NTRU Cryptosystem: Recent Developments

Ron Steinfeld
School of IT
Monash University, Australia
(partly based on joint work with Damien Stehlé, ENS Lyon, France)

Johann Radon Institute (RICAM), Linz, Austria, December 2013

Outline of the talk

1- Introduction
- Background: Why study NTRU?

2- NTRU Cryptosystem: Review

3- Recent Developments on NTRU Security
- NTRU variant provably as secure as worst-case lattice problems
- Tools: Discrete Gaussians, Fourier analysis, Ring-LWE

4- Recent Developments on NTRU Applications
- Fully-Homomorphic Encryption (FHE) from NTRU
- Cryptographic Multilinear Maps from NTRU

5- Concluding Remarks

The NTRU Cryptosystem

NTRUEncrypt: A public-key encryption scheme.

- 1996: Proposed by Hoffstein, Pipher & Silverman.
- 1997: Lattice attacks by Coppersmith & Shamir.
- 1998: Revised by Hoffstein et al.

In the last 15 years:

- Several minor improvements to the lattice attacks.
- Attacks for isolated sets of parameters.
- **But the design has proved very robust.**

In the last 3 years (this talk):

- Variants with a provable security foundation
- Variants with new functionality

The NTRU Cryptosystem

NTRUEncrypt: A public-key encryption scheme.

- 1996: Proposed by Hoffstein, Pipher & Silverman.
- 1997: Lattice attacks by Coppersmith & Shamir.
- 1998: Revised by Hoffstein et al.

In the last 15 years:

- Several minor improvements to the lattice attacks.
- Attacks for isolated sets of parameters.
- **But the design has proved very robust.**

In the last 3 years (this talk):

- Variants with a provable security foundation
- Variants with new functionality

The NTRU Cryptosystem

NTRUEncrypt: A public-key encryption scheme.

- 1996: Proposed by Hoffstein, Pipher & Silverman.
- 1997: Lattice attacks by Coppersmith & Shamir.
- 1998: Revised by Hoffstein et al.

In the last 15 years:

- Several minor improvements to the lattice attacks.
- Attacks for isolated sets of parameters.
- **But the design has proved very robust.**

In the last 3 years (this talk):

- Variants with a provable security foundation
- Variants with new functionality

Why study NTRU Cryptosystem?

- Standardized: IEEE P1363.
- Commercialized: Security Innovation.
- Super-fast (comparison to 1024-bit RSA, based on an NTRU brochure):
  - Encryption ~ 10 times faster
  - Decryption ~ 100 times faster
  - Asymptotically: $\widetilde{O}(\lambda)$ versus $\widetilde{O}(\lambda^6)$, for security 2^λ
- Interesting security features:
  - No integer factoring nor discrete logs
  - Seems to resist practical attacks
  - Seems to resist quantum attacks

Why study NTRU Cryptosystem?

- Standardized: IEEE P1363.
- Commercialized: Security Innovation.
- Super-fast (comparison to 1024-bit RSA, based on an NTRU brochure):
  - Encryption $\sim$ 10 times faster
  - Decryption $\sim$ 100 times faster
  - Asymptotically: $\widetilde{O}(\lambda)$ versus $\widetilde{O}(\lambda^6)$, for security 2^λ
- Interesting security features:
  - No integer factoring nor discrete logs
  - Seems to resist practical attacks
  - Seems to resist quantum attacks

Why study NTRU Cryptosystem?

- Standardized: IEEE P1363.
- Commercialized: Security Innovation.
- Super-fast (comparison to 1024-bit RSA, based on an NTRU brochure):
  - Encryption $\sim$ 10 times faster
  - Decryption $\sim$ 100 times faster
  - Asymptotically: $\widetilde{O}(\lambda)$ versus $\widetilde{O}(\lambda^6)$, for security 2^λ
- Interesting security features:
  - No integer factoring nor discrete logs
  - Seems to resist practical attacks
  - Seems to resist quantum attacks

Polynomial Rings

Take $\phi \in \mathbb{Z}[x]$ monic of degree n.

$$R^{\phi} := \Big[\mathbb{Z}[x]/(\phi), +, \times\Big].$$

Interesting ϕ's:

- $\phi = x^n - 1 \rightarrow R^-$, $\phi = x^n + 1 \rightarrow R^+$.
- For n a power of 2, the ring R^+ is isomorphic to the ring of integers of $K = \mathbb{Q}[e^{i\pi/n}]$:

$$K \simeq \mathbb{Q}[x]/(x^n + 1)$$
$$\mathcal{O}_K \simeq \mathbb{Z}[x]/(x^n + 1).$$

$\Rightarrow$ Rich algebraic structure (great for design and proofs).

Polynomial Rings

Take $\phi \in \mathbb{Z}[x]$ monic of degree n.

$$R^{\phi} := \Big[\mathbb{Z}[x]/(\phi), +, \times\Big].$$

Interesting ϕ's:

- $\phi = x^n - 1 \rightarrow R^-, \quad \phi = x^n + 1 \rightarrow R^+$.
- For n a power of 2, the ring R^+ is isomorphic to the ring of integers of $K = \mathbb{Q}[e^{i\pi/n}]$:

$$\begin{aligned} K &\simeq \mathbb{Q}[x]/(x^n+1) \\ \mathcal{O}_K &\simeq \mathbb{Z}[x]/(x^n+1). \end{aligned}$$

$\Rightarrow$ Rich algebraic structure (great for design and proofs).

Polynomial Rings

Take $\phi \in \mathbb{Z}[x]$ monic of degree n.

$$R^{\phi} := \Big[\mathbb{Z}[x]/(\phi), +, \times\Big].$$

Interesting ϕ's:

- $\phi = x^n - 1 \rightarrow R^-$, $\phi = x^n + 1 \rightarrow R^+$.
- For n a power of 2, the ring R^+ is isomorphic to the ring of integers of $K = \mathbb{Q}[e^{i\pi/n}]$:

$$\begin{aligned} K &\simeq \mathbb{Q}[x]/(x^n + 1) \\ \mathcal{O}_K &\simeq \mathbb{Z}[x]/(x^n + 1). \end{aligned}$$

$\Rightarrow$ Rich algebraic structure (great for design and proofs).

Polynomial Rings

Let $q \geq 2$ and $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$.

$$R_q^\phi := \left[\mathbb{Z}_q[x]/(\phi), +, \times\right].$$

- Arithmetic in R_q^ϕ costs $\widetilde{O}(n \log q)$.
- R_q^+ is isomorphic to $\mathcal{O}_K/(q)$.

The key to decryption correctness

If $f \in R^\phi$ is known to have coefficients in $(-q/2, q/2)$, then

$f \bmod q$ uniquely determines f.

Polynomial Rings

Let $q \geq 2$ and $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$.

$$R_q^\phi := \left[\mathbb{Z}_q[x]/(\phi), +, \times\right].$$

- Arithmetic in R_q^ϕ costs $\widetilde{O}(n \log q)$.
- R_q^+ is isomorphic to $\mathcal{O}_K/(q)$.

The key to decryption correctness

If $f \in R^\phi$ is known to have coefficients in $(-q/2, q/2)$, then

$$f \bmod q \quad \text{uniquely determines} \quad f.$$

Polynomial Rings

Let $q \geq 2$ and $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$.

$$R_q^\phi := \left[\mathbb{Z}_q[x]/(\phi), +, \times\right].$$

- Arithmetic in R_q^ϕ costs $\widetilde{O}(n \log q)$.
- R_q^+ is isomorphic to $\mathcal{O}_K/(q)$.

The key to decryption correctness

If $f \in R^\phi$ is known to have coefficients in $(-q/2, q/2)$, then

$$f \bmod q \quad \text{uniquely determines} \quad f.$$

NTRU Cryptosystem: Key Generation

Parameters: n prime, $q \approx n$ a power of 2, p small, $\phi = x^n - 1$.
(e.g. $(n, q, p) = (503, 256, 3)$).

- Secret key sk: $f, g \in R^-$ sampled indep. from distrib. χ_σ with:
  - f is invertible mod q and mod p
  - The coeffs of f and g are small
    - $\mathrm{Supp}(\chi_\sigma) = \{-1, 0, 1\}^n$.
- Public key pk: $h = g/f \bmod q$.

Security intuition

Given $h \in R_q^-$, finding $g, f \in R^-$ small s.t. $h = g/f \ [q]$ is hard.

NTRU Cryptosystem: Key Generation

Parameters: n prime, $q \approx n$ a power of 2, p small, $\phi = x^n - 1$.
(e.g. $(n, q, p) = (503, 256, 3)$).

- Secret key sk: $f, g \in R^-$ sampled indep. from distrib. χ_σ with:
  - f is invertible mod q and mod p
  - The coeffs of f and g are small
    - $\mathrm{Supp}(\chi_\sigma) = \{-1, 0, 1\}^n$.
- Public key pk: $h = g/f \bmod q$.

Security intuition

Given $h \in R_q^-$, finding $g, f \in R^-$ small s.t. $h = g/f \ [q]$ is hard.

NTRU Cryptosystem: Key Generation

Parameters: n prime, $q \approx n$ a power of 2, p small, $\phi = x^n - 1$.
(e.g. $(n, q, p) = (503, 256, 3)$).

