Reduced Memory Meet-in-the-Middle Attack against the NTRU Private Key

Christine van Vredendaal

Eindhoven, University of Technology

c.v.vredendaal@tue.nl

Twelfth Algorithmic Number Theory Symposium
University of Kaiserslautern, August 29, 2016

NTRU

- NTRU [HPS06] is a proposal for a post-quantum cryptosystem
- Fast due to being polynomial-ring based

> *"[NTRU] is considerably faster; that is something we acknowledge."*
>
> — Ari Juels, Chief Scientist, RSA Labs

NTRU

- NTRU [HPS06] is a proposal for a post-quantum cryptosystem
- Fast due to being polynomial-ring based

> *"[NTRU] is considerably faster; that is something we acknowledge."*
>
> — Ari Juels, Chief Scientist, RSA Labs

- Best known attacks;
  - meet-in-the-middle attack (mitm)
  - lattice-basis reduction (e.g. LLL, BKZ, sieving)
  - combination thereof (hybrid)
- Parameters are chosen based on analysis of these attacks

NTRU cryptosystem

- Define parameters; prime N, large modulus q, small modulus p
- Do math in the quotient ring $\mathbb{Z}[X]/(X^N - 1)$

NTRU cryptosystem

- Define parameters; prime N, large modulus q, small modulus p
- Do math in the quotient ring $\mathbb{Z}[X]/(X^N - 1)$
- Private keys small in this ring (parameter d)

$$f, g = a_0 + a_1 X + \ldots a_{N-1} X^{N-1} \text{ with } a_i \in \{0, 1\}$$

- Public key

$$h \equiv p \cdot f^{-1} g \bmod q$$

NTRU cryptosystem

- Define parameters; prime N, large modulus q, small modulus p
- Do math in the quotient ring $\mathbb{Z}[X]/(X^N - 1)$
- Private keys small in this ring (parameter d)

$$f, g = a_0 + a_1 X + \ldots a_{N-1} X^{N-1} \text{ with } a_i \in \{0, 1\}$$

- Public key

$$h \equiv p \cdot f^{-1} g \bmod q$$

- Encryption (message m, nonce r)

$$e \equiv r \cdot h + m \bmod q$$

- Decryption (simplified)

$$m = f \cdot e \bmod q \bmod p$$

Odlyzko's meet-in-the-middle attack on NTRU

- Idea: split the possibilities for $f \in \mathcal{F} = \mathcal{F}_1 \oplus \mathcal{F}_2$

$$h = (f_1 + f_2)^{-1} g \mod q$$
$$f_1 \cdot h = g - f_2 \cdot h \mod q.$$

- If there was no g: collision search in $f_1 \cdot h$ and $-f_2 \cdot h$

Odlyzko's meet-in-the-middle attack on NTRU

- Idea: split the possibilities for $f \in \mathcal{F} = \mathcal{F}_1 \oplus \mathcal{F}_2$

$$h = (f_1 + f_2)^{-1} g \mod q$$
$$f_1 \cdot h = g - f_2 \cdot h \mod q.$$

- If there was no g: collision search in $f_1 \cdot h$ and $-f_2 \cdot h$
- Solution: look for collisions in $c(f_1 \cdot h)$ and $c(-f_2 \cdot h)$ with

$$c(a_0 + a_1 x + \cdots + a_{N-1} x^{N-1}) = (\mathbb{1}(a_0 \geq q/2), \ldots, \mathbb{1}(a_{N-1} \geq q/2))$$

- We call $c(f)$ the address of a polynomial f

Odlyzko's meet-in-the-middle attack on NTRU

- Idea: split the possibilities for $f \in \mathcal{F} = \mathcal{F}_1 \oplus \mathcal{F}_2$

$$h = (f_1 + f_2)^{-1} g \mod q$$
$$f_1 \cdot h = g - f_2 \cdot h \mod q.$$

- If there was no g: collision search in $f_1 \cdot h$ and $-f_2 \cdot h$
- Solution: look for collisions in $c(f_1 \cdot h)$ and $c(-f_2 \cdot h)$ with

$$c(a_0 + a_1 x + \cdots + a_{N-1} x^{N-1}) = (\mathbb{1}(a_0 \geq q/2), \ldots, \mathbb{1}(a_{N-1} \geq q/2))$$

