HAETAE: Bridging Algebraic Number Theory to Post-Quantum Digital Signatures

Jung Hee Cheon[1,2], **Hyeongmin Choe**[1], Julien Devevey[3], Tim Güneysu[4], Dongyeon Hong[2], Markus Krausz[4], Georg Land[4], Junbum Shin[2], Damien Stehlé[3,5], MinJune Yi[1,2]

[1]Seoul National University, [2]CryptoLab Inc., [3]École Normale Supérieure de Lyon, [4]Ruhr Universität Bochum, [5]Institut Universitaire de France

2024 Algebra Camp
February 5, 2024

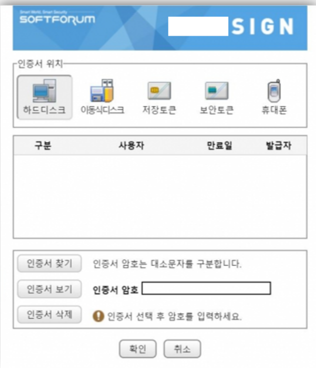
Table of Contents

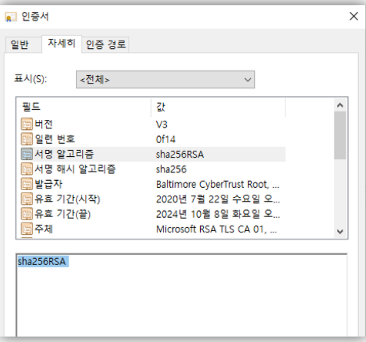
1. Post-Quantum Cryptography:
    - What is Cryptography?
    - Lattice hard problems

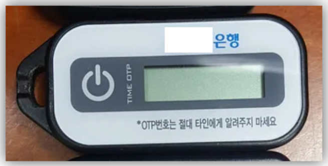
2. Digital Signatures:
    - What is a digital signature?
    - Lattice-based digital signatures
    - Rejection sampling

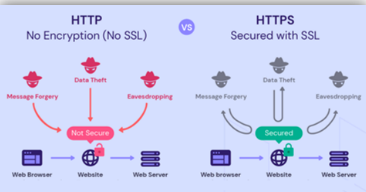
3. HAETAE:
    - Hyperball bimodal rejection sampling
    - Comparison to SotA lattice signatures
    - Current status

What is Cryptography?

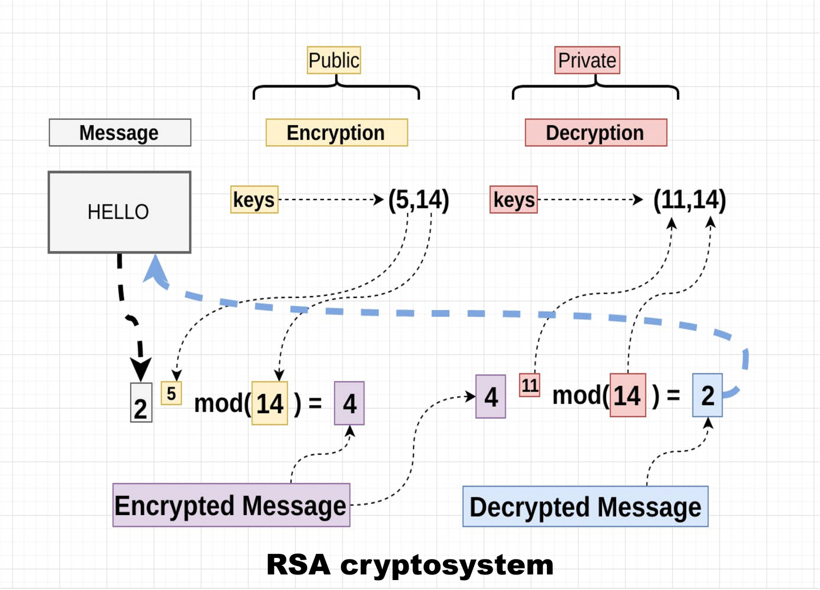




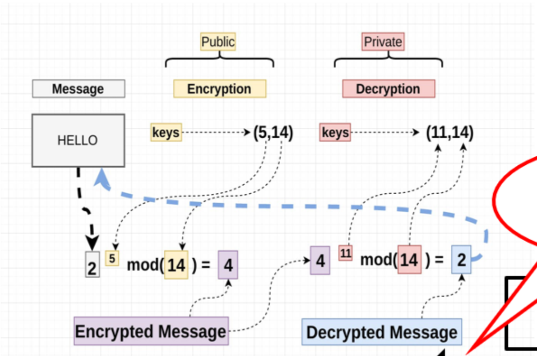




What is Cryptography?



What is Cryptography?



RSA cryptosystem

Reduction!

Integer Factorization problem

RSA problem: ***for given*** $a, e, n \in \mathbb{Z}$, ***find*** $b \in \mathbb{Z}$ ***satisfying*** $a \equiv b^e \bmod n$.

What is Cryptography?

Cryptosystem ← **Reduction** ← **Hard problems**

- RSA Encryption/Signature

- Diffie-Hellman Key Exchange

- ECDSA/ECDH

⋮

- Integer Factorization

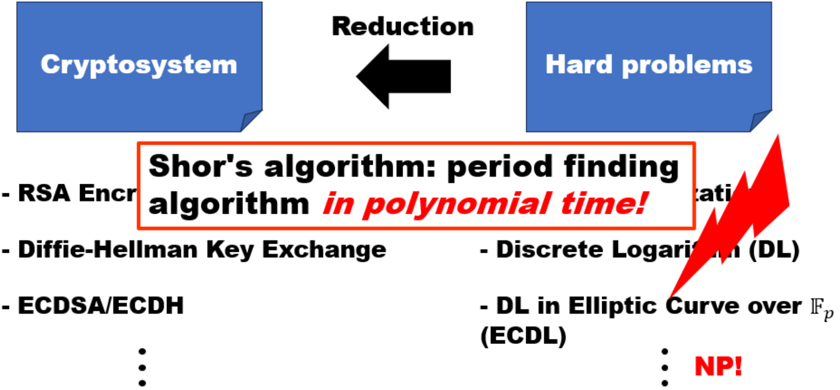
- Discrete Logarithm (DL)

- DL in Elliptic Curve over $\mathbb{F}_p$ (ECDL)

⋮ **NP!**

Post-Quntum Cryptography

However, including the quantum algorithms...



