***GT-Grace***

*27 September 2022*

# A Search for Approximate Trapdoors in Lattice-Based Cryptosystem


TOHOKU UNIVERSITY

*Laboratory of Information Security*

Anaëlle Le Dévéhat

anaelle.le.devehat.s8@dc.tohoku.ac.jp

Supervisors :

- Prof. Shizuya, Hiroki
- Prof. Hasegawa, Shingo


***[GPV08] :*** *Le Dévéhat, A., Shizuya, H., Hasegawa, S. (2021). On the Higher-Bit Version of Approximate Inhomogeneous Short Integer Solution Problem. CANS 2021. Lecture Notes in Computer Science(), vol 13099. Springer, Cham. https://doi.org/10.1007/978-3-030-92548-2_14*

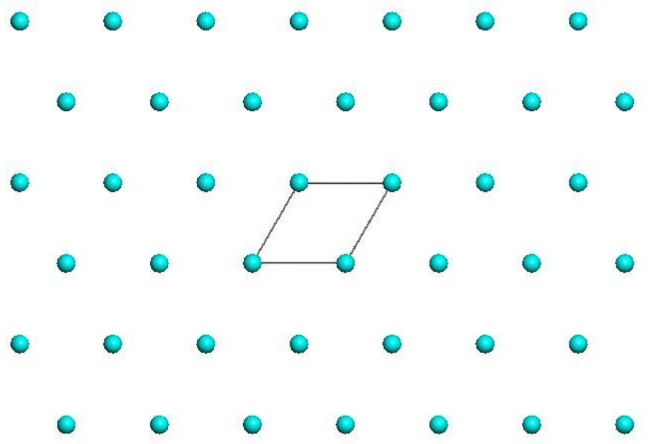
# Contents

# Lattice-based cryptography

**What ?**

A lattice is a Discrete additive subgroup of $\mathbb{R}^n$.

➔ We focus on integer lattices



**Why in cryptography ?**

- Simple and efficient : highly parallel ; elegant design
- Quantum-resistant
- Enjoys good average-case to worst-case hardness reductions
- Allows to construct some advanced cryptographic primitives

# Lattices problems

We base our hardness on average-case lattices problems that reducts to worst-case ones. Security of a construction is based on the hardness assumption of these underlying problems.

**(I)SIS :**

(Inhomogeneous) Short Integer Solution Problem

**LWE :**

Learning with errors Problem

Find a short vector in the q-ary lattice defined by integer matrix A as :

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

# Lattice trapdoors

**Lattice-based trapdoor functions**

for $A \leftarrow \mathbb{Z}_q^{n \times m}$ and with appropriate parameters,

| | |
|---|---|
| $f_A(x) = Ax \quad mod\ q$ | $g_A(x) = s^t A + e^t \quad mod\ q$ |
| ( "short" $x$ ) | ( "very short" $e$ ) |

➔ Use of a "strong" trapdoor to invert $f_A$ **(SIS)** and/or $g_A$ **(LWE)** permits the construction of various cryptographic primitives.

**Cryptographic functionality of such trapdoors**

- Solve worst-case hardness problems
- Sample from a discrete Gaussian distribution of "rather small" width, over any desired coset of the lattice :

$$\boxed{\Lambda_u^{\perp}(A) := \{\, x \in \mathbb{Z}^m : Ax = u\ (mod\ q)\ \} = f_A^{-1}(u)}$$



**Our focus :**

Lattice trapdoors and its application to "Hash-and-Sign" Signatures

5

# Gaussian distribution

- For any $s > 0$, define the **Gaussian function on $\mathbb{R}^n$**:

$$\forall x \in \mathbb{R}^n,\ \rho_s(x) = e^{-\pi||x||^2/s^2}$$

- For any $c \in \mathbb{R}^n$, real $s > 0$, and n-dimensional lattice $\Lambda$, define the **Discrete Gaussian distribution** $D_{\Lambda+c,s}$ as:

$$\forall x \in \Lambda + c,\ D_{\Lambda+c,s}(x) = \frac{\rho_s(x)}{\rho_s(\Lambda + c)}$$

- For any semi-definite $\Sigma = TT^t$, define the **Non-spherical Gaussian function on $\mathbb{R}^n$**:

$$\forall x \in span(T) = span(\Sigma),\ \rho_\Sigma(x) = e^{-\pi \boldsymbol{x}^t \boldsymbol{\Sigma}^+ \boldsymbol{x}}$$