- We call $c(f)$ the address of a polynomial f
- Due to small coeficients of g, small probability of

$$c(-f_2 \cdot h) \neq c(g - f_2 \cdot h) \Rightarrow c(f_1 \cdot h) \neq c(-f_2 \cdot h)$$

Odlyzko's meet-in-the-middle attack on NTRU

- Idea: split the possibilities for $f \in \mathcal{F} = \mathcal{F}_1 \oplus \mathcal{F}_2$

$$h = (f_1 + f_2)^{-1} g \mod q$$
$$f_1 \cdot h = g - f_2 \cdot h \mod q.$$

- If there was no g: collision search in $f_1 \cdot h$ and $-f_2 \cdot h$
- Solution: look for collisions in $c(f_1 \cdot h)$ and $c(-f_2 \cdot h)$ with

$$c(a_0 + a_1 x + \cdots + a_{N-1} x^{N-1}) = (\mathbb{1}(a_0 \geq q/2), \ldots, \mathbb{1}(a_{N-1} \geq q/2))$$

- We call $c(f)$ the address of a polynomial f
- Due to small coeficients of g, small probability of

$$c(-f_2 \cdot h) \neq c(g - f_2 \cdot h) \Rightarrow c(f_1 \cdot h) \neq c(-f_2 \cdot h)$$

- Take this into account by checking where $a_i = \lfloor q/2 \rfloor - 1$.

Mitm: the Algorithm

begin

$L \leftarrow$ Enumerate all $f_2 \in \mathcal{F}_2$ and store $(f_2, c_1(-f_2 \cdot h), c_2(-f_2 \cdot h), \dots)$;

while *not found* **do**

$f_1 \leftarrow_R \mathcal{F}_1$;

$a \leftarrow c(f_1 \cdot h)$;

if $a \in L$ *for some* f_2 **then**

if $(f_1 + f_2) \cdot h$ *is binary* **then**

return $f = (f_1 + f_2)$, $g = (f_1 + f_2) \cdot h$

Security of NTRU against Odlyzko's meet-in-the-middle attack

- General running time / memory mitm

$$L = \mathcal{O}\left(\frac{\sqrt{|\mathcal{F}|}}{\sqrt{s}}\right).$$

- $\mathcal{F}$ is the space of solutions
- s is the number of solutions, i.e rotations:

$$f_1 \cdot h = g - f_2 \cdot h \mod q \Rightarrow x^i f_1 \cdot h = x^i g - x^i f_2 \cdot h \mod q.$$

Security of NTRU against Odlyzko's meet-in-the-middle attack

- General running time / memory mitm

$$L = \mathcal{O}\left(\frac{\sqrt{|\mathcal{F}|}}{\sqrt{s}}\right).$$

- $\mathcal{F}$ is the space of solutions
- s is the number of solutions, i.e rotations:

$$f_1 \cdot h = g - f_2 \cdot h \mod q \Rightarrow x^i f_1 \cdot h = x^i g - x^i f_2 \cdot h \mod q.$$

- But is memory = running time?
- Can we get a time/memory trade-off?

Some motivation

Regarding Odlyzko's meet-in-the-middle attack in a CRYPTO 2007 paper

"*Odlyzkos attack on the ees251ep6 parameter set will require too many operations and/or too much storage to be feasible, and hence the parameter set is more than adequate for a $k = 80$ security level.* **Of these two constraints the storage requirement is by far the larger obstacle in todays hardware.**"

— Nick Howgrave-Graham, NTRU Cryptosystems, Inc.

A little more motivation

- "2^{70} *bytes of storage is approximately* 1500 *times as expensive as* 2^{80} *floating point operations*"

  — Dan Bernstein, Tanja Lange [BL14]

A little more motivation

- "2^{70} *bytes of storage is approximately* 1500 *times as expensive as* 2^{80} *floating point operations*"