- Secret key sk: $f, g \in R^-$ sampled indep. from distrib. χ_σ with:
  - f is invertible mod q and mod p
  - The coeffs of f and g are small
    - $\mathrm{Supp}(\chi_\sigma) = \{-1, 0, 1\}^n$.
- Public key pk: $h = g/f \bmod q$.

Security intuition

Given $h \in R_q^-$, finding $g, f \in R^-$ small s.t. $h = g/f \ [q]$ is hard.

NTRU Cryptosystem: Key Generation

Parameters: n prime, $q \approx n$ a power of 2, p small, $\phi = x^n - 1$.
(e.g. $(n, q, p) = (503, 256, 3)$).

- Secret key sk: $f, g \in R^-$ sampled indep. from distrib. χ_σ with:
  - f is invertible mod q and mod p
  - The coeffs of f and g are small
    - $\mathrm{Supp}(\chi_\sigma) = \{-1, 0, 1\}^n$.
- Public key pk: $h = g/f \bmod q$.

Security intuition

Given $h \in R_q^-$, finding $g, f \in R^-$ small s.t. $h = g/f$ $[q]$ is hard.

NTRU Cryptosystem: Encryption and Decryption

- sk: $f, g \in R^-$ small with f invertible mod q and mod p
- pk: $h = g/f \bmod q$

Encryption of $M \in R$ with coeffs in $\{0, \ldots, p-1\}$:

- Sample $s, e \in R_q^-$ from distrib. χ_ρ, χ_β resp. with small coeffs – $\mathrm{Supp}(\chi_\rho) = \{-1, 0, 1\}^n, \mathrm{Supp}(\chi_\beta) = \{0\}$).
- Send $C := p(hs + e) + M \bmod q$

Decryption of $C \in R_q^-$:

- $f \times C = p(gs + fe) + fM \bmod q$
- Smallness $\Rightarrow$ equality holds over R^-
- $(f \times C \bmod q) \bmod p = fM \bmod p$
- Multiply by the inverse of f mod p

Security intuition

The mask $p(hs + e)$ hides the plaintext M in the ciphertext C.

Lattices Background: Approx-SVP

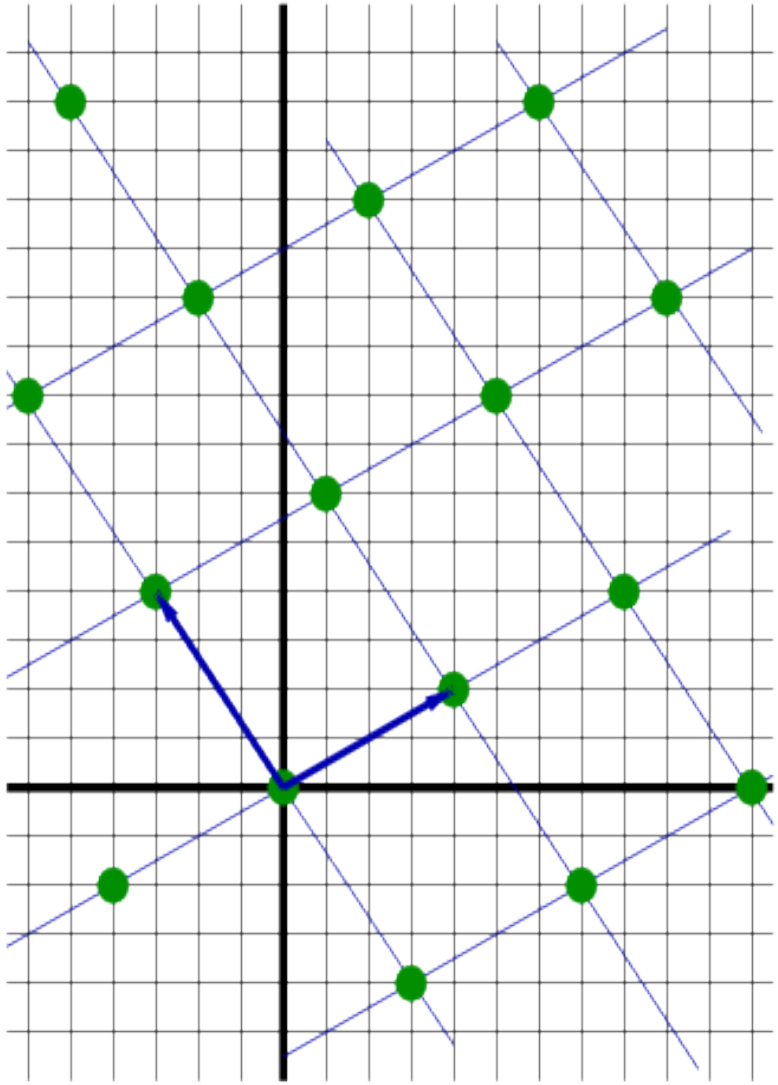
Lattice $\equiv \{\sum_{i \le n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$,
for some lin. independent $\mathbf{b}_i$'s.

Minimum: $\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \mathbf{0})$

γ-SVP

Find $\mathbf{b} \in L$ with: $0 < \|\mathbf{b}\| \le \gamma \cdot \lambda(L)$.

- No known sub-exp. algorithm for $\gamma = Poly(n)$.
- Not even quantumly.
- Seems harder than Int-Fac and DLog.



Lattices Background: Approx-SVP

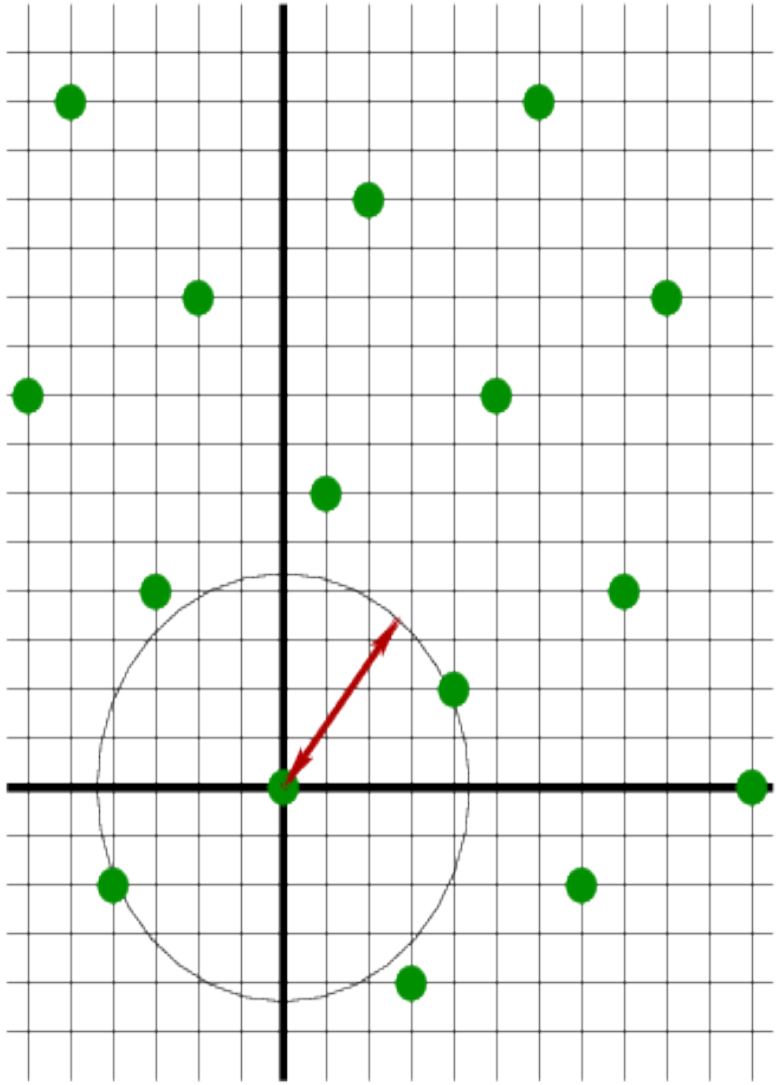
Lattice $\equiv \{\sum_{i \le n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$,
for some lin. independent $\mathbf{b}_i$'s.

Minimum: $\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \mathbf{0})$

γ-SVP

Find $\mathbf{b} \in L$ with: $0 < \|\mathbf{b}\| \le \gamma \cdot \lambda(L)$.

- No known sub-exp. algorithm for $\gamma = \mathcal{P}oly(n)$.
- Not even quantumly.
- Seems harder than Int-Fac and DLog.



