Une preuve de sécurité pour le cryptosystème NTRU

Damien Stehlé et Ron Steinfeld

CNRS – ENS de Lyon
Macquarie University

Grenoble, Mai 2012

The NTRU cryptographic functions

NTRUEncrypt: A public-key encryption scheme.

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

NTRUSign: A digital signature scheme.

- 2001: Hoffstein et al propose NSS.
- 2001 & 2002: Broken by Gentry, Jonsson, Stern & Szydlo.
- 2003: HoHGPiSiWh propose NTRUSign.
- 2003 & 2004: Many partial attacks.
- 2006: Total break of one of the two variants of NTRUSign, by Nguyen & Regev.

The NTRU cryptographic functions

NTRUEncrypt: A public-key encryption scheme.

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

NTRUSign: A digital signature scheme.

- 2001: Hoffstein et al propose NSS.
- 2001 & 2002: Broken by Gentry, Jonsson, Stern & Szydlo.
- 2003: HoHGPiSiWh propose NTRUSign.
- 2003 & 2004: Many partial attacks.
- 2006: Total break of one of the two variants of NTRUSign, by Nguyen & Regev.

Why studying NTRUEncrypt?

- Standardized & commercialized.
- 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:
  - Does not rely on the hardness of Int-Fac or DLog.
  - Seems to resist practical attacks.
  - Seems to resist quantum attacks.

Why studying NTRUEncrypt?

- Standardized & commercialized.
- 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:
  - Does not rely on the hardness of Int-Fac or DLog.
  - Seems to resist practical attacks.
  - Seems to resist quantum attacks.

Our main result

An IND-CPA variant of NTRUEncrypt

It is possible to modify NTRUEncrypt so that:

- Encryption and decryption of λ bits still cost $\widetilde{O}(\lambda)$.
- Any semantic attack with run-time T leads to a $\mathcal{P}oly(n, T)$ quantum algorithm for $\mathcal{P}oly(n)$-Ideal-SVP.

Our main result

An IND-CPA variant of NTRUEncrypt

It is possible to modify NTRUEncrypt so that:

- Encryption and decryption of λ bits still cost $\widetilde{O}(\lambda)$.
- Any semantic attack with run-time T leads to a $\mathcal{P}oly(n, T)$ quantum algorithm for $\mathcal{P}oly(n)$-Ideal-SVP.

Our main result

An IND-CPA variant of NTRUEncrypt

It is possible to modify NTRUEncrypt so that:

- Encryption and decryption of λ bits still cost $\widetilde{O}(\lambda)$.
- Any semantic attack with run-time T leads to a $\mathcal{P}oly(n, T)$ quantum algorithm for $\mathcal{P}oly(n)$-Ideal-SVP.

- Semantic security (IND-CPA): Given the public parameters, the attacker cannot distinguish between the encryptions of two plaintexts of his choice.
- Similar result for NTRUSign, in the random oracle model and with a non-quantum security proof.

Outline of the talk

1- **Regular** NTRUEncrypt.

2- The Ideal-SVP and R-LWE problems.

3- The modified NTRUEncrypt.

4- Modifying NTRUSign.

Polynomial Rings: Generalizing $\mathbb{Z}$

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

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

Interesting Φ's:

- $\Phi = x^n - 1 \rightarrow R^-$, $\Phi = x^n + 1 \rightarrow R^+$.
- $x^n + 1$ irreducible if n is a power of 2.
- In this case, R^{Φ} is isomorphic to the ring of integers of the cyclotomic number field:

$$\mathbb{Q}[e^{i\pi/n}] \simeq \mathbb{Q}[x]/(\Phi).$$

Polynomial Rings: Generalizing $\mathbb{Z}$

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

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

Interesting Φ's:

- $\Phi = x^n - 1 \rightarrow R^-$, $\Phi = x^n + 1 \rightarrow R^+$.
- $x^n + 1$ irreducible if n is a power of 2.
- In this case, R^{Φ} is isomorphic to the ring of integers of the cyclotomic number field:

$$\mathbb{Q}[e^{i\pi/n}] \simeq \mathbb{Q}[x]/(\Phi).$$

Polynomial Rings: Generalizing $\mathbb{Z}$

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

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

Interesting Φ's:

- $\Phi = x^n - 1 \ \rightarrow \ R^-, \quad \Phi = x^n + 1 \ \rightarrow \ R^+.$
- $x^n + 1$ irreducible if n is a power of 2.
- In this case, R^{Φ} is isomorphic to the ring of integers of the cyclotomic number field:

$$\mathbb{Q}[e^{i\pi/n}] \ \simeq \ \mathbb{Q}[x]/(\Phi).$$

Polynomial Rings: Generalizing $\mathbb{Z}/q\mathbb{Z}$

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

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

- Arithmetic in R_q^Φ costs $\widetilde{O}(n \log q)$.
- R_q^- and R_q^+ defined similarly.
- If $\Phi = x^n \pm 1$ has n distinct linear factors modulo prime q, then R_q^Φ comes with a natural FFT.

If $f \in R^\Phi$ has coefficients in $(-q/2, q/2]$, then $f \bmod q$ is f.

Polynomial Rings: Generalizing $\mathbb{Z}/q\mathbb{Z}$

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

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

- Arithmetic in R_q^Φ costs $\widetilde{O}(n \log q)$.
- R_q^- and R_q^+ defined similarly.
- If $\Phi = x^n \pm 1$ has n distinct linear factors modulo prime q, then R_q^Φ comes with a natural FFT.

The key to decryption correctness

If $f \in R^\Phi$ has coefficients in $(-q/2, q/2)$, then $(f \bmod q)$ is f.

Polynomial Rings: Generalizing $\mathbb{Z}/q\mathbb{Z}$

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

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

- Arithmetic in R_q^Φ costs $\widetilde{O}(n \log q)$.
- R_q^- and R_q^+ defined similarly.
- If $\Phi = x^n \pm 1$ has n distinct linear factors modulo prime q, then R_q^Φ comes with a natural FFT.

The key to decryption correctness

If $f \in R^\Phi$ has coefficients in $(-q/2, q/2)$, then $(f \bmod q)$ is f.

Polynomial Rings: Generalizing $\mathbb{Z}/q\mathbb{Z}$

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

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

- Arithmetic in R_q^Φ costs $\widetilde{O}(n \log q)$.
- R_q^- and R_q^+ defined similarly.
- If $\Phi = x^n \pm 1$ has n distinct linear factors modulo prime q, then R_q^Φ comes with a natural FFT.

The key to decryption correctness

If $f \in R^\Phi$ has coefficients in $(-q/2, q/2)$, then $(f \bmod q)$ is f.

