Quantum-Safe HIBE: Does It Cost a Latte?

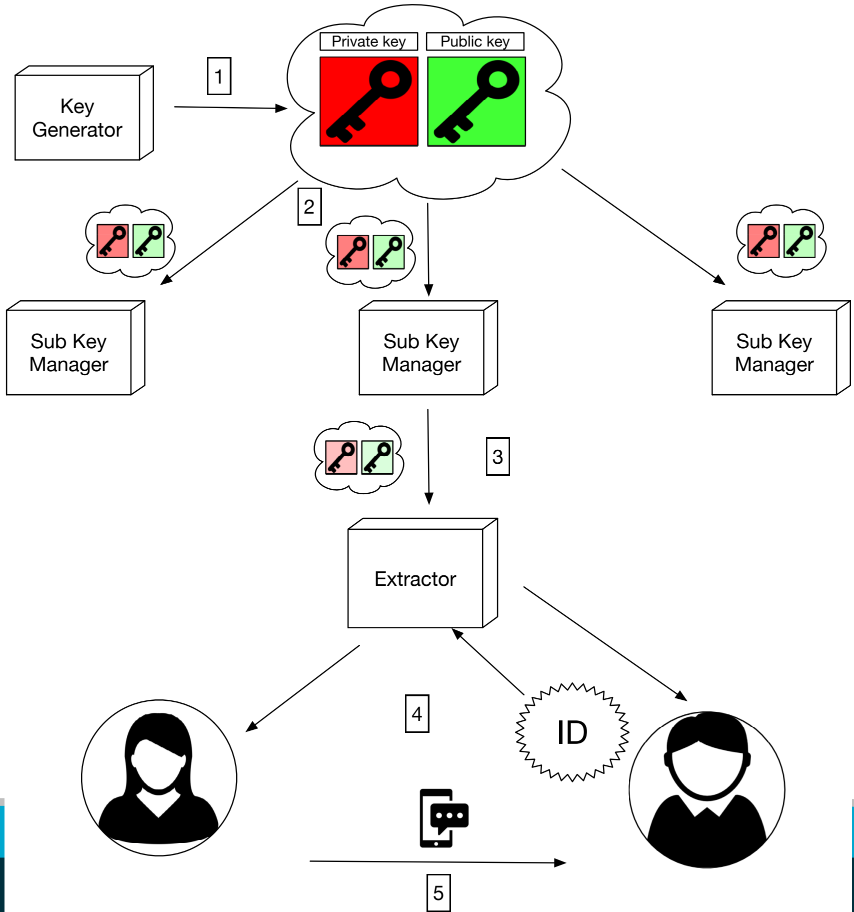
Raymond K. Zhao[1,3], Sarah McCarthy[2,4], Ron Steinfeld[3], Amin Sakzad[3], and Máire O'Neill[4]

[1]CSIRO's Data61; [2]University of Waterloo; [3]Monash University; [4]Queen's University Belfast

whoami

- PostDoc in the CSIRO's Data61.
- Before joining the CSIRO: PhD candidate, then Research Assistant in the Monash University.
- Research interest: Implementation of Post-quantum Cryptography (PQC).

Hierarchical Identity-based Encryption (HIBE)



Hierarchical Identity-based Encryption (HIBE)

1. KeyGen: The master key generator establishes the master public and private keys.
2. Delegate: Through a delegation function, the master key generator creates a public/private key pair for the sub-key manager. This gives it the ability to delegate further key pairs, and extract user private keys at that level.
3. Delegate: The sub-key manager delegates a further public/private key to the next level of the hierarchy.
4. Extract: The extractor uses their public/private key pair to extract and share user public/private keys, as in the single-level IBE scheme.
5. Encrypt/Decrypt: Encryption/decryption works as a regular encryption scheme.

Latte Post-quantum HIBE

- DLP IBE [DLP14] based on the NTRU lattice + Lattice basis delegation [CHKP10].
- Endorsed by the European Telecommunications Standards Institute (ETSI) [ETS19].
- However, only the Encrypt/Decrypt were implemented and evaluated in [ETS19].

Our contributions:

- First complete optimised practical implementation and benchmarking of Latte.
- Precision analysis of Latte.

Preliminaries

Definition 1 (Lattice)**.** An n-dimension lattice $\Lambda(\mathbf{B})$ is the set of all integer linear combinations of some basis set $\mathbf{B}$, where $\mathbf{B} = \{\mathbf{b}_i\}_{i=0}^{n-1} \subseteq \mathbb{R}^n$ and $\mathbf{b}_0, \ldots, \mathbf{b}_{n-1}$ are linearly independent: $\Lambda(\mathbf{B}) := \left\{\sum_{i=0}^{n-1} c_i \mathbf{b}_i : c_i \in \mathbb{Z}\right\}$.

Definition 2 (NTRU Lattice [DLP14])**.** Let q be a positive integer. Let polynomial ring $\mathfrak{R} := \mathbb{Z}[x]/\langle x^N + 1\rangle$. Let $\mathbf{f}, \mathbf{g} \in \mathfrak{R}$ and $\mathbf{h} := \mathbf{g}/\mathbf{f} \bmod q$. The NTRU lattice associated to $\mathbf{h}$ and q is $\Lambda_{\text{NTRU}} := \{\mathbf{x} \in \mathfrak{R}^2 : \mathbf{x} \cdot (1, \mathbf{h}) = \mathbf{0} \bmod q\}$.