# Prior works

## [GPV08]

Trapdoor : **short base** $S$ for $\Lambda^{\perp}(A)$

i.e $AS=0 \mod q$

- Formal proof of unforgeability in the random-oracle model
- Randomized approach : Gaussian sampler

Problems :

➔ Generation of A with S is slow and complicated

➔ Inefficient inversion algorithms

## [MP12] G-trapdoor

Trapdoor : $R$ → **Not** a base for $\Lambda^{\perp}(A)$

$$A\begin{pmatrix} R \\ I \end{pmatrix} = G \mod q$$

- Introduction of gadget matrix $G$ : **easy to invert** $f_G$
- Maps coset from $\Lambda^{\perp}(A)$ to cosets from $\Lambda^{\perp}(G)$ : more efficient gaussian sampler

Problem :

➔ Practical inefficiency due to large sizes (key and signature sizes for "Hash-and-Sign"signature)

## [CGM19] F-trapdoor

Trapdoor : **Approximate version of a G-trapdoor**

- Introduction of the Approximate setting
- Reduce considerably the key and signature sizes by allowing an error on the sampled signature.

Problem :

➔ Despite its optimization, key and signature sizes are still too large

# A comparison with NIST standardization process digital signatures candidates

| [MP12] | [CGM19] | qTesla : *rejection sampling approach* | Dilithium : *rejection sampling approach* | Falcon : *NTRU lattices* |
|---|---|---|---|---|
| 88-bit security :<br><br>Public key : 19.5 kB<br>Signature : 13.5 kB<br><br>128-bit security :<br><br>Public key : > 35 kB<br>Signature : > 25 kB | **88-bit security :**<br><br>**Public key : 5 kB**<br>**Signature : 4.45 kB**<br><br>184-bit security :<br><br>Public key : 11.25 kB<br>Signature : 9.38 kB | 128-bit security :<br><br>Public key : 4.03 kB<br>Signature : 3.05 kB | 121-bit security :<br><br>Public key : 1.32 kB<br>Signature : 2.42 kB | 133-bit security :<br><br>Public key : 0.90 kB<br>Signature : 0.66 kB |

Question : How to further downsize the public-key and signature for “hash-and-sign” signatures ?

# How to further downsize the public key and signature for “Hash-and-Sign” digital signatures ?

- To make “Hash-and-Sign” digital signatures from GPV line of work competitive and practical for post–quantum standardization

- Even though some methods are more advanced, they all suffer from downsides either in systems simplicity, running times, storage...

- ***As cryptanalysis of post-quantum cryptosystems is not yet well understood, it is essential to develop different schemes relying on different assumptions and/or construction methods.***

**Why ?**

# Our results

**1.** Definition of the higher-bit approximate ISIS problem

- Reduction to the ISIS problem
- Permits to discard low-weighted bits of coefficients in the matrix $A$ which defines Ajtai's function. (Downsize modulus)

**2.** An adaptation of **[CGM19]** trapdoor generation and preimage sampling algorithms to fit the "higher-bit" setting

- Public matrix $A$ belongs to $\mathbb{Z}_{q/b^d}^{n\times m}$ rather than $\mathbb{Z}_q^{n\times m}$
- Sampled preimage belongs to $\mathbb{Z}_{q/b^d}^{m}$ rather than $\mathbb{Z}_q^{m}$

# Our results

## **3.** Instantiation of hash-and-sign digital signature

- sEUF-CMA secure
- **Trade-off** between security and memory space :

  *We expect our construction to reduce the public key and signature sizes by about half at the expense of a reasonable drop in the security level.*

*Implementation*

Analysis :

➔ Our scheme is fit to obtain some **intermediate level of security** compared to those of [CGM19]

➔ Estimate **155-bit security level rather than 88-bit security** as in [CGM19] for **better key sizes** *(but bigger running times)*

# Our results

**4.** Combination of our work with a non-spherical Gaussian sampler (**[JHT21]**)

- New higher-bit approximate preimage sampling algorithm
- Instantiation of a **sEUF-CMA secure “Hash-and-Sign” digital signature**

*We expect this second construction to further reduce the signature size.*

*Implementation*

Analysis :

➔ Theoretical improvements in objects’ length bounds and in the digital signature security level.