Description of NTRUEncrypt, Part I

Parameters: n prime, $q \approx n$ a power of 2.
E.g.: $(n, q) = (503, 256)$.

- Secret key sk: $f, g \in R^{-}$ such that:
  - f is invertible mod q and mod 3.
  - The coeffs of f and g are in $\{-1, 0, 1\}$.
- 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.

Description of NTRUEncrypt, Part I

Parameters: n prime, $q \approx n$ a power of 2.
E.g.: $(n, q) = (503, 256)$.

- Secret key sk: $f, g \in R^{-}$ such that:
  - f is invertible mod q and mod 3.
  - The coeffs of f and g are in $\{-1, 0, 1\}$.
- 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.

Description of NTRUEncrypt, Part I

Parameters: n prime, $q \approx n$ a power of 2.
E.g.: $(n, q) = (503, 256)$.

- Secret key sk: $f, g \in R^{-}$ such that:
  - f is invertible mod q and mod 3.
  - The coeffs of f and g are in $\{-1, 0, 1\}$.
- 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.

Description of NTRUEncrypt, Part I

Parameters: n prime, $q \approx n$ a power of 2.
E.g.: $(n, q) = (503, 256)$.

- Secret key sk: $f, g \in R^{-}$ such that:
  - f is invertible mod q and mod 3.
  - The coeffs of f and g are in $\{-1, 0, 1\}$.
- 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.

Description of NTRUEncrypt, Part II

- sk: $f, g \in R$ small with f invertible mod q and mod 3.
- pk: $h = g/f \bmod q$.

Description of NTRUEncrypt, Part II

- sk: $f, g \in R$ small with f invertible mod q and mod 3.
- pk: $h = g/f \bmod q$.

Encryption of $M \in \{0,1\}[x]/(x^n - 1)$:

- Sample $s \in R_q^-$ with coeffs in $\{-1, 0, 1\}$,
- Return $C := 3hs + M \bmod q$.

Description of NTRUEncrypt, Part II

- sk: $f, g \in R$ small with f invertible mod q and mod 3.
- pk: $h = g/f \bmod q$.

Encryption of $M \in \{0,1\}[x]/(x^n - 1)$:

- Sample $s \in R_q^-$ with coeffs in $\{-1, 0, 1\}$,
- Return $C := 3hs + M \bmod q$.

Decryption of $C \in R_q^-$:

- $f \times C = 3gs + fM \bmod q$.
- g, M, f, s small $\Rightarrow$ equality holds over R^-.
- $(f \times C \bmod q) \bmod 3 = fM \bmod 3$.
- Multiply by the inverse of f mod 3.

Security intuition

Given $C \in R_q$, finding $M, s \in R$ small s.t. $C = 3hs + M$ $[q]$ is hard.

Description of NTRUEncrypt, Part II

- sk: $f, g \in R$ small with f invertible mod q and mod 3.
- pk: $h = g/f \bmod q$.

Encryption of $M \in \{0,1\}[x]/(x^n - 1)$:

- Sample $s \in R_q^-$ with coeffs in $\{-1, 0, 1\}$,
- Return $C := 3hs + M \bmod q$.

Decryption of $C \in R_q^-$:

- $f \times C = 3gs + fM \bmod q$.
- g, M, f, s small $\Rightarrow$ equality holds over R^-.
- $(f \times C \bmod q) \bmod 3 = fM \bmod 3$.
- Multiply by the inverse of f mod 3.

Security intuition

Given $C \in R_q$, finding $M, s \in R$ small s.t. $C = 3hs + M$ $[q]$ is hard.

Outline of the talk

1- Regular NTRUEncrypt.

2- **The Ideal-SVP and R-LWE problems.**

3- The modified NTRUEncrypt.

4- Modifying NTRUSign.

Ideals in R^Φ

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

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

- Let's identify polynomials to vectors via their coefficients:

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

- Ideal I is mapped to an integer lattice.
- A Φ-ideal lattice is a lattice corresponding to an ideal of R^Φ.

Ideals in R^Φ

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

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

- Let's identify polynomials to vectors via their coefficients:

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

- Ideal I is mapped to an integer lattice.
- A Φ-ideal lattice is a lattice corresponding to an ideal of R^Φ.

Ideals in R^Φ

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

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

- Let's identify polynomials to vectors via their coefficients:

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

- Ideal I is mapped to an integer lattice.
- A Φ-ideal lattice is a lattice corresponding to an ideal of R^Φ.

(Integral) lattices and the Shortest Vector Problem

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

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

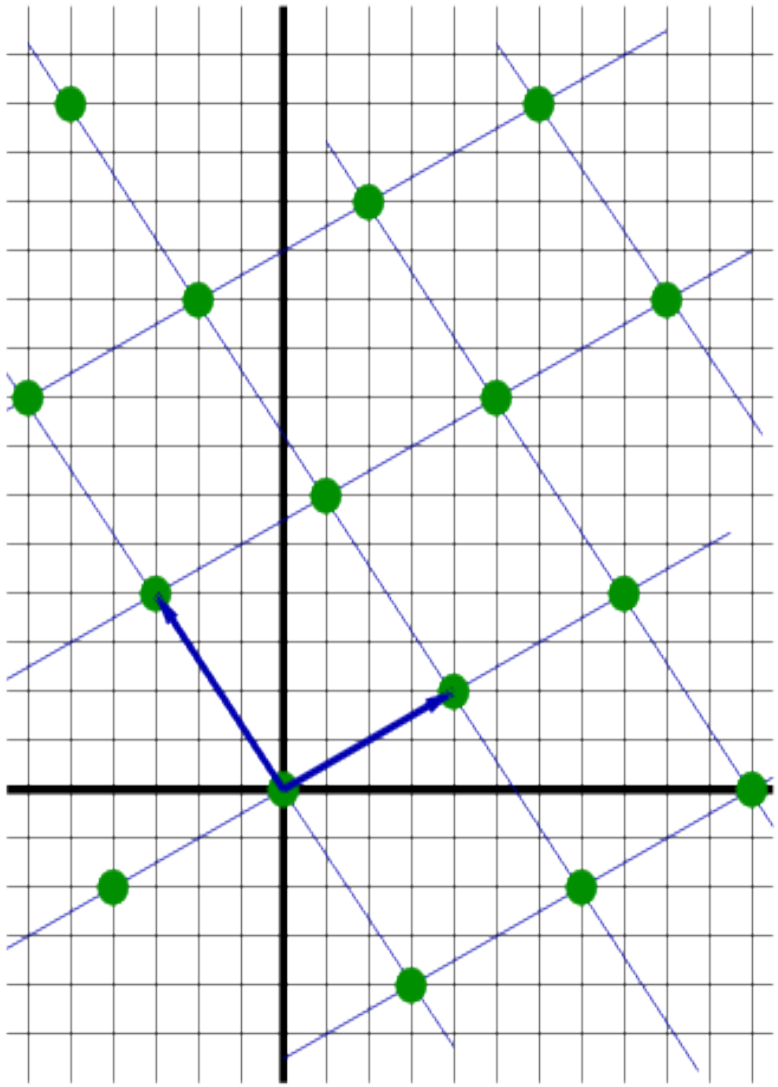
γ-SVP (computational variant)

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

No known sub-exp. algo. for $\gamma = \mathcal{P}oly(n)$.