Lattices Background: Approx-SVP

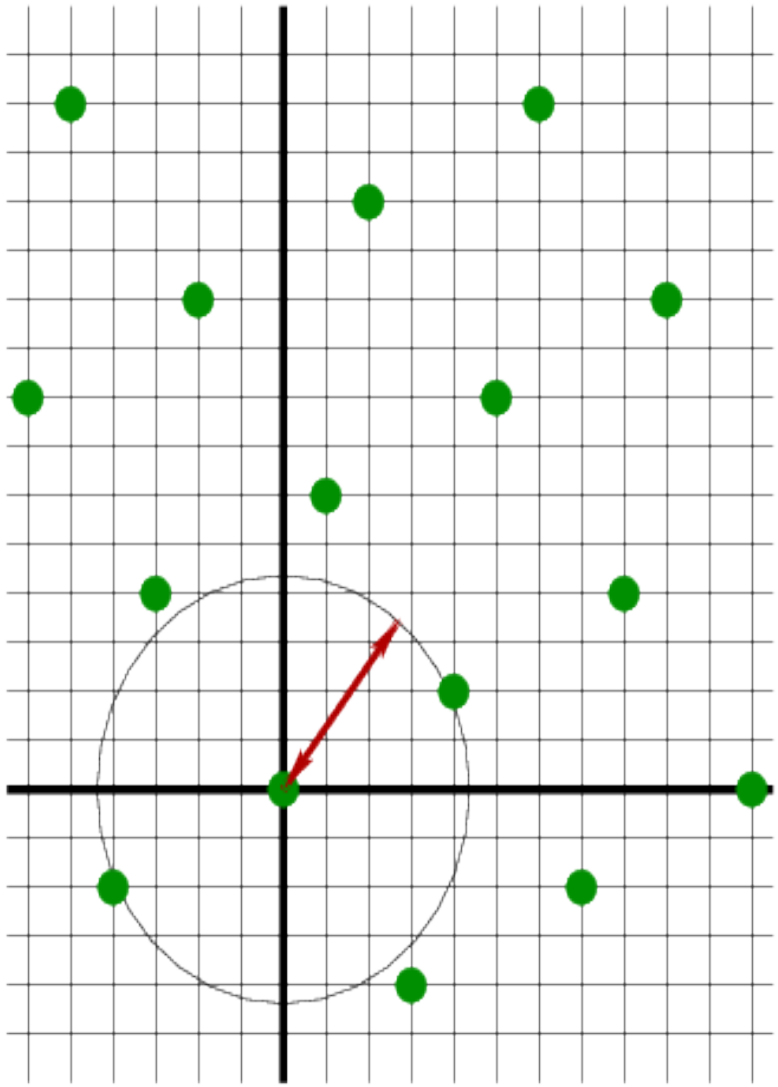
Lattice $\equiv \{\sum_{i \le n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$,
for some lin. independent $\mathbf{b}_i$'s.

Minimum: $\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \mathbf{0})$

γ-SVP

Find $\mathbf{b} \in L$ with: $0 < \|\mathbf{b}\| \le \gamma \cdot \lambda(L)$.

- No known sub-exp. algorithm for $\gamma = \mathcal{P}oly(n)$.
- Not even quantumly.
- Seems harder than Int-Fac and DLog.



Lattice Background: Approx-Ideal-SVP

- $I \subseteq R^\phi$ is an ideal if:

$$\forall a, b \in I, \forall r \in R^\phi : \quad a + b \cdot r \in I.$$

- We identify polynomials to vectors via their coefficients:

$$\begin{array}{ccc} R^\phi & \rightarrow & \mathbb{Z}^n \\ \sum_{i<n} f_i x^i & \mapsto & (f_0, \ldots, f_{n-1})^t \end{array}$$

- An ideal I is mapped to an integer lattice.

$\mathcal{P}oly(n)$-Ideal-SVP: $\mathcal{P}oly(n)$-SVP restricted to ideal lattices.

No known computational advantage for this family of inputs.

Lattice Background: Approx-Ideal-SVP

- $I \subseteq R^\phi$ is an ideal if:

$$\forall a, b \in I, \forall r \in R^\phi : \quad a + b \cdot r \in I.$$

- We identify polynomials to vectors via their coefficients:

$$\begin{array}{ccc} R^\phi & \rightarrow & \mathbb{Z}^n \\ \sum_{i<n} f_i x^i & \mapsto & (f_0, \ldots, f_{n-1})^t \end{array}$$

- An ideal I is mapped to an integer lattice.

$\mathcal{P}oly(n)$-Ideal-SVP: $\mathcal{P}oly(n)$-SVP restricted to ideal lattices.

No known computational advantage for this family of inputs.

Lattice Background: Approx-Ideal-SVP

- $I \subseteq R^\phi$ is an ideal if:

$$\forall a, b \in I, \forall r \in R^\phi : \quad a + b \cdot r \in I.$$

- We identify polynomials to vectors via their coefficients:

$$\begin{array}{ccc} R^\phi & \rightarrow & \mathbb{Z}^n \\ \sum_{i<n} f_i x^i & \mapsto & (f_0, \ldots, f_{n-1})^t \end{array}$$

- An ideal I is mapped to an integer lattice.

$\mathcal{P}oly(n)$-Ideal-SVP: $\mathcal{P}oly(n)$-SVP restricted to ideal lattices.

No known computational advantage for this family of inputs.

Lattice Background: Approx-Ideal-SVP

- $I \subseteq R^\phi$ is an ideal if:

$$\forall a, b \in I, \forall r \in R^\phi : \quad a + b \cdot r \in I.$$

- We identify polynomials to vectors via their coefficients:

$$\begin{array}{ccc} R^\phi & \rightarrow & \mathbb{Z}^n \\ \sum_{i<n} f_i x^i & \mapsto & (f_0, \ldots, f_{n-1})^t \end{array}$$

- An ideal I is mapped to an integer lattice.

$\mathcal{P}oly(n)$-Ideal-SVP: $\mathcal{P}oly(n)$-SVP restricted to ideal lattices.

No known computational advantage for this family of inputs.

Security of NTRU: Lattice Attacks

Coppersmith-Shamir Lattice attack: Given $h = g/f \in R_q$, small secret key $(f, g) \in R^2$) satisfies:

$$f \cdot h - g = 0 \bmod q.$$

Set of all solutions $(f', f'h + qR) \in R^2$ to above is a $2n$-dim. $\mathbb{Z}$-lattice L_{NTRU} with row basis

$$\begin{bmatrix} I_n & \mathrm{rot}(h) \\ 0 & qI_n \end{bmatrix}, \tag{1}$$

called the NTRU lattice.

Attack: $\|(f, g)\| = \lambda_1(L_{\mathrm{NTRU}}) \rightarrow$ Run γ-SVP on L_{NTRU}, hope to get small multiple of (f, g).

Catch: $(q, 0) \in L_{\mathrm{NTRU}}$, so need $\gamma \leq q/\|(f, g)\| = O(\mathcal{P}oly(n))$.

- Recall: Best known alg. for $\mathcal{P}oly(n)$-SVP take time $2^{\Omega(n)}$!

Security of NTRU: Lattice Attacks

Coppersmith-Shamir Lattice attack: Given $h = g/f \in R_q$, small secret key $(f, g) \in R^2$) satisfies:

$$f \cdot h - g = 0 \bmod q.$$

Set of all solutions $(f', f'h + qR) \in R^2$ to above is a $2n$-dim. $\mathbb{Z}$-lattice L_{NTRU} with row basis

$$\begin{bmatrix} I_n & \mathrm{rot}(h) \\ 0 & qI_n \end{bmatrix}, \tag{1}$$

called the NTRU lattice.

Attack: $\|(f, g)\| = \lambda_1(L_{\mathrm{NTRU}}) \rightarrow$ Run γ-SVP on L_{NTRU}, hope to get small multiple of (f, g).

Catch: $(q, 0) \in L_{\mathrm{NTRU}}$, so need $\gamma \leq q/\|(f, g)\| = \mathcal{O}(\mathcal{P}oly(n))$.

- Recall: Best known alg. for $\mathcal{P}oly(n)$-SVP take time $2^{\Omega(n)}$!

Security of NTRU: Lattice Attacks

Coppersmith-Shamir Lattice attack: Given $h = g/f \in R_q$, small secret key $(f, g) \in R^2$) satisfies:

$$f \cdot h - g = 0 \bmod q.$$

Set of all solutions $(f', f'h + qR) \in R^2$ to above is a $2n$-dim. $\mathbb{Z}$-lattice L_{NTRU} with row basis

$$\begin{bmatrix} I_n & \mathrm{rot}(h) \\ 0 & qI_n \end{bmatrix}, \tag{1}$$

called the NTRU lattice.
Attack: $\|(f, g)\| = \lambda_1(L_{\mathrm{NTRU}}) \rightarrow$ Run γ-SVP on L_{NTRU}, hope to get small multiple of (f, g).
Catch: $(q, 0) \in L_{\mathrm{NTRU}}$, so need $\gamma \leq q/\|(f, g)\| = O(\mathcal{P}oly(n))$.

- Recall: Best known alg. for $\mathcal{P}oly(n)$-SVP take time $2^{\Omega(n)}$!

NTRU variant provably as secure as worst-case lattice problems

Motivation:

- NTRU lattices have a special algebraic structure.
- What if an efficient approx-SVP algorithm could be tailored to NTRU lattices?
- Could there exist a non-negligible fraction of "weak" NTRU lattices?

Theorem[Stehlé, Steinfeld 2011]

There is a choice of parameters for NTRU Cryptosystem so that:

- Encryption/decryption of λ bits still cost $\widetilde{O}(\lambda)$,
- Any polynomial-time IND attack leads to a polynomial-time quantum algorithm for $\mathcal{P}oly(n)$-Ideal-SVP that works for all inputs.

NTRU variant provably as secure as worst-case lattice problems

Motivation:

- NTRU lattices have a special algebraic structure.
- What if an efficient approx-SVP algorithm could be tailored to NTRU lattices?
- Could there exist a non-negligible fraction of "weak" NTRU lattices?

Theorem[Stehlé, Steinfeld 2011]

There is a choice of parameters for NTRU Cryptosystem so that:

- Encryption/decryption of λ bits still cost $\widetilde{O}(\lambda)$,
- Any polynomial-time IND attack leads to a polynomial-time quantum algorithm for $\mathcal{P}oly(n)$-Ideal-SVP that works for all inputs.