Post-Quntum Cryptography

Post-Quantum Cryptography ← **Reduction** ← **Hard problems (even) against Quantum Algorithms**

- Lattice-based cryptography

- Code-based cryptography

⋮

- Shortest/Closest Vector Problem (SVP/CVP)

- Syndrome Decoding Problem (SDP)

⋮

NP-hard!*

1. Post-Quantum Cryptography:
   - What is Cryptography?
   - Lattice hard problems

2. Digital Signatures:
   - What is a digital signature?
   - Lattice-based digital signatures
   - Rejection sampling

3. HAETAE:
   - Hyperball bimodal rejection sampling
   - Comparison to SotA lattice signatures
   - Current status

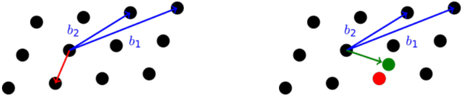
Lattice hard problems

Useful hard problems:

- LWE, Ring-LWE, and Module-LWE
- SIS, Ring-SIS, and Module-SIS

NP-hard problems:

- Shortest Vector Problem (SVP)
- Closet Vector Problem (CVP)



SVP and CVP in dimension two.

Reductions:

Schemes $\Leftarrow$ Useful hard problems $\Leftarrow$ NP-hard problems

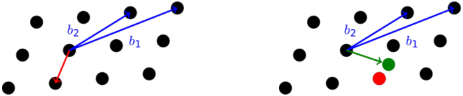
Lattice hard problems

Useful hard problems:

- LWE, Ring-LWE, and Module-LWE
- SIS, Ring-SIS, and Module-SIS

NP-hard problems:

- Shortest Vector Problem (SVP)
- Closet Vector Problem (CVP)



SVP and CVP in dimension two.

Reductions:

Schemes $\Leftarrow$ Useful hard problems $\Leftarrow$ NP-hard problems

Lattice hard problems

Useful hard problems:

- LWE, Ring-LWE, and Module-LWE
- SIS, Ring-SIS, and Module-SIS

NP-hard problems:

- Shortest Vector Problem (SVP)
- Closet Vector Problem (CVP)

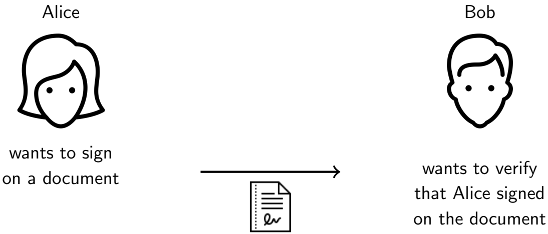
SVP and CVP in dimension two.

Reductions:

$$\text{Schemes} \Leftarrow \text{Useful hard problems} \Leftarrow \text{NP-hard problems}$$

Digital signatures

Conventional signatures work as:



Digital signatures

Conventional signatures work as:

Alice

wants to sign
on a document

Bob

wants to verify
that Alice signed
on the document

Digital signatures work as:

$(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}$ and broadcast vk

Alice (knows sk)

signature $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$

$\xrightarrow[(m,\sigma)]{}$

Bob (knows vk)

$\mathsf{Verify}(\mathsf{vk}, m, \sigma)$
$=$ accept (or reject)

Digital signatures

Digital signatures work as:

(sk, vk) $\leftarrow$ KeyGen and broadcast vk

Alice (knows sk)

Bob (knows vk)

signature $\sigma \leftarrow$ Sign(sk, m)

$\xrightarrow[(m, \sigma)]{}$

Verify(vk, m, σ)
= accept (or reject)

Necessary properties:

- **Correctness**:

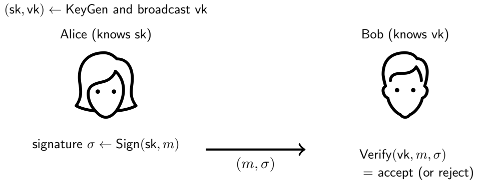
$$\mathsf{Verify}(\mathsf{vk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = \mathsf{accept}$$

- **Unforgeability**: No one else than Alice can make a new signature. More formally,

for a given verification key and some message-signature pairs, no adversary can forge a new valid signature.

Digital signatures

Digital signatures work as:



Necessary properties:

- **Correctness**:

$$\mathsf{Verify}(\mathsf{vk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = \mathsf{accept}$$

- **Unforgeability**: No one else than Alice can make a new signature.
  More formally,

  for a given verification key and some message-signature pairs, no adversary can forge a new valid signature.

