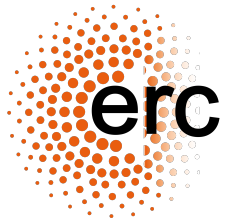
The NTRU encryption scheme, and ideal lattices

Damien Stehlé

Based on joint work with Ron Steinfeld

Šibenik, June 2015

perso.ens-lyon.fr/damien.stehle/NTRU.html



The NTRU encryption scheme

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 20 years:

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

The NTRU encryption scheme

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 20 years:

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

Why studying NTRUEncrypt?

NTRUEncrypt is a practical scheme that seems secure.

- Standardized: IEEE P1363.
- Commercialized: Security Innovation.
- Super-fast:
  - Encryption: a bit 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 studying NTRUEncrypt?

NTRUEncrypt is a practical scheme that seems secure.

- Standardized: IEEE P1363.
- Commercialized: Security Innovation.
- Super-fast:
  - Encryption: a bit 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 studying NTRUEncrypt?

NTRUEncrypt is a practical scheme that seems secure.

- Standardized: IEEE P1363.
- Commercialized: Security Innovation.
- Super-fast:
  - Encryption: a bit 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

And NTRUSign?

NTRUSign is a digital signature counterpart of NTRUEncrypt.

- 2001: First proposal, called NSS [HoPiSi01].
- 2001: Cryptanalysis, by Gentry, Jonsson, Stern and Szydlo.
- 2001: First repair.
- 2002: Re-broken, by Gentry and Szydlo.
- Since then: many breaks and repairs.

- Standardized and commercialized.
  - Super-fast, before the Nguyen-Regev attack (2006).
  - It may be thwarted, but with big performance impact.

Fixed NTRUSign not competitive, e.g. compared to BLISS.

And NTRUSign?

NTRUSign is a digital signature counterpart of NTRUEncrypt.

- 2001: First proposal, called NSS [HoPiSi01].
- 2001: Cryptanalysis, by Gentry, Jonsson, Stern and Szydlo.
- 2001: First repair.
- 2002: Re-broken, by Gentry and Szydlo.
- Since then: many breaks and repairs.
- Standardized and commercialized.
- Superseded before the NTRU Patent Expired (2012).
- It may be thwarted, but with big performance losses.
- And NTRUSign is not compatible with NTRUEncrypt.

And NTRUSign?

NTRUSign is a digital signature counterpart of NTRUEncrypt.

- 2001: First proposal, called NSS [HoPiSi01].
- 2001: Cryptanalysis, by Gentry, Jonsson, Stern and Szydlo.
- 2001: First repair.
- 2002: Re-broken, by Gentry and Szydlo.
- Since then: many breaks and repairs.
- Standardized and commercialized.
- Superseded before the NTRU Patent Expired (2012).
- It may be thwarted, but with big performance losses.
- Not NTRUSign-like signatures are considered today.

And NTRUSign?

NTRUSign is a digital signature counterpart of NTRUEncrypt.

- 2001: First proposal, called NSS [HoPiSi01].
- 2001: Cryptanalysis, by Gentry, Jonsson, Stern and Szydlo.
- 2001: First repair.
- 2002: Re-broken, by Gentry and Szydlo.
- Since then: many breaks and repairs.

- Standardized and commercialized.
- Super-fast, before the Nguyen-Regev attack (2009).
- It may be thwarted, but with big performance impact.

Fixed NTRUSign not competitive, e.g. compared to BLISS.

And NTRUSign?

NTRUSign is a digital signature counterpart of NTRUEncrypt.

- 2001: First proposal, called NSS [HoPiSi01].
- 2001: Cryptanalysis, by Gentry, Jonsson, Stern and Szydlo.
- 2001: First repair.
- 2002: Re-broken, by Gentry and Szydlo.
- Since then: many breaks and repairs.

- Standardized and commercialized.
- Super-fast, before the Nguyen-Regev attack (2009).
- It may be thwarted, but with big performance impact.

Fixed NTRUSign not competitive, e.g. compared to BLISS.

And NTRUSign?

NTRUSign is a digital signature counterpart of NTRUEncrypt.

- 2001: First proposal, called NSS [HoPiSi01].
- 2001: Cryptanalysis, by Gentry, Jonsson, Stern and Szydlo.
- 2001: First repair.
- 2002: Re-broken, by Gentry and Szydlo.
- Since then: many breaks and repairs.

- Standardized and commercialized.