➔ Very low practical improvement. We might assume that the higher-bit setting subsumes the optimizations brought in **[JHT21]**

# Contents

# Approximate ISIS problem

$\underline{ApproxISIS\ n,m,q,\alpha\ ,\beta}$ :

For any $n, m, q \in \mathbb{N}$ and $\alpha, \beta \in \mathbb{R}$, define the approximate inhomogeneous short integer solution problem $Approx.ISIS_{n,m,q,\alpha,\beta}$ as follows.
Given $A \in \mathbb{Z}_q^{nxm}$, $y \in \mathbb{Z}_q^{n}$ find a vector $x \in \mathbb{Z}_q^{m}$ such that $||x|| \leq \beta$ and there is a vector $z \in \mathbb{Z}^n$ satisfying :

$$||z|| \leq \alpha \text{ and } Ax = y + z \pmod q$$

- $LWE_{n,m,q,\theta,U(\mathbb{Z}_q),\chi} \leq_p Approx.ISIS_{n,m,q,\alpha,\beta}$
- $ISIS_{n,n+m,q,\beta} \geq_p Approx.ISIS_{n,m,q,\alpha+\beta,\beta}$
- $ISIS_{n,n+m,q,\alpha+\beta} \leq_p Approx.ISIS_{n,m,q,\alpha,\beta}$

# Approximate trapdoors

- Define the **Approximate gadget-matrix** $\boldsymbol{F}$ :

$$F := I_n \otimes f^t \in \mathbb{Z}^{n \times w}$$

where

$$f := (b^l, b^{l+1}, \ldots, b^{k-1})^t \in \mathbb{Z}_q^{(k-l)}$$

- Sample a **Public-Key** ***A*** with a **Secret-Key** $\mathcal{R}$ :
    - $\hat{A} \leftarrow \mathbb{U}(\mathbb{Z}_q^{n \times m}),\ \mathcal{R} \leftarrow \chi^{2n \times m}$
    - Let $\bar{A} := [I_n, \hat{A}]$ and form $A := [\bar{A} \mid F - \bar{A}\mathcal{R}] \in \mathbb{Z}_q^{n \times m}$

→ We can map short cosets representatives of $\Lambda^T(F)$ to approximate short cosets representatives of $\Lambda^T(A)$ using the approximate trapdoor $\mathcal{R}$

# Approximate trapdoors

**Algorithm 3:** APPROX.SAMPLEPRE.

**Input:** $(\mathbf{A}, \mathbf{R}, \mathbf{u}, s)$

**Output:** An approximate preimage of $\mathbf{u}$ for $\mathbf{A}$, $\mathbf{y} \in \mathbb{Z}^m$.

1. Sample a perturbation $\mathbf{p} \leftarrow D_{\mathbb{Z}^m, \sqrt{\Sigma_p}}$.
2. Form $\mathbf{v} = \mathbf{u} - \mathbf{A}\mathbf{p} \in \mathbb{Z}_q^n$.
3. Sample the approximate gadget preimage $\mathbf{z} \in \mathbb{Z}^{n(k-l)}$ as $\mathbf{z} \leftarrow \text{GSAMP.CUT}(\mathbf{v}, \sigma)$.
4. Form $\mathbf{y} := \mathbf{p} + \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \mathbf{z} \in \mathbb{Z}^m$.
5. **return** $\mathbf{y}$.

**Generalization** of the **perturbation based** discrete Gaussian sampler by Micciano and Peikert

*The perturbation p* :

$$\Sigma_p := s^2\mathbf{I} - \sigma^2 \begin{bmatrix} \mathbf{R}\mathbf{R}^t & \mathbf{R}^t \\ \mathbf{R} & \mathbf{I} \end{bmatrix}$$

Where

$$\sigma \geq \eta_\varepsilon(\Lambda^\perp(\mathbf{G}))$$

**Samples** an approximate preimage y of u from a **spherical** discrete Gaussian

# Contents

# Our idea

**G-trapdoor : exact trapdoor**

$$G = I_n \otimes g^t \in \mathbb{Z}_q^{n\mathsf{x}nk}$$

where $g^t = (1, b, \ldots, b^{k-1})$ and $k = [\log_b q]$

Truncation
of lower bits

**F-trapdoor : approximate trapdoor**

$$F = I_n \otimes f^t \in \mathbb{Z}_q^{n\mathsf{x}n(k-l)}$$

where $f^t = (b^l, b^{l+1}, \ldots, b^{k-1})$ and $k = [\log_b q]$

**We extend this idea :**

- We can interpret approximate gadget matrix $F$ as an object from $b^l \times \mathbb{Z}_{q/b^l}^{n\times nk}$
- Generalization of this approximation on all objects in the signature scheme