Digital signatures

Digital signatures work as:

(sk, vk) $\leftarrow$ KeyGen and broadcast vk

Alice (knows sk)

Bob (knows vk)

signature $\sigma \leftarrow$ Sign(sk, m)

$\xrightarrow[(m,\sigma)]{}$

Verify(vk, m, σ) = accept (or reject)

Necessary properties:

- **Correctness**:

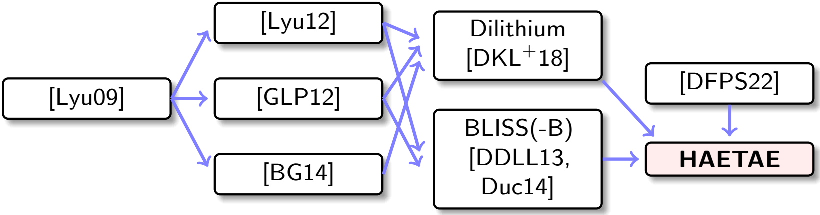
$$\mathsf{Verify}(\mathsf{vk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = \mathsf{accept}$$

- **Unforgeability**: No one else than Alice can make a new signature. More formally,

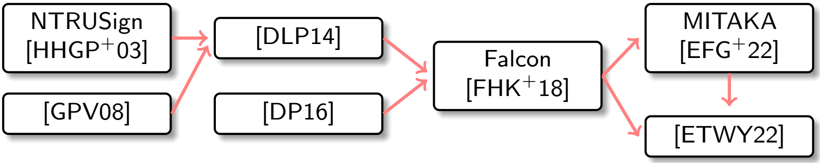
for a given verification key and some message-signature pairs, no adversary can forge a new valid signature.

Lattice-based signatures

Fiat-Shamir with abort



Hash-and-Sign



Lattice-based signatures

Fiat-Shamir with abort:

Key: ($\mathbf{s}$: small secret, $\mathbf{t} = \mathbf{A}\mathbf{s} \bmod q$: public)

Sign: $(c = H(\mathbf{A}\mathbf{y} \bmod q,\ m),\ \mathbf{z} = \mathbf{y} + c\mathbf{s})$ for short $\mathbf{y}$, with rejection sampling

Verify: check whether $c = H(\mathbf{A}\mathbf{z} - c\mathbf{t} \bmod q, m)$ and $\mathbf{z}$ is short.

Correctness of FSwA:

- $\mathbf{y}$, $\mathbf{s}$: short, and $c = H(\cdot)$: binary $\Rightarrow c\mathbf{s}$: short. $\Rightarrow \mathbf{z} = \mathbf{y} + c\mathbf{s}$: short.
- $\mathbf{A}\mathbf{z} - c\mathbf{t} = \mathbf{A}(\mathbf{y} + c\mathbf{s}) - c\mathbf{t} = \mathbf{A}\mathbf{y} + c(\mathbf{A}\mathbf{s} - \mathbf{t}) = \mathbf{A}\mathbf{y} \bmod q$.

Lattice-based signatures

Fiat-Shamir with abort:

Key: ($\mathbf{s}$: small secret, $\mathbf{t} = \mathbf{A}\mathbf{s} \bmod q$: public)

Sign: $(c = H(\mathbf{A}\mathbf{y} \bmod q,\ m),\ \mathbf{z} = \mathbf{y} + c\mathbf{s})$ for short $\mathbf{y}$, with rejection sampling

Verify: check whether $c = H(\mathbf{A}\mathbf{z} - c\mathbf{t} \bmod q, m)$ and $\mathbf{z}$ is short.

Correctness of FSwA:

- $\mathbf{y}$, $\mathbf{s}$: short, and $c = H(\cdot)$: binary $\Rightarrow c\mathbf{s}$: short. $\Rightarrow \mathbf{z} = \mathbf{y} + c\mathbf{s}$: short.
- $\mathbf{A}\mathbf{z} - c\mathbf{t} = \mathbf{A}(\mathbf{y} + c\mathbf{s}) - c\mathbf{t} = \mathbf{A}\mathbf{y} + c(\mathbf{A}\mathbf{s} - \mathbf{t}) = \mathbf{A}\mathbf{y} \bmod q.$

Lattice-based signatures

Fiat-Shamir with abort:

Key: ($\mathbf{s}$: small secret, $\mathbf{t} = \mathbf{A}\mathbf{s} \bmod q$: public)

Sign: $(c = H(\mathbf{A}\mathbf{y} \bmod q,\ m),\ \mathbf{z} = \mathbf{y} + c\mathbf{s})$ for short $\mathbf{y}$, with rejection sampling

Verify: check whether $c = H(\mathbf{A}\mathbf{z} - c\mathbf{t} \bmod q, m)$ and $\mathbf{z}$ is short.

Correctness of FSwA:

- $\mathbf{y}$, $\mathbf{s}$: short, and $c = H(\cdot)$: binary $\Rightarrow c\mathbf{s}$: short. $\Rightarrow \mathbf{z} = \mathbf{y} + c\mathbf{s}$: short.
- $\mathbf{A}\mathbf{z} - c\mathbf{t} = \mathbf{A}(\mathbf{y} + c\mathbf{s}) - c\mathbf{t} = \mathbf{A}\mathbf{y} + c(\mathbf{A}\mathbf{s} - \mathbf{t}) = \mathbf{A}\mathbf{y} \bmod q$.

Lattice-based signatures

Fiat-Shamir with abort:

Key: ($\mathbf{s}$: small secret, $\mathbf{t} = \mathbf{A}\mathbf{s} \bmod q$: public)

Sign: $(c = H(\mathbf{A}\mathbf{y} \bmod q,\ m),\ \mathbf{z} = \mathbf{y} + c\mathbf{s})$ for short $\mathbf{y}$, with rejection sampling

Verify: check whether $c = H(\mathbf{A}\mathbf{z} - c\mathbf{t} \bmod q, m)$ and $\mathbf{z}$ is short.

Unforgeability of FSwA:

- **key** is secure $\Leftarrow$ Module-LWE,
- signature $(c, \mathbf{z})$ **do not leak the secret** $\mathbf{s}$ due to rejection sampling,
- **no new signatures** can be sampled **without** $\mathbf{s}$ $\Leftarrow$ Module-SIS,

even in the use of quantum algorithms.