Definition 3 (Discrete Gaussian)**.** Let $\rho_{\mathbf{c},\sigma}(\mathbf{x}) := \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2}\right)$ be the n-dimensional (continuous) Gaussian function on $\mathbb{R}^n$ with center $\mathbf{c} \in \mathbb{R}^n$ and standard deviation σ. We denote the discrete Gaussian distribution on lattice Λ with center $\mathbf{c} \in \mathbb{R}^n$ and standard deviation σ by $\mathcal{D}_{\Lambda,\mathbf{c},\sigma}(\mathbf{x}) := \frac{\rho_{\mathbf{c},\sigma}(\mathbf{x})}{\sum_{\mathbf{k}\in\Lambda} \rho_{\mathbf{c},\sigma}(\mathbf{k})}$.

Note: Λ is omitted when $\Lambda = \mathbb{Z}$; $\mathbf{c}$ is omitted when $\mathbf{c} = \mathbf{0}$.

NTRU Lattice Trapdoor

- Hardness Assumption (informal): Given a long (in terms of the Euclidean norm) basis $\mathbf{B}_{\text{long}}$ of Λ_{NTRU}, it is hard to find a short basis $\mathbf{B}_{\text{short}}$ of Λ_{NTRU} (equivalent to finding short lattice vectors).
- Assume $\mathbf{f}, \mathbf{g}$ are short. For Λ_{NTRU} associated to $\mathbf{h} = \mathbf{g}/\mathbf{f} \bmod q \in \mathbb{Z}_q^N$, we have [DLP14]:

$$\mathbf{B}_{\text{long}} := \begin{bmatrix} -\mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{0}_N \end{bmatrix}, \mathbf{B}_{\text{short}} := \begin{bmatrix} \mathcal{A}(\mathbf{g}) & -\mathcal{A}(\mathbf{f}) \\ \mathcal{A}(\mathbf{G}) & -\mathcal{A}(\mathbf{F}) \end{bmatrix},$$

for some sufficiently short (in the same order as $\mathbf{f}, \mathbf{g}$) $\mathbf{F}, \mathbf{G} \in \Re$ such that $\det \left[\begin{smallmatrix} \mathbf{g} & -\mathbf{f} \\ \mathbf{G} & -\mathbf{F} \end{smallmatrix}\right] = q$, i.e. $\mathbf{fG} - \mathbf{gF} = q \bmod x^N + 1$, where $\mathcal{A}(\mathbf{f})$ refers to the anti-circulant matrix associated with polynomial $\mathbf{f}$:

$$\mathcal{A}(\mathbf{f}) = \begin{bmatrix} \mathbf{f}_0 & \mathbf{f}_1 & \dots & \mathbf{f}_{N-1} \\ -\mathbf{f}_{N-1} & \mathbf{f}_0 & \dots & \mathbf{f}_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{f}_1 & -\mathbf{f}_2 & \dots & \mathbf{f}_0 \end{bmatrix}.$$

NTRU Lattice Trapdoor (cont.)

Trapdoor function [GPV08]: Given basis $\mathbf{B}$ of Λ_{NTRU}, sample short $\mathbf{v} \hookleftarrow \mathcal{D}_{\Lambda_{\mathsf{NTRU}},\mathbf{c},\sigma}$.

- The minimal σ one can sample has: $\sigma_{min} \approx ||\tilde{\mathbf{B}}|| \propto ||\mathbf{B}||$, where $\tilde{\mathbf{B}}$ is the Gram-Schmidt orthogonalised basis of $\mathbf{B}$ (constant factors in σ are omitted).
- For small $\sigma \approx ||\tilde{\mathbf{B}}_{\mathsf{short}}||$:
  - Easy to sample given $\mathbf{B}_{\mathsf{short}}$.
  - Hard to sample given $\mathbf{B}_{\mathsf{long}}$ ($||\tilde{\mathbf{B}}_{\mathsf{long}}|| \gg ||\tilde{\mathbf{B}}_{\mathsf{short}}||$).
  - Easy to verify $\mathbf{v}$ is short and $\mathbf{v} \in \Lambda_{\mathsf{NTRU}}$.
- $\mathbf{B}_{\mathsf{short}}$ is the trapdoor.

Applications: Falcon signature [PFH+17], DLP IBE [DLP14], Latte HIBE [ETS19], ...

We now introduce (our modified) Latte functions.

Latte Function: KeyGen

Essentially the NTRU lattice trapdoor generation.

1. (*) $\mathbf{f}, \mathbf{g} \hookleftarrow \mathcal{D}_{\sigma_0}^N$ for $\sigma_0 \approx \sqrt{q/2N}$ [DLP14].
2. If $||\tilde{\mathbf{B}}_{\text{short}}|| > \sigma_0\sqrt{2N}$, goto Step 1.
   - Can be done *before* Step 3 [DLP14].
3. (*) Find sufficiently short $\mathbf{F}, \mathbf{G}$ such that $\mathbf{fG}-\mathbf{gF} = q \mod x^N+1$. If unable to find, goto Step 1.
4. $\mathbf{h} := \mathbf{g}/\mathbf{f} \mod q$. If $\mathbf{f}$ is irrevertible, goto Step 1.

Master Public Key: $\mathbf{h}$ (essentially $\mathbf{B}_{\text{long}}$).

Master Private Key: $\mathbf{S}_0 := \begin{bmatrix} \mathbf{g} & -\mathbf{f} \\ \mathbf{G} & -\mathbf{F} \end{bmatrix}$ (essentially $\mathbf{B}_{\text{short}}$).

Latte Function: Delegate (cont.)