Indistinguishability (IND) Security Definition

Modern security definition for public-key encryption (against passive eavesdropping)

For $b \in \{0, 1\}$, two phase game G_b with adversary A:

- Phase 1:
  - A is given public key pk from key generation algorithm $\mathrm{KG}(n)$.
  - A outputs two challenge messages m_0, m_1.
- Phase 2:
  - A is given challenge ciphertext $c_b = \mathrm{Enc}(pk, m_b)$.
  - A outputs an estimate b' for bit b.

Indistinguishability (IND) Security

For all $\mathcal{P}oly(n)$-time A,

$$Adv(A) \stackrel{\text{def}}{=} |\Pr_{G_1}[b' = 1] - \Pr_{G_0}[b' = 1]| = n^{-\omega(1)} = \mathrm{neg}(n).$$

Indistinguishability (IND) Security Definition

Modern security definition for public-key encryption (against passive eavesdropping)

For $b \in \{0, 1\}$, two phase game G_b with adversary A:

- Phase 1:
  - A is given public key pk from key generation algorithm $\mathrm{KG}(n)$.
  - A outputs two challenge messages m_0, m_1.
- Phase 2:
  - A is given challenge ciphertext $c_b = \mathrm{Enc}(pk, m_b)$.
  - A outputs an estimate b' for bit b.

Indistinguishability (IND) Security

For all $\mathcal{P}oly(n)$-time A,

$$Adv(A) \stackrel{\text{def}}{=} |\Pr_{G_1}[b' = 1] - \Pr_{G_0}[b' = 1]| = n^{-\omega(1)} = \mathrm{neg}(n).$$

Indistinguishability (IND) Security Definition

Modern security definition for public-key encryption (against passive eavesdropping)

For $b \in \{0,1\}$, two phase game G_b with adversary A:

- Phase 1:
  - A is given public key pk from key generation algorithm $\mathrm{KG}(n)$.
  - A outputs two challenge messages m_0, m_1.
- Phase 2:
  - A is given challenge ciphertext $c_b = \mathrm{Enc}(pk, m_b)$.
  - A outputs an estimate b' for bit b.

Indistinguishability (IND) Security

For all $\mathcal{P}oly(n)$-time A,

$$Adv(A) \stackrel{\text{def}}{=} |\Pr_{G_1}[b' = 1] - \Pr_{G_0}[b' = 1]| = n^{-\omega(1)} = \mathrm{neg}(n).$$

Security of NTRU: Computational/Statistical Problems

Essentially two ways to break the IND security of NTRU:

- Crack the **public key**:

NTRU Decision Key Cracking Problem $\mathtt{DNKC}_{n,q,\phi,\chi_\sigma}$

Given (n, q, ϕ) and h, distinguish

- NTRU key distribution $D_0 = \{h = g/f \in R_q : f, g \hookleftarrow \chi_\sigma\}$.
- Uniform key distribution $D_1 = \{h \hookleftarrow U(R_q^*)\}$.

- Crack the **ciphertext** for a uniform key:

NTRU Decision Ciphertext Cracking Problem $\mathtt{DNCC}_{n,q,\phi,\chi_\rho,\chi_\beta}$

Given (n, q, ϕ), h sampled from $U(R_q^*)$, and c, distinguish

- NTRU ciphertext distribution $D_0 = \{c = hs + e : s \hookleftarrow \chi_\rho, e \hookleftarrow \chi_\beta\}$.
- Uniform distribution $D_1 = \{c \hookleftarrow U(R_q)\}$.

Security of NTRU: Computational/Statistical Problems

Essentially two ways to break the IND security of NTRU:

- Crack the **public key**:

NTRU Decision Key Cracking Problem $\mathtt{DNKC}_{n,q,\phi,\chi_\sigma}$

Given (n, q, ϕ) and h, distinguish

- NTRU key distribution $D_0 = \{h = g/f \in R_q : f, g \hookleftarrow \chi_\sigma\}$.
- Uniform key distribution $D_1 = \{h \hookleftarrow U(R_q^*)\}$.

- Crack the **ciphertext** for a uniform key:

NTRU Decision Ciphertext Cracking Problem $\mathtt{DNCC}_{n,q,\phi,\chi_\rho,\chi_\beta}$

Given (n, q, ϕ), h sampled from $U(R_q^*)$, and c, distinguish

- NTRU ciphertext distribution $D_0 = \{c = hs + e : s \hookleftarrow \chi_\rho, e \hookleftarrow \chi_\beta\}$.
- Uniform distribution $D_1 = \{c \hookleftarrow U(R_q)\}$.

IND Security of NTRU: Sufficient Condition

Proposition (Adapted from [StSt11])

If DNKC and DNCC are both hard, then NTRUcryptosystem achieves semantic (IND) security.

Proof by contradiction – three 'games' with adversary A:

- IND_b – pk: $h = g/f$, ciph: $c_b = p \cdot (hs + e) + m_b$, $p_b = \Pr_{\text{IND}_b}[A(h, c_b) = 1]$.
- IND'_b – pk: $h \leftarrow\!\!\shortmid\; U(R_q^*)$, ciph: $c_b = p \cdot (hs + e) + m_b$, $p'_b = \Pr_{\text{IND'}_b}[A(h, c_b) = 1]$.
  - $|p'_b - p_b| = \text{non-neg}(n) \rightarrow A$ breaks DNKC.
- IND''_b – pk: $h \leftarrow\!\!\shortmid\; U(R_q^*)$, ciph: $c_b = p \cdot U(R_q) + m_b$, $p'' = \Pr_{\text{IND''}_b}[A(h, c_b) = 1]$.
  - $|p''_b - p'_b| = \text{non-neg}(n) \rightarrow A$ breaks DNCC.

Else, A can distinguish IND''_0 from IND''_1: contradiction – $p \cdot U(R_q)$ term perfectly hides m_b!

How to make both DNKCand DNCCproblems hard?

StSt11 strategy to prove hardness of DNKC and DNCC problems:

- Choose χ_σ for f, g to make DNKC statistically hard.
  - $f, g \hookleftarrow \chi_\sigma \to h = g/f$ almost uniformly distributed on R_q^*.
  - Must work in statistical region: $|\mathrm{Supp}(\chi_\sigma)| > |R_q^*| \to \sigma > \sqrt{q}$.
  - Use a (modified) discrete Gaussian distribution χ_σ.
  - Proof ingredients: Fourier analysis, duality of lattices, algebraic structure of R_q.
- Choose $\chi_\rho = \chi_\beta$ for s, e to make DNCC computationally hard.
  - Change rings: $R_q^- = \mathbb{Z}_q[x]/(x^n - 1) \to R_q^+ = \mathbb{Z}_q[x]/(x^n + 1)$, $n = 2^k$.
  - $h \hookleftarrow U(R_q^*), s, e \hookleftarrow \chi_\beta \to (h, c = hs + e)$ computationaly indist. from $U(R_q^* \times R_q)$, if $\approx q/\beta$-Ideal-SVP is hard.
  - Must work in computational region: $|\mathrm{Supp}(\chi_\beta)| < |R_q| \to \beta < \sqrt{q}$.
  - Use a rounded Gaussian distribution χ_β.

The modified scheme

Parameters: n, q a power of 2, $R = R^-$.

Key generation:

- sk: $f, g \in R$ with:
  - f invertible mod q and p.
  - Coeffs of f and g in $\{-1, 0, 1\}$
- pk: $h = g/f \bmod q$.

Encryption of $M \in R$ with coeffs in $\{0, 1\}$:

- $C := phs + M \bmod q$, with coeffs of s in $\{-1, 0, 1\}$.

Decryption of $C \in R_q$:

- $f \times C \bmod q \;=\; pgs + fM \qquad$ (over R)
- $(f \times C \bmod q) \bmod p \;=\; fM \bmod p$.
- Multiply by the inverse of f mod p.

The modified scheme

Parameters: n a power of 2, q prime, $R = R^+$.

Key generation:

- sk: $f, g \in R$ with:
  - f invertible mod q and p.
  - Coeffs of f and g of magnitude $\approx \sqrt{q}$
- pk: $h = g/f \bmod q$.

Encryption of $M \in R$ with coeffs in $\{0, 1\}$:

- $C := p(hs + e) + M \bmod q$, with coeffs of s, e of magnitude $\approx \beta$.

Decryption of $C \in R_q$:

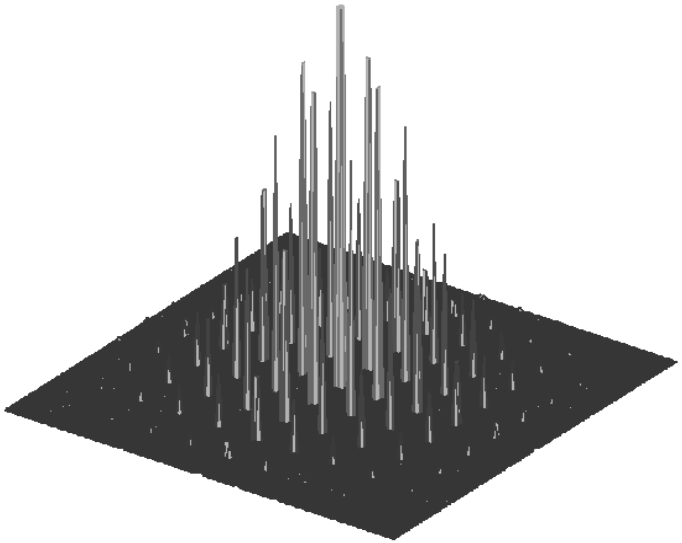
- $f \times C \bmod q \;=\; p(gs + fe) + fM \qquad$ (over R)
- $(f \times C \bmod q) \bmod p \;=\; fM \bmod p$.
- Multiply by the inverse of $f \bmod p$.

