf{A}\,\mathbf{s}$

Prover $\mathcal{P}$

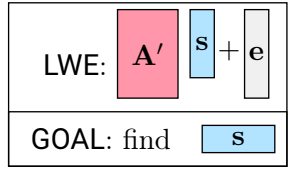
Verifier $\mathcal{V}$



Binding

It reduces to find a SIS.

An idea of a lattice-based Zero-Knowledge proof



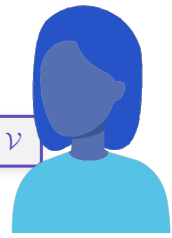
$$\mathbf{A} = \left[\mathbf{A}' \mid \mathbf{I}_n\right]$$

Small secret $\mathbf{s}$,
Small randomness $\mathbf{e}$,
Def. $\mathbf{s} := \mathbf{s} \| \mathbf{e}$,

$\mathbf{t} = \mathbf{A}\,\mathbf{s}$

Prover $\mathcal{P}$

Verifier $\mathcal{V}$



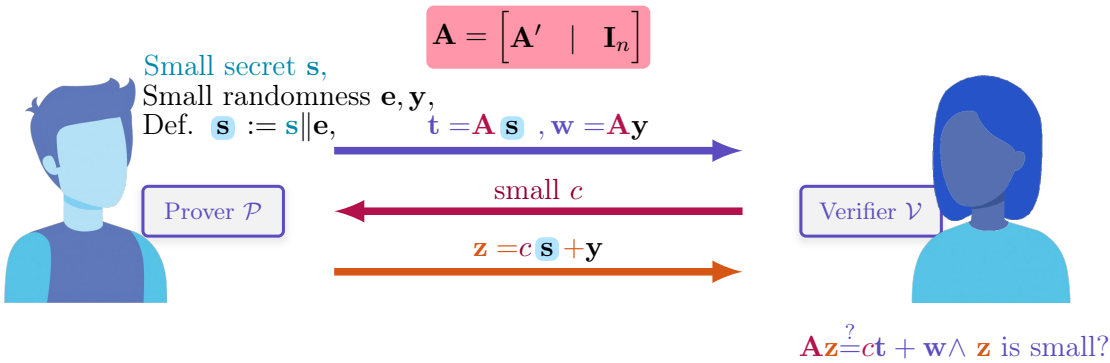
Binding

It reduces to find a SIS.

Hiding

It reduces to solve LWE.

An idea of a lattice-based Zero-Knowledge proof



Completeness

If $\mathbf{s}, \mathbf{y}$ smalls then $\mathbf{z}$ is small.

An idea of a lattice-based Zero-Knowledge proof

SIS: $\mathbf{A}$ $\mathbf{s}$ $= \mathbf{0}$

GOAL: find $\mathbf{s}$

$$\mathbf{A} = \left[\mathbf{A}' \mid \mathbf{I}_n\right]$$

Small secret $\mathbf{s}$,
Small randomness $\mathbf{e}, \mathbf{y}$,
Def. $\mathbf{s} := \mathbf{s} \| \mathbf{e}$,

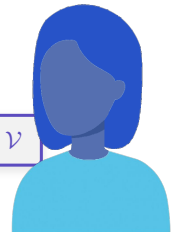
Prover $\mathcal{P}$

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

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?

Completeness

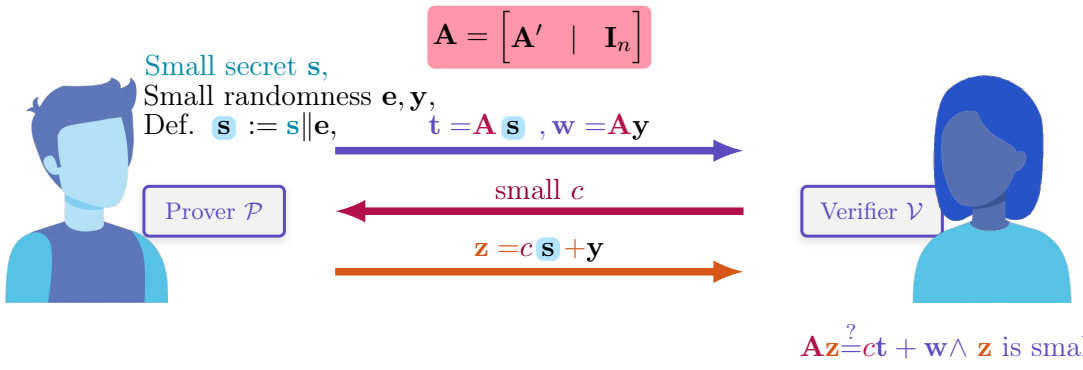
If $\mathbf{s}, \mathbf{y}$ smalls then $\mathbf{z}$ is small.