Idea [CHKP10]: From level $\ell - 1$ to ℓ, for $\mathbf{A}_\ell := H(\text{ID}_1||\ldots||\text{ID}_\ell)$ (hash of the chain of identities), given a secret basis $\mathbf{S}_{\ell-1}$,

1. Basis Extension: Extend $\mathbf{S}_{\ell-1}$ to a higher-dimensional basis $\mathbf{B}$ containing information of $\mathbf{A}_\ell$, such that $||\tilde{\mathbf{B}}|| = ||\tilde{\mathbf{S}}_{\ell-1}||$.
2. Basis Re-randomization: Sample linearly independent vectors $\mathbf{s}_i \hookleftarrow \mathcal{D}_{\Lambda(\mathbf{B}),\sigma_\ell}$ for some $\sigma_\ell \approx ||\tilde{\mathbf{S}}_{\ell-1}||$ to hide info of $\mathbf{S}_{\ell-1}$.

For NTRU basis, both can be achieved together by sampling from $\mathcal{D}_{\mathbf{c}+\Lambda(\mathbf{S}_{\ell-1}),\sigma_\ell}$ for some coset $\mathbf{c}$ [ETS19].

e.g. From $\mathbf{S}_0$ to $\mathbf{S}_1$, given $\mathbf{A}_1 \in \mathbb{Z}_q^N$:

1. For $i := 0, 1$:
   (a) (*) $\mathbf{s}_{i,2} \hookleftarrow \mathcal{D}_{\sigma_1}^N$.
   (b) (*) $(\mathbf{s}_{i,0}, \mathbf{s}_{i,1}) \hookleftarrow \mathcal{D}_{\mathbf{c}+\Lambda(\mathbf{S}_0),\sigma_1}$, for $\mathbf{c} := -\mathbf{s}_{i,2} \cdot \mathbf{A}_1 \bmod q$.
   (c) If $||\mathbf{s}_{i,0}, \mathbf{s}_{i,1}, \mathbf{s}_{i,2}|| > \sigma_1\sqrt{3N}$, goto Step (a).
2. (*) Find sufficiently short $(\mathbf{s}_{2,0}, \mathbf{s}_{2,1}, \mathbf{s}_{2,2})$ such that $\det(\mathbf{S}_1) = q$ for $\mathbf{S}_1 := \{\mathbf{s}_{i,j}\}$. If unable to find, goto Step 1.

Latte Function: Delegate (cont.)

Because $\Lambda(\mathbf{S}_0) = \{\mathbf{x} \in \Re^2 : \mathbf{x} \cdot (1, \mathbf{h}) = \mathbf{0} \bmod q\}$

$$\mathbf{s}_{i,0} + \mathbf{s}_{i,1} \cdot \mathbf{h} = -\mathbf{s}_{i,2} \cdot \mathbf{A}_1 \implies \mathbf{s}_{i,0} + \mathbf{s}_{i,1} \cdot \mathbf{h} + \mathbf{s}_{i,2} \cdot \mathbf{A}_1 = \mathbf{0} \bmod q,$$

$\implies (\mathbf{s}_{i,0}, \mathbf{s}_{i,1}, \mathbf{s}_{i,2})$ is a lattice vector in a ModNTRU lattice [CPS+20]:

$$\Lambda(\mathbf{S}_1) := \{\mathbf{x} \in \Re^3 : \mathbf{x} \cdot (1, \mathbf{h}, \mathbf{A}_1) = \mathbf{0} \bmod q\}.$$

Public (long) basis: $\begin{bmatrix} -\mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{0}_N \end{bmatrix} \rightarrow \begin{bmatrix} -\mathcal{A}(\mathbf{A}_1) & \mathbf{0}_N & \mathbf{I}_N \\ -\mathcal{A}(\mathbf{h}) & \mathbf{I}_N & \mathbf{0}_N \\ q\mathbf{I}_N & \mathbf{0}_N & \mathbf{0}_N \end{bmatrix}$

Private (short) basis: $\begin{bmatrix} \mathcal{A}(\mathbf{g}) & -\mathcal{A}(\mathbf{f}) \\ \mathcal{A}(\mathbf{G}) & -\mathcal{A}(\mathbf{F}) \end{bmatrix} \rightarrow \begin{bmatrix} \mathcal{A}(\mathbf{s}_{0,0}) & \mathcal{A}(\mathbf{s}_{0,1}) & \mathcal{A}(\mathbf{s}_{0,2}) \\ \mathcal{A}(\mathbf{s}_{1,0}) & \mathcal{A}(\mathbf{s}_{1,1}) & \mathcal{A}(\mathbf{s}_{1,2}) \\ \mathcal{A}(\mathbf{s}_{2,0}) & \mathcal{A}(\mathbf{s}_{2,1}) & \mathcal{A}(\mathbf{s}_{2,2}) \end{bmatrix}$

Latte Function: Extract

From level $\ell - 1$ to a user at level ℓ, given $\mathbf{A}'_\ell := H_E(\mathsf{ID}_1||\ldots||\mathsf{ID}_\ell) \in \mathbb{Z}_q^N$ (a different hash function than $\mathbf{A}_\ell$ used by the Delegate):

- (*) $(\mathbf{t}_0, \mathbf{t}_1, \ldots, \mathbf{t}_\ell) \hookleftarrow \mathcal{D}_{\mathbf{c}+\Lambda(\mathbf{S}_{\ell-1}),\sigma_\ell}$, for $\mathbf{c} := \mathbf{A}'_\ell$ and $\sigma_\ell \approx \|\tilde{\mathbf{S}}_{\ell-1}\|$, using a seed derived from $\mathsf{ID}_1||\ldots||\mathsf{ID}_\ell$.

$$\mathbf{t}_0 + \mathbf{t}_1 \cdot \mathbf{h} + \mathbf{t}_2 \cdot \mathbf{A}_1 + \cdots + \mathbf{t}_\ell \cdot \mathbf{A}_{\ell-1} = \mathbf{A}'_\ell \bmod q.$$

- User private key: $(\mathbf{t}_1, \ldots, \mathbf{t}_\ell)$.