The distribution $\chi_\sigma = D_\sigma^\times$ of f and g

1. Sample f from the discrete Gaussian $D_{\mathbb{Z}^n,\sigma}$ (using [GePeVa'08]):

$$\forall x \in \mathbb{Z}^n : \quad D_{\mathbb{Z}^n,\sigma}[x] \sim \exp\left(-\pi \frac{\|x\|^2}{\sigma^2}\right).$$

2. If f is not invertible in R_q, restart.



- Discrete Gaussian with odd support.
- If $f \hookleftarrow D_\sigma^\times$, then $\|f\| \leq \sigma\sqrt{n}$, with overwhelming prob.
- Here, we need $\sigma \geq \sqrt{q}$.

- We also want f invertible mod p: handled by tweaking $D_\sigma^\times$.

Making $h = g/f$ statistically close to uniform

Main technical contribution of StSt11

If $\sigma \geq n \cdot q^{\frac{1}{2}+\varepsilon}$ with $\varepsilon > 0$, then:

$$\Delta\left[\frac{D_\sigma^\times}{D_\sigma^\times} \bmod q\ ,\ U(R_q^\times)\right] \ \leq\ q^{-\Omega(\varepsilon \cdot n)},$$

where $\Delta(D_1, D_2) = \frac{1}{2}\sum_t |D_1(t) - D_2(t)|$ is the stat. distance.

- We don't get uniformity in R_q but only in $R_q^\times$.
- Proof based on smoothing phenomenon of Gaussians modulo lattices (MR04).

Making $h = g/f$ statistically close to uniform

Main technical contribution of StSt11

If $\sigma \geq n \cdot q^{\frac{1}{2}+\varepsilon}$ with $\varepsilon > 0$, then:

$$\Delta\left[\frac{D_\sigma^\times}{D_\sigma^\times} \bmod q\ ,\ U(R_q^\times)\right] \ \leq\ q^{-\Omega(\varepsilon \cdot n)},$$

where $\Delta(D_1, D_2) = \frac{1}{2}\sum_t |D_1(t) - D_2(t)|$ is the stat. distance.

- We don't get uniformity in R_q but only in $R_q^\times$.
- Proof based on smoothing phenomenon of Gaussians modulo lattices (MR04).

Making $h = g/f$ statistically close to uniform

Main technical contribution of StSt11

If $\sigma \geq n \cdot q^{\frac{1}{2}+\varepsilon}$ with $\varepsilon > 0$, then:

$$\Delta\left[\frac{D_\sigma^\times}{D_\sigma^\times} \bmod q\ ,\ U(R_q^\times)\right] \ \leq\ q^{-\Omega(\varepsilon \cdot n)},$$

where $\Delta(D_1, D_2) = \frac{1}{2}\sum_t |D_1(t) - D_2(t)|$ is the stat. distance.

- We don't get uniformity in R_q but only in $R_q^\times$.
- Proof based on smoothing phenomenon of Gaussians modulo lattices (MR04).

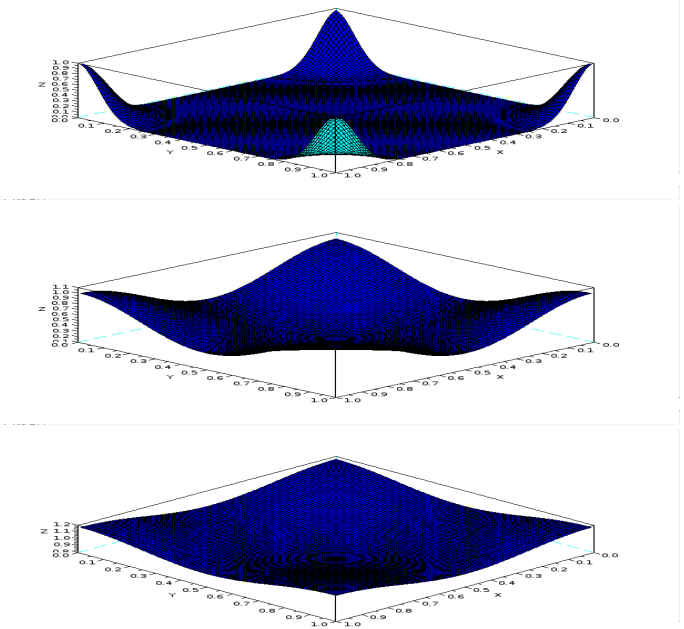
Gaussians Modulo Lattices: Smoothing Phenomenon

Take a continuous Gaussian density function $\nu_\sigma(x)$ on $\mathbb{R}^n$, width parameter σ:

$$\nu_\sigma(x) = \sigma^{-n} \cdot \rho_\sigma(x), \rho(x) \stackrel{\text{def}}{=} e^{-\pi\|x\|^2/\sigma^2},$$

and reduce it modulo a lattice $L \subseteq \mathbb{R}^n$:

$$\nu'_\sigma(x) \stackrel{\text{def}}{=} (\nu_\sigma \bmod L)(x) = \sum_{v \in L} \nu_\sigma(x+v).$$



Two regions for ν'_σ depending on σ:

- Wavy region: σ small
- Smooth region: σ large

As σ increases, ν'_σ approaches uniformity – Micciancio-Regev (2004) studied this smoothing phenomenon.

Quantifying Smoothing Phenomenon: Fourier Analysis

ν'_σ naturally extends to an L-periodic function on $\mathbb{R}^n$:

$$\nu'_\sigma(x) \stackrel{\text{def}}{=} \sum_{v \in L} \nu_\sigma(x+v) \to \nu'_\sigma(x+t) = \nu'_\sigma(x) \ \forall \ t \in L.$$

L-Periodic functions have a Fourier series decomposition:

$$\nu'_\sigma(x) = \det \hat{L} \cdot \sum_{w \in \hat{L}} c_{\sigma,w} e^{2\pi i \langle x,w \rangle},$$

- $\hat{L}$ is the dual lattice: $\{w \in \mathbb{R}^n : \forall v \in L, \langle w, v \rangle \in \mathbb{Z}\}$.
- w'th Fourier component $\psi_w(x) = e^{2\pi i \langle x,w \rangle}$
  - $w = 0$ – uniform component, $w \in \hat{L} \setminus \{0\}$ – non-uniform wavy component, wave period $= 1/\|w\|$ in direction of w
- w'th Fourier coefficient $c_{\sigma,w}$ is Fourier transform of ν_σ evaluated at w: $c_{\sigma,w} = \int_{\mathbb{R}^n} \nu_\sigma(x) e^{2\pi i \langle x,w \rangle} dx = \rho_{1/\sigma}(w)$.

Quantifying Smoothing Phenomenon: Fourier Analysis

$$\nu'_\sigma(x) = \det \hat{L} \cdot \sum_{w \in \hat{L}} \rho_{1/\sigma}(w) e^{2\pi i \langle x, w \rangle},$$

Stat. dist. $\Delta \stackrel{\text{def}}{=} \int_{P(L)} |\nu'_\sigma(x) - \det L^{-1}| dx$ of ν'_σ to uniform:

$$\Delta \leq S_\sigma(L) \stackrel{\text{def}}{=} \sum_{w \in \hat{L} \setminus 0} \rho_{1/\sigma}(w).$$

ε-Smoothing Parameter $\eta_\varepsilon(L)$ of lattice L

$\eta_\varepsilon(L) \stackrel{\text{def}}{=}$ Smallest σ such that $S_\sigma(L) \leq \varepsilon$.

Smooth region: $1/\sigma < \lambda_1(\hat{L})$ – terms in $S_\sigma(L)$ in 'tail' of $\rho_{1/\sigma}$.

Theorem [MR04]

Fix $\varepsilon > 0$. We have:

$$\eta_\varepsilon(L) \leq \sqrt{n \ln(2n(1 + 1/\varepsilon))} \cdot \lambda(\hat{L})^{-1}.$$

[GPV08]: Extends to discrete Gaussians: $\nu_\sigma \to D_{\mathbb{Z}^n,\sigma}$ for $L \subseteq \mathbb{Z}^n$.

Proving uniformity of $h = g/f$ (1)

Outline of proof

- Goal: $\Delta = \frac{1}{2}\sum_{h \in R_q^*} |\Pr_{f,g}[g/f = h] - |R_q^*|^{-1}| \leq \varepsilon$.
- Sufficient term-wise condition:
  $|\Pr_{f,g}[g/f = h] - |R_q^*|^{-1}| < |R_q^*|^{-1} \cdot \varepsilon$.
- Since $g/f = h$ equivalent to $fh - g = 0$, suffices to show

$$|\Pr_{f,g}[fh - g = 0] - |R_q^*|^{-1}| < |R_q^*|^{-1} \cdot \varepsilon.$$

- **Observation:** $\Pr_{f,g}[fh - g = 0]$ is prob. that (f, g) falls in NTRU lattice

$$L_h \stackrel{\text{def}}{=} \{(f, g) \in R^2 : fh - g = 0 \bmod q\}.$$

- $\rightarrow$ suffices to show distrib. $\chi_\sigma = (D^*_{\mathbb{Z}^n,\sigma})^2$ of (f, g) reduced modulo lattice L_h is close to uniform on $\mathbb{Z}^{2n}/L_h$.
- $\rightarrow$ Can almost directly apply smoothing!