⇨ **Truncation of CGM scheme to downsize the modulus $q$ to $q/b^d$ $(d < l)$**

# Higher-bit approximate ISIS

<u>$H.ApproxISIS\ n,m,q,d,\alpha,\beta$</u> :

Given $A \in \mathbb{Z}_{q/b^d}^{n\times m}$ , $y \in \mathbb{Z}_q^n$, find a vector $x \in \mathbb{Z}^m$ such that $||x|| \le \beta$ and there is a vector $z \in \mathbb{Z}^n$ satisfying :

$$||z|| \le \alpha \text{ and } b^d Ax = y + z \ (mod\ q)$$

Specific to this problem

error due to approximation (same as in [CGM19])

$$ISIS_{n,n+m,q,\beta} \ge_p H.Approx.ISIS_{n,m,q,d,\alpha+\beta,\beta}$$
$$H.Approx.ISIS_{n,m,q,d,\alpha,\beta} \ge_p ISIS_{n,n+m,q,\alpha+(\sqrt{n}b^d+1)\beta}$$

Proven via a reduction to the Approximate ISIS from [CGM19]

*greater reduction loss*

# Higher-bit setting construction

**Change in Public matrix and sampled preimage**
*(For a syndrome u)*

[CGM19] :

Trapdoor = $R$

$\simeq G^H$
[MP12]

**PM** = $A_0 := [\bar{A} \mid F - \bar{A}R]$
**Prei** = $y_0 \leftarrow$ *Gaussian Sampler*

generalization

This work : $q = b^k$

Trapdoor = $R$ ; **PM** = $A_0{}^H / b^d$

**Prei** = $y_0 \ (mod\ b^{k-d})$

**Impact on the error term**

Define $e_0$ and $e_{new}$ as the following :

$e_0 = u - A_0 y_0 \ (mod\ q);$

$e_{new} = A_0{}^L y_0 \ (mod\ q);$

➔ We find that :

$$\boxed{e = e_0 + e_{new} \ (mod\ q)}$$

*due to our modification on A*

We adapt the trapdoor generation and preimage sampling algorithms from [CGM19] to force them into the higher-bit setting

$A^L = A\ (mod\ b^d)$
$A^H = A - A^L$

# "Hash-and-Sign" signature scheme

*The key-generation algorithm samples* $A \in \mathbb{Z}_{q/b^d}^{n\times m}$ *together with its* ($\alpha$ ; $\beta$)*-approximate trapdoor* $R$ *and the matrix* $A_0^L \in \mathbb{Z}_{b^d}^{n\times m}$

$sk$ = trapdoor $R$
$A_0^L$

1. Gets the hash of a message $\mu$
2. Uses the Gaussian sampler to get an approximate preimage $y$ for $H(\mu)$ by Ajtai function defined w.r.t $b^d A$
3. Outputs $y$ as signature

$pk = A$
$\alpha$ ; $\beta$

Checks if :

1. $\|y\| \leq \beta$
2. $\| b^d Ax - H(\mu)\| \leq \alpha$

If so, it accepts.

**sEUF-CMA secure** assuming the hardness of

$SIS_{n,n+m,q,2[\alpha+(\sqrt{n}\,b^d+1)\beta]}$ and $LWE_{n,n,q,\chi,U(\mathbb{Z}_q),\chi}$

# Contents

# A trade-off between size and security

| | [CGM19] | *This work* |
|---|---|---|
| Norm of a short solution in the underlying SIS problem | $2(s\sqrt{m} + b^l\sigma\sqrt{n})$ | $2(s\sqrt{m} + b^l\sigma\sqrt{n}) + 4\sqrt{nm}\, b^d s$ |
| Signature size (in bits) | $m \times k \times log_2(b)$ | $m \times (k - d) \times log_2(b)$ |
| Public key size (in bits) | $m \times n \times k \times log_2(b)$ | $m \times n \times (k - d) \times log_2(b)$ |