(Know.) Soundness

2 valids π implied know.,

unicity is reduced to SIS.

An idea of a lattice-based Zero-Knowledge proof



Completeness

If $\mathbf{s}, \mathbf{y}$ smalls then $\mathbf{z}$ is small.

(Know.) Soundness

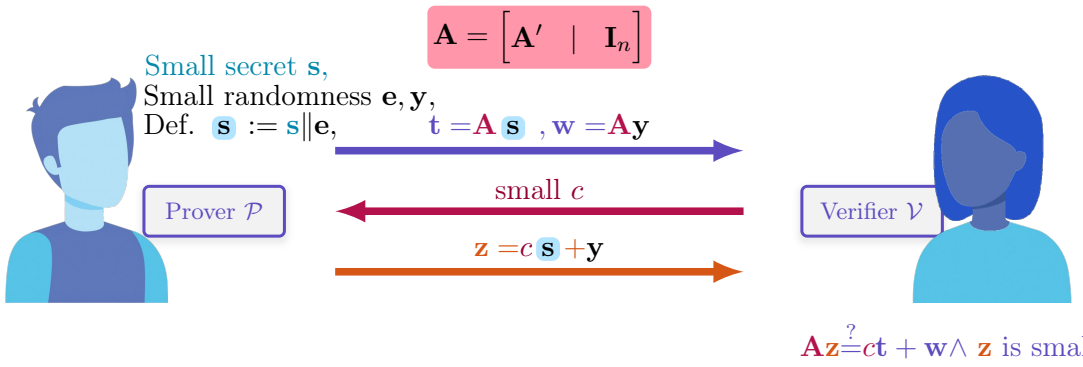
2 valids π implied know.,

unicity is reduced to SIS.

Zero-Knowledge

Hiding + π.

An idea of a lattice-based Zero-Knowledge proof



Completeness

If $\mathbf{s}, \mathbf{y}$ smalls then $\mathbf{z}$ is small.

(Know.) Soundness

2 valids π implied know.,

unicity is reduced to SIS.

Zero-Knowledge

It is not! $\mathbf{z}$ leaks $c\mathbf{s}$.

Transmitting without revealing

Noise flooding:

GOAL: Flood $\mathrm{resp}(\mathbf{s}, c)$ by adding a more significant error $\mathbf{y}'$.

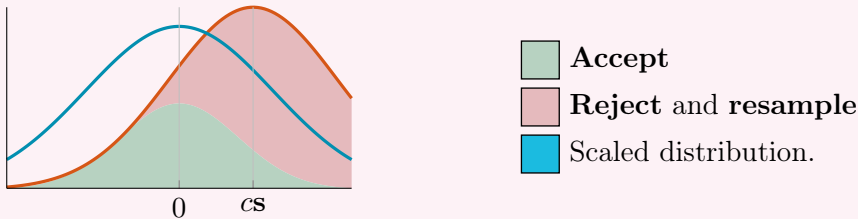
1. It will be difficult to recover $c\mathbf{s}$ from $\mathbf{z}$ and therefore $\mathbf{s}$.
2. The response $\mathbf{z}$ will not be small. Therefore, it is not applicable.

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 $\mathbf{z}$ leaks info about $\mathbf{s}$.
2. **Verifier** sees a publicly known distribution from its point of view.



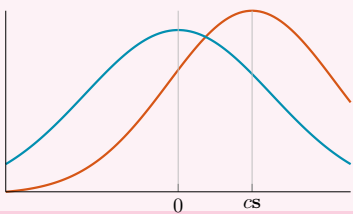
[Lyu09] Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures.; Lyubashevsky, 2009

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 $\mathbf{z}$ leaks info about $\mathbf{s}$.
2. **Verifier** sees a publicly known distribution from its point of view.



Real distribution of $\mathbf{z}$,
Distribution of $\mathbf{z}$ from $\mathcal{V}$.

[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 $\mathrm{resp}(\mathbf{s}, c)$.