Latte Function: Encrypt (simplified)

Ring Learning with Errors (RLWE) encryption [LPR10].

For message encoded to $\mathbf{m} \in \{0, (q-1)/2\}^N$, the ciphertext:

$$(*) \begin{bmatrix} \mathbf{C}_h \\ \mathbf{C}_1 \\ \vdots \\ \mathbf{C}_{\ell-1} \\ \mathbf{C}_\ell \end{bmatrix} := \begin{bmatrix} \mathbf{h} \\ \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_{\ell-1} \\ \mathbf{A}'_\ell \end{bmatrix} \cdot \mathbf{e} + \begin{bmatrix} \mathbf{e}_h \\ \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_{\ell-1} \\ \mathbf{e}_\ell \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{m} \end{bmatrix} \bmod q,$$

where $\mathbf{e}, \mathbf{e}_h, \mathbf{e}_1, \ldots, \mathbf{e}_\ell$ are sampled from a binomial distribution with center 0 and standard deviation 2.

Latte Function: Decrypt (simplified)

$$
\begin{aligned}
(*)\mathbf{V} :&= \mathbf{C}_\ell - \mathbf{C}_h \cdot \mathbf{t}_1 - \mathbf{C}_1 \cdot \mathbf{t}_2 - \cdots - \mathbf{C}_{\ell-1} \cdot \mathbf{t}_\ell \bmod q \\
&= \mathbf{A}'_\ell \cdot \mathbf{e} + \mathbf{e}_\ell + \mathbf{m} - (\mathbf{h} \cdot \mathbf{e} + \mathbf{e}_h) \cdot \mathbf{t}_1 - (\mathbf{A}_1 \cdot \mathbf{e} + \mathbf{e}_1) \cdot \mathbf{t}_2 - \cdots \\
&= \mathbf{e}_\ell + \mathbf{m} - \mathbf{t}_1 \cdot \mathbf{e}_h - \mathbf{t}_2 \cdot \mathbf{e}_1 - \cdots - \mathbf{t}_\ell \cdot \mathbf{e}_{\ell-1} + \mathbf{t}_0 \cdot \mathbf{e}.
\end{aligned}
$$

The last equation holds because $\mathbf{t}_0 + \mathbf{t}_1 \cdot \mathbf{h} + \mathbf{t}_2 \cdot \mathbf{A}_1 + \cdots + \mathbf{t}_\ell \cdot \mathbf{A}_{\ell-1} = \mathbf{A}'_\ell \bmod q$ by Extract.

- Round coefficients of $\mathbf{V}$ to the nearest integer in $\{0, (q-1)/2\}$.
- Parameters are chosen so the error terms are small (with coefficients $< q/4$), i.e. the decryption failure rate is negligible.

Acutal Latte is a Key Encapsulation Mechanism (KEM).

Implementation

How to efficiently implement the steps with blue asterisk?

- Discrete Gaussian sampling:
  - $\mathcal{D}_\sigma$ (KeyGen, Delegate).
  - $\mathcal{D}_{\mathbf{c}+\Lambda(\mathbf{S}_i),\sigma}$ (Delegate, Extract).
- Find short (Mod)NTRU solution (i.e. last row of $\mathbf{S}_i$) for $\det(\mathbf{S}_i) = q$ (KeyGen, Delegate).
- Polynomial ring arithmetic in $\mathfrak{R}_q := \mathbb{Z}_q[x]/\langle x^N + 1\rangle$ (Encrypt, Decrypt).

Some previous works have been done under different scenarios.

Polynomial Ring Arithmetic

- Number Theoretic Transform (NTT):
  - Essentially the Fast Fourier Transform (FFT) over $\mathbb{Z}_q$ with quasilinear time complexity.
  - For $\mathbf{a}, \mathbf{b} \in \mathfrak{R}_q$ with power-of-2 N and prime $q \equiv 1 \pmod{2N}$:
    - NTT$(\mathbf{a}) = \mathbf{a}(\zeta^i) \bmod q$ for $i \in \{0, \ldots, N-1\}$, where ζ is the $2N$-th root of unity of $\mathbb{Z}_q$; NTT$^{-1}(\hat{\mathbf{a}}) = 1/N \cdot \hat{\mathbf{a}}(\zeta^{-i}) \bmod q$.
    - $\mathbf{a} \pm \mathbf{b} =$ NTT$^{-1}($NTT$(\mathbf{a}) \pm$ NTT$(\mathbf{b}))$.
    - $\mathbf{a} \cdot \mathbf{b} =$ NTT$^{-1}($NTT$(\mathbf{a}) \circ$ NTT$(\mathbf{b}))$, where $\circ$ is the point-wise multiplication mod q.
- We adopt the Plantard's modular reduction [Pla21].
- We keep polynomials in their NTT form whenever possible to reduce the number of NTTs in Encrypt/Decrypt.
  - Master public key $\mathbf{h}$, Identities $\mathbf{A}_i$, User private key $\mathbf{t}_i$, Ciphertext $\mathbf{C}_i$.