$n$ : security parameter
$m$ : vector dimension
$b$ : base
$s, \sigma$ : Gaussian distributions widths

# Implementation results

| | $F$-trapdoor [CGM19] | $F$-trapdoor [CGM19] | This work | This work | This work |
|---|---|---|---|---|---|
| $n$ | 512 | 1024 | 512 | 1024 | 1024 |
| $k = \lceil log_b q \rceil$ | 8 | 9 | 16 | 16 | 9 |
| $m$ | 3072 | 6144 | 3584 | 7168 | 6144 |
| $b$ | 4 | 4 | 2 | 2 | 4 |
| $l$ | 4 | 5 | 11 | 11 | 5 |
| $d$ | - | - | 11 | 11 | 5 |
| $\|\|x\|\|_2$ | 138244.3 | 296473.0 | 1072.2 | 1535.5 | 11495.9 |
| $\|\|e\|\|_2$ | 20627.9 | 1502259.7 | 428806.9 | 607601.6 | 2452040.3 |
| ***PK* (kB)** | **5.12** | **11.52** | **1.92** | **3.84** | **5.12** |
| ***Sig* (kB)** | **4.5** | **9.4** | **2.25** | **4.5** | **6.1** |
| *LWE* | 104.7 | 192.7 | 104.7 | 192.7 | 192.7 |
| ***AISIS*** | **87.8** | **183.7** | **75.0** | **155.4** | **140.5** |

: n= 512

: n=1024

Advantages :

➔ **Better security level** for **better Public key and Signature sizes** at the expense of a *higher security parameter* $n$.

➔ Allows to obtain **different security levels** with more appropriate public key and signature sizes. *(for a same security parameter n)*

Disadvantage :

➔ To achieve more than 88-bit security, we **set** ***n=1024*** which can lead to **longer running times.** (Even bigger for more than 155-bit security.)

# A comparison with NIST standardization process digital signatures candidates

| **This work** | **[CGM19]** | **qTesla :** *rejection sampling approach* | **Dilithium :** *rejection sampling approach* | **Falcon :** *NTRU lattices* |
|---|---|---|---|---|
| **75-bit security :**<br><br>**Public key : 1.92 kB**<br>**Signature : 2.25 kB**<br><br>**155-bit security :**<br><br>**Public key : 3.84 kB**<br>**Signature : 4.5 kB** | **88-bit security :**<br><br>**Public key : 5 kB**<br>**Signature : 4.45 kB**<br><br>184-bit security :<br><br>Public key : 11.25 kB<br>Signature : 9.38 kB | 128-bit security :<br><br>Public key : 4.03 kB<br>Signature : 3.05 kB | 121-bit security :<br><br>Public key : 1.32 kB<br>Signature : 2.42 kB | 133-bit security :<br><br>Public key : 0.9 kB<br>Signature : 0.66 kB |

Result : We get ***pk*** and ***sig*** sizes closer (or even of same level) to those of NIST 2-round standardization process digital signatures.

# Contents

# Recent related work (2021)

**[JHT21]**

Construction : **Modified version of [CGM19]**

$D_{\mathbb{Z}^m,s}$

Spherical preimage Gaussian sampler

$$\Sigma = s^2 I_m$$

$D_{\mathbb{Z}^m,\sqrt{\Sigma}}$

**Non-Spherical** preimage Gaussian sampler

$$\Sigma = s_0{}^2 I_{2n} \oplus s_1{}^2 I_{kn}$$

*More precise distribution*
*→ smaller signatures*

*Different distortion on the sampled 2n first entries than on kn last entries*

Problems :

- ➔ Signature size still not competitive
- ➔ No impact on Public key

# New Higher-bit setting construction

**Change in sampled preimage**
*(For a syndrome u)*

[JHT21] :

*same as* [CGM19]

Trapdoor = $R$

**PM** = $A_0 := [\bar{A} \mid F - \bar{A}R]$
**Prei** = $y_0 \leftarrow$ **Non-spherical *Gaussian Sampler***

generalization

This work : $q = b^k$

Trapdoor = $R$ ; **PM** = $A_0^H / b^d$

**Prei** = $y_0 \ (mod\ b^{k-d})$

We adapt the preimage sampling algorithm from [JHT21] to force it into the higher-bit setting