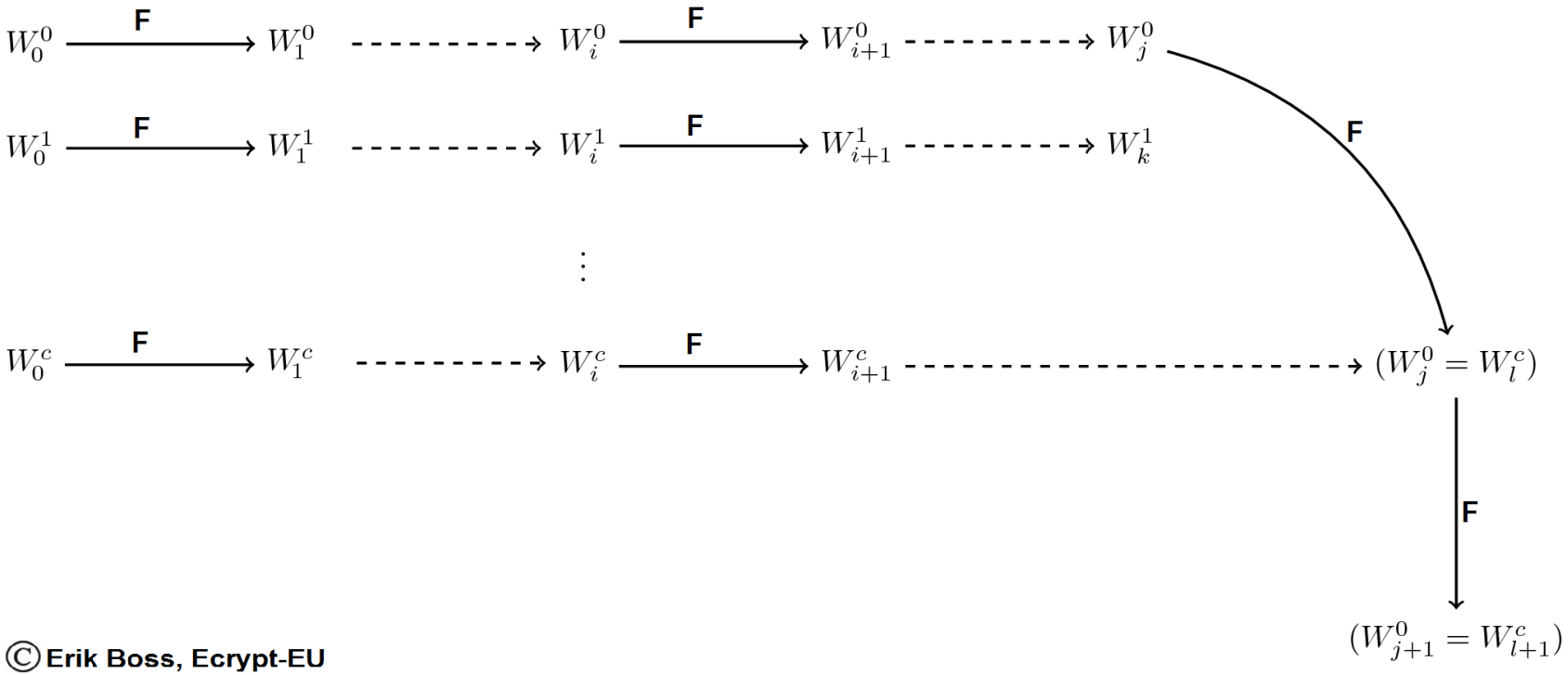
  — Dan Bernstein, Tanja Lange [BL14]
- Amazon EC2 Cloud: 2^{60} operations and 2^{36} bits of storage for 3000$
- Sorting numbers is more expensive than their computation

Memory reduction in collision search

- CRYPTO'97 paper van Oorschot/Wiener: reduce the memory of a collision search in a space $\mathcal{S}$

Memory reduction in collision search

- CRYPTO'97 paper van Oorschot/Wiener: reduce the memory of a collision search in a space $\mathcal{S}$
- It works by taking a function $F : \mathcal{S} \to \mathcal{S}$ and creating trails of values



Collision search result

Variable memory for regular mitm attack was also in [OW99]

Heuristic (From OW99)

Let $F : \mathcal{S} \to \mathcal{S}$ and $w \geq 2^{10}$ the number of triples (W_0, W_t, t) that can be stored. Then the (conjectured) optimum proportion of distinguished points is $\theta \approx 2.25\sqrt{w/|\mathcal{S}|}$, and one should generate about $10w$ trails per version of F. The expected number of iterations of F required to complete a meet-in-the-middle attack using these parameters is $(2.5|\mathcal{S}|^{3/2}/w^{1/2})r$, and the expected number of memory accesses is $4.5|\mathcal{S}|$.