Proving uniformity of $h = g/f$ (2)

Q: How to deal with non-lattice support of $\chi_\sigma = (D^*_{\mathbb{Z}^n,\sigma})^2$?

A: Decompose it in terms of lattices:

- $\mathbb{Z}^{2n} \cap (R_q^*)^2 = \mathbb{Z}^{2n} \setminus \bigcup_{S \subseteq \{1,\ldots,n\}} I_S$
  - I_S denotes the ideal of R_q generated by $\prod_{i \in S} \phi_i(x)$
  - $\phi_1, \ldots, \phi_n$ denote the irreducible factors of $\phi = x^n + 1 \bmod q$
- $L_h^* \cap (R_q^*)^2 = L_h \setminus \bigcup_{S \subseteq \{1,\ldots,n\}} L_h(I_S)$
  - $L_h(I_S) \stackrel{\text{def}}{=} L_h \cap (I_S \times I_S)$.

Apply inclusion-exclusion to reduce termwise to lattice smoothing:

$$\Pr_{f,g}[(f,g) \in L_h] = \frac{\sum_{S \subseteq \{1,\ldots,n\}} (-1)^{|S|} \cdot D_{\mathbb{Z}^{2n},\sigma}(L_h(I_S))}{\left(\sum_{S \subseteq \{1,\ldots,n\}} (-1)^{|S|} \cdot D_{\mathbb{Z}^n,\sigma}(I_S)\right)^2}.$$

Proving uniformity of $h = g/f$ (3)

Apply smoothing-parameter method – need lower bound on minimum of dual $\lambda(\widehat{L_h(I_S)})$.
Dual of $L_h(I_S)$ has a simple algebraic description!
Counting argument based on algebraic structure of $R_q \rightarrow$ probabilistic bound on minimum.

Smoothing paramter of gen. NTRU lattices $L_h(I_S)$ [StSt11]

Fix $\varepsilon, \varepsilon' > 0$. Let $n \geq 8$ be a power of 2 such that $\phi = x^n + 1$ splits into n linear factors modulo a prime $q \geq 5$ and let $S \subseteq [n]$ with $|S| \leq \varepsilon' n$. Then for all except a fraction $\leq 2^{8n} q^{-2\varepsilon n}$ of $h \in (R_q^*)$, we have

$$\eta_\varepsilon(L_h(I_S)) \leq \sqrt{n \ln(4n(1+1/\varepsilon))/\pi} \cdot q^{\frac{1}{2}+\varepsilon'/2+\varepsilon}$$

Hardness of DNCC: The R-LWE Problem

- The error distribution ν_β:
  - n-dimensional Gaussian of standard deviation $\beta \ll q$,
  - rounded to $\mathbb{Z}^n$,
  - looked at as an element of R^+.

  $\Rightarrow$ Small element of R^+.

- The R-LWE distribution D_β:
  - Sample $a \hookleftarrow U(R_q^+)$, $s \hookleftarrow \nu_\beta$, $e \hookleftarrow \nu_\beta$,
  - Return $(a, as + e) \in R_q^+ \times R_q^+$.

R-LWE (simplified)

Distinguish between D_β and $U(R_q^+ \times R_q^+)$.

$\rightarrow$ If $\chi_\beta = \chi_\rho = \nu_\beta$, DNCC coincides with R-LWE!

Hardness of DNCC: The R-LWE Problem

- The error distribution ν_β:
  - n-dimensional Gaussian of standard deviation $\beta \ll q$,
  - rounded to $\mathbb{Z}^n$,
  - looked at as an element of R^+.

  $\Rightarrow$ Small element of R^+.

- The R-LWE distribution D_β:
  - Sample $a \hookleftarrow U(R_q^+)$, $s \hookleftarrow \nu_\beta$, $e \hookleftarrow \nu_\beta$,
  - Return $(a, as + e) \in R_q^+ \times R_q^+$.

R-LWE (simplified)

Distinguish between D_β and $U(R_q^+ \times R_q^+)$.

$\rightarrow$ If $\chi_\beta = \chi_\rho = \nu_\beta$, DNCC coincides with R-LWE!

Hardness of DNCC: The R-LWE Problem

- The error distribution ν_β:
  - n-dimensional Gaussian of standard deviation $\beta \ll q$,
  - rounded to $\mathbb{Z}^n$,
  - looked at as an element of R^+.

  $\Rightarrow$ Small element of R^+.

- The R-LWE distribution D_β:
  - Sample $a \hookleftarrow U(R_q^+)$, $s \hookleftarrow \nu_\beta$, $e \hookleftarrow \nu_\beta$,
  - Return $(a, as + e) \in R_q^+ \times R_q^+$.

R-LWE (simplified)

Distinguish between D_β and $U(R_q^+ \times R_q^+)$.

$\rightarrow$ If $\chi_\beta = \chi_\rho = \nu_\beta$, DNCC coincides with R-LWE!

Computational hardness of R-LWE

Hardness of R-LWE [LyPeRe'10]

$\mathcal{P}oly(n)$-Ideal-SVP reduces to R-LWE in quantum polynomial-time.

- Security under R-LWE implies security under Ideal-SVP.
- $\mathcal{P}oly(n)$-Ideal-SVP is conjectured hard, even using quantum computations.

Computational hardness of R-LWE

Hardness of R-LWE [LyPeRe'10]

$\mathcal{P}oly(n)$-Ideal-SVP reduces to R-LWE in quantum polynomial-time.

- Security under R-LWE implies security under Ideal-SVP.
- $\mathcal{P}oly(n)$-Ideal-SVP is conjectured hard, even using quantum computations.

Homomorphic Encryption

Homomorphic encryption with respect to function f – Given:

- Ciphertexts for messages $m_1, \ldots, m_t$ (encrypted under pk_A):
  - $c_1 = \text{Enc}_{pk_A}(m_1), \ldots, c_t = \text{Enc}_{pk_A}(m_t)$
- a function $f(m_1, \ldots, m_t)$,
- and Alice's public key pk_A,

Bob can compute: $c = \text{Enc}_{pk_A}(m)$ for $m = f(m_1, \ldots, m_t)$.

Primary Application: Private 'cloud computing':

- **Setup:** Alice generates (sk_A, pk_A), keeps sk_A private, stores $c_i = \text{Enc}_{pk_A}(m_i)$ on cloud server (Bob).
- **Query:** Alice issues to Bob search query f: $f(m_1, \ldots, m_t) = \{m_i : \text{Title}(m_i) = \text{'Bank Statement'}\}$.
- **Response:** Bob uses f and $c_1, \ldots, c_t$ to compute $c = \text{Enc}_{pk_A}(f(m_1, \ldots, m_t))$, and returns c to Alice.
- **Decryption:** Alice uses sk_A to decrypts c and obtain $m = f(m_1, \ldots, m_t)$.

Homomorphic Encryption

Homomorphic encryption with respect to function f – Given:

- Ciphertexts for messages $m_1, \ldots, m_t$ (encrypted under pk_A):
  - $c_1 = \text{Enc}_{pk_A}(m_1), \ldots, c_t = \text{Enc}_{pk_A}(m_t)$
- a function $f(m_1, \ldots, m_t)$,
- and Alice's public key pk_A,

Bob can compute: $c = \text{Enc}_{pk_A}(m)$ for $m = f(m_1, \ldots, m_t)$.

Primary Application: Private 'cloud computing':

- **Setup:** Alice generates (sk_A, pk_A), keeps sk_A private, stores $c_i = \text{Enc}_{pk_A}(m_i)$ on cloud server (Bob).
- **Query:** Alice issues to Bob search query f: $f(m_1, \ldots, m_t) = \{m_i : \text{Title}(m_i) = \text{'Bank Statement'}\}$.
- **Response:** Bob uses f and $c_1, \ldots, c_t$ to compute $c = \text{Enc}_{pk_A}(f(m_1, \ldots, m_t))$, and returns c to Alice.
- **Decryption:** Alice uses sk_A to decrypts c and obtain $m = f(m_1, \ldots, m_t)$.

Homomorphic Encryption: Realization

History:

- Concept proposed in the 1970s [RAD78]
- 1970s – 2009: Very limited classes of functions f allowed.
- 2009: First plausible scheme supporting arbitrary functions f (Fully Homomorphic Encryption - FHE)
  - Based on hardness of approx-Ideal-SVP variants
  - Inefficient – Huge parameters
- 2010-2013: Significant improvements: Most efficient schemes based on R-LWE (BV11,BGV12,B12,LTV12,GHS13)

One of the more efficient schemes (LTV12): NTRU variant

- Ciphertext same as standard NTRU encryption
- Hardness Assumptions:
  - DNKC problem in the computational region: $\sigma < q^{1/2}$.
  - 'circular' variant of R-LWE problem = DNCC problem.