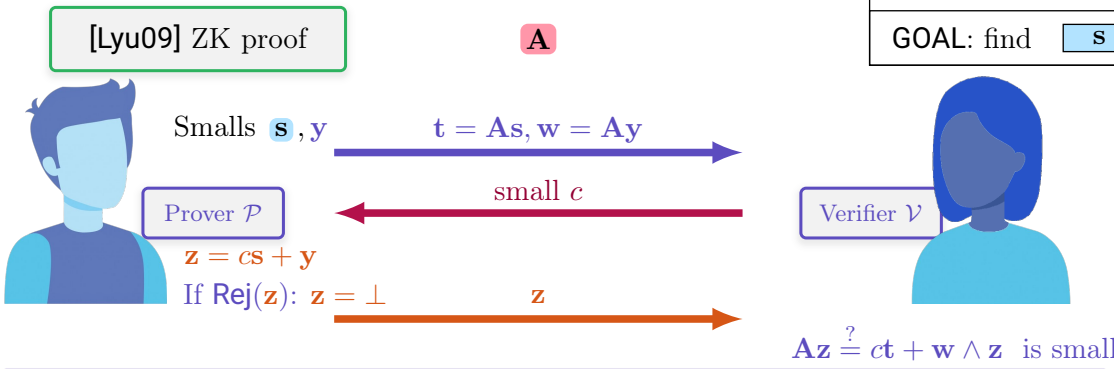
1. **Prover** aborts and restarts the entire protocol when $\mathbf{z}$ leaks info about $\mathbf{s}$.
2. **Verifier** sees a publicly known distribution from its point of view.

Problems (highlighted by some papers [CGL+24,ZCQ+26])

1. Non-constant execution time,
2. Must involve costly masking to protect it from side-channel attacks,
3. Hard to safe-implement (due to processor/hardware optimisations).

[Lyu09] Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures.; Lyubashevsky, 2009
[CGL+24] Improved High-Order Masked Generation of Masking Vector and Rejection Sampling in Dilithium; Coron et al., 2024
[ZCQ+26] Rejection Matters: Efficient Non-Profiling Side-Channel Attack on ML-DSA via Exploiting Public Templates; Zhao et al. 2026

Lattice-based Zero-Knowledge proof

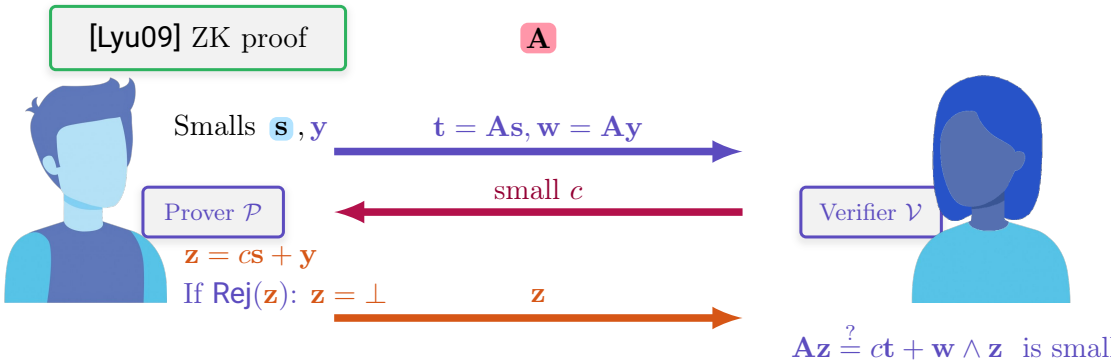


The protocol is zero-knowledge ... by abortion

Based on the hardness of LWE and instances of rejection sampling.

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

Lattice-based Zero-Knowledge proof



Our Goal: Transform the proof to deal without abortion

Idea: ensure that $\mathbf{z}$ does not reveal anything (no matter what c and $\mathbf{s}$ are)