- Distinguished points dictate the length of the trails
- Here r is the time for 1 iteration of F
- A version of F is a randomization of F

Applying collision search to NTRU mitm attack

- Problem 1 for NTRU: the storage of multiple addresses

Applying collision search to NTRU mitm attack

- Problem 1 for NTRU: the storage of multiple addresses

Lemma

Suppose f and g are randomly chosen of degree $N-1$ with d coefficients set to 1. Under the assumption that the public key h is uniformly distributed over $\mathcal{R}$, the probability that g will not change the address c of $-f_2 h$ is $(1-\frac{d}{Nq})^N \approx e^{-\frac{d}{q}}$.

- If $q \approx 4N$ and $d \approx 2N/3$ then this probability is ≈ 0.85.

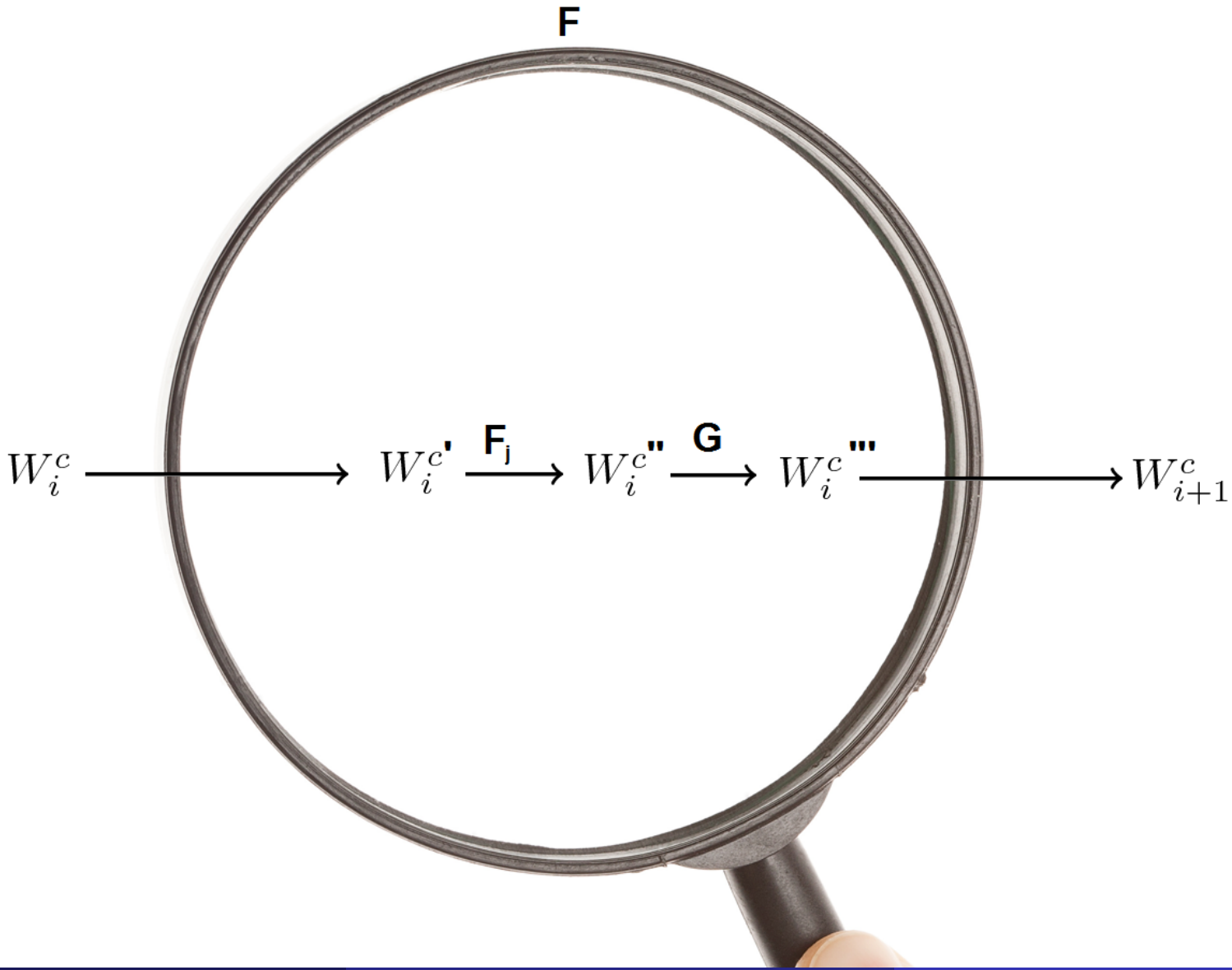
Applying collision search to NTRU mitm attack