Lattice-based signatures

Fiat-Shamir with abort:

Key: ($\mathbf{s}$: small secret, $\mathbf{t} = \mathbf{A}\mathbf{s} \bmod q$: public)

Sign: $(c = H(\mathbf{A}\mathbf{y} \bmod q,\ m),\ \mathbf{z} = \mathbf{y} + c\mathbf{s})$ for short $\mathbf{y}$, with rejection sampling

Verify: check whether $c = H(\mathbf{A}\mathbf{z} - c\mathbf{t} \bmod q, m)$ and $\mathbf{z}$ is short.

Unforgeability of FSwA:

- **key** is secure $\Leftarrow$ Module-LWE,
- signature $(c, \mathbf{z})$ **do not leak the secret** $\mathbf{s}$ due to rejection sampling,
- **no new signatures** can be sampled **without** $\mathbf{s}$ $\Leftarrow$ Module-SIS,

even in the use of quantum algorithms.

Lattice-based signatures

Fiat-Shamir with abort:

Key: ($\mathbf{s}$: small secret, $\mathbf{t} = \mathbf{A}\mathbf{s} \bmod q$: public)

Sign: $(c = H(\mathbf{A}\mathbf{y} \bmod q,\ m),\ \mathbf{z} = \mathbf{y} + c\mathbf{s})$ for short $\mathbf{y}$, with rejection sampling

Verify: check whether $c = H(\mathbf{A}\mathbf{z} - c\mathbf{t} \bmod q, m)$ and $\mathbf{z}$ is short.

Unforgeability of FSwA:

- **key** is secure $\Leftarrow$ Module-LWE,
- signature $(c, \mathbf{z})$ **do not leak the secret** s due to rejection sampling,
- **no new signatures** can be sampled **without** s $\Leftarrow$ Module-SIS,

even in the use of quantum algorithms.

Lattice-based signatures

Fiat-Shamir with abort:

Key: ($\mathbf{s}$: small secret, $\mathbf{t} = \mathbf{A}\mathbf{s} \bmod q$: public)

Sign: $(c = H(\mathbf{A}\mathbf{y} \bmod q,\ m),\ \mathbf{z} = \mathbf{y} + c\mathbf{s})$ for short $\mathbf{y}$, with rejection sampling

Verify: check whether $c = H(\mathbf{A}\mathbf{z} - c\mathbf{t} \bmod q, m)$ and $\mathbf{z}$ is short.

Unforgeability of FSwA:

- **key** is secure $\Leftarrow$ Module-LWE,
- signature $(c, \mathbf{z})$ **do not leak the secret** s due to rejection sampling,
- **no new signatures** can be sampled **without** $\mathrm{s} \Leftarrow$ Module-SIS,

even in the use of quantum algorithms.

1. Post-Quantum Cryptography:
   - What is Cryptography?
   - Lattice hard problems

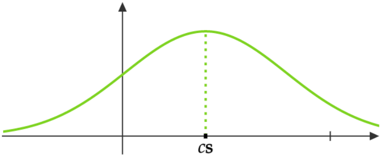
2. Digital Signatures:
   - What is a digital signature?
   - Lattice-based digital signatures
   - Rejection sampling

3. HAETAE:
   - Hyperball bimodal rejection sampling
   - Comparison to SotA lattice signatures
   - Current status

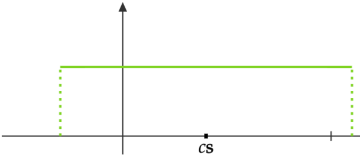
Rejection sampling

Leakage from $(c, \mathbf{z} = \mathbf{y} + c\mathbf{s})$?

With ∞ pairs of $(c, \mathbf{z} = \mathbf{y} + c\mathbf{s})$, we can collect $\mathbf{z}$ for the same c:



$y \leftarrow \mathcal{N}(0, \sigma^2)$



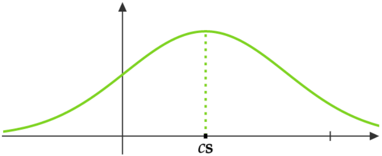
$y \leftarrow U[-a, a]$

$\Rightarrow$ Recover $\mathbf{s}$ from $c\mathbf{s}$.

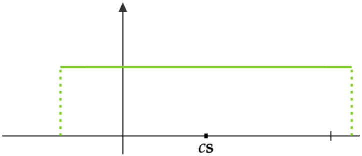
Rejection sampling

Leakage from $(c, \mathbf{z} = \mathbf{y} + c\mathbf{s})$?

With ∞ pairs of $(c, \mathbf{z} = \mathbf{y} + c\mathbf{s})$, we can collect $\mathbf{z}$ for the same c:



$y \leftarrow \mathcal{N}(0, \sigma^2)$



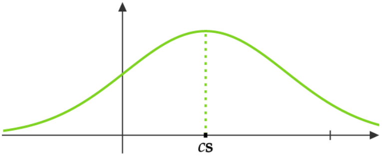
$y \leftarrow U[-a, a]$

$\Rightarrow$ Recover s from cs.

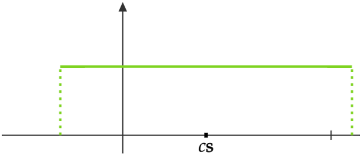
Rejection sampling

Leakage from $(c, \mathbf{z} = \mathbf{y} + c\mathbf{s})$?

With ∞ pairs of $(c, \mathbf{z} = \mathbf{y} + c\mathbf{s})$, we can collect $\mathbf{z}$ for the same c:



$y \leftarrow \mathcal{N}(0, \sigma^2)$



$y \leftarrow U[-a, a]$

$\Rightarrow$ Recover $\mathbf{s}$ from $c\mathbf{s}$.