- Super-fast, before the Nguyen-Regev attack (2009).
- It may be thwarted, but with big performance impact.

Fixed NTRUSign not competitive, e.g. compared to BLISS.

Outline of the talk

1- **Regular** NTRUEncrypt
2- Attacks on NTRUEncrypt
3- The Ideal-SVP and Ring-LWE problems
4- A provably secure NTRUEncrypt

Polynomial Rings: Generalizing $\mathbb{Z}$

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

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

Some 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: Generalizing $\mathbb{Z}$

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

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

Some 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: Generalizing $\mathbb{Z}$

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

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

Some 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: Generalizing $\mathbb{Z}/q\mathbb{Z}$

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

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

- 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: Generalizing $\mathbb{Z}/q\mathbb{Z}$

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

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

- 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: Generalizing $\mathbb{Z}/q\mathbb{Z}$

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

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

- 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

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

Description of NTRUEncrypt: Key Generation

Parameters: n, q a power of 2 (e.g. $(n, q) = (503, 256)$).

- Secret key sk: $f, g \in R^-$ with:
  - 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: Key Generation

Parameters: n, q a power of 2 (e.g. $(n, q) = (503, 256)$).

- Secret key sk: $f, g \in R^-$ with:
  - 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: Key Generation

Parameters: n, q a power of 2 (e.g. $(n, q) = (503, 256)$).

- Secret key sk: $f, g \in R^-$ with:
  - 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: Key Generation

Parameters: n, q a power of 2 (e.g. $(n, q) = (503, 256)$).

- Secret key sk: $f, g \in R^-$ with:
  - 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: Encryption and Decryption

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

Description of NTRUEncrypt: Encryption and Decryption

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

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

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

Description of NTRUEncrypt: Encryption and Decryption

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

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

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

Decryption of $C \in R_q^-$:

- $f \times C = 3gs + fM \bmod q$
- Smallness $\Rightarrow$ equality holds over R^-
- $(f \times C \bmod q) \bmod 3 = fM \bmod 3$
- Multiply by the inverse of f mod 3

Security intuition

The mask $3hs$ hides the plaintext M in the ciphertext C.

Description of NTRUEncrypt: Encryption and Decryption

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

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

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

Decryption of $C \in R_q^-$:

- $f \times C = 3gs + fM \bmod q$
- Smallness $\Rightarrow$ equality holds over R^-
- $(f \times C \bmod q) \bmod 3 = fM \bmod 3$
- Multiply by the inverse of f mod 3

Security intuition

The mask $3hs$ hides the plaintext M in the ciphertext C.

NTRU in practice

Quite a few implementation hacks:

- How many 0's in $f, g, s \longrightarrow d_f, d_g, d_s$.
- If $f = 1 + 3f'$ for a small f', then the final multiplication by the inverse of f mod 3 is for free.
- Replacing "3" by "$x + 2$" increases performance.
- More hacks for NAEP
  (variant meant to be IND-CCA in the Random Oracle Model).

Outline of the talk

1- Regular NTRUEncrypt
2- **Attacks on** NTRUEncrypt
3- The Ideal-SVP and Ring-LWE problems
4- A provably secure NTRUEncrypt

On the NTRUEncrypt key pair

The key-pair $(h, (f, g))$ of NTRUEncrypt satisfies:

$$\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot \begin{bmatrix} f \\ - \end{bmatrix} = \begin{bmatrix} f \\ f \cdot h \bmod q \end{bmatrix}.$$

Secret key $\begin{bmatrix} f \\ g \end{bmatrix}$ is a short vector in the image of $\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix}$.

We look at R-multiples of the columns, and try to find a short combination...

On the NTRUEncrypt key pair

The key-pair $(h, (f, g))$ of NTRUEncrypt satisfies:

$$\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot \begin{bmatrix} f \\ \_ \end{bmatrix} = \begin{bmatrix} f \\ f \cdot h \bmod q \end{bmatrix}.$$

Secret key $\begin{bmatrix} f \\ g \end{bmatrix}$ is a short vector in the image of $\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix}$.

We look at R-multiples of the columns, and try to find a short combination...

On the NTRUEncrypt key pair

The key-pair $(h, (f, g))$ of NTRUEncrypt satisfies:

$$\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot \begin{bmatrix} f \\ - \end{bmatrix} = \begin{bmatrix} f \\ f \cdot h \bmod q \end{bmatrix}.$$

Secret key $\begin{bmatrix} f \\ g \end{bmatrix}$ is a short vector in the image of $\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix}$.