Homomorphic Encryption: NTRU Variant (LTV12)

Observation: NTRU has natural 'semi homomorphic' properties

Given ciphertexts $c_i = h \cdot s_i + pe_i + m_i \in R_q$, $i \in \{1, 2\}$:

- **Addition** (ciphertexts over R_q, messages over R_p):
  - $c = c_1 + c_2 = h \cdot (s_1 + s_2) + p(e_1 + e_2) + (m_1 + m_2)$.
  - c decrypts with $sk = f$ to message $m = m_1 + m_2 \in R_p$
  - 'Semi-homomorphic' catch: Need $\|p(gs + fe) + m\|_\infty < q/2$), with $s = s_1 + s_2$, $e = e_1 + e_2$.
- **Multiplication** (ciphertexts over R_q, messages over R_p):
  - $c = c_1 \cdot c_2 =$ $h^2 s_1 s_2 + h(s_1 e_2' + s_2 e_1') + p(e_1 e_2 + e_1 m_2 + e_2 m_1) + m_1 m_2$.
  - c decrypts with $sk = f^2$ to message $m = m_1 \cdot m_2 \in R_p$.
  - 'Semi-homomorphic' catch: Need $\|(pg)^2 s_1 s_2 + (pfg)(s_1 e_2' + s_2 e_1') + f^2(p(e_1 e_2 + e_1 m_2 + e_2 m_1) + m_1 m_2)\|_\infty < q/2$.

Multiplication is the bottleneck: Even 1 mult requires $(\sigma \cdot \beta \mathcal{P}oly(n))^2 \leq q/2$.

Homomorphic Encryption: NTRU Variant (LTV12)

Observation: NTRU has natural 'semi homomorphic' properties

Given ciphertexts $c_i = h \cdot s_i + pe_i + m_i \in R_q$, $i \in \{1, 2\}$:

- **Addition** (ciphertexts over R_q, messages over R_p):
  - $c = c_1 + c_2 = h \cdot (s_1 + s_2) + p(e_1 + e_2) + (m_1 + m_2)$.
  - c decrypts with $sk = f$ to message $m = m_1 + m_2 \in R_p$
  - 'Semi-homomorphic' catch: Need $\|p(gs + fe) + m\|_\infty < q/2$), with $s = s_1 + s_2$, $e = e_1 + e_2$.
- **Multiplication** (ciphertexts over R_q, messages over R_p):
  - $c = c_1 \cdot c_2 =$ $h^2 s_1 s_2 + h(s_1 e_2' + s_2 e_1') + p(e_1 e_2 + e_1 m_2 + e_2 m_1) + m_1 m_2$.
  - c decrypts with $sk = f^2$ to message $m = m_1 \cdot m_2 \in R_p$.
  - 'Semi-homomorphic' catch: Need $\|(pg)^2 s_1 s_2 + (pfg)(s_1 e_2' + s_2 e_1') + f^2(p(e_1 e_2 + e_1 m_2 + e_2 m_1) + m_1 m_2)\|_\infty < q/2$.

Multiplication is the bottleneck: Even 1 mult requires $(\sigma \cdot \beta \mathcal{P}oly(n))^2 \leq q/2$.

Homomorphic Encryption: NTRU Variant (LTV12)

Observation: NTRU has natural 'semi homomorphic' properties

Given ciphertexts $c_i = h \cdot s_i + pe_i + m_i \in R_q$, $i \in \{1, 2\}$:

- **Addition** (ciphertexts over R_q, messages over R_p):
  - $c = c_1 + c_2 = h \cdot (s_1 + s_2) + p(e_1 + e_2) + (m_1 + m_2)$.
  - c decrypts with $sk = f$ to message $m = m_1 + m_2 \in R_p$
  - 'Semi-homomorphic' catch: Need $\|p(gs + fe) + m\|_\infty < q/2$), with $s = s_1 + s_2$, $e = e_1 + e_2$.
- **Multiplication** (ciphertexts over R_q, messages over R_p):
  - $c = c_1 \cdot c_2 =$ $h^2 s_1 s_2 + h(s_1 e_2' + s_2 e_1') + p(e_1 e_2 + e_1 m_2 + e_2 m_1) + m_1 m_2$.
  - c decrypts with $sk = f^2$ to message $m = m_1 \cdot m_2 \in R_p$.
  - 'Semi-homomorphic' catch: Need $\|(pg)^2 s_1 s_2 + (pfg)(s_1 e_2' + s_2 e_1') + f^2(p(e_1 e_2 + e_1 m_2 + e_2 m_1) + m_1 m_2)\|_\infty < q/2$.

Multiplication is the bottleneck: Even 1 mult requires $(\sigma \cdot \beta \mathcal{P}oly(n))^2 \leq q/2$.

Homomorphic Encryption: NTRU Variant (LTV12)

For depth d homomorphic multiplications, 'doubly exponential norm blowup' difficulty:

- Need to decrypt with $sk = f^{2^d}$ – need $(\sigma\beta\mathcal{P}oly(n))^{2^d} < q/2$.
- For hardness of DNKC, allows d only up to $O(\log n)$!

Improvement to support larger d (based on BV11): Relinearization

- Apply relinearization procedure to $c = c_1 \cdot c_2$ after each homomorphic multiplication
- Produce $\hat{c}$ encrypting same message $m_1 \cdot m_2 \in R_p$ as c, but decryptable with the f, not f^2 $\rightarrow$ avoid the exponential degree blowup for sk.

Homomorphic Encryption: NTRU Variant (LTV12)

For depth d homomorphic multiplications, 'doubly exponential norm blowup' difficulty:

- Need to decrypt with $sk = f^{2^d}$ – need $(\sigma\beta\mathcal{P}oly(n))^{2^d} < q/2$.
- For hardness of DNKC, allows d only up to $O(\log n)$!

Improvement to support larger d (based on BV11): Relinearization

- Apply relinearization procedure to $c = c_1 \cdot c_2$ after each homomorphic multiplication
- Produce $\hat{c}$ encrypting same message $m_1 \cdot m_2 \in R_p$ as c, but decryptable with the f, not f^2 $\rightarrow$ avoid the exponential degree blowup for sk.

Homomorphic Encryption: NTRU Variant (LTV12)

Idea of Relinearization:

- Modify key generation – Alice publishes $\approx \log q$ additional ring elements ζ_τ : 'pseudo-encryptions' of f^2:

$$\zeta_\tau = h \cdot s_\tau + p e_\tau + 2^\tau f^2 \in R_q \text{ for } \tau = 0, \ldots, \lfloor \log q \rfloor,$$

- Relinearization procedure – split c into its binary reprsentation $\sum_\tau c_\tau 2^\tau$ and compute

$$\hat{c} = \sum_\tau c_\tau \cdot \zeta_\tau = h \cdot (\sum_\tau c_\tau s_\tau) + p \cdot (\sum_\tau c_\tau e_\tau) + f^2 \cdot (\sum_\tau c_\tau 2^\tau).$$

Second Improvement (based on BV11): Modulus Reduction.

- Scales down the ciphertext $\hat{c}$ from R_q to $R_{q'}$ with $q' < q$
- Noise in $\hat{c}'$ also scaled down by the ratio q'/q.

Overall: Can support f of mult. depth $d = O(n^\varepsilon)$.

- With 'bootstrapping' technique [G09], arbitrary f (FHE).

Homomorphic Encryption: NTRU Variant (LTV12)

Idea of Relinearization:

- Modify key generation – Alice publishes $\approx \log q$ additional ring elements ζ_τ : 'pseudo-encryptions' of f^2:

$$\zeta_\tau = h \cdot s_\tau + p e_\tau + 2^\tau f^2 \in R_q \text{ for } \tau = 0, \ldots, \lfloor \log q \rfloor,$$

- Relinearization procedure – split c into its binary reprsentation $\sum_\tau c_\tau 2^\tau$ and compute

$$\hat{c} = \sum_\tau c_\tau \cdot \zeta_\tau = h \cdot (\sum_\tau c_\tau s_\tau) + p \cdot (\sum_\tau c_\tau e_\tau) + f^2 \cdot (\sum_\tau c_\tau 2^\tau).$$

Second Improvement (based on BV11): Modulus Reduction.

- Scales down the ciphertext $\hat{c}$ from R_q to $R_{q'}$ with $q' < q$
- Noise in $\hat{c}'$ also scaled down by the ratio q'/q.

Overall: Can support f of mult. depth $d = O(n^\varepsilon)$.

- With 'bootstrapping' technique [G09], arbitrary f (FHE).

Homomorphic Encryption: NTRU Variant (LTV12)

Idea of Relinearization:

- Modify key generation – Alice publishes $\approx \log q$ additional ring elements ζ_τ : 'pseudo-encryptions' of f^2:

$$\zeta_\tau = h \cdot s_\tau + p e_\tau + 2^\tau f^2 \in R_q \text{ for } \tau = 0, \ldots, \lfloor \log q \rfloor,$$

- Relinearization procedure – split c into its binary reprsentation $\sum_\tau c_\tau 2^\tau$ and compute

$$\hat{c} = \sum_\tau c_\tau \cdot \zeta_\tau = h \cdot (\sum_\tau c_\tau s_\tau) + p \cdot (\sum_\tau c_\tau e_\tau) + f^2 \cdot (\sum_\tau c_\tau 2^\tau).$$

Second Improvement (based on BV11): Modulus Reduction.