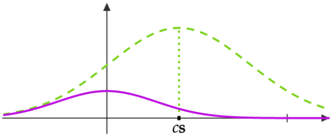
Rejection sampling

Rejection sampling

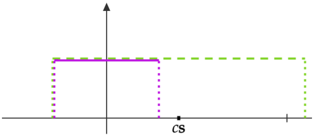
$$D_{\text{source}} = \{(c, \mathbf{z})\} \xrightarrow[\text{prob. } p(c,\mathbf{z})]{\text{reject with}} D_{\text{target}}$$

distribution of $(c, \mathbf{z})$, possibly leak $\mathbf{s}$

new distribution, independent of $\mathbf{s}$



$y \leftarrow \mathcal{N}(0, \sigma^2)$



$y \leftarrow U[-a, a]$

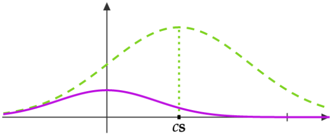
Rejection sampling

Rejection sampling

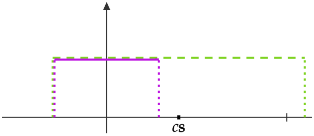
$D_{\text{source}} = \{(c, \mathbf{z})\}$ $\xrightarrow[\text{prob. } p(c,\mathbf{z})]{\text{reject with}}$ D_{target}

distribution of $(c, \mathbf{z})$, possibly leak $\mathbf{s}$

new distribution, independent of $\mathbf{s}$



$y \leftarrow \mathcal{N}(0, \sigma^2)$



$y \leftarrow U[-a, a]$

Rejection sampling

The **FSwA signatures** are commonly given as follows:

1. $\mathbf{y} \leftarrow D_0$
2. $c \leftarrow H(\mathbf{Ay}, m)$
3. $\mathbf{z} \leftarrow \mathbf{y} + c\mathbf{s}$
4. with probability $\frac{p_{\text{target}}(c,\mathbf{z})}{M \cdot p_{\text{source}}(c,\mathbf{z})}$, return $\sigma = (c, \mathbf{z})$, else go to step 1

M: bounding factor for the probability to be ≤ 1.

Final distribution $\sim D_{\text{target}}$.

Run-time $\propto M$.

Rejection sampling

The **FSwA signatures** are commonly given as follows:

1. $\mathbf{y} \leftarrow D_0$
2. $c \leftarrow H(\mathbf{Ay}, m)$
3. $\mathbf{z} \leftarrow \mathbf{y} + c\mathbf{s}$
4. with probability $\frac{p_{\text{target}}(c,\mathbf{z})}{M \cdot p_{\text{source}}(c,\mathbf{z})}$, return $\sigma = (c, \mathbf{z})$, else go to step 1

M: bounding factor for the probability to be ≤ 1.

Final distribution $\sim D_{\text{target}}$.

Run-time $\propto M$.

Rejection sampling

The **FSwA signatures** are commonly given as follows:

1. $\mathbf{y} \leftarrow D_0$
2. $c \leftarrow H(\mathbf{Ay}, m)$
3. $\mathbf{z} \leftarrow \mathbf{y} + c\mathbf{s}$
4. with probability $\frac{p_{\text{target}}(c,\mathbf{z})}{M \cdot p_{\text{source}}(c,\mathbf{z})}$, return $\sigma = (c, \mathbf{z})$, else go to step 1

M: bounding factor for the probability to be ≤ 1.

Final distribution $\sim D_{\text{target}}$.

Run-time $\propto M$.

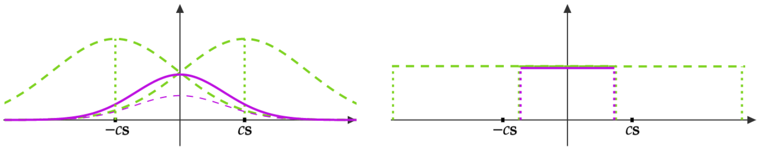
Bimodal rejection sampling

Run-time $\propto M$ ($\approx$ green area / purple area).

To decrease M, [DDLL13] uses

$$\mathbf{z} = \mathbf{y} + (-1)^b c\mathbf{s} \bmod 2q$$

instead of $\mathbf{z} = \mathbf{y} + c\mathbf{s} \bmod q$:



$y \leftarrow \mathcal{N}(0, \sigma^2)$ $y \leftarrow U[-a, a]$

Note, no change for the uniform case.

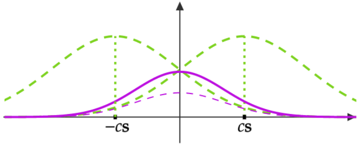
Bimodal rejection sampling

Run-time $\propto M$ ($\approx$ green area / purple area).

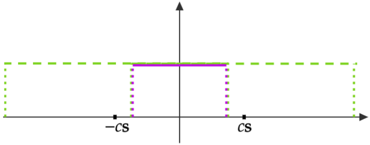
To decrease M, [DDLL13] uses

$$\mathbf{z} = \mathbf{y} + (-1)^b c\mathbf{s} \bmod 2q$$

instead of $\mathbf{z} = \mathbf{y} + c\mathbf{s} \bmod q$:



$y \leftarrow \mathcal{N}(0, \sigma^2)$



$y \leftarrow U[-a, a]$

Note, no change for the uniform case.

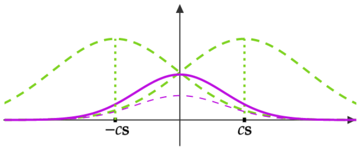
Bimodal rejection sampling

Run-time $\propto M$ ($\approx$ green area / purple area).