We look at R-multiples of the columns, and try to find a short combination...

On the NTRUEncrypt key pair

Secret key $\begin{bmatrix} f \\ g \end{bmatrix}$ is a short vector in the image of $\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix}$.

Let's identify polynomials to vectors via their coefficients:

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

Let L be the image of $\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot R^2$.

- L is a lattice of dimension $2n$.
- $(f, g)^T$ is a short vector in L.
- If they are chosen binary and sparse, then L contains exceptionally short vectors.

$\Rightarrow$ **Lattice reduction gives a key recovery attack.**

On the NTRUEncrypt key pair

Secret key $\begin{bmatrix} f \\ g \end{bmatrix}$ is a short vector in the image of $\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix}$.

Let's identify polynomials to vectors via their coefficients:

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

Let L be the image of $\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot R^2$.

- L is a lattice of dimension $2n$.
- $(f, g)^T$ is a short vector in L.
- If they are chosen binary and sparse, then L contains exceptionally short vectors.

$\Rightarrow$ **Lattice reduction gives a key recovery attack.**

On the NTRUEncrypt ciphertext

We have: $C = 3h \cdot s + M \mod q$, for some small unknown s.
We can write:

$$\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot \begin{bmatrix} 3s \\ - \end{bmatrix} + \begin{bmatrix} -3s \\ M \end{bmatrix} = \begin{bmatrix} 0 \\ C \end{bmatrix}.$$

- The vector $(-3s, M)^T$ has small coefficients.
- By mapping to the integers, we get an instance of the Bounded Distance Decoding problem:
  given a vector close to L, recover the closest lattice point.
- Here $L = \begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot R^2$, and input vector is given by C.

$\Rightarrow$ **Lattice reduction gives a message recovery attack.**

On the NTRUEncrypt ciphertext

We have: $C = 3h \cdot s + M \mod q$, for some small unknown s.
We can write:

$$\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot \begin{bmatrix} 3s \\ - \end{bmatrix} + \begin{bmatrix} -3s \\ M \end{bmatrix} = \begin{bmatrix} 0 \\ C \end{bmatrix}.$$

- The vector $(-3s, M)^T$ has small coefficients.
- By mapping to the integers, we get an instance of the Bounded Distance Decoding problem:
  given a vector close to L, recover the closest lattice point.
- Here $L = \begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot R^2$, and input vector is given by C.

$\Rightarrow$ **Lattice reduction gives a message recovery attack.**

Chosen plaintext attack for $\Phi = x^n - 1$

Parameters may be set to thwart (improved versions of) these lattice reduction attacks.
But there are easy poly-time chosen plaintext attacks.

We have: $C = 3h \cdot s + M \mod (q, x^n - 1)$, for some small s.

- Note that $C(1) = 3h(1) \cdot s(1) + M(1) \mod q$ [1-dim NTRU!]
- Find $f_1, g_1 \in \mathbb{Z}$ of magnitude $\approx \sqrt{q}$ s.t. $h(1) = f_1/g_1 \mod q$.
- Use 1-dim decryption, to recover $M(1) \mod 3$.

The factorization of Φ is used for cryptanalysis. It should be a fundamental ring.

Chosen plaintext attack for $\Phi = x^n - 1$

Parameters may be set to thwart (improved versions of) these lattice reduction attacks.
But there are easy poly-time chosen plaintext attacks.

We have: $C = 3h \cdot s + M \mod (q, x^n - 1)$, for some small s.

- Note that $C(1) = 3h(1) \cdot s(1) + M(1) \bmod q$ [1-dim NTRU!]
- Find $f_1, g_1 \in \mathbb{Z}$ of magnitude $\approx \sqrt{q}$ s.t. $h(1) = f_1/g_1 \bmod q$.
- Use 1-dim decryption, to recover $M(1) \bmod 3$.

The factorisation of Φ is used to map the n-dimensional ring to a 1-dimensional ring.

Chosen plaintext attack for $\Phi = x^n - 1$

Parameters may be set to thwart (improved versions of) these lattice reduction attacks.
But there are easy poly-time chosen plaintext attacks.

We have: $C = 3h \cdot s + M \mod (q, x^n - 1)$, for some small s.

