

# Masking NTRU

NTRU pqc scheme and potential masking scheme for physical security

**Lukas Prokop**

2021-09-16

<https://lukas-prokop.at/talks/seminar-masking-ntru/>

IAIK – Graz University of Technology

## A post-quantum scheme

---

**1996** CRYPTO'96 rump session

preliminary draft by J. Hoffstein, J. Pipher, and J.H. Silverman<sup>1</sup>

*Properties:* probabilistic public key cryptosystem as defined by Goldwasser and Micali, turned into deterministic one, 0% decoding failure

**1998** “NTRU: A Ring-Based Public Key Cryptosystem” by the same authors<sup>2</sup>

---

<sup>1</sup><https://web.securityinnovation.com/hubfs/files/ntru-orig.pdf>

<sup>2</sup><https://doi.org/10.1007/BFb0054868>

**2003** EESS #1: Implementation aspects of NTRU-Encrypt and NTRUSign v2.0<sup>3</sup>

**2017** “High-Speed Key Encapsulation from NTRU” by Andreas Hülsing, Joost Rijneveld, John M. Schanck, Peter Schwabe<sup>4</sup>

NIST Post-Quantum Standardization Process:

**2017** Round 1: KEMs {NTRUEncrypt, NTRU-HRSS-KEM, NTRU Prime}

**2019** Round 2: KEMs {NTRU, NTRU Prime}

**2020** Round 3: KEMs {NTRU (finalist), NTRU Prime (alt)}

---

<sup>3</sup><https://web.archive.org/web/20030723065225/http://grouper.ieee.org/groups/1363/lattPK/submissions/EESS1v2.pdf>

<sup>4</sup>[https://doi.org/10.1007/978-3-319-66787-4\\_12](https://doi.org/10.1007/978-3-319-66787-4_12)

NTRU Prime = Streamlined NTRU Prime + NTRU LPRime

*NTRU Prime was advanced to the third round but not as a finalist. Additional motivation for NTRU Prime's unique choice of algebraic structure could be gained by new progress in algebraic cryptanalysis of cyclotomic structures during the third round, provided that it undermines NIST's confidence in cyclotomic structures but clearly does not extend to NTRUprime's choice of  $\mathbb{Z}_q[x]/(x^p - x - 1)$ .*

*—NISTIR 8309: Status report on the second round of the NIST PQC Standardization Process*

**Parameter set**  $N$ ,  $p$ , and  $q$  are integers

s.t.  $\gcd(p, q) = 1$  and  $q \gg p$ .

Let  $R := \mathbb{Z}[X]/(X^N - 1)$  define ring<sup>5</sup>  $(R, \oplus, \otimes)$ .

**Key generation** pick random polynomials  $f$  and  $g$  in  $R$

s.t.  $f_p^{-1} \in \mathbb{Z}[X]/(p)$  and  $f_q^{-1} \in \mathbb{Z}[X]/(q)$  (i.e. mult. inverses exist)

$\implies f_p^{-1} \otimes f \equiv 1 \pmod{p}$  and  $f_q^{-1} \otimes f \equiv 1 \pmod{q}$

$$h := f_q^{-1} \otimes g \pmod{q}$$

public key =  $(h)$ , secret key =  $(f)$

---

<sup>5</sup> $(X^N - 1)$  is not a cyclotomic with  $N > 1$ . To the best of knowledge, this ring must be commutative.

**Encrypt** Let  $m$  be an  $N$ -bits message<sup>6</sup>. Pick random polynomial  $\phi$  in  $R$ .

$$e := p \cdot \phi \otimes h + m \cdot \left\lfloor \frac{q}{2} \right\rfloor \bmod q$$

ciphertext =  $(e)$

**Decrypt**

$$a := f \otimes e \bmod q$$

$$m = f_p^{-1} \otimes a \bmod p$$

recovered plaintext =  $(m)$

---

<sup>6</sup>Representation with rounding factor added by myself. Bits implies Binary NTRU (see later slides).

$$\begin{aligned}
 m &= f_p^{-1} \otimes a && \text{mod } p \\
 &= f_p^{-1} \otimes (f \otimes e \text{ mod } q) && \text{mod } p \\
 &= f_p^{-1} \otimes (f \otimes (p \cdot \phi \otimes h + m \cdot \lfloor \frac{q}{2} \rfloor \text{ mod } q) \text{ mod } q) && \text{mod } p \\
 &= f_p^{-1} \otimes (f \otimes (p \cdot \phi \otimes (f_q^{-1} \otimes g \text{ mod } q) + m \cdot \lfloor \frac{q}{2} \rfloor \text{ mod } q) \text{ mod } q) && \text{mod } p \\
 \hline
 &= f_p^{-1} \otimes (p \cdot \phi \otimes (f \otimes f_q^{-1} \otimes g \text{ mod } q) + f \otimes m \cdot \lfloor \frac{q}{2} \rfloor \text{ mod } q) \text{ mod } q && \text{mod } p \\
 &= f_p^{-1} \otimes (p \cdot \phi \otimes g + f \otimes m \cdot \lfloor \frac{q}{2} \rfloor \text{ mod } q) \text{ mod } q && \text{mod } p \\
 &= (f_p^{-1} \otimes p \cdot \phi \otimes g) + (f_p^{-1} \otimes f \otimes m \cdot \lfloor \frac{q}{2} \rfloor \text{ mod } q) && \text{mod } p \\
 &= m \cdot \lfloor \frac{q}{2} \rfloor \text{ mod } q = m
 \end{aligned}$$