γ-Ideal-SVP:

- γ-SVP restricted to Φ-ideal lattices.
- Does not seem easier than SVP.



(Integral) lattices and the Shortest Vector Problem

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

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

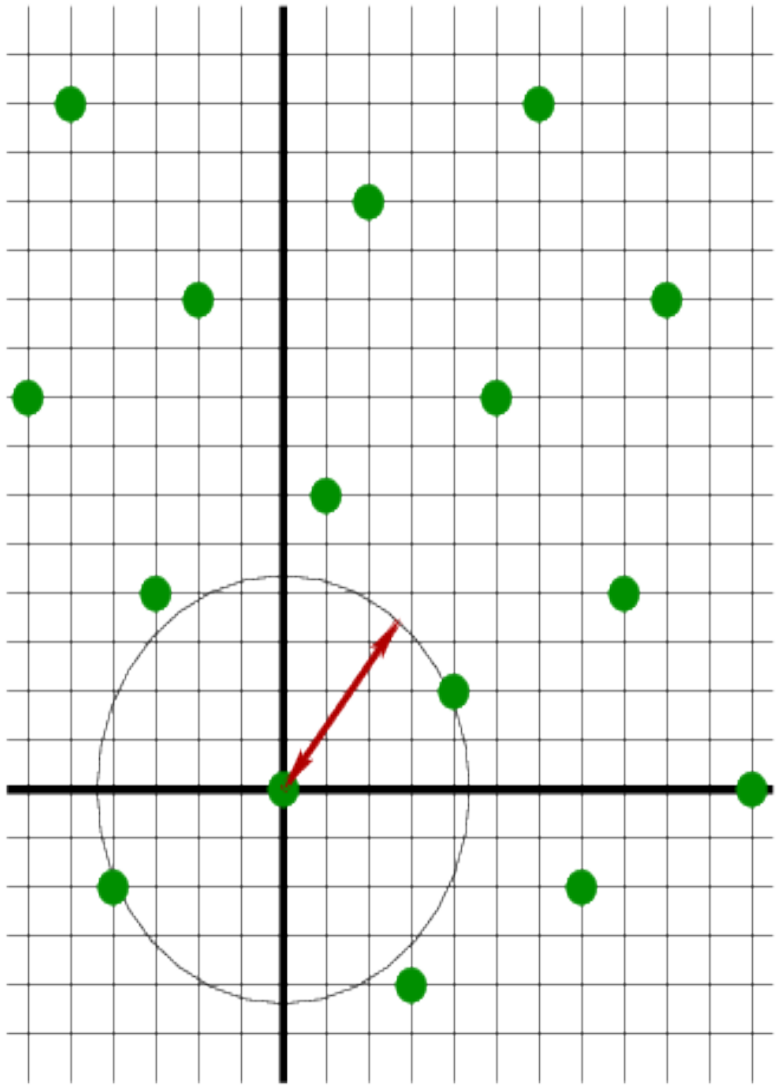
γ-SVP (computational variant)

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

No known sub-exp. algo. for $\gamma = \mathcal{P}oly(n)$.

γ-Ideal-SVP:

- γ-SVP restricted to Φ-ideal lattices.
- Does not seem easier than SVP.



(Integral) lattices and the Shortest Vector Problem

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

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

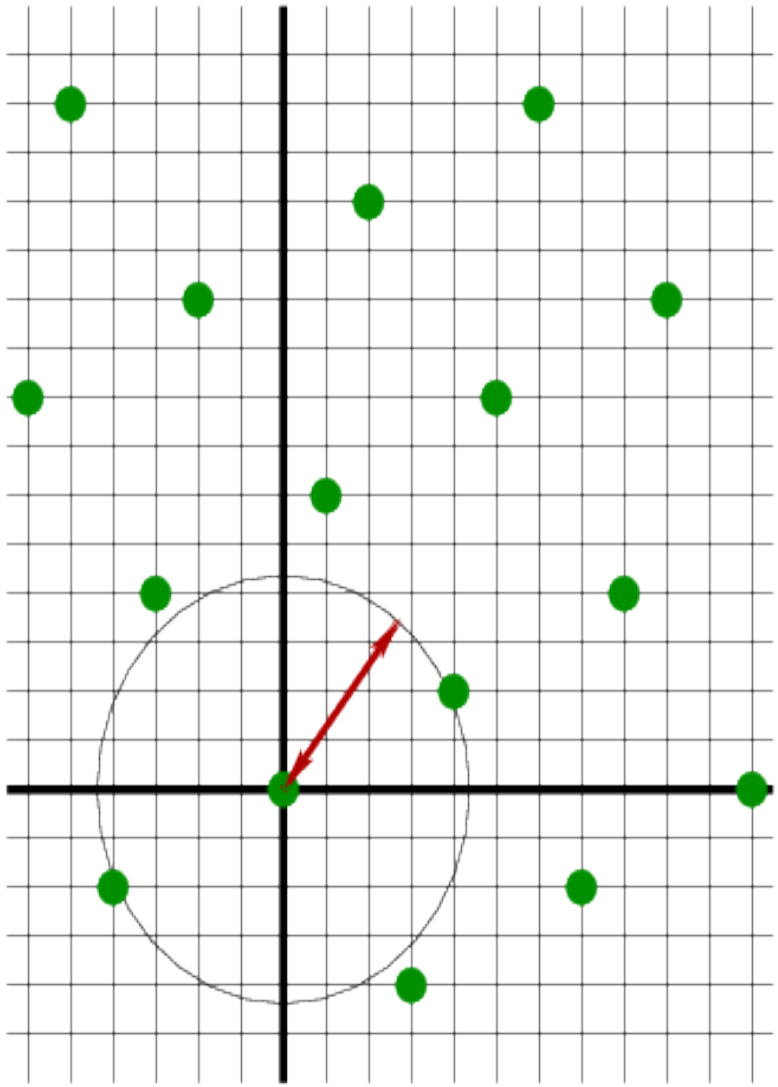
γ-SVP (computational variant)

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

No known sub-exp. algo. for $\gamma = \mathcal{P}oly(n)$.