- Note that $C(1) = 3h(1) \cdot s(1) + M(1) \bmod q$ [1-dim NTRU!]
- Find $f_1, g_1 \in \mathbb{Z}$ of magnitude $\approx \sqrt{q}$ s.t. $h(1) = f_1/g_1 \bmod q$.
- Use 1-dim decryption, to recover $M(1) \bmod 3$.

The factorisation of Φ is used to map the n-dimensional ring to a 1-dimensional ring.

Chosen plaintext attack for $\Phi = x^n - 1$

Parameters may be set to thwart (improved versions of) these lattice reduction attacks.
But there are easy poly-time chosen plaintext attacks.

We have: $C = 3h \cdot s + M \mod (q, x^n - 1)$, for some small s.

- Note that $C(1) = 3h(1) \cdot s(1) + M(1) \mod q$ [1-dim NTRU!]
- Find $f_1, g_1 \in \mathbb{Z}$ of magnitude $\approx \sqrt{q}$ s.t. $h(1) = f_1/g_1 \mod q$.
- Use 1-dim decryption, to recover $M(1) \mod 3$.

The factorisation of Φ is used to map the n-dimensional ring to a 1-dimensional ring.

Outline of the talk

1- Regular NTRUEncrypt
2- Attacks on NTRUEncrypt
3- **The Ideal-SVP and Ring-LWE problems**
4- A provably secure NTRUEncrypt

From now on, we use

$$R = \mathbb{Z}[x]/(x^n + 1),$$

with n a power of 2.

$\mathcal{P}oly(n)$-Ideal-SVP

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

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

- We identify polynomials to vectors via their coefficients:

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

- An ideal I is mapped to an integer lattice.

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

$\mathcal{P}oly(n)$-Ideal-SVP

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

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

- We identify polynomials to vectors via their coefficients:

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

- An ideal I is mapped to an integer lattice.

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

$\mathcal{P}oly(n)$-Ideal-SVP

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

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

- We identify polynomials to vectors via their coefficients:

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

- An ideal I is mapped to an integer lattice.

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

$\mathcal{P}oly(n)$-Ideal-SVP

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

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

- We identify polynomials to vectors via their coefficients:

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

- An ideal I is mapped to an integer lattice.

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

Are these easier lattices?

First weakness.

- If I is an ideal lattice, we know a good approximation of the norm of any shortest non-zero vector.
- For $\Phi = x^n + 1$, we have a $\sqrt{n}$-factor approximation.
- Why? If s is a shortest non-zero vector of I, then

$$s \cdot R \subseteq I \text{ is a full-dimensional sublattice .}$$

Second weakness: [Cramer-Ducas-Peikert-Regev'15].

- If I is of the form $I = s \cdot R$ where s is quite short, then one can recover s in sub-exponential time in $n = \dim I$.
- Quantumly, this can be done in polynomial time.
- Limited to very special ideal lattices.

$\mathcal{P}oly(n)$-Ideal-SVP is believed to be as hard as $\mathcal{P}oly(n)$-SVP.

Are these easier lattices?

First weakness.

- If I is an ideal lattice, we know a good approximation of the norm of any shortest non-zero vector.
- For $\Phi = x^n + 1$, we have a $\sqrt{n}$-factor approximation.
- Why? If s is a shortest non-zero vector of I, then

$$s \cdot R \subseteq I \text{ is a full-dimensional sublattice .}$$

Second weakness: [Cramer-Ducas-Peikert-Regev'15].

- If I is of the form $I = s \cdot R$ where s is quite short, then one can recover s in sub-exponential time in $n = \dim I$.
- Quantumly, this can be done in polynomial time.
- Limited to very special ideal lattices.

$\mathcal{P}oly(n)$-Ideal-SVP is believed to be as hard as $\mathcal{P}oly(n)$-SVP.

Are these easier lattices?

First weakness.

- If I is an ideal lattice, we know a good approximation of the norm of any shortest non-zero vector.
- For $\Phi = x^n + 1$, we have a $\sqrt{n}$-factor approximation.
- Why? If s is a shortest non-zero vector of I, then

$$s \cdot R \subseteq I \text{ is a full-dimensional sublattice .}$$

Second weakness: [Cramer-Ducas-Peikert-Regev'15].

- If I is of the form $I = s \cdot R$ where s is quite short, then one can recover s in sub-exponential time in $n = \dim I$.
- Quantumly, this can be done in polynomial time.
- Limited to very special ideal lattices.

$\mathcal{P}oly(n)$-Ideal-SVP is believed to be as hard as $\mathcal{P}oly(n)$-SVP.

Does NTRU involve ideal lattices?