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

Public-key Cryptography

Lattice-based Cryptography

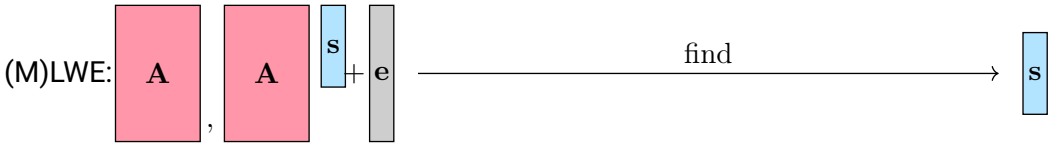
Lattice-based Signatures

Zero-Knowledge Proof without Aborts

Commit-and-prove Proof without Aborts

Relaxed lattice assumption

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

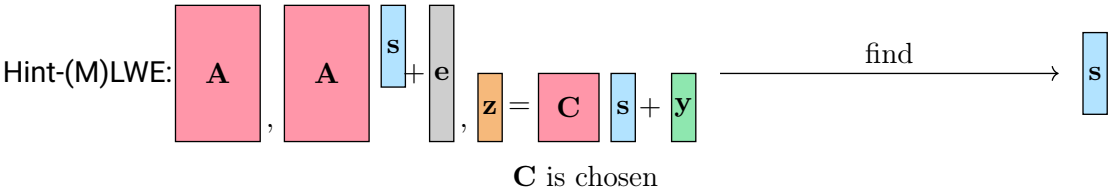


[Reg05] On lattices, Learning With Errors, random linear codes, and cryptography; Regev, 2005

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

Relaxed lattice assumption

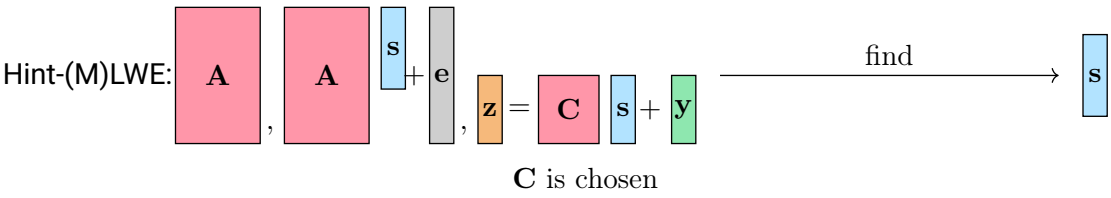
(Module) Learning With Errors (M)LWE with hints about the secret/error: Hint-(M)LWE [KLSS23]



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

Relaxed lattice assumption

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



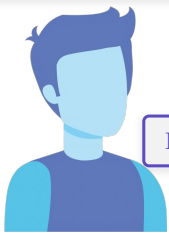
[KLSS23]: MLWE $\Longrightarrow$ Hint-MLWE, when secrets and errors follow Discrete Gaussians.

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

Lattice-based Zero-Knowledge proof

Knowledge of an Ajtai opening

$\mathbf{A}$



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

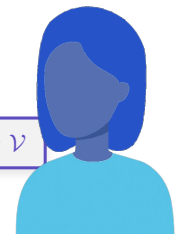
Prover $\mathcal{P}$

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

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

Consider that the response $\mathbf{z}$ is the hint in Hint-MLWE

Not directly applicable as $\mathbf{s}$ is not fully resampled at each execution.

Our modification of the ZK proof

In [DR25]: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

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

In [Lyu09]: $\mathbf{z} := c\,\mathbf{s} + \mathbf{y}$

- Use rejection sampling to ensure zero-knowledge.

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[Lyu09] Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures.; Lyubashevsky, 2009

Our modification of the ZK proof

In [DR25]: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- 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 [Lyu09]: $\mathbf{z} := c\,\mathbf{s} + \mathbf{y}$

- Use rejection sampling to ensure zero-knowledge.

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[Lyu09] Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures.; Lyubashevsky, 2009
[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the ZK proof

In [DR25]: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- 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].

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

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

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the ZK proof

In [DR25]: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- 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).

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the ZK proof

In [DR25]: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- 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 $\hat{\mathbf{s}} = \left[\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 3 & 1 \end{bmatrix} \begin{bmatrix} 0 & 4 \end{bmatrix}\right]$ (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 $\hat{\mathbf{e}} = \left[\begin{bmatrix} 5 & 3 \end{bmatrix} \begin{bmatrix} 1 & 4 \end{bmatrix} \begin{bmatrix} -4 & 1 \end{bmatrix}\right]$.

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the ZK proof

In [DR25]: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- 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 $\hat{\mathbf{s}} = \left[\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 3 & 1 \end{bmatrix} \begin{bmatrix} 0 & 4 \end{bmatrix}\right]$ (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 $\hat{\mathbf{e}} = \left[\begin{bmatrix} 5 & 3 \end{bmatrix} \begin{bmatrix} 1 & 4 \end{bmatrix} \begin{bmatrix} -4 & 1 \end{bmatrix}\right]$.