$A^L = A \ (mod\ b^d)$
$A^H = A - A^L$

# New “Hash-and-Sign” signature scheme

*The key-generation algorithm samples* $A \in \mathbb{Z}_{q/b^d}^{n\times m}$ *together with its* ($\alpha$ ; $\beta$)*-approximate trapdoor* $R$ *and the matrix* $A_0^L \in \mathbb{Z}_{b^d}^{n\times m}$

$sk$ = trapdoor $R$

$A_0^L$

1. Gets the hash of a message $\mu$

2. Uses the **<u>Non-spherical Gaussian sampler</u>** to get an approximate preimage $y$ for $H(\mu)$ by Ajtai function defined w.r.t $b^d A$

3. Outputs $y$ as signature

$pk = A$

$\alpha ; \beta$

Checks if :

1. $\|y\| \leq \beta$

2. $\| b^d Ax - H(\mu)\| \leq \alpha$

If so, it accepts.

**sEUF-CMA secure** assuming the hardness of

$SIS_{n,n+m,q,2[\alpha + (\sqrt{n}\, b^d + 1)\beta]}$ and $LWE_{n,n,q,\chi,U(\mathbb{Z}_q),\chi}$

# Better theoretical length bounds

| | **Construction 2** *Non-spherical Gaussian sampler* | **Construction 1** *Spherical Gaussian sampler* |
|---|---|---|
| signature term $\boldsymbol{y}$ | $s_0\sqrt{2n} + s_1\sqrt{kn}$ | $s\sqrt{m}$ |
| error term $\boldsymbol{e}$ | $b^l\sigma\sqrt{n} + nb^d(s_0\sqrt{2} + s_1\sqrt{k})$ | $b^l\sigma\sqrt{n} + \sqrt{nm}b^d s$ |

$n$ : security parameter
$m$ : vector dimension
$b$ : base
$s, \sigma$ : Gaussian distributions widths

Expectations :

- ➔ Better security
- ➔ Better practical signature size

# Implementation results

| | Construction 1 | [JHT21] | Construction 2 | Construction 2 |
|---|---|---|---|---|
| $n$ | 1024 | 1024 | 1024 | 1024 |
| $k = \lceil log_b q \rceil$ | 16 | 9 | 16 | 9 |
| $m$ | 7168 | 6144 | 7168 | 6144 |
| $b$ | 2 | 4 | 2 | 4 |
| $l$ | 11 | 5 | 11 | 5 |
| $d$ | 11 | - | 11 | 5 |
| $\|\|x\|\|_2$ | 1535.5 | 536010 | 1544.0 | 12732.6 |
| $\|\|e\|\|_2$ | 607601.6 | 173254 | 603592.8 | 2448537.1 |
| ***PK* (kB)** | **3.84** | **11.25** | **3.84** | **5.12** |
| ***Sig* (kB)** | **4.5** | **5.75** | **4.4** | **5.50** |
| *LWE* | 192.7 | 218.0 | 192.7 | 192.7 |
| ***AISIS*** | **155.4** | **168.82** | **155.4** | **140.5** |

: previous constructions

: new construction

Analysis :

➔ **Very small optimization** obtained in the signature size ( about $0.1\ kB$).

➔ **No gain** in the security level (same signature norm).

➔ However, using the **higher-bit setting brings important improvement** to the original scheme from [JHT21].

Possible explanation :

Our bitwise optimization already removes the unnecessary information in the sampled signature. Thus, there is no need for a more precise Gaussian sampler.

# Conclusion

- Definition of the Higher-bit approximate ISIS. It can **downsize the modulus** at the price of a **trade-off** between sizes and security level.

- *<u>For a same security parameter,</u>* our setting brings **optimized objects with different levels of security** than in prior works.

  *<u>For a higher security parameter,</u>* we achieve a **win-win scenario** and obtain better sizes along with better security level (but higher running time).

SIGNATURE SCHEME

- Adaptation of the higher-bit setting with a **non-spherical Gaussian sampler** : Better theoretical objects norms .

# Future works

01 **Improve the reduction loss in the Higher-bit Approximate ISIS problem**

02 **Construct a more efficient digital signature implementation code**

- ➔ In this work, our implementation is only a tool for the sake of comparison.

03 **Explore the possible applications of the higher-bit approximate setting in other advanced lattice cryptosystems**

- ➔ Extend the Bonsaï techniques in the approximate setting.