No!

- The NTRU lattice $\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot R^2$ is...
  a 2-dimensional module over R.
- An ideal lattice is 1-dimensional over R.
- It could be that ideal lattice problems are easy to solve, but NTRU remains hard to break.

Does NTRU involve ideal lattices?

No!

- The NTRU lattice $\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot R^2$ is...
  a 2-dimensional module over R.
- An ideal lattice is 1-dimensional over R.
- It could be that ideal lattice problems are easy to solve, but NTRU remains hard to break.

Does NTRU involve ideal lattices?

No!

- The NTRU lattice $\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot R^2$ is...
  a 2-dimensional module over R.
- An ideal lattice is 1-dimensional over R.
- It could be that ideal lattice problems are easy to solve, but NTRU remains hard to break.

Does NTRU involve ideal lattices?

No!

- The NTRU lattice $\begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \cdot R^2$ is...
  a 2-dimensional module over R.
- An ideal lattice is 1-dimensional over R.
- It could be that ideal lattice problems are easy to solve, but NTRU remains hard to break.

The Ring-LWE Problem

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

  $\Rightarrow$ Element of R with small coefficients.

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

(Simplified) R-LWE [Lyubashevsky-Peikert-Regev'11]

Distinguish between D_α and $U(R_q \times R_q)$.

The Ring-LWE Problem

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

  $\Rightarrow$ Element of R with small coefficients.
- The R-LWE distribution D_α:
  - Sample $a \hookleftarrow U(R_q)$, $s \hookleftarrow \nu_\alpha$, $e \hookleftarrow \nu_\alpha$,
  - Return $(a, as + e) \in R_q \times R_q$.

(Simplified) R-LWE [Lyubashevsky-Peikert-Regev'11]

Distinguish between D_α and $U(R_q \times R_q)$.

The Ring-LWE Problem

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

  $\Rightarrow$ Element of R with small coefficients.
- The R-LWE distribution D_α:
  - Sample $a \hookleftarrow U(R_q)$, $s \hookleftarrow \nu_\alpha$, $e \hookleftarrow \nu_\alpha$,
  - Return $(a, as + e) \in R_q \times R_q$.

(Simplified) R-LWE [Lyubashevsky-Peikert-Regev'11]

Distinguish between D_α and $U(R_q \times R_q)$.

R-LWE is hard [Lyubashevsky-Peikert-Regev'11]

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.

- The arithmetic restriction on q can be removed [Langlois-Stehlé'14]
- It could be that $\mathcal{P}oly(n)$-Ideal-SVP is easy, but R-LWE remains hard to solve.

R-LWE is hard [Lyubashevsky-Peikert-Regev'11]

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.

- The arithmetic restriction on q can be removed [Langlois-Stehlé'14]
- It could be that $\mathcal{P}oly(n)$-Ideal-SVP is easy, but R-LWE remains hard to solve.

Outline of the talk

1- Regular NTRUEncrypt
2- Attacks on NTRUEncrypt
3- The Ideal-SVP and Ring-LWE problems
4- **A provably secure** NTRUEncrypt

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:

- $(a, as + e)$ is pseudo-random when $a \hookleftarrow U(R_q^+), s, e \hookleftarrow \nu_\alpha$.
- Let's change rings and replace "(h, hs)" by "$(h, hs + e)$"!

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:

- $(a, as + e)$ is pseudo-random when $a \hookleftarrow U(R_q^+), s, e \hookleftarrow \nu_\alpha$.
- Let's change rings and replace "(h, hs)" by "$(h, hs + e)$"!

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:

- $(a, as + e)$ is pseudo-random when $a \hookleftarrow U(R_q^+), s, e \hookleftarrow \nu_\alpha$.
- Let's change rings and replace "(h, hs)" by "$(h, hs + e)$"!

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:

- $(a, as + e)$ is pseudo-random when $a \hookleftarrow U(R_q^+), s, e \hookleftarrow \nu_\alpha$.
- Let's change rings and replace "(h, hs)" by "$(h, hs + e)$"!

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 3
  - Coeffs of f and g in $\{-1, 0, 1\}$
- pk: $h = g/f \bmod q$.

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

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

Decryption of $C \in R_q$:

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

The modified scheme

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

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

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

Decryption of $C \in R_q$:

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

Is it that simple?