## First NTRU variants

Distinguished since 1998<sup>7</sup>.

**Binary NTRU**  $(N, p, q, d)$   $d$  polynomial coefficients of  $f, g, \phi$ , and  $m$  are one.

$N - d$  coefficients are zero.

**Symmetric NTRU**  $(N, p, q, d, r)$  The number of polynomial coefficients of  $f, g, \phi$ , and  $m$  equal ...

$$\underbrace{-r, -r + 1, \dots, -1}_d, \underbrace{0}_{N-rd-1}, \underbrace{1}_{d+1}, \underbrace{2, 3, \dots, r}_d$$

Special case  $r = 1$  called “ternary” or “trinary”.

---

<sup>7</sup> “NTRU: A Ring-Based Public Key Cryptosystem” by J. Hoffstein, J. Pipher, and J.H. Silverman (1998)

Decisional hardness problem establishing the security of NTRU:

*Let  $f$  and  $g$  be two small elements of ring  $R$  (or uniformly random).*

*Let  $h = f/g$ .*

*Given  $h$ . Decide*

- *whether  $h$  is a random ring element*
- *or  $f$  and  $g$  exist.*

## Parameter sets

NIST security level <sup>8</sup>	1	3	5	3
	ntruhs2048509	ntruhs2048677	ntruhs4096821	ntruhrs701
$n$	509	677	821	701
$q$	2048	2048	4096	8192
public key size	699 [672]	930 [992]	1230 [1312]	1138 [992]
private key size	935 [832]	1234 [1248]	1590 [1664]	1450 [1248]
ciphertext size	699 [736]	930 [1088]	1230 [1472]	1138 [1088]
shared secret size	256	256	256	256

Sizes always in bytes. Hash function is always SHA3\_256. Values in braces for uncompressed Saber variants for reference.

<sup>8</sup>Assuming the *local* model acc. to Round 3 spec. page 31.

**HPS** fixed-weight sample space

$n$  is prime

$p = 3$ ,  $q$  is a power of two

**HRSS** arbitrary-weight sample space

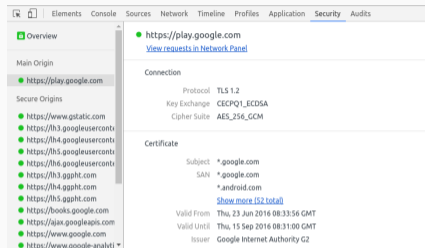
$n$  is prime

$p = 3$ ,  $q = 2^{\lceil 7/2 + \log_2(n) \rceil}$

CPU cycles <sup>9</sup>	KeyGen	Encaps	Decaps
ntruhs2048509 (1)	77 698 713	645 329	542 439
lightsaber (1)	459 965	651 273	678 810
ntruhs2048677 (3)	144 383 491	955 902	836 959
ntruhrss701 (3)	154 676 705	402 784	890 231
saber (3)	896 035	1 161 849	1 204 633
ntruhs4096821 (5)	211 758 452	1 205 662	1 066 879
firesaber (5)	1 448 776	1 786 930	1 853 339

Measurements for the  $m4$  implementation. Distribution (98 %, 1 %, 1 %) versus Saber's (28 %, 36 %, 37 %).

<sup>9</sup>“pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4” by M. J. Kannwischer, J. Rijneveld, P. Schwabe, K. Stoffelen (2019) for round 2 candidates



“Experimenting with Post-Quantum Cryptography” image by Google Security

CECPQ = **C**ombined **E**lliptic-**C**urve and **P**ost-**Q**uantum

**CECPQ1** X25519 and NewHope; 2016; Google Chrome 54 beta only

**CECPQ2** X25519 and HRSS; 2019; Cloudflare and Google Chrome Canary

**CECPQ2b** X25519 and SIKE/p434; 2019

# Masking NTRU

---

1. [LSCH10] “Countermeasures against Power Analysis Attacks for the NTRU Public Key Cryptosystem” by M.K. Lee, J.E. Song, D. Choi, D.G. Han (2010)
  - “typical [. . .] NTRU is vulnerable to the simple power analysis and the correlation power analysis including a second-order power attack”
  - “present novel countermeasures to prevent these attacks”
  - “perform experiments to estimate the performance overheads of our countermeasures”
  - “overheads in required memory and execution time are only 8.17 % and 9.56 %, respectively, over a Tmote Sky equipped with an MSP430 processor.”
  - Convolution Product Computation attacked (binary NTRU used; generalizable)
  - Countermeasures (masking is not mentioned):
    - randomization of the temporary data stored in  $t$
    - blinding the public data  $c$
    - randomization of the secret data  $b$