γ-Ideal-SVP:

- γ-SVP restricted to Φ-ideal lattices.
- Does not seem easier than SVP.



The R-LWE problem [LyPeRe'10]

A couple of distributions:

- The noise distribution. For $\alpha > 0$, we define ν_α as the n-dimensional normal law of standard deviation α, rounded to $\mathbb{Z}^n$ and interpreted as an element of R^+.
- The R-LWE distribution. We define D_α as the distribution obtained as follows:

$$\text{Sample } a \hookleftarrow U(R_q^+),\ s \hookleftarrow \nu_\alpha,\ e \hookleftarrow \nu_\alpha,$$
$$\text{Return } (a, as + e) \in R_q^+ \times R_q^+.$$

R-LWE$_{q,\alpha}$ (Decisional variant with one sample)

Tell whether a given (a, b) is sampled from D_α or $U(R_q^+ \times R_q^+)$.

The R-LWE problem [LyPeRe'10]

A couple of distributions:

- The noise distribution. For $\alpha > 0$, we define ν_α as the n-dimensional normal law of standard deviation α, rounded to $\mathbb{Z}^n$ and interpreted as an element of R^+.
- The R-LWE distribution. We define D_α as the distribution obtained as follows:

$$\text{Sample } a \hookleftarrow U(R_q^+),\ s \hookleftarrow \nu_\alpha,\ e \hookleftarrow \nu_\alpha,$$
$$\text{Return } (a, as + e) \in R_q^+ \times R_q^+.$$

R-LWE$_{q,\alpha}$ (Decisional variant with one sample)

Tell whether a given (a, b) is sampled from D_α or $U(R_q^+ \times R_q^+)$.

R-LWE is hard [LyPeRe'10]

R-LWE$_{q,\alpha}$

Tell whether a given (a, b) is sampled from D_α or $U(R_q^+ \times R_q^+)$.

R-LWE is no easier than $\mathcal{P}oly(n)$-Ideal-SVP

Take $q = \mathcal{P}oly(n)$ with $q = 1 \bmod 2n$, and $\alpha = q/\mathcal{P}oly(n)$.
Solving R-LWE$_{q,\alpha}$ with non-negligible advantage is computationally infeasible, assuming the quantum hardness of $\mathcal{P}oly(n)$-Ideal-SVP.

- Sampling from ν_α costs $\widetilde{O}(n)$.
- Samples from ν_α are small with very high probability: their Euclidean norms are $\leq \sqrt{n} \cdot \alpha$ with probability $\geq 1 - 2^{-n}$.

R-LWE is hard [LyPeRe'10]

R-LWE$_{q,\alpha}$

Tell whether a given (a, b) is sampled from D_α or $U(R_q^+ \times R_q^+)$.

R-LWE is no easier than $\mathcal{P}oly(n)$-Ideal-SVP

Take $q = \mathcal{P}oly(n)$ with $q = 1 \bmod 2n$, and $\alpha = q/\mathcal{P}oly(n)$.
Solving R-LWE$_{q,\alpha}$ with non-negligible advantage is computationally infeasible, assuming the quantum hardness of $\mathcal{P}oly(n)$-Ideal-SVP.

- Sampling from ν_α costs $\widetilde{O}(n)$.
- Samples from ν_α are small with very high probability: their Euclidean norms are $\leq \sqrt{n} \cdot \alpha$ with probability $\geq 1 - 2^{-n}$.

Outline of the talk

1- Regular NTRUEncrypt.

2- The Ideal-SVP and R-LWE problems.

3- **The modified NTRUEncrypt.**

4- Modifying NTRUSign.

Some intuition

NTRUEncrypt:

- pk: $h = g/f \in R_q^-$ with f, g small.
- Enc: $M \mapsto 3hs + M \bmod q$, where s is small.
- IND-CPA: we would like $(h, 3hs)$ to be pseudo-random.
- It's not! Divide RHS by h and check for smallness.

R-LWE hardness:

- $(a, as + e) \approx^c U(R_q^+ \times R_q^+)$, where $a \hookleftarrow U(R_q^+), s, e \hookleftarrow \nu_\alpha$.
- Let's change rings and replace "(h, hs)" by "$(a, as + e)$"!

Some intuition

NTRUEncrypt:

- pk: $h = g/f \in R_q^-$ with f, g small.
- Enc: $M \mapsto 3hs + M \bmod q$, where s is small.
- IND-CPA: we would like $(h, 3hs)$ to be pseudo-random.
- It's not! Divide RHS by h and check for smallness.

R-LWE hardness:

- $(a, as + e) \approx^c U(R_q^+ \times R_q^+)$, where $a \hookleftarrow U(R_q^+), s, e \hookleftarrow \nu_\alpha$.
- Let's change rings and replace "(h, hs)" by "$(a, as + e)$"!

Some intuition

NTRUEncrypt:

- pk: $h = g/f \in R_q^-$ with f, g small.
- Enc: $M \mapsto 3hs + M \bmod q$, where s is small.
- IND-CPA: we would like $(h, 3hs)$ to be pseudo-random.
- It's not! Divide RHS by h and check for smallness.

R-LWE hardness:

- $(a, as + e) \approx^c U(R_q^+ \times R_q^+)$, where $a \hookleftarrow U(R_q^+), s, e \hookleftarrow \nu_\alpha$.
- Let's change rings and replace "(h, hs)" by "$(a, as + e)$"!

Some intuition

NTRUEncrypt:

- pk: $h = g/f \in R_q^-$ with f, g small.
- Enc: $M \mapsto 3hs + M \bmod q$, where s is small.
- IND-CPA: we would like $(h, 3hs)$ to be pseudo-random.
- It's not! Divide RHS by h and check for smallness.

R-LWE hardness:

- $(a, as + e) \approx^c U(R_q^+ \times R_q^+)$, where $a \hookleftarrow U(R_q^+), s, e \hookleftarrow \nu_\alpha$.
- Let's change rings and replace "(h, hs)" by "$(a, as + e)$"!

Is it that simple?

- Enc: $M \mapsto 3hs + M \bmod q$, where s is small.
- R-LWE: $(a, as + e)$, where $a \hookleftarrow U(R_q^+), s, e \hookleftarrow \nu_\alpha$.
- Changing rings and replacing "(h, hs)" by "$(a, as + e)$"?

Good news:

- s, e are small $\Rightarrow$ decryption still works.
- q prime $\Rightarrow$ multiplying by $p = 3$ preserves pseudo-randomness.
- Everything remains (asymptotically) efficient.