- Scales down the ciphertext $\hat{c}$ from R_q to $R_{q'}$ with $q' < q$
- Noise in $\hat{c}'$ also scaled down by the ratio q'/q.

Overall: Can support f of mult. depth $d = O(n^\varepsilon)$.

- With 'bootstrapping' technique [G09], arbitrary f (FHE).

Homomorphic Encryption: NTRU Variant (LTV12)

Relinearization $\rightarrow$ security now relies on new variant of DNKC:

NTRU Decision Circular Key Cracking Problem $\mathtt{DNCKC}_{n,q,\phi,\chi_\sigma,\chi_\beta,\ell}$

Given (n, q, ϕ) and $(h, \{\zeta_\tau\}_\tau)$, distinguish

- NTRU circular key distribution $D_0 = \{(h = g/f \in R_q, \zeta_\tau = h \cdot s_\tau + pe_\tau + 2^\tau f^2 \in R_q : f, g \hookleftarrow \chi_\sigma, s_\tau, e_\tau \hookleftarrow \chi_\beta\}$.
- Uniform key distribution $D_1 = U(R_q^*) \times U(R_q^\ell)$.

Q: Hardness??

Cryptographic Multilinear Maps

Example Motivation: Non-interactive Key exchange
Classical Diffie-Hellman Non-Interactive 2-party Key Exchange (1976)

- Publish a cyclic group G (generator g, order q) where Discrete Log (DL) problem is hard.
- Alice chooses random $x_1 \in \mathbb{Z}_q$, publishes $y_1 = g^{x_1}$.
- Bob chooses random $x_2 \in \mathbb{Z}_q$, publishes $y_2 = g^{x_2}$.
- Both Alice and Bob compute agreed secret key $K = g^{x_1 x_2} = y_1^{x_2} = y_2^{x_1}$.

Cryptographic Multilinear Maps

Q: How to generalize Diffie-Hellman to $N > 2$ parties?

A[J00,BS02]: Use a group where DL is hard and there is an efficient $(N-1)$-linear map $e : G_1 \times \cdots \times G_{N-1} \to G_T$:

$$e(g^{x_1}, g^{x_2}, \ldots, g^{x_{N-1}}) = e(g,g)^{x_1 \cdots x_{N-1}} \forall x_1, \ldots, x_{N-1} \in \mathbb{Z}_q.$$

N-party Non-Interactive Key Exchange

- Publish a cyclic group G (generator g, order q) where Discrete Log (DL) problem is hard, with an efficient $(N-1)$-linear map e.
- For $i \in \{1, \ldots, N\}$, party P_i chooses random $x_i \in \mathbb{Z}_q$, publishes $y_i = g^{x_i}$.
- All parties can compute agreed secret key $K = e(g, \ldots, g)^{x_1 \cdots x_N} = e(y_2, y_3, \ldots, y_N)^{x_1}$.

Cryptographic Multilinear Maps

Q: How to generalize Diffie-Hellman to $N > 2$ parties?
A[J00,BS02]: Use a group where DL is hard and there is an efficient $(N-1)$-linear map $e : G_1 \times \cdots \times G_{N-1} \to G_T$:

$$e(g^{x_1}, g^{x_2}, \ldots, g^{x_{N-1}}) = e(g,g)^{x_1 \cdots x_{N-1}} \forall x_1, \ldots, x_{N-1} \in \mathbb{Z}_q.$$

N-party Non-Interactive Key Exchange

- Publish a cyclic group G (generator g, order q) where Discrete Log (DL) problem is hard, with an efficient $(N-1)$-linear map e.
- For $i \in \{1, \ldots, N\}$, party P_i chooses random $x_i \in \mathbb{Z}_q$, publishes $y_i = g^{x_i}$.
- All parties can compute agreed secret key $K = e(g, \ldots, g)^{x_1 \cdots x_N} = e(y_2, y_3, \ldots, y_N)^{x_1}$.

Cryptographic Multilinear Maps

Q: How to generalize Diffie-Hellman to $N > 2$ parties?
A[J00,BS02]: Use a group where DL is hard and there is an efficient $(N-1)$-linear map $e : G_1 \times \cdots \times G_{N-1} \to G_T$:

$$e(g^{x_1}, g^{x_2}, \ldots, g^{x_{N-1}}) = e(g,g)^{x_1 \cdots x_{N-1}} \forall x_1, \ldots, x_{N-1} \in \mathbb{Z}_q.$$

N-party Non-Interactive Key Exchange

- Publish a cyclic group G (generator g, order q) where Discrete Log (DL) problem is hard, with an efficient $(N-1)$-linear map e.
- For $i \in \{1, \ldots, N\}$, party P_i chooses random $x_i \in \mathbb{Z}_q$, publishes $y_i = g^{x_i}$.
- All parties can compute agreed secret key $K = e(g, \ldots, g)^{x_1 \cdots x_N} = e(y_2, y_3, \ldots, y_N)^{x_1}$.

k-linear Maps: History

- 2000: Bilinear ($k = 2$) via Weil pairings on algebraic curves, applications:
  - 2000: 3-party non-interactive key agreement [J00]
  - 2000-2001: Identity-Based Encryption (IBE) [SK00,BF01]
  - 2001: Short signatures [BS01]
  - 2000-2013: **lots** of others
- 2002: Applications for k-linear maps [BS02]
  - $(k + 1)$-party non-interactive key agreement
  - Efficient Broadcast Encryption
  - and others...
- 2012: First plausible realization for $k > 2$, via ideal lattices [GGH12], applications:
  - 2012-2013: Functional Encryption for arbitrary functions
  - 2013: Program obfuscation notions for arbitrary functions
  - and others...

k-linear Maps: GGH Realization

[GGH12] realization: not quite a k-linear map, but essentially the same

Technically, a k-graded encoding scheme:

- Replace groups $\mathbb{Z}_q, G$ by
  - Rings R_p, R_q.
- Replace $x \to g^x$ by
  - $x \to \mathrm{Enc}_1(x; \rho)$ – randomized 'level 1 encoding' of x.
- Replace $e(g^{x_1}, \ldots, g^{x_k}) = e(g, \ldots, g)^{x_1 \cdots x_k}$ by
  - Homomorphic up to 'level k':
    $\mathrm{Enc}_1(x_1; \rho_1) \cdots \mathrm{Enc}_1(x_k; \rho_k) = \mathrm{Enc}_k(x_1 \cdots x_k; \rho)$ for some ρ.
  - Randomness-independent extraction at level k –
    $\mathrm{Ext} : R_q \to \{0,1\}^\ell$: $\mathrm{Ext}(\mathrm{Enc}_k(x; \rho)) = r(x) \in \{0,1\}^n$ is independent of randomness ρ, and uniformly random for $x \hookleftarrow U(R_p)$.

GGH k-graded encoded scheme (LSS13 simplified - GGHLite)

Public Params: Choose a secret 'small' $p \in R$. Publish (h_1, h_2, e_k), with:

- $h_1 = pg_1/f, h_2 = pg_2/f \in R_q$, with 'small' $f, g_1, g_2 \hookleftarrow \chi_\sigma$ with $\sigma < q^{1/k}$.
- $e_k = uf^k/p \in R_q$, for u of norm $\|u\| = \mathcal{P}oly(n) \cdot q^{1/2}$.

Level 1 Encoding of $m \in R_p$:

$c = \mathrm{Enc}_1(m) = h_1 s_1 + h_2 s_2 + m \in R_q$, with s_i 'small'.

- Note: $\mathrm{Enc}_1(m) = pg'/f + m$ for a small g'
- By homomorphic property, $\mathrm{Enc}_1(m_1) \cdots \mathrm{Enc}_1(m_k) = pg'/f^k + m = \mathrm{Enc}_1(m)$, with $m = m_1 \cdots m_k \bmod p$.

Level k Representative Extraction: $\mathrm{Ext}(e_k, c) = MSB_\ell(e_k \cdot c)$.

- Note: $c = \mathrm{Enc}_1(m; \rho) = pg'/f^k + m \to \mathrm{Ext}(e_k, c) = MSB_\ell(ug' + uf^k/p \cdot m) = MSB_\ell(uf^k/p \cdot m)$ since ug' small.

GGH k-graded encoded scheme (LSS13 simplified - GGHLite)

Scheme security depends on hardness of a new variant of DNKC problem:

k-graded NTRU Discrete-Log Problem $\mathtt{DNDL}_{n,q,\phi,\chi_\sigma,\chi_\beta,\ell}$

Given

- $(n, q, h_1 = pg_1/f, h_2 = pg_2/f, e_k = uf^k/p)$
- $c = h_1 s_1 + h_2 s_2 + m \in R_q$,

find m' with $\|m'\|$ 'small' (less than q) such that $m' = m \bmod p$.

Note:

- Without knowing e_k, similar to standard DNCC problem.
- With e_k, may be easier (e.g. checking c is encoding of m is easy!).

Conclusions

Intersting recent developments in both security analysis and applications of NTRU.

Important open problems:

- Hardness of NTRU key cracking problem in the computational region $\sigma < q^{1/2}$ (applications: efficient parameters, FHE, multilinear maps)
  - Starting point: statistical properties of $h = g/f \in R_q$ in this region?
- Hardness of circular NTRU ciphertext cracking problem (application: FHE)
- Hardness of k-graded NTRU discrete-log problem (application: multilinear maps)