- Problem 2 for NTRU: the function

$$W_i^c \xrightarrow{\mathbf{F}} W_{i+1}^c$$

Applying collision search to NTRU mitm attack

- Problem 2 for NTRU: the function



Memory reduction function in NTRU mitm

- Ingredients:
  - $I > \max\{|\mathcal{F}_1|, |\mathcal{F}_2|\}$
  - $\mathcal{S} = [0, I) \times \{0, 1\}$
  - $I_i : [0, I) \to \mathcal{F}_i \ \ (i = 1, 2)$
  - H hash function with $|\mathrm{codomain}(H)| \gg I$

Memory reduction function in NTRU mitm

- Ingredients:
  - $I > \max\{|\mathcal{F}_1|, |\mathcal{F}_2|\}$
  - $\mathcal{S} = [0, I) \times \{0, 1\}$
  - $I_i : [0, I) \to \mathcal{F}_i \ \ (i = 1, 2)$
  - H hash function with $|\mathrm{codomain}(H)| \gg I$
- Define the functions:

$$
\begin{aligned}
F_i : [0, I) &\to \{0, 1\}^N \\
x &\to c(\pm I_i(x) \cdot h) \\
G : \{0, 1\}^n &\to \mathcal{I} \\
x &\to (H(x) \bmod |I|) \times \mathrm{MSB}(H(x)).
\end{aligned}
$$

Memory reduction function in NTRU mitm

- Ingredients:
  - $I > \max\{|\mathcal{F}_1|, |\mathcal{F}_2|\}$
  - $\mathcal{S} = [0, I) \times \{0, 1\}$
  - $I_i : [0, I) \to \mathcal{F}_i \ \ (i = 1, 2)$
  - H hash function with $|\mathrm{codomain}(H)| \gg I$
- Define the functions:

$$\begin{aligned} F_i : [0, I) &\to \{0, 1\}^N \\ x &\to c(\pm I_i(x) \cdot h) \\ G : \{0, 1\}^n &\to \mathcal{I} \\ x &\to (H(x) \bmod |I|) \times \mathrm{MSB}(H(x)). \end{aligned}$$

- Now $F : \mathcal{S} \to \mathcal{S}$ with $F(x, i) = G(F_{i+1}(x))$

Collision search in (binary) NTRU

Heuristic

Let w be the number of triples (W_0, W_t, t) for which there is available memory. Let $\mathcal{D}$ be a set of distinguished points with $|\mathcal{D}|/|\mathcal{F}| = \theta = \alpha\sqrt{w/|\mathcal{F}|} = 2.25\sqrt{w/|\mathcal{F}|}$.
Then the algorithm is expected to run in

$$L^* = 5r\sqrt{\frac{2\binom{n}{d}\binom{n/2}{d/2}}{nw}},$$

operations, where r is the number of operations needed for a function evaluation of F.

Current parameters?

- Best attack(s) on current NTRU parameters involve the meet-in-the-middle attack
- We extrapolate what the heuristics mean for the hybrid attack
- We assume that memory is 2^{10} times as expensive as a multiplication

Current parameters?

- Best attack(s) on current NTRU parameters involve the meet-in-the-middle attack
- We extrapolate what the heuristics mean for the hybrid attack
- We assume that memory is 2^{10} times as expensive as a multiplication

sec. goal	N	sec. est.	hybrid mitm memory req.	direct mitm memory req.	memory cap	operation count
112	401	116	117	145	109	119
128	439	133	133	147	125	136
192	593	193	204	193	189	199
256	743	256	279	256	252	262

Table : An overview of NTRU parameters. All security and memory values are $\log_2$ values. For each parameterset $q = 2048$.

Figure : Also a collision.

Questions?