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

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the ZK proof

In [DR25]: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- 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 $b = 4$.

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

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Our modification of the ZK proof

In [DR25]: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

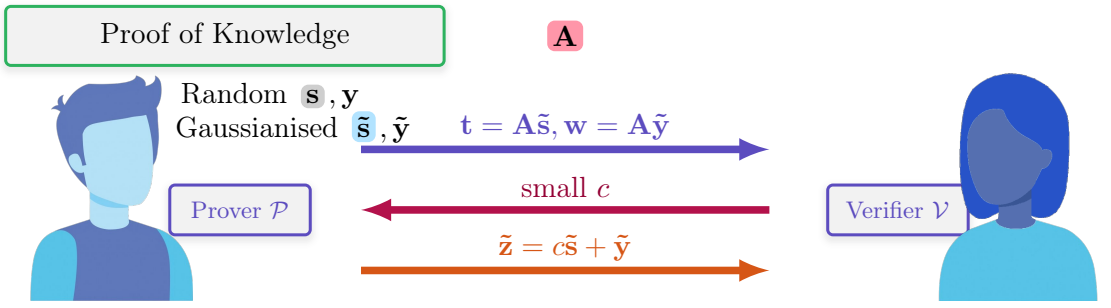
- Use $\tilde{\mathbf{z}}$ as hint in Hint-MLWE ensuring that it is built on a Gaussian element.
- Use the Gaussianised $\tilde{\mathbf{s}} = \mathsf{Gauss}(\mathbf{s})$ (such that $\mathbf{G}_b\tilde{\mathbf{s}} = \mathbf{s}$) using [HSS24].

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

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

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[HSS24] Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions; Hwang, Seo, and Song, 2024

Rejection-free proof of knowledge [DR25]



The response $\mathbf{z}$ can be considered as the hint in Hint-MLWE

By committing to $\tilde{\mathbf{s}}$ which is deterministically decoded into $\mathbf{s}$.

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025

Public-key Cryptography

Lattice-based Cryptography

Lattice-based Signatures

Zero-Knowledge Proof without Aborts

Commit-and-prove Proof without Aborts

Extension to a commit-and-prove protocol

Black-box use to prove different relations

We have a proof of knowledge of $\mathbf{s}$ but we can also prove:

$$\mathbf{s}^\top \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^\top \mathbf{s} + r_0 = 0$$

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

Extension to a commit-and-prove protocol

In [DR25]: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- 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 $\mathbf{s}$.

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[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

Extension to a commit-and-prove protocol

In [DR25]: $\tilde{\mathbf{z}} := c\,\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$

- 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 $\mathbf{s}$.

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[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

Extension to a commit-and-prove protocol

Black-box use to prove different relations

We have a proof of knowledge of $\mathbf{s}$ but we can also prove:

$$\mathbf{s}^\top \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^\top \mathbf{s} + r_0 = 0$$

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

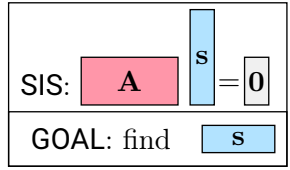
[DR25]: Modifying relations over $\mathbf{s}$ to relation over $\tilde{\mathbf{s}}$: remind that $\mathbf{s} = \mathbf{G}_b \tilde{\mathbf{s}}$.

- In **[LNP22]**: the authors used $\mathbf{z} = c\mathbf{s} + \mathbf{y}$ to prove $\mathbf{s}^\top \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^\top \mathbf{s} + r_0 = 0$.
- We exploit the same strategy for $\tilde{\mathbf{z}} = c\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$ to prove $\tilde{\mathbf{s}}^\top \mathbf{R'}_2 \tilde{\mathbf{s}} + \mathbf{r'}_1^\top \tilde{\mathbf{s}} + r_0 = 0$.
- Finally, $\tilde{\mathbf{s}}^\top (\mathbf{G}_b^\top \mathbf{R}_2 \mathbf{G}_b) \tilde{\mathbf{s}} + (\mathbf{r}_1^\top \mathbf{G}_b) \tilde{\mathbf{s}} + r_0 = \mathbf{s}^\top \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^\top \mathbf{s} + r_0$.