The catch:

- Relying on R-LWE requires h uniform in R_q^+.

Is it that simple?

- Enc: $M \mapsto 3hs + M \bmod q$, where s is small.
- R-LWE: $(a, as + e)$, where $a \hookleftarrow U(R_q^+), s, e \hookleftarrow \nu_\alpha$.
- Changing rings and replacing "(h, hs)" by "$(a, as + e)$"?

Good news:

- s, e are small $\Rightarrow$ decryption still works.
- q prime $\Rightarrow$ multiplying by $p = 3$ preserves pseudo-randomness.
- Everything remains (asymptotically) efficient.

The catch:

- Relying on R-LWE requires h uniform in R_q^+.

Is it that simple?

- Enc: $M \mapsto 3hs + M \bmod q$, where s is small.
- R-LWE: $(a, as + e)$, where $a \hookleftarrow U(R_q^+), s, e \hookleftarrow \nu_\alpha$.
- Changing rings and replacing "(h, hs)" by "$(a, as + e)$"?

Good news:

- s, e are small $\Rightarrow$ decryption still works.
- q prime $\Rightarrow$ multiplying by $p = 3$ preserves pseudo-randomness.
- Everything remains (asymptotically) efficient.

The catch:

- Relying on R-LWE requires h uniform in R_q^+.

The modified scheme

Parameters: n prime, $q \approx n$ a power of 2.

Key generation:

- sk: $f, g \in R^-$ with:
  - f invertible mod q and 3.
  - Coeffs of f and g in $\{-1, 0, 1\}$.
- pk: $h = g/f \bmod q$.

Encryption of $M \in \{0,1\}[x]/(x^n - 1)$:

- $C := 3hs + M \bmod q$, with coeffs of s in $\{-1, 0, 1\}$.

Decryption of $C \in R_q^-$:

- $f \times C \bmod q \;=\; 3gs + fM$.
- $(f \times C \bmod q) \bmod 3 \;=\; fM \bmod 3$.
- Multiply by the inverse of f mod 3.

The modified scheme

Parameters: n a power of 2, $q = \mathcal{P}oly(n)$ prime s.t. $q = 1\ [2n]$.

Key generation:

- sk: $f, g \in R^+$ with:
  - f invertible mod q and 2.
  - Coeffs of f and g of magnitude $\approx \sqrt{q}$.
- pk: $h = g/f \bmod q$.

Encryption of $M \in \{0,1\}[x]/(x^n+1)$:

- $C := 2(hs + e) + M \bmod q$, with $s, e \hookleftarrow \nu_\alpha$.

Decryption of $C \in R_q^+$:

- $f \times C \bmod q \ = \ 2(gs + fe) + fM$.
- $(f \times C \bmod q) \bmod 2 \ = \ fM \bmod 2$.
- Multiply by the inverse of f mod 2.

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

- We want h uniform while having f and g with small coeffs.
- If we want a chance, we need the magnitudes to be $\geq \sqrt{q}$.

The distribution $D_\sigma^\times$ used for 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.

- It's a discrete Gaussian with a non-lattice support.
- We also want f invertible mod 2: handled by tweaking $D_\sigma^\times$.

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

- We want h uniform while having f and g with small coeffs.
- If we want a chance, we need the magnitudes to be $\geq \sqrt{q}$.

The distribution $D_\sigma^\times$ used for 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.

- It's a discrete Gaussian with a non-lattice support.
- We also want f invertible mod 2: handled by tweaking $D_\sigma^\times$.

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

- We want h uniform while having f and g with small coeffs.
- If we want a chance, we need the magnitudes to be $\geq \sqrt{q}$.

The distribution $D_\sigma^\times$ used for 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.

- It's a discrete Gaussian with a non-lattice support.
- We also want f invertible mod 2: handled by tweaking $D_\sigma^\times$.

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

Our main technical contribution

If $\sigma = \widetilde{\Omega}(n \cdot q^{\frac{1}{2}+\varepsilon})$ 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.

- If $f \hookleftarrow D_\sigma^\times$, then $\|f\| \leq \sigma\sqrt{n}$, with overwhelming probability.
- We don't get uniformity in R_q but only in $R_q^\times$.
- R-LWE is still hard if h is restricted to $U(R_q^\times)$.

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

Our main technical contribution

If $\sigma = \widetilde{\Omega}(n \cdot q^{\frac{1}{2}+\varepsilon})$ 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.

- If $f \hookleftarrow D_\sigma^\times$, then $\|f\| \leq \sigma\sqrt{n}$, with overwhelming probability.
- We don't get uniformity in R_q but only in $R_q^\times$.
- R-LWE is still hard if h is restricted to $U(R_q^\times)$.

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

Our main technical contribution

If $\sigma = \widetilde{\Omega}(n \cdot q^{\frac{1}{2}+\varepsilon})$ 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.

- If $f \hookleftarrow D_\sigma^\times$, then $\|f\| \leq \sigma\sqrt{n}$, with overwhelming probability.
- We don't get uniformity in R_q but only in $R_q^\times$.
- R-LWE is still hard if h is restricted to $U(R_q^\times)$.

Proving uniformity in two steps

Step 1: We show that if $a_i \hookleftarrow U(R_q^\times)$ and $t_i \hookleftarrow D_\sigma$:

$$\Delta\Big[(a_1, a_2, t_1 a_1 + t_2 a_2); U(R_q^\times \times R_q^\times \times R_q)\Big] \;\leq\; q^{-\Omega(\varepsilon n)}.$$

Step 2: We observe that for $a = -a_2/a_1$:

$$\Pr_{t_1,t_2}[t_1/t_2 = a\ [q]] = \Pr_{t_1,t_2}[a_1 t_1 + a_2 t_2 = 0\ [q]],$$

where the t_i's are from $D_\sigma^\times$.

- Step 2 would be easy if the t_i's were sampled from a lattice.
- But $\{\mathbf{x} \in \mathbb{Z}^n, x \in R_q^\times\}$ is not a lattice.
- Handled by inclusion-exclusion, involving the ideals of R_q.

Proving uniformity in two steps

Step 1: We show that if $a_i \hookleftarrow U(R_q^\times)$ and $t_i \hookleftarrow D_\sigma$:

$$\Delta\Big[(a_1, a_2, t_1 a_1 + t_2 a_2); U(R_q^\times \times R_q^\times \times R_q)\Big] \leq q^{-\Omega(\varepsilon n)}.$$

Step 2: We observe that for $a = -a_2/a_1$:

$$\Pr_{t_1,t_2}[t_1/t_2 = a\ [q]] = \Pr_{t_1,t_2}[a_1 t_1 + a_2 t_2 = 0\ [q]],$$

where the t_i's are from $D_\sigma^\times$.