Pseudo-randomness of $(h, hs + e)$

$\Downarrow$

Pseudo-randomness of $(h, 2(hs + e))$

$\Downarrow$

M is computationally hidden in $(h, 2(hs + e) + M)$

$\Downarrow$

IND-CPA security of the modified scheme

There is a catch!

Relying on R-LWE requires h uniform in R_q^+.
But here h is the quotient of two small polynomials in R^+...

See: `perso.ens-lyon.fr/damien.stehle/NTRU.html`

Is it that simple?

Pseudo-randomness of $(h, hs + e)$

$\Downarrow$

Pseudo-randomness of $(h, 2(hs + e))$

$\Downarrow$

M is computationally hidden in $(h, 2(hs + e) + M)$

$\Downarrow$

IND-CPA security of the modified scheme

There is a catch!

Relying on R-LWE requires h uniform in R_q^+.
But here h is the quotient of two small polynomials in R^+...

See: `perso.ens-lyon.fr/damien.stehle/NTRU.html`

A provably secure variant of NTRUEncrypt

It is possible to modify NTRUEncrypt so that:

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

What's the interest of this result?

What we prove:

- There is a variant of NTRUEncrypt that is secure under the assumption that $\mathcal{P}oly(n)$-Ideal-SVP is hard.
- It's asymptotically as efficient as the original scheme.

What's the interest of this result?

What we prove:

- There is a variant of NTRUEncrypt that is secure under the assumption that $\mathcal{P}oly(n)$-Ideal-SVP is hard.
- It's asymptotically as efficient as the original scheme.

It does not mean we should blindly replace NTRUEncrypt by this variant: It is much less practical!

What's the interest of this result?

What we prove:

- There is a variant of NTRUEncrypt that is secure under the assumption that $\mathcal{P}oly(n)$-Ideal-SVP is hard.
- It's asymptotically as efficient as the original scheme.

It does not mean we should blindly replace NTRUEncrypt by this variant: It is much less practical!

What it suggests:

- The general design of NTRUEncrypt is sound.
- It hints to cheap modifications towards more security
  - Change the underlying ring.
  - Replace hs by $hs + e$, to thwart trivial CPA attacks.
  - Take less small coefficients for f, g, s, e.

Outline of the talk

1- Regular NTRUEncrypt
2- The Ideal-SVP and R-LWE problems
3- A provably secure NTRUEncrypt
4- On NTRUSign

Conclusion

- NTRUEncrypt has resisted well against years of cryptanalytic efforts.
- The design is sound: a mild modification admits a security proof under standard hardness assumptions.

- The security of NTRU is related to lattices defined from polynomial rings.
- Deep connection with algebraic number theory.

Conclusion

- NTRUEncrypt has resisted well against years of cryptanalytic efforts.
- The design is sound: a mild modification admits a security proof under standard hardness assumptions.

- The security of NTRU is related to lattices defined from polynomial rings.
- Deep connection with algebraic number theory.

Open problems

Underlying hardness assumptions.

- Is $\mathcal{P}oly(n)$-Ideal-SVP really so hard?
- Are R-LWE and NTRU security equivalent to $\mathcal{P}oly(n)$-Ideal-SVP?

What about practice?

- Which modifications to achieve good efficiency and security?
- What are the limits of the best known practical attacks?
- How to set parameters?

Design

- Can we design more advanded primitives, from NTRU?
- [Lopez-Alt, Tromer, Vaikuntanathan'12] Application to fully homomorphic encryption and multi-party computation.

Open problems

Underlying hardness assumptions.

- Is $\mathcal{P}oly(n)$-Ideal-SVP really so hard?
- Are R-LWE and NTRU security equivalent to $\mathcal{P}oly(n)$-Ideal-SVP?

What about practice?

- Which modifications to achieve good efficiency and security?
- What are the limits of the best known practical attacks?
- How to set parameters?

Design

- Can we design more advanded primitives, from NTRU?
- [Lopez-Alt, Tromer, Vaikuntanathan'12] Application to fully homomorphic encryption and multi-party computation.

Open problems

Underlying hardness assumptions.

- Is $\mathcal{P}oly(n)$-Ideal-SVP really so hard?
- Are R-LWE and NTRU security equivalent to $\mathcal{P}oly(n)$-Ideal-SVP?

What about practice?

- Which modifications to achieve good efficiency and security?
- What are the limits of the best known practical attacks?
- How to set parameters?

Design

- Can we design more advanded primitives, from NTRU?
- [Lopez-Alt, Tromer, Vaikuntanathan'12] Application to fully homomorphic encryption and multi-party computation.