Find Short NTRU Solution

NTRUSolve [PP19] in Falcon: To find $\mathbf{F}, \mathbf{G}$ such that $\mathbf{fG} - \mathbf{gF} = q$, use tower of rings:

1. Use field norm recursively to map $\mathbf{f}, \mathbf{g}$ to $\mathbb{Z}$.
2. Perform xgcd over $\mathbb{Z}$ to find $\mathbf{F}', \mathbf{G}' \in \mathbb{Z}$.
3. Lift $\mathbf{F}', \mathbf{G}'$ back to $\mathbf{F}, \mathbf{G} \in \mathfrak{R}$ with length reduction ($(\mathbf{F}, \mathbf{G}) - \mathbf{k} \cdot (\mathbf{f}, \mathbf{g})$ for some $\mathbf{k} \in \mathfrak{R}$).

$$
\begin{array}{ccccc}
\mathbb{Z}[x]/(x^n+1) & \ni & f, g & \to & F, G \\
\cup\!\nmid & & \downarrow & & \uparrow \\
\mathbb{Z}[x]/(x^{n/2}+1) & \ni & \mathrm{N}(f), \mathrm{N}(g) & \to & F^{[1]}, G^{[1]} \\
\cup\!\nmid & & \downarrow & & \uparrow \\
\mathbb{Z}[x]/(x^{n/4}+1) & \ni & \mathrm{N}^2(f), \mathrm{N}^2(g) & \to & F^{[2]}, G^{[2]} \\
\cup\!\nmid & & \downarrow & & \uparrow \\
\vdots & \vdots & \vdots & & \vdots \\
\cup\!\nmid & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathrm{N}^\ell(f), \mathrm{N}^\ell(g) & \to & F^{[\ell]}, G^{[\ell]}
\end{array}
$$

Find Short ModNTRU Solution

ModFalcon [CPS+20]: Use Schur complement.

e.g. for $\mathbf{S}_1 = \left[\begin{array}{c|cc} \mathbf{s}_{0,0} & \mathbf{s}_{0,1} & \mathbf{s}_{0,2} \\ \mathbf{s}_{1,0} & \mathbf{s}_{1,1} & \mathbf{s}_{1,2} \\ \hline \mathbf{s}_{2,0} & \mathbf{s}_{2,1} & \mathbf{s}_{2,2} \end{array}\right] = \begin{bmatrix} \mathbf{v}^\mathsf{T} & \mathbf{M} \\ \mathbf{G} & \mathbf{F}' \end{bmatrix}$, if $\mathbf{M}$ is invertible,

$$\begin{aligned} \det(\mathbf{S}_1) &= \det(\mathbf{G} - \mathbf{F}'\mathbf{M}^{-1}\mathbf{v}^\mathsf{T}) \det(\mathbf{M}) \\ &= (\mathbf{G} - \mathbf{F}'\mathbf{M}^{-1}\mathbf{v}^\mathsf{T}) \det(\mathbf{M}) \\ &= \mathbf{G} \det(\mathbf{M}) - \mathbf{F}'\mathrm{adj}(\mathbf{M})\mathbf{v}^\mathsf{T}. \end{aligned}$$

Choose $\mathbf{F}' := (\mathbf{F}, \mathbf{0})$.

$$\det(\mathbf{S}_1) = \det(\mathbf{M}) \cdot \mathbf{G} - \mathbf{F} \cdot \mathbf{u}_0 = q,$$

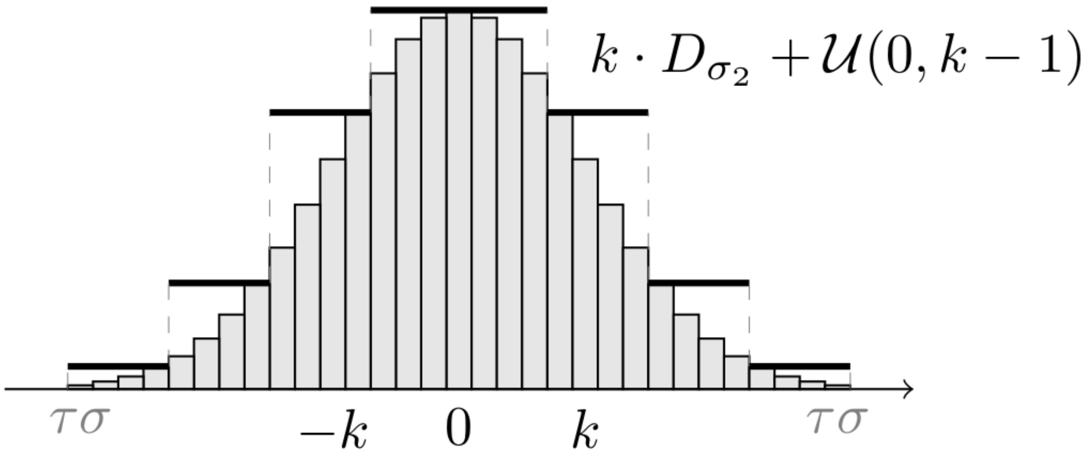
where $\mathbf{u}_0$ is the first coordinate of $\mathrm{adj}(\mathbf{M}) \cdot \mathbf{v}^\mathsf{T}$.

1. Use NTRUSolve to find $\mathbf{F}, \mathbf{G}$, with input $\det(\mathbf{M}), \mathbf{u}_0$.
   - Problem: Coefficient size of $\mathbf{F}, \mathbf{G}$ are in the same order of input $\det(\mathbf{M}), \mathbf{u}_0$, i.e. in the order of q^2.
2. Use Cramer's rule [ETS19] for length reduction.

Discrete Gaussian Sampling for $\mathcal{D}_\sigma$

We use our FACCT sampler [ZSS20b].