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025

[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

We have a proof of knowledge of $\mathbf{s}$ but we can also prove:

$$\mathbf{s}^\top \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^\top \mathbf{s} + r_0 = 0 \qquad \mathbf{R}\mathbf{s} = \mathbf{t}$$

One example: SIS (but we can do the same for LWE)

- Knowledge of $\mathbf{A}\mathbf{s} = \mathbf{0}$ such that $0 < \|\mathbf{s}\| < \beta$ ($\mathbf{s}$ is a SIS solution),
- But $\|\mathbf{s}\| < \beta$ is a quadratic formula: $\beta - \|\mathbf{s}\| = a^2 + b^2 + c^2 + d^2 > 0$,
- $+$ proof of no wraparound modulo q.

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[LNP22] Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General; Lyubashevsky, Nguyen, and Plançon, 2022

Conclusion

We provide an implementation usable (as an addon) of the LaZer library from [LSS24]:

Protocol	Phase	$(q \approx 2^{60}, \mathsf{length}(\mathbf{s}) = 12)$	$(q \approx 2^{80}, \mathsf{length}(\mathbf{s}) = 120)$
[LNP22]	**Prove**	324.22	4462.29
[**D**R25]		**59.45**	**3961.24**
[LNP22]	**Verify**	18.54	1629.12
[**D**R25]		**41.74**	**3624.34**

Mean times (in million cycles), with q the modulo for 128 bits of security.

[**D**R25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[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

We provide an implementation usable (as an addon) of the LaZer library from [LSS24]:

Protocol	Phase	$(q \approx 2^{60}, \mathsf{length}(\mathbf{s}) = 12)$	$(q \approx 2^{80}, \mathsf{length}(\mathbf{s}) = 120)$
[LNP22]	**Prove**	324.22	4462.29
[**D**R25]		**59.45**	**3961.24**
[LNP22]	**Verify**	18.54	1629.12
[**D**R25]		**41.74**	**3624.34**

Verifying phase at most $\times 3$

Mean times (in million cycles), with q the modulo for 128 bits of security.

[**D**R25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[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

We provide an implementation usable (as an addon) of the LaZer library from [LSS24]:

Protocol	Phase	$(q \approx 2^{60}, \mathsf{length}(\mathbf{s}) = 12)$	$(q \approx 2^{80}, \mathsf{length}(\mathbf{s}) = 120)$
[LNP22]	**Prove**	324.22	4462.29
[**D**R25]		**59.45**	**3961.24**
[LNP22]	**Verify**	18.54	1629.12
[**D**R25]		**41.74**	**3624.34**

Proving phase

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

Mean times (in million cycles), with q the modulo for 128 bits of security.

[**D**R25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[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

We provide an implementation usable (as an addon) of the LaZer library from [LSS24]:

Protocol	Phase	$(q \approx 2^{60}, \mathsf{length}(\mathbf{s}) = 12)$	$(q \approx 2^{80}, \mathsf{length}(\mathbf{s}) = 120)$
[LNP22]	**Prove**	324.22	4462.29
[**D**R25]		**59.45**	**3961.24**
[LNP22]	**Verify**	18.54	1629.12
[**D**R25]		**41.74**	**3624.34**

Mean times (in million cycles), with q the modulo for 128 bits of security.

Open questions:

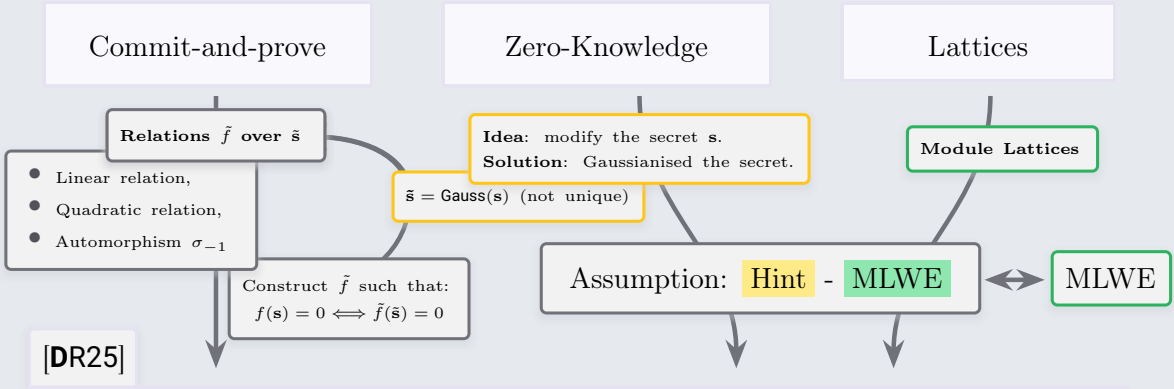
- Enable norm proofs to this rejection-free model with changing scheme,
- Transform it in the standard model by defining its associated Trapdoor Σ protocols.