TOHOKU
UNIVERSITY

*Laboratory of Information Security*

# *Thank you for listening !*

# *Any questions ?*

Anaëlle Le  Dévéhat

anaelle.le.devehat.s8@dc.tohoku.ac.jp

# Bibliography


**[GPV08] :** Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. STOC '08, ACM (2008). https://doi.org/10.1145/1374376.1374407

**[MP12] :** Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: EUROCRYPT 2012. vol. LNCS 7237, pp. 700-718. Springer (2012). https://doi.org/10.1007/978-3-642-29011-4 41

**[CGM19] :** Chen, Y., Genise, N., Mukherjee, P.: Approximate trapdoors for lattices and smaller hash-and-sign signatures. In: ASIACRYPT 2019. vol. LNCS 11923, pp. 3-32. Springer (2019). https://doi.org/10.1007/978-3-030-34618-8 1

**[JHT21] :** Jia, H., Hu, Y., Tang, C. : Lattice-based hash-and-sign signatures using approximate trapdoor, revisited. In : IET Information Security. John Wiley Sons Ltd (2021). https://doi.org/10.1049/ise2.12039

# Parameters

**Parameters we choose : *(n, b, q ,l ,d)***

**Implementation**

- ❑ $m = 2n + n(k - l)$
- ❑ $\sigma = \text{sqrt}(b^2 + 1)\log_2(n)$
- ❑ $s = (s_1\_R + 1)\sigma$
- ❑ $\chi = D_{\mathbb{Z}^m,\tau}$ where $\tau = 2.6$ or $2.8$

**Theoretical conditions**

- ❑ q : power of b
- ❑ $q > n^c$ where $c \geq 2$
- ❑ $n > 128$
- ❑ $0 < l < d$
- ❑ n : power of 2
- ❑ $\sigma = \text{sqrt}(b^2 + 1)\Omega(\text{sqrt}(\log_2(n)))$
- ❑ $\chi$ is a distributions such that the associated LWE problem is hard

# Apply the higher-bit setting to LWE

**Definition (LWE Assumption [Reg05]).** *Let $\lambda$ be the security parameter, $n = n(\lambda), m = m(\lambda), q = q(\lambda)$ be integers and let $\chi = \chi(\lambda)$ be a distribution over $\mathbb{Z}_q$. The* $\mathrm{LWE}_{n,m,q,\chi}$ *assumption says that, if we choose $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ then the following distributions are computationally indistinguishable:*

$$(\mathbf{A}, \mathbf{A}\cdot\mathbf{s} + \mathbf{e}) \overset{\text{comp}}{\approx} (\mathbf{A}, \mathbf{u}).$$

$$\downarrow$$

**Definition (LWR [BPR12]).** *Let $\lambda$ be the security parameter, $n = n(\lambda), m = m(\lambda), q = q(\lambda), p = p(\lambda)$ be integers. The* $\mathrm{LWR}_{n,m,q,p}$ *problem states that for $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ the following distributions are computationally indistinguishable:* $(\mathbf{A}, \lfloor \mathbf{A}\cdot\mathbf{s}\rceil_p) \overset{\text{comp}}{\approx} (\mathbf{A}, \lfloor \mathbf{u}\rceil_p)$.

$$\lfloor\cdot\rceil_p \;:\; \mathbb{Z}_q \to \mathbb{Z}_p \;:\; x \mapsto \lfloor (p/q)\cdot x\rceil$$

# Worst-case hardness

**Table 1.** Comparing the three families of SVP and CVP solvers.

| | Time complexity upper bound | Space complexity upper bound | Remarks |
|---|---|---|---|
| Sec. 3 | $2^{2n+o(n)}$ | $2^{n+o(n)}$ | Deterministic |
| Sec. 4, SVP | $2^{2.465n+o(n)}$ | $2^{1.325n+o(n)}$ | Monte-Carlo |
| Sec. 4, CVP | $(2+1/\varepsilon)^{O(n)}$ | $(2+1/\varepsilon)^{O(n)}$ | Monte-Carlo<br>solves $(1+\varepsilon)$-CVP only |
| Sec. 5, SVP | $n^{n/(2e)+o(n)}$ | $\mathcal{P}oly(n)$ | Deterministic |
| Sec. 5, CVP | $n^{n/2+o(n)}$ | $\mathcal{P}oly(n)$ | Deterministic |