- Essentially a constant-time variant of the BLISS sampler [DDLL13].
  - Rejection sampling over a distribution close to $\mathcal{D}_\sigma$.
  - The rejection step needs to compute $\exp(x)$.
- Our FACCT sampler developed a fast, compact, and constant-time polynomial approximation technique to compute $\exp(x)$ with sufficient precision.
  - Adopted by the Falcon signature [PRR19].



Discrete Gaussian Sampling for $\mathcal{D}_{\mathbf{c}+\Lambda(\mathbf{S}_i),\sigma}$

Equivalent to $\mathbf{c} - \mathbf{v}$ for $\mathbf{v} \hookleftarrow \mathcal{D}_{\Lambda(\mathbf{S}_i),\mathbf{c},\sigma}$ [GPV08].

Definition 4 (Gram-Schmidt Orthogonal Decomposition [DP16])**.** Let $\mathbf{B} \in \mathbb{R}^{n\times n}$ be a full-rank matrix. There exists a Gram-Schmidt Orthogonal (GSO) Decomposition $\mathbf{B} = \mathbf{L} \cdot \tilde{\mathbf{B}}$, where $\mathbf{L}$ is unit lower triangular and rows $\tilde{\mathbf{B}}_i$ of $\tilde{\mathbf{B}}$ are pairwise orthogonal.

Given input $\mathbf{t} \in \mathbb{R}^n$, to sample $\mathbf{zB} \hookleftarrow \mathcal{D}_{\Lambda(\mathbf{B}),\mathbf{tB},\sigma}$ (variant of [GPV08] in [DP16]):

For $j = n-1, \ldots, 0$:

1. $\mathbf{t}'_j := \mathbf{t}_j + \sum_{i>j}(\mathbf{t}_i - \mathbf{z}_i)\mathbf{L}_{i,j}$.
2. $\mathbf{z}_j \hookleftarrow \mathcal{D}_{\mathbf{t}'_j,\sigma_j}$ with $\sigma_j := \sigma/\|\tilde{\mathbf{B}}_j\|$.

Let $\mathbf{t} := \mathbf{c} \cdot \mathbf{S}_i^{-1}$. Then $(\mathbf{t} - \mathbf{z})\mathbf{S}_i$ follows $\mathcal{D}_{\mathbf{c}+\Lambda(\mathbf{S}_i),\sigma}$.

- Quadratic time complexity.

Fast Fourier Sampling

Definition 5 ($\mathbf{LDL}^*$ Decomposition [DP16])**.** Let the full-rank Gram matrix $\mathbf{G} = \mathbf{B}\mathbf{B}^*$ where $\mathbf{B} \in \mathbb{R}^{n \times n}$. There exists an $\mathbf{LDL}^*$ Decomposition $\mathbf{G} = \mathbf{LDL}^*$, where $\mathbf{L}$ is a lower triangular matrix with 1 on its diagonal and $\mathbf{D}$ is a diagonal matrix.

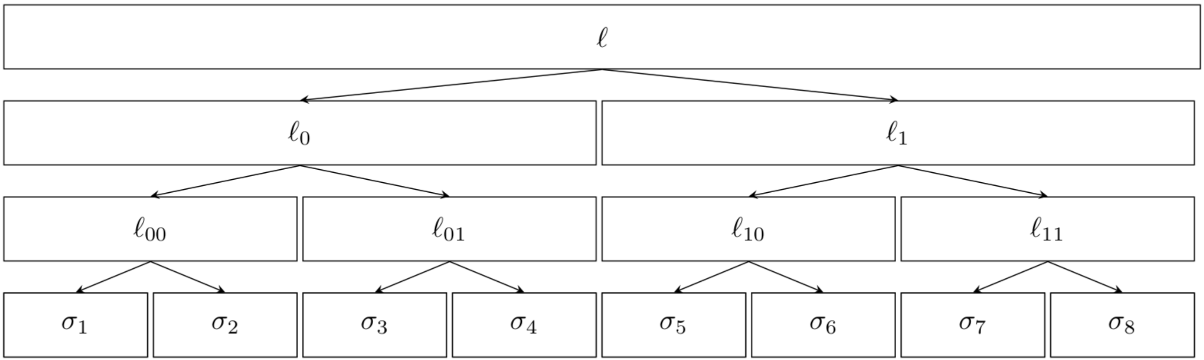
- For $\mathbf{B} = \mathbf{L} \cdot \tilde{\mathbf{B}}$, $\mathbf{L} \cdot (\tilde{\mathbf{B}}\tilde{\mathbf{B}}^*) \cdot \mathbf{L}^*$ is the $\mathbf{LDL}^*$ decomposition of $\mathbf{G} = \mathbf{B}\mathbf{B}^*$.
- The diagonal of $\mathbf{D}$ is $||\tilde{\mathbf{B}}_i||^2$.

For $\mathbf{B} \in \mathfrak{R}^{d \times d}$, the $\mathbf{LDL}^*$ decomposition can exploit the tower of rings [DP16].

- Can work in the Fourier domain.
- The decomposition results form a tree structure, with leaves values being (permuted) $||\tilde{\mathbf{B}}_i||^2$.
- Quasilinear time complexity.

Fast Fourier Sampling (cont.)

- Similarly, Falcon signature [PFH$^+$17] shows the sampling algorithm can also exploit the tower of rings in the Fourier domain with quasilinear time complexity.
- For the Fast Fourier $\mathbf{LDL}^*$ tree of (Mod)NTRU bases in Latte, we prove:
  - $\mathbf{D}$ only contains real numbers.
  - Values in $\mathbf{D}$ can be computed from the $\mathbf{D}$ values of its parent node.
- $\mathbf{D}$ can be solely computed via real number arithmetic (i.e. without complex number arithmetic).