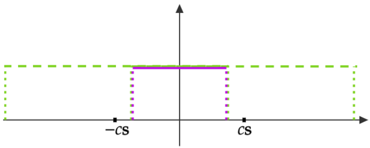
To decrease M, [DDLL13] uses

$$\mathbf{z} = \mathbf{y} + (-1)^b c\mathbf{s} \bmod 2q$$

instead of $\mathbf{z} = \mathbf{y} + c\mathbf{s} \bmod q$:



$y \leftarrow \mathcal{N}(0, \sigma^2)$



$y \leftarrow U[-a, a]$

Note, no change for the uniform case.

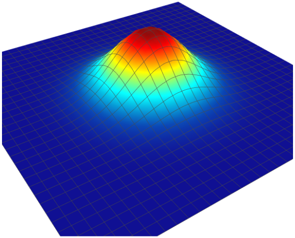
1. Post-Quantum Cryptography:
   - What is Cryptography?
   - Lattice hard problems

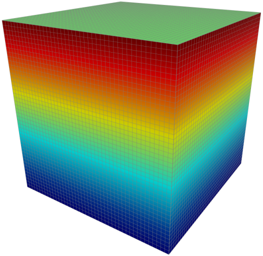
2. Digital Signatures:
   - What is a digital signature?
   - Lattice-based digital signatures
   - Rejection sampling

3. HAETAE:
   - Hyperball bimodal rejection sampling
   - Comparison to SotA lattice signatures
   - Current status

Hyperball bimodal rejection sampling

Previously, the randomness $\mathbf{y}$ was chosen from either discrete Gaussian or uniform hypercube[1].



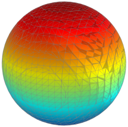


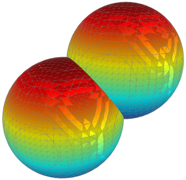
[1] The vectors $\mathbf{y}$ and $\mathbf{z}$ are high-dimensional vectors, so uniform in an interval is indeed a uniform hypercube.

Hyperball bimodal rejection sampling

We, instead, use **uniform hyperball** distribution for sampling $\mathbf{y}$ [DFPS22];

- to exploit optimal M,
- to reduce signature and verification key sizes,





based on the **bimodal approach** [DDLL13].

Hyperball bimodal rejection sampling

We, instead, use **uniform hyperball** distribution for sampling $\mathbf{y}$ [DFPS22];

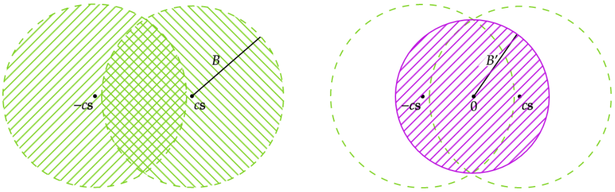
- to exploit optimal M,
- to reduce signature and verification key sizes,

based on the **bimodal approach** [DDLL13].

Hyperball bimodal rejection sampling

We reject $(c, \mathbf{z}) \sim D_\mathsf{s}$ (with p.d.f. p_s) to a target distribution D_t (with p.d.f. p_t), where

- p_s: uniform in hyperballs of radii B centered at $\pm c\mathbf{s}$
  - union of two large balls
- p_t: uniform in a smaller hyperball of radii B' centered at zero
  - a smaller ball in the middle

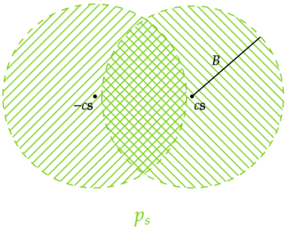


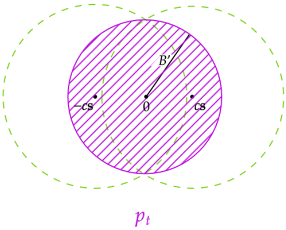
p_s p_t

Hyperball bimodal rejection sampling

We reject $(c, \mathbf{z}) \sim D_{\mathsf{s}}$ (with p.d.f. p_{s}) to a target distribution D_{t} (with p.d.f. p_{t}), where

- p_{s}: uniform in hyperballs of radii B centered at $\pm c\mathbf{s}$
  - union of two large balls
- p_{t}: uniform in a smaller hyperball of radii B' centered at zero
  - a smaller ball in the middle





Hyperball bimodal rejection sampling

- $p_{\mathsf{s}}(\mathbf{x}) = \frac{1}{2\cdot\mathsf{vol}(\mathcal{B}(B))} \cdot \chi_{\|\mathbf{z}-c\mathbf{s}\|<B} + \frac{1}{2\cdot\mathsf{vol}(\mathcal{B}(B))} \cdot \chi_{\|\mathbf{z}+c\mathbf{s}\|<B}$,
- $p_{\mathsf{t}}(\mathbf{x}) = \frac{1}{\mathsf{vol}(\mathcal{B}(B))} \cdot \chi_{\|\mathbf{z}\|<B'}$.

$$\Rightarrow p(\mathbf{x}) = \frac{p_{\mathsf{t}}(\mathbf{x})}{M \cdot p_{\mathsf{s}}(\mathbf{x})} = \frac{\chi_{\|\mathbf{z}\|<B'}}{\chi_{\|\mathbf{z}-c\mathbf{s}\|<B} + \chi_{\|\mathbf{z}+c\mathbf{s}\|<B}}$$

$$= \begin{cases} 0 & \text{if } \mathbf{z} \notin \mathcal{B}(B'), \\ 1/2 & \text{if } \mathbf{z} \in \mathcal{B}(B') \cap \mathcal{B}(B, c\mathbf{s}) \cap \mathcal{B}(B, -c\mathbf{s}), \\ 1 & \text{if } \mathbf{z} \in \mathcal{B}(B') \setminus (\mathcal{B}(B, c\mathbf{s}) \cap \mathcal{B}(B, -c\mathbf{s})), \end{cases}$$

for some $M > 0$.