- Step 2 would be easy if the t_i's were sampled from a lattice.
- But $\{\mathbf{x} \in \mathbb{Z}^n, x \in R_q^\times\}$ is not a lattice.
- Handled by inclusion-exclusion, involving the ideals of R_q.

Proving uniformity in two steps

Step 1: We show that if $a_i \hookleftarrow U(R_q^\times)$ and $t_i \hookleftarrow D_\sigma$:

$$\Delta\Big[(a_1, a_2, t_1 a_1 + t_2 a_2); U(R_q^\times \times R_q^\times \times R_q)\Big] \leq q^{-\Omega(\varepsilon n)}.$$

Step 2: We observe that for $a = -a_2/a_1$:

$$\Pr_{t_1,t_2}[t_1/t_2 = a\ [q]] = \Pr_{t_1,t_2}[a_1 t_1 + a_2 t_2 = 0\ [q]],$$

where the t_i's are from $D_\sigma^\times$.

- Step 2 would be easy if the t_i's were sampled from a lattice.
- But $\{\mathbf{x} \in \mathbb{Z}^n, x \in R_q^\times\}$ is not a lattice.
- Handled by inclusion-exclusion, involving the ideals of R_q.

Proving uniformity in two steps

Step 1: We show that if $a_i \hookleftarrow U(R_q^\times)$ and $t_i \hookleftarrow D_\sigma$:

$$\Delta\Big[(a_1, a_2, t_1 a_1 + t_2 a_2); U(R_q^\times \times R_q^\times \times R_q)\Big] \leq q^{-\Omega(\varepsilon n)}.$$

Step 2: We observe that for $a = -a_2/a_1$:

$$\Pr_{t_1,t_2}[t_1/t_2 = a\ [q]] = \Pr_{t_1,t_2}[a_1 t_1 + a_2 t_2 = 0\ [q]],$$

where the t_i's are from $D_\sigma^\times$.

- Step 2 would be easy if the t_i's were sampled from a lattice.
- But $\{\mathbf{x} \in \mathbb{Z}^n, x \in R_q^\times\}$ is not a lattice.
- Handled by inclusion-exclusion, involving the ideals of R_q.

Outline of the talk

1- Regular NTRUEncrypt.

2- The Ideal-SVP and R-LWE problems.

3- The modified NTRUEncrypt.

4- **Modifying** NTRUSign**.**

The NTRU lattice

Recall that $R = \mathbb{Z}[x]/(x^n + 1)$ and $R_q = \mathbb{Z}_q[x]/(x^n + 1)$.

Given $h = g/f \in R_q$, we consider the lattice spanned by:

The NTRU lattice

Recall that $R = \mathbb{Z}[x]/(x^n + 1)$ and $R_q = \mathbb{Z}_q[x]/(x^n + 1)$.

Given $h = g/f \in R_q$, we consider the lattice spanned by:

$$\left[\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{h} & \mathbf{q} \end{array}\right] = \left[\begin{array}{cccc|cccc} 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ \hline h_0 & -h_{n-1} & \dots & -h_1 & q & 0 & \dots & 0 \\ h_1 & h_0 & \dots & -h_2 & 0 & q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{n-1} & h_{n-2} & \dots & h_0 & 0 & 0 & \dots & q \end{array}\right]$$

The NTRU R-module/$\mathbb{Z}$-lattice

Matrix $+/\times$ are consistent with $+/\times$ in ring R:

$$\left[\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{h} & \mathbf{q} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{f} \\ \hline - \end{array}\right] = \left[\begin{array}{c} f \\ f \cdot h \bmod q \end{array}\right].$$

Secret key $(f, g)^t$ is a short lattice vector of:

$$\begin{aligned} L &= R \cdot \left[\frac{\mathbf{1}}{\mathbf{h}}\right] + R \cdot \left[\frac{\mathbf{0}}{\mathbf{q}}\right] \\ &\subseteq \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in R^2 : hx_1 + x_2 = 0 \ [q] \right\}. \end{aligned}$$

The NTRU R-module/$\mathbb{Z}$-lattice

Matrix $+/\times$ are consistent with $+/\times$ in ring R:

$$\left[\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{h} & \mathbf{q} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{f} \\ \hline - \end{array}\right] = \left[\begin{array}{c} f \\ f \cdot h \bmod q \end{array}\right].$$

Secret key $(f, g)^t$ is a short lattice vector of:

$$\begin{aligned} L &= R \cdot \left[\frac{\mathbf{1}}{\mathbf{h}}\right] + R \cdot \left[\frac{\mathbf{0}}{\mathbf{q}}\right] \\ &\subseteq \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in R^2 : hx_1 + x_2 = 0 \ [q] \right\}. \end{aligned}$$

NTRUSign

Assume we have a small module basis of L:

$$\left[\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{h} & \mathbf{q} \end{array}\right] \cdot U = \left[\begin{array}{c|c} \mathbf{f} & \mathbf{F} \\ \hline \mathbf{g} & \mathbf{G} \end{array}\right], \quad \text{for some } U \in GL_2(R).$$

NTRUSign follows the hash-and-sign paradigm:

- Public key: h; secret key: small module basis.
- To sign M, use sk to get $s_1, s_2 \in R$ small with

$$hs_1 + s_2 = \mathcal{H}(M) \ [q].$$

- To verify (M, s_1, s_2): check

$$hs_1 + s_2 = \mathcal{H}(M) \ [q] \quad \text{and} \quad \|s_1\|, \|s_2\| \text{ small}.$$

Security based on a variant of R-LWE, in the random oracle model.

NTRUSign

Assume we have a small module basis of L:

$$\left[\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{h} & \mathbf{q} \end{array}\right] \cdot U = \left[\begin{array}{c|c} \mathbf{f} & \mathbf{F} \\ \hline \mathbf{g} & \mathbf{G} \end{array}\right], \quad \text{for some } U \in GL_2(R).$$

NTRUSign follows the hash-and-sign paradigm:

- Public key: h; secret key: small module basis.
- To sign M, use sk to get $s_1, s_2 \in R$ small with

$$hs_1 + s_2 = \mathcal{H}(M) \ [q].$$

- To verify (M, s_1, s_2): check

$$hs_1 + s_2 = \mathcal{H}(M) \ [q] \quad \text{and} \quad \|s_1\|, \|s_2\| \text{ small.}$$

Security based on a variant of R-LWE, in the random oracle model.