Discrete Gaussian Sampling for $\mathcal{D}_{c,\sigma}$

Problem: For $\mathbf{S}_1$, $(\tilde{\mathbf{S}}_1)_{2N}$ is much shorter than $(\tilde{\mathbf{S}}_1)_0$, $(\tilde{\mathbf{S}}_1)_N$.

- The sampler for $\mathcal{D}_{c,\sigma}$ [HPRR20] used by Falcon has rejection rate proportional to $\sigma_{max}/\sigma_{min}$.
- Because $\sigma_j = \sigma/\|(\tilde{\mathbf{S}}_1)_j\|$, the gap between σ_{min} and σ_{max} is large for $\mathbf{S}_1$.
  - Not a problem for $\mathbf{S}_0$ (and Falcon), because σ_0 in KeyGen is chosen so $\|(\tilde{\mathbf{S}}_0)_0\|$, $\|(\tilde{\mathbf{S}}_0)_N\|$ are close [DLP14].

We use a variant [SZJ+21] of our COSAC sampler [ZSS20a].

- Rejection sampling on a center-shifted rounded Gaussian distribution (i.e. round sample from a continuous Gaussian distribution to the nearest integer).

Latte Security vs Precision Analysis

How to choose the precision of Discrete Gaussian sampling arithmetic?

- Want to minimize precision for efficiency
- **Q:** How low can we reduce precision while still maintaining security?
- **A:** Analyze concrete provable security reduction success probability degradation with precision. Choose precision to lose $\leq L \approx 2$ bits of security (wrt infinite precision security).
- We use:
  - **RD-Based Security Reduction:** Rényi Divergence analysis techniques [BLRL+18, Pre17] for Latte security degradation wrt arithmetic error bounds.
  - **Statistical model for Gaussian sampler arithmetic errors:** estimating Gaussian sampler arithmetic error bounds resulting from precision roundoff.

Latte Security vs Precision Analysis

RD-Based Security Reduction: We use RD-based arguments to relate bit security loss L of REAL (finite precision) LATTE with

- Precision p_D of discrete $\mathbb{Z}$-sampler at leaves of ffSampling.
- Precision p_f of fp arithmetic in ffSampling.

wrt to security of IDEAL (infinite precision) LATTE with $p_D, p_f = \infty$.

Our bound on L depends on:

- number $Q_{\max}$ of $\mathcal{A}$'s delegate/extract queries
- RD of precision-p_D discrete $\mathbb{Z}$-sampler from ideal (infinite precision) distribution.
  - known dependence on p_D from COSAC sampler RD analysis [ZSS20a].
- precision-p_f fp arithmetic errors bounds $\Delta^U_{t^{(i)}}, \delta^U_{\sigma^{(i)}}$ for leaf $\mathbb{Z}$-sampler centre and std dev parameters.
  - **Question:** What is the relation of $\Delta^U_{t^{(i)}}, \delta^U_{\sigma^{(i)}}$ to p_f?

Latte Security vs Precision Analysis

Statistical model for Gaussian sampler arithmetic errors:

To get tight bounds on $\Delta^U_{t^{(i)}}, \delta^U_{\sigma^{(i)}}$ in terms of p_f we introduce a heuristic statistical (numerical) model:

- model the finite precision fp errors throughout the algorithm as independent random additive error with a Gaussian distribution.
- At each fp arithmetic operation, given the mean and standard deviation of the Gaussian-distributed inputs, propagate them through the fp operation to compute the mean and standard deviation parameters of the output.

Using this model, we compute estimates for fp arithmetic errors bounds (tail bounds on the Gaussian distributed errors from the statistical model) for $\Delta^U_{t^{(i)}}, \delta^U_{\sigma^{(i)}}$.

Latte Security vs Precision Analysis

Security vs Precision Results for Latte:

We computed the max. no. of Latte attack delegate/extact queries $Q_{max} = \min(Q^B_{max}, Q^C_{max})$ to keep the security loss $L \le 2$ bits due to finite precision p_D and p_{fp}.

TABLE II
LATTE SECURITY IMPACT OF FINITE PRECISION BASED ON EMPIRICAL ERROR ESTIMATION RESULTS FROM OUR STATISTICAL MODEL.

			$\ell = 1$					$\ell = 2$				
Set	$\mathbf{p}_{fp}$	$\mathbf{p}_{\mathcal{D}}$	$\ln C_K$	$\Delta^U_{\mathbf{z}}$	$Q^C_{\max}$	Q^C_U	$Q^B_{\max}$	$\ln C_K$	$\Delta^U_{\mathbf{z}}$	$Q^C_{\max}$	Q^C_U	$Q^B_{\max}$
LATTE-1	53	48	2^{-46}	2^{-23}	2^{46}	2^{39}	2^{74}	-	-	-	-	-
LATTE-2	53	48	2^{-42}	2^{-21}	2^{42}	2^{33}	2^{72}	-	-	-	-	-
LATTE-3	113	96	2^{-156}	2^{-71}	2^{156}	2^{149}	2^{171}	2^{-95}	2^{-35}	2^{95}	2^{88}	2^{75}
LATTE-4	113	96	2^{-149}	2^{-68}	2^{149}	2^{142}	2^{169}	2^{-85}	2^{-30}	2^{85}	2^{77}	2^{66}

Conclusion:

- Standard double precision fp ($p_{fp} = 53$ bit) sufficient for Latte-1 and Latte-2 up to 2^{42} delegate/extract queries.
- 113-bit fp precision sufficient for Latte-3/Latte-4 up to 2^{66} delegate/extract queries.