[**D**R25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025
[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

Final overview

Thanks you for listening. Feel free to ask questions.

eprint:2025/2239



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

[DR25] Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE; Douteau and Roux-Langlois, 2025

Module Lattices

Consider a structured variant to replace $\mathbb{Z}_q^d$ with $d = 2^k > 0$.

How ? Defining $\mathcal{R} := \mathbb{Z}[X]/\langle X^d + 1\rangle$ such that $\mathbb{Z}^d \cong \mathcal{R}$. Let $a \in \mathcal{R}$, then we can view $a = a_0 + a_1 X + \cdots + a_{d-1} X^{d-1} \in \mathcal{R}$ and $\vec{a} = \mathsf{cf}(a) = (a_0, \ldots, a_{d-1}) \in \mathbb{Z}^d$.

For the sum: $\mathsf{cf}(a + b) = \vec{a} + \vec{b}$

For the product:

$$\mathsf{cf}(a \cdot b) = \mathsf{Rot}(a) \cdot \vec{b}^\top = \begin{bmatrix} a_0 & -a_{d-1} & \cdots & -a_1 \\ a_1 & a_0 & \cdots & -a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{d-1} & a_{d-2} & \cdots & a_0 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{d-1} \end{bmatrix}$$

Now, enhance this construction to matrices with coefficients being polynomials $(\mathcal{M}_{n,m}(\mathcal{R}))$. It can be view as a structured lattice with dimension nd.

Construction

Why it works? 1/2

Why this encoding?

$\mathcal{R}_q := \mathbb{Z}_q[X]/\langle X^{2d}+1\rangle \simeq \mathbb{Z}_q^{2d}$ and $\mathcal{R}/\langle X^d - b\rangle \simeq \mathbb{Z}_q^d$ as $\mathbb{Z}$-module (i.e. a lattice).

Is q forced to be $q = b^2+1$?

Actually yes and no. The encoding prime q only needs to be constructed as $q = b^k + 1$.

In LNP22, they use a challenge space that requires $q \equiv 5 \mod 8$ (and this imply $k = 2$).

Another solution is to modify the challenge space (we tried).

But with this restriction, do you have huge constraint of the modulo q? YES!

There is no existing parameter regime for $q < 2^50$. (LWE Estimator used).

[APS15] On the concrete hardness of Learning with Errors (LWE Estimator); Albrecht, Player, and Scott, 2015

Construction

Why it works? 2/2

Modify the challenge space: did you try?

We need too many things: exponential size, stable with σ_{-1}, all differences need to be invertible.

Retrieve z from $\tilde{\mathbf{z}}$: As $\mathbf{G}\tilde{\mathbf{s}} = \mathbf{s}$, why we just don't do the same to $\tilde{\mathbf{z}}$?

We have $\mathbf{y}$ that is randomnly sampled from $\mathbb{Z}_q^n$. Then, $\tilde{\mathbf{y}}$ is a Gaussianised version of $\mathbf{y}$ such that $\mathbf{G}\tilde{\mathbf{y}} = \mathbf{y}$. Finally, $\tilde{\mathbf{z}} = c\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$. Then $\mathbf{G}\tilde{\mathbf{z}} = c\mathbf{s} + \mathbf{y}$ (noise flooded).

Why do we care about σ_{-1}?

σ_{-1} is an automorphism of $\mathcal{R}$ and $\langle \mathbf{s}, \mathbf{s} \rangle = \mathsf{cst}(\sigma_{-1}(\mathbf{s}) \cdot \mathbf{s})$. We have $\widetilde{\sigma_{-1}(\mathbf{s})} = \sigma_{-1}(\tilde{\mathbf{s}})$.

Construction

Why is it useful?

Why do we care about removing the rejection sampling? The rejection sampling is a bottleneck for the proving time. For some advanced construction, parameters implies high rejection rate and then unfeasability by the Lyubashesvsky' method.