Hyperball bimodal rejection sampling

- $p_{\mathsf{s}}(\mathbf{x}) = \frac{1}{2\cdot\mathsf{vol}(\mathcal{B}(B))} \cdot \chi_{\|\mathbf{z}-c\mathbf{s}\|<B} + \frac{1}{2\cdot\mathsf{vol}(\mathcal{B}(B))} \cdot \chi_{\|\mathbf{z}+c\mathbf{s}\|<B}$,
- $p_{\mathsf{t}}(\mathbf{x}) = \frac{1}{\mathsf{vol}(\mathcal{B}(B))} \cdot \chi_{\|\mathbf{z}\|<B'}$.

$$\Rightarrow p(\mathbf{x}) = \frac{p_{\mathsf{t}}(\mathbf{x})}{M \cdot p_{\mathsf{s}}(\mathbf{x})} = \frac{\chi_{\|\mathbf{z}\|<B'}}{\chi_{\|\mathbf{z}-c\mathbf{s}\|<B} + \chi_{\|\mathbf{z}+c\mathbf{s}\|<B}}$$

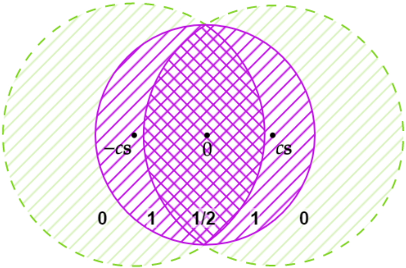
$$= \begin{cases} 0 & \text{if } \mathbf{z} \notin \mathcal{B}(B'), \\ 1/2 & \text{if } \mathbf{z} \in \mathcal{B}(B') \cap \mathcal{B}(B, c\mathbf{s}) \cap \mathcal{B}(B, -c\mathbf{s}), \\ 1 & \text{if } \mathbf{z} \in \mathcal{B}(B') \setminus (\mathcal{B}(B, c\mathbf{s}) \cap \mathcal{B}(B, -c\mathbf{s})), \end{cases}$$

for some $M > 0$.

Hyperball bimodal rejection sampling

That is, we return $\mathbf{x} = (c, \mathbf{z})$ with probability

- 0: if $\|\mathbf{z}\| \geq B'$,
- 1/2: else if $\|\mathbf{z} - c\mathbf{s}\| < B$ and $\|\mathbf{z} + c\mathbf{s}\| < B$,
- 1: otherwise.



1. Post-Quantum Cryptography:
   - What is Cryptography?
   - Lattice hard problems

2. Digital Signatures:
   - What is a digital signature?
   - Lattice-based digital signatures
   - Rejection sampling

3. HAETAE:
   - Hyperball bimodal rejection sampling
   - Comparison to SotA lattice signatures
   - Current status

Comparison to SotA lattice signatures.

For 120-bit classical security. Sizes are in bytes.

Scheme	*sig*	*vk*	KeyGen	Sign	
				sampling	rejection
Dilithium-2	2420	1312	fast	Hypercube	$\|\cdot\|_\infty < B$
Bliss-1024[2]	1700	1792	fast	dGaussian at 0	reject with prob. f(sk, Sig)
HAETAE120	1468	1056	fast	dHyperball at 0	$\|\cdot\|_2 < B$
Mitaka-512[3]	713	896	slow	dGaussian at 0 & intGaussian at $H(m)$	none
Falcon-512	666	897	slow	dGaussian at $H(m)$	none

Table: Comparison between different lattice-based signature schemes.

[2] modified Bliss (to ≥ 120 bit-security) in Dilithium paper.

[3] Mitaka-512 has 102 bits of security

1. Post-Quantum Cryptography:
   - What is Cryptography?
   - Lattice hard problems

2. Digital Signatures:
   - What is a digital signature?
   - Lattice-based digital signatures
   - Rejection sampling

3. HAETAE:
   - Hyperball bimodal rejection sampling
   - Comparison to SotA lattice signatures
   - Current status

Current Status

NIST PQC

- Competition for USA standard PQC schemes.
- HAETAE is one of the candidates in *Additional Signatures* track.

KPQC

- Competition for Korean standard PQC schemes.
- HAETAE is advanced to Round 2, one of four candidates in *Digital Signatures* track.

Thank you!

Any question?

References I

[BG14] Shi Bai and Steven D Galbraith.
An improved compression technique for signatures based on learning with errors.
In Cryptographers' Track at the RSA Conference, pages 28–47. Springer, 2014.

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky.
Lattice signatures and bimodal gaussians.
In Annual Cryptology Conference, pages 40–56. Springer, 2013.

[DFPS22] Julien Devevey, Omar Fawzi, Alain Passelègue, and Damien Stehlé.
On rejection sampling in lyubashevsky's signature scheme.
Cryptology ePrint Archive, Number 2022/1249, 2022.
To be appeared in Asiacrypt, 2022. https://eprint.iacr.org/2022/1249.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé.
Crystals-dilithium: A lattice-based digital signature scheme.
IACR Transactions on Cryptographic Hardware and Embedded Systems, pages 238–268, 2018.

[DLP14] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest.
Efficient identity-based encryption over ntru lattices.
In International Conference on the Theory and Application of Cryptology and Information Security, pages 22–41. Springer, 2014.

References II

[DP16] Léo Ducas and Thomas Prest.
Fast fourier orthogonalization.
In Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, pages 191–198, 2016.

[Duc14] Léo Ducas.
Accelerating bliss: the geometry of ternary polynomials.
Cryptology ePrint Archive, Paper 2014/874, 2014.
https://eprint.iacr.org/2014/874.