NTRUSign

Assume we have a small module basis of L:

$$\left[\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{h} & \mathbf{q} \end{array}\right] \cdot U = \left[\begin{array}{c|c} \mathbf{f} & \mathbf{F} \\ \hline \mathbf{g} & \mathbf{G} \end{array}\right], \quad \text{for some } U \in GL_2(R).$$

NTRUSign follows the hash-and-sign paradigm:

- Public key: h; secret key: small module basis.
- To sign M, use sk to get $s_1, s_2 \in R$ small with

$$hs_1 + s_2 = \mathcal{H}(M) \ [q].$$

- To verify (M, s_1, s_2): check

$$hs_1 + s_2 = \mathcal{H}(M) \ [q] \quad \text{and} \quad \|s_1\|, \|s_2\| \text{ small.}$$

Security based on a variant of R-LWE, in the random oracle model.

Two problems with NTRUSign

- Public key: h; secret key: small module basis.
- To sign M, use sk to get $s_1, s_2 \in R$ small with $hs_1 + s_2 = \mathcal{H}(M)$ $[q]$.

Problem 1: The "perturbations".

- Choose t_1, t_2 with $ht_1 + t_2 = \mathcal{H}(M)$ $[q]$, and then perturb (t_1, t_2) by a random lattice point.
- No perturbation $\Rightarrow$ secret key is revealed [NgRe'08].

$\Rightarrow$ Fixed by using a discrete Gaussian [GePeVa'08,Peikert'10].

Problem 2: The key-pair.

- NTRU's extension of short vector to short basis is heuristic ...

Two problems with NTRUSign

- Public key: h; secret key: small module basis.
- To sign M, use sk to get $s_1, s_2 \in R$ small with $hs_1 + s_2 = \mathcal{H}(M)\ [q]$.

Problem 1: The "perturbations".

- Choose t_1, t_2 with $ht_1 + t_2 = \mathcal{H}(M)\ [q]$, and then perturb (t_1, t_2) by a random lattice point.
- No perturbation $\Rightarrow$ secret key is revealed [NgRe'08].

$\Rightarrow$ Fixed by using a discrete Gaussian [GePeVa'08,Peikert'10].

Problem 2: The key-pair.

- NTRU's extension of short vector to short basis is heuristic ...

Remaining difficulty: getting a small module basis

We have h, f, g such that:

- h is (essentially) uniform in $R_q^\times$.
- $(f, g)^t$ is a small non-zero vector in the module spanned by:

$$\left[\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{h} & \mathbf{q} \end{array}\right]$$

We want to find F, G s.t.:

$$\left[\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{h} & \mathbf{q} \end{array}\right] \cdot U = \left[\begin{array}{c|c} \mathbf{f} & \mathbf{F} \\ \hline \mathbf{g} & \mathbf{G} \end{array}\right], \quad \text{for some } U \in GL_2(R).$$

Remaining difficulty: getting a small module basis

We have h, f, g such that:

- h is (essentially) uniform in $R_q^\times$.
- $(f, g)^t$ is a small non-zero vector in the module spanned by:

$$\left[\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{h} & \mathbf{q} \end{array}\right]$$

We want to find F, G s.t.:

$$\left[\begin{array}{c|c} \mathbf{1} & \mathbf{0} \\ \hline \mathbf{h} & \mathbf{q} \end{array}\right] \cdot U = \left[\begin{array}{c|c} \mathbf{f} & \mathbf{F} \\ \hline \mathbf{g} & \mathbf{G} \end{array}\right], \quad \text{for some } U \in GL_2(R).$$

The NTRUSign secret key extension procedure

1. It is likely that f and g are coprime in R.
$$\exists u_1, u_2 \in R,\ u_1 f + u_2 g = 1.$$
2. Take $F_q = q u_2$ and $G_q = -q u_1$.
3. $\begin{bmatrix} \mathbf{f} & \mathbf{F_q} \\ \mathbf{g} & \mathbf{G_q} \end{bmatrix}$ is a basis of the NTRU module.
4. Make it small by reducing the second vector wrt the first one:
$$\mathbf{b_2} := \mathbf{b_2} - \left\lfloor \frac{\langle \mathbf{b_1}, \mathbf{b_2} \rangle}{\langle \mathbf{b_1}, \mathbf{b_1} \rangle} \right\rceil \mathbf{b_1},$$
with $\langle \cdot, \cdot \rangle$ and $\lfloor \cdot \rceil$ taken over R.

This procedure is efficient, but heuristic because of Step 1.

The NTRUSign secret key extension procedure

1. It is likely that f and g are coprime in R.
$$\exists u_1, u_2 \in R,\ u_1 f + u_2 g = 1.$$
2. Take $F_q = q u_2$ and $G_q = -q u_1$.
3. $\left[\begin{array}{c|c} \mathbf{f} & \mathbf{F_q} \\ \hline \mathbf{g} & \mathbf{G_q} \end{array}\right]$ is a basis of the NTRU module.
4. Make it small by reducing the second vector wrt the first one:
$$\mathbf{b}_2 := \mathbf{b}_2 - \left\lfloor \frac{\langle \mathbf{b}_1, \mathbf{b}_2 \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} \right\rceil \mathbf{b}_1,$$
with $\langle \cdot, \cdot \rangle$ and $\lfloor \cdot \rceil$ taken over R.

This procedure is efficient, but heuristic because of Step 1.

The NTRUSign secret key extension procedure

1. It is likely that f and g are coprime in R.
$$\exists u_1, u_2 \in R,\ u_1 f + u_2 g = 1.$$
2. Take $F_q = q u_2$ and $G_q = -q u_1$.
3. $\left[\begin{array}{c|c} \mathbf{f} & \mathbf{F_q} \\ \hline \mathbf{g} & \mathbf{G_q} \end{array}\right]$ is a basis of the NTRU module.
4. Make it small by reducing the second vector wrt the first one:
$$\mathbf{b_2} := \mathbf{b_2} - \left\lfloor \frac{\langle \mathbf{b_1}, \mathbf{b_2} \rangle}{\langle \mathbf{b_1}, \mathbf{b_1} \rangle} \right\rceil \mathbf{b_1},$$
with $\langle \cdot, \cdot \rangle$ and $\lfloor \cdot \rceil$ taken over R.

This procedure is efficient, but heuristic because of Step 1.