Our Latte Parameters

Set	Sec.	N	$\log_2 q$	σ_ℓ		
				$\ell = 0$	$\ell = 1$	$\ell = 2$
LATTE-1	128	1024	24	106.2	5513.3	-
LATTE-2	256	2048	25	106.2	7900.2	-
LATTE-3	80	1024	36	6777.6	351968.4	22559988.0
LATTE-4	160	2048	38	9583.7	713167.	64997288.2

- LATTE-1 and LATTE-2 have 1 level (essentially an IBE).
- LATTE-3 and LATTE-4 have 2 levels.

Our Latte Benchmark Results

Speed (op/s):

Set	KeyGen	$\ell = 1$				$\ell = 2$		
		Ext	Enc	Dec	Del	Ext	Enc	Dec
LATTE-1	9.4	1361.8	23061.4	18041.3	-	-	-	-
LATTE-2	3.3	636.9	10690.7	8456.4	-	-	-	-
LATTE-3	5.7	36.3	14331.1	12134.7	2.4	20.0	11429.8	9713.4
LATTE-4	1.7	17.1	6846.6	5785.6	0.8	9.4	5450.2	4642.1

- Delegate takes ≈ 1 second, significantly faster than the order of minutes estimated in [ETS19].
- Encrypt/Decrypt are very fast (up to 9.7x faster than [ETS19]).

Key/Ciphertext Sizes (bytes):

Set	Master Public Key	Master Private Key	User Private Key		Ciphertext		Delegated Public Key	Delegated Private Key
			$\ell = 1$	$\ell = 2$	$\ell = 1$	$\ell = 2$		
LATTE-1	3072	12288	3072	-	6176	-	-	-
LATTE-2	6400	25600	6400	-	12832	-	-	-
LATTE-3	4608	18432	4608	9216	9248	13856	9216	41472
LATTE-4	9728	38912	9728	19456	19488	29216	19456	87552

References

[BLRL+18] S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé, and R. Steinfeld. Improved security proofs in lattice-based cryptography: Using the rényi divergence rather than the statistical distance. *J. Cryptology*, 31(2):610–640, 2018.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, 2010.

[CPS+20] Chitchanok Chuengsatiansup, Thomas Prest, Damien Stehlé, Alexandre Wallet, and Keita Xagawa. Modfalcon: Compact signatures based on module-ntru lattices. In *AsiaCCS*, pages 853–866. ACM, 2020.

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal Gaussians. In *CRYPTO (1)*, volume 8042 of *Lecture Notes in Computer Science*, pages 40–56. Springer, 2013.

[DLP14] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over NTRU lattices. In *ASIACRYPT (2)*, volume 8874 of *Lecture Notes in Computer Science*, pages 22–41. Springer, 2014.

[DP16] Léo Ducas and Thomas Prest. Fast fourier orthogonalization. In *ISSAC*, pages 191–198. ACM, 2016.

[ETS19] ETSI. Quantum-Safe Identity-based Encryption. Technical report, The European Telecommunications Standards Institute, Sophia-Antipolis, France, 2019.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. ACM, 2008.

[HPRR20] James Howe, Thomas Prest, Thomas Ricosset, and Mélissa Rossi. Isochronous Gaussian sampling: From inception to implementation. In *PQCrypto*, volume 12100 of *LNCS*, pages 53–71. Springer, 2020.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, 2010.

[PFH+17] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. `https://falcon-sign.info/`, 2017. Accessed: 2024-02-29.

[Pla21] Thomas Plantard. Efficient word size modular arithmetic. *IEEE Trans. Emerg. Top. Comput.*, 9(3):1506–1518, 2021.

[PP19] Thomas Pornin and Thomas Prest. More efficient algorithms for the NTRU key generation using the field norm. In *Public Key Cryptography (2)*, volume 11443 of *Lecture Notes in Computer Science*, pages 504–533. Springer, 2019.

[Pre17] Thomas Prest. Sharper bounds in lattice-based cryptography using the Rényi divergence. In *ASIACRYPT (1)*, volume 10624 of *Lecture Notes in Computer Science*, pages 347–374. Springer, 2017.

[PRR19] Thomas Prest, Thomas Ricosset, and Mélissa Rossi. Simple, fast and constant-time Gaussian sampling over the integers for Falcon. Second PQC Standardization Conference, `https://csrc.nist.gov/CSRC/media/Events/Second-PQC-Standardization-Conference/documents/accepted-papers/rossi-simple-fast-constant.pdf`, 2019. Accessed: 2019-08-13.

[SZJ+21] Shuo Sun, Yongbin Zhou, Yunfeng Ji, Rui Zhang, and Yang Tao. Generic, efficient and isochronous gaussian sampling over the integers. *IACR Cryptol. ePrint Arch.*, 2021:199, 2021.

[ZSS20a] Raymond K. Zhao, Ron Steinfeld, and Amin Sakzad. COSAC: Compact and scalable arbitrary-centered discrete Gaussian sampling over integers. In *PQCrypto 2020*, 2020.

[ZSS20b] Raymond K. Zhao, Ron Steinfeld, and Amin Sakzad. FACCT: fast, compact, and constant-time discrete Gaussian sampler over integers. *IEEE Trans. Computers*, 69(1):126–137, 2020.

Thank You

[1]CSIRO's Data61; [2]University of Waterloo; [3]Monash University; [4]Queen's University Belfast
www.csiro.au