[EFG+22] Thomas Espitau, Pierre-Alain Fouque, François Gérard, Mélissa Rossi, Akira Takahashi, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu.
Mitaka: A simpler, parallelizable, maskable variant of.
In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 222–253. Springer, 2022.

[ETWY22] Thomas Espitau, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu.
Shorter hash-and-sign lattice-based signatures.
In Yevgeniy Dodis and Thomas Shrimpton, editors, Advances in Cryptology – CRYPTO, 2022.

References III

[FHK+18] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang.
Falcon: Fast-fourier lattice-based compact signatures over ntru.
Submission to the NIST's post-quantum cryptography standardization process, 36(5), 2018.

[GLP12] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann.
Practical lattice-based cryptography: A signature scheme for embedded systems.
In International Workshop on Cryptographic Hardware and Embedded Systems, pages 530–547. Springer, 2012.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan.
Trapdoors for hard lattices and new cryptographic constructions.
In Proceedings of the fortieth annual ACM symposium on Theory of computing, pages 197–206, 2008.

[HHGP+03] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H Silverman, and William Whyte.
Ntrusign: Digital signatures using the ntru lattice.
In Cryptographers' track at the RSA conference, pages 122–140. Springer, 2003.

References IV

[Lyu09] Vadim Lyubashevsky.
Fiat-shamir with aborts: Applications to lattice and factoring-based signatures.
In International Conference on the Theory and Application of Cryptology and Information Security, pages 598–616. Springer, 2009.

[Lyu12] Vadim Lyubashevsky.
Lattice signatures without trapdoors.
In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 738–755. Springer, 2012.

HAETAE description (high-level)

KeyGen(1^λ)

1: $\mathbf{A}_{\mathsf{gen}} \leftarrow \mathcal{R}_q^{k\times(\ell-1)}$ and $(\mathbf{s}_{\mathsf{gen}}, \mathbf{e}_{\mathsf{gen}}) \leftarrow S_\eta^{\ell-1} \times S_\eta^k$
2: $\mathbf{b} = \mathbf{A}_{\mathsf{gen}} \cdot \mathbf{s}_{\mathsf{gen}} + \mathbf{e}_{\mathsf{gen}} \in \mathcal{R}_q^k$
3: $\mathbf{A} = (-2\mathbf{b} + q\mathbf{j} \mid 2\mathbf{A}_{\mathsf{gen}} \mid 2\mathbf{Id}_k) \bmod 2q$ and write $\mathbf{A} = (\mathbf{A}_1 \mid 2\mathbf{Id}_k)$
4: $\mathbf{s} = (1, \mathbf{s}_{\mathsf{gen}}, \mathbf{e}_{\mathsf{gen}})$
5: if $\sigma_{\mathsf{max}}(\mathsf{rot}(\mathbf{s}_{\mathsf{gen}})) > \gamma$, then restart
6: Return $\mathsf{sk} = \mathbf{s}$, $\mathsf{vk} = \mathbf{A}$

Sign(sk, M)

1: $\mathbf{y} \leftarrow U(\mathcal{B}_{(1/N)\mathcal{R},(k+\ell)}(B))$
2: $c = H(\mathsf{HighBits}_{2q}^{\mathsf{hint}}(\mathbf{A}\lfloor\mathbf{y}\rceil, \alpha), \mathsf{LSB}(\lfloor y_0\rceil), M) \in \mathcal{R}_2$
3: $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2) = \mathbf{y} + (-1)^b c \cdot \mathbf{s}$ for $b \leftarrow U(\{0,1\})$
4: $\mathbf{h} = \mathsf{HighBits}_{2q}^{\mathsf{hint}}(\mathbf{A}\lfloor\mathbf{z}\rceil - qc\mathbf{j}, \alpha) - \mathsf{HighBits}_{2q}^{\mathsf{hint}}(\mathbf{A}_1\lfloor\mathbf{z}_1\rceil - qc\mathbf{j}, \alpha) \bmod^+ \frac{2(q-1)}{\alpha}$
5: if $\|\mathbf{z}\|_2 \geq B'$, then restart
6: if $\|2\mathbf{z} - \mathbf{y}\|_2 < B$, then restart with probability $1/2$
7: Return $\sigma = (\mathsf{Encode}(\mathsf{HighBits}(\lfloor\mathbf{z}_1\rceil, a)), \mathsf{LowBits}(\lfloor\mathbf{z}_1\rceil, a), \mathsf{Encode}(\mathbf{h}), c)$

Verify$(\mathsf{vk}, M, \sigma = (x, \mathbf{v}, h, c))$

1: $\tilde{\mathbf{z}}_1 = \mathsf{Decode}(x) \cdot a + \mathbf{v}$ and $\tilde{\mathbf{h}} = \mathsf{Decode}(h)$
2: $\mathbf{w} = \tilde{\mathbf{h}} + \mathsf{HighBits}_{2q}^{\mathsf{hint}}(\mathbf{A}_1 \tilde{\mathbf{z}}_1 - qc\mathbf{j}, \alpha) \bmod^+ \frac{2(q-1)}{\alpha}$
3: $w' = \mathsf{LSB}(\tilde{z}_0 - c)$
4: $\tilde{\mathbf{z}}_2 = \left[\mathbf{w} \cdot \alpha + w'\mathbf{j} - (\mathbf{A}_1 \tilde{z}_1 - qc\mathbf{j})\right] / 2 \bmod^{\pm} q$
5: $\tilde{\mathbf{z}} = (\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2)$
6: Return $(\ c = H(\mathbf{w}, w', M)\ ) \wedge (\ \|\tilde{\mathbf{z}}\| < B''\ )$