The NTRUSign secret key extension procedure

1. It is likely that f and g are coprime in R.
$$\exists u_1, u_2 \in R,\ u_1 f + u_2 g = 1.$$
2. Take $F_q = q u_2$ and $G_q = -q u_1$.
3. $\left[\begin{array}{c|c} \mathbf{f} & \mathbf{F_q} \\ \hline \mathbf{g} & \mathbf{G_q} \end{array}\right]$ is a basis of the NTRU module.
4. Make it small by reducing the second vector wrt the first one:
$$\mathbf{b_2} := \mathbf{b_2} - \left\lfloor \frac{\langle \mathbf{b_1}, \mathbf{b_2} \rangle}{\langle \mathbf{b_1}, \mathbf{b_1} \rangle} \right\rceil \mathbf{b_1},$$
with $\langle \cdot, \cdot \rangle$ and $\lfloor \cdot \rceil$ taken over R.

This procedure is efficient, but heuristic because of Step 1.

Making the NTRUSign secret key extension rigorous

Is it likely that f and g are coprime in R?

- In our case, $f, g \hookleftarrow D_\sigma^\times$.
- Two "random" integers are coprime with prob. $1/\zeta(2)$, where

$$\zeta(2) = \sum_{k \in \mathbb{Z}} \frac{1}{p^2} = \frac{\pi^2}{6}.$$

- We adapt this to R and D_σ: if σ is large enough, the probability that $f, g \hookleftarrow D_\sigma$ are coprime is:

$$\geq \frac{1}{2\zeta_R(2)} \text{ where } \zeta_R(2) := \sum_{I \text{ ideal of } R} \frac{1}{(\det I)^2} = O(1).$$

- Moving from D_σ to $D_\sigma^\times$ is easy.

Making the NTRUSign secret key extension rigorous

Is it likely that f and g are coprime in R?

- In our case, $f, g \hookleftarrow D_\sigma^\times$.
- Two "random" integers are coprime with prob. $1/\zeta(2)$, where

$$\zeta(2) = \sum_{k \in \mathbb{Z}} \frac{1}{p^2} = \frac{\pi^2}{6}.$$

- We adapt this to R and D_σ: if σ is large enough, the probability that $f, g \hookleftarrow D_\sigma$ are coprime is:

$$\geq \frac{1}{2\zeta_R(2)} \quad \text{where} \quad \zeta_R(2) := \sum_{I \text{ ideal of } R} \frac{1}{(\det I)^2} = O(1).$$

- Moving from D_σ to $D_\sigma^\times$ is easy.

Making the NTRUSign secret key extension rigorous

Is it likely that f and g are coprime in R?

- In our case, $f, g \hookleftarrow D_\sigma^\times$.
- Two "random" integers are coprime with prob. $1/\zeta(2)$, where

$$\zeta(2) = \sum_{k \in \mathbb{Z}} \frac{1}{p^2} = \frac{\pi^2}{6}.$$

- We adapt this to R and D_σ: if σ is large enough, the probability that $f, g \hookleftarrow D_\sigma$ are coprime is:

$$\geq \frac{1}{2\zeta_R(2)} \quad \text{where} \quad \zeta_R(2) := \sum_{I \text{ ideal of } R} \frac{1}{(\det I)^2} = O(1).$$

- Moving from D_σ to $D_\sigma^\times$ is easy.

Making the NTRUSign secret key extension rigorous

Is it likely that f and g are coprime in R?

- In our case, $f, g \hookleftarrow D_\sigma^\times$.
- Two "random" integers are coprime with prob. $1/\zeta(2)$, where

$$\zeta(2) = \sum_{k \in \mathbb{Z}} \frac{1}{p^2} = \frac{\pi^2}{6}.$$

- We adapt this to R and D_σ: if σ is large enough, the probability that $f, g \hookleftarrow D_\sigma$ are coprime is:

$$\geq \frac{1}{2\zeta_R(2)} \quad \text{where} \quad \zeta_R(2) := \sum_{I \text{ ideal of } R} \frac{1}{(\det I)^2} = O(1).$$

- Moving from D_σ to $D_\sigma^\times$ is easy.

Making the NTRUSign secret key extension rigorous

Is it likely that f and g are coprime in R?

- In our case, $f, g \hookleftarrow D_\sigma^\times$.
- Two "random" integers are coprime with prob. $1/\zeta(2)$, where

$$\zeta(2) = \sum_{k \in \mathbb{Z}} \frac{1}{p^2} = \frac{\pi^2}{6}.$$

- We adapt this to R and D_σ: if σ is large enough, the probability that $f, g \hookleftarrow D_\sigma$ are coprime is:

$$\geq \frac{1}{2\zeta_R(2)} \quad \text{where} \quad \zeta_R(2) := \sum_{I \text{ ideal of } R} \frac{1}{(\det I)^2} = O(1).$$

- Moving from D_σ to $D_\sigma^\times$ is easy.

The modified NTRUSign key generation procedure

1. Sample f, g from $D_\sigma^\times$.
2. If f, g are not co-prime in R, restart.
3. Public key: $h = g/f \ [q]$.
4. Get secret key using NTRU's key extension procedure.

Outline of the talk

1- Regular NTRUEncrypt.

2- The Ideal-SVP and R-LWE problems.

3- The modified NTRUEncrypt.

4- Modifying NTRUSign.

What's the interest of this result?

What we prove:

- There are variants of NTRUEncrypt and NTRUSign that are secure under the assumption that $\mathcal{P}oly(n)$-Ideal-SVP is hard.
- They're asymptotically as efficient as the original schemes.

What's the interest of this result?

What we prove:

- There are variants of NTRUEncrypt and NTRUSign that are secure under the assumption that $\mathcal{P}oly(n)$-Ideal-SVP is hard.
- They're asymptotically as efficient as the original schemes.

It does not mean we should blindly move to the provable variants: They are most likely less practical.

What's the interest of this result?

What we prove:

- There are variants of NTRUEncrypt and NTRUSign that are secure under the assumption that $\mathcal{P}oly(n)$-Ideal-SVP is hard.
- They're asymptotically as efficient as the original schemes.

It does not mean we should blindly move to the provable variants: They are most likely less practical.

What it means:

- The general design of NTRUEncrypt is sound.
- It hints that we could
  - replace hs by $hs + e$, to thwart trivial semantic attacks.
  - take less small coeffs for f, g, s, e, to improve security.

Work in progress and open problems

- ✓ A provably IND-CCA variant of NTRUEncrypt.
- What about practice?
  - Which optimisations do not lower security?
  - What are the limits of the best known practical attacks? How do we extrapolate these limits to reach given security levels?
- Is $\mathcal{P}oly(n)$-Ideal-SVP really so hard?

Work in progress and open problems

- ✓ A provably IND-CCA variant of NTRUEncrypt.

- What about practice?
  - Which optimisations do not lower security?
  - What are the limits of the best known practical attacks? How do we extrapolate these limits to reach given security levels?
- Is $\mathcal{P}oly(n)$-Ideal-SVP really so hard?