Is it applicable to others schemes using Ajtai/ABDLOP? to folding schemes? Our comm. scheme works and can replaced any Ajtai/ABDLOP comm. scheme. But it is also important to highlight that the witness committed is 2 times bigger with ours. Folding scheme does not have the ZK property we were interested in. There is no rejection sampling (think about folding as proven delegated computations) ... but yes.

Implementation

You said "implementation" isn't it?

Information about the implementation

In `c`, not using the full capacity as the LaZer library as it required AVX512 instructions. We obtain the results with a compilation of our code on a laptop with Intel Core Ultra 7 165H, using `gcc version 11.4.0` with the options `-03 -march=native, -mtune=native` and `-pthread` to ensure optimality of the results.

Is it viable?

It's a POC because asymptotic comparison was not interesting.

Information about your comparison table

Following slides.

Implementation

Parameters

Sets of parameters						
Variable	**Description**	**Set 1**	**Set 2**	**Set 3**	**Set 4**	**Set 5**
λ	Security parameter	128				
d	Ring degree	64				
$\log p$	Log of the ring moduli	60	120	80	80	60
$m_{1_{\mathsf{LNP}}}$	Number of low-norm messages	10	10	100	0	100
ℓ_{LNP}	Number of unbounded messages	2	2	0	20	20
$m_{1_{\mathsf{new}}}$	Number of messages $= m_{1_{\mathsf{LNP}}} + \ell_{\mathsf{LNP}}$	12	12	100	20	120
N	Number of quadratic equations to prove	3				
Security parameters						
κ	$\|c\|_\infty \leq \kappa$ with $c \in \mathcal{C}$	8				
η	$\|c\mathbf{r}\| \leq \eta\|\mathbf{r}\|$ with $c \in \mathcal{C}$ and $\mathbf{r}$ a vector of $\mathcal{R}$	140				
n	Height of $\mathbf{A}_1$ and $\mathbf{A}_2$	32	38	35	34	34
m_2	Length of the randomness $\mathbf{s}_2$	71	114	86	85	73
Dilithium compression						
γ	Largest parameter to cut low-order bits of $\mathbf{w}$	See in implementation				
D	Number of low-order bits cut from $\mathbf{t_A}$	31	61	42	41	32
Computed parameters						
b	Base decomposition	See in implementation				
p	Prime $p := b^2 + 1$ congruent to 5 modulo 8					
σ_1	SD of the randomized encoding of $\mathbf{s}_1$					
$\sqrt{2}\mathfrak{s}_1$	SD of the randomized encoding of the mask $\mathbf{y}_{\mathsf{rand}}$					
σ_2	SD of the randomness $\mathbf{s}_2$					
$\mathfrak{s}_2$	SD of the mask of $\mathbf{y}_2$					

Table 2: Sets of parameters involved in the proof frameworks. The first section of parameters are manually set, others are either computed using `/scripts` of our code inspired by the LaZer Library or using LWE-Estimator [APS15].

Implementation

Results

Size of the transcripts (in kilobytes KB)						
Protocol	Protocol/Phase	Param. 1	Param. 2	Param. 3	Param. 4	Param. 5
LNP22	**Opening proof**	13.22	20.77	30.53	×	41.05
	+ 3 Quad.	13.93	21.86	31.31	×	41.68
	+ 3 Eval.	15.16	22.96	32.09	×	42.93
Ours	**Opening proof**	24.23	45.69	103.19	37.33	96.52
	+ 3 Quad.	24.83	46.76	103.95	38.08	97.12
	+ 3 Eval.	25.42	47.82	104.70	38.83	97.72
Mean times (in million cycles)						
LNP22	**KeyGen**	7.63	10.32	19.32	×	39.35
	Commit	7.29	20.89	31.33	×	39.42
	Prove	324.22	352.22	4289.68	×	4462.29
	Verify	18.54	39.17	1744.10	×	1629.12
Ours	**KeyGen**	19.01	47.04	66.30	29.55	62.01
	Commit	19.86	86.54	102.47	45.73	67.15
	Prove	59.45	134.26	4248.67	220.98	3961.24
	Verify	41.74	96.79	3897.43	180.31	3624.34

Table 3: The first part of the Table explains the size of the transcripts in kilobytes for different parameter sets explained in Appendix C.1. In the second part, the results show our benchmarks for the mean running time of 100 executions of each phase for an execution of different instances of the protocol for the 5 different sets of parameters detailed in Table 2. The running time of the Prove part of LNP22 using the LSS24 implementation takes into account several executions due to rejection sampling.