2. [SMS19] “Practical Evaluation of Masking for NTRUEncrypt on ARM Cortex-M4” by T. Schamberger, O. Mischke, J. Sepúlveda (COSADE 2019)
- “provide a practical evaluation of masking applied to index-based multiplication”
  - “a modern parameter set using trinary polynomials”
  - Cortex-M4 SIMD instructions used to mitigate overhead
  - “no observable first-order leakage using a HW model and two million measurement traces”
  - “Successful second-order attacks are demonstrated” “
  - ”we show that applying both our low cost masking countermeasure together with a known and equally efficient shuffling scheme can provide a good trade-off achieving a high level of security without a large performance penalty”

$$a(x) \otimes b(x) = \sum_{k=0}^{N-1} \left( \sum_{i+j=k \pmod{N}} a_i b_j \right) x^k$$

- “we will utilize the Hamming distance power model because it is well suited to describe the power consumption of our target processor MSP430F157” [LSCH10]  
“we use the hamming weight power model” ... “STM32F303RCT7” [SMS19]
- [LSCH10] attacks trinary variant; [SMS19] attacks binary variant; CPA could not be reproduced
- Both attack “ $f \otimes e \bmod p$ ” in the decryption step
- [SMS19] considers masking: use arithmetically shared masks, compute product per share, combine.
- Also parallel implementation considered: uses SIMD instr./DSP of Cortex-M4

*Countermeasure paper:* A. Wang, C. Wang, X. Zheng, W. Tian, R. Xu, G. Zhang: “Random key rotation: Side-channel countermeasure of NTRU cryptosystem for resource-limited devices” (2017)

<u>KeyGen'(seed)</u>	<u>Encrypt(h, (r, m))</u>	<u>Decrypt((f, f<sub>p</sub>, h<sub>q</sub>), c)</u>
1. (f, g) ← Sample_fg(seed)	1. m' ← Lift(m)	1. if c ≠ 0 (mod (q, Φ <sub>1</sub> )) return (0, 0, 1)
2. f <sub>q</sub> ← (1/f) mod (q, Φ <sub>n</sub> )	2. c ← (r · h + m') mod (q, Φ <sub>1</sub> Φ <sub>n</sub> )	2. a ← (c · f) mod (q, Φ <sub>1</sub> Φ <sub>n</sub> )
3. h ← (3 · g · f <sub>q</sub> ) mod (q, Φ <sub>1</sub> Φ <sub>n</sub> )	3. return c	3. m ← (a · f <sub>p</sub> ) mod (3, Φ <sub>n</sub> )
4. h <sub>q</sub> ← (1/h) mod (q, Φ <sub>n</sub> )		4. m' ← Lift(m)
5. f <sub>p</sub> ← (1/f) mod (3, Φ <sub>n</sub> )		5. r ← ((c - m') · h <sub>q</sub> ) mod (q, Φ <sub>n</sub> )
6. return ((f, f <sub>p</sub> , h <sub>q</sub> ), h)		6. if (r, m) ∈ ℒ <sub>r</sub> × ℒ <sub>m</sub> return (r, m, 0)
		7. else return (0, 0, 1)

Figure 9: The DPKE for the NTRU submission.

<u>KeyGen(seed)</u>	<u>Encapsulate(h)</u>	<u>Decapsulate((f, f<sub>p</sub>, h<sub>q</sub>, s), c)</u>
1. ((f, f <sub>p</sub> , h <sub>q</sub> ), h) ← KeyGen'(seed)	1. coins ← <sub>\$</sub> {0, 1} <sup>256</sup>	1. (r, m, fail) ← Decrypt((f, f <sub>p</sub> , h <sub>q</sub> ), c)
2. s ← <sub>\$</sub> {0, 1} <sup>256</sup>	2. (r, m) ← Sample_rm(coins)	2. k <sub>1</sub> ← H <sub>1</sub> (r, m)
3. return ((f, f <sub>p</sub> , h <sub>q</sub> , s), h)	3. c ← Encrypt(h, (r, m))	3. k <sub>2</sub> ← H <sub>2</sub> (s, c)
	4. k ← H <sub>1</sub> (r, m)	4. if fail = 0 return k <sub>1</sub>
	5. return (c, k)	5. else return k <sub>2</sub>

Figure 10: The KEM for the NTRU submission.

**Thank you!**

# Masking NTRU

NTRU pqc scheme and potential masking scheme for physical security

**Lukas Prokop**

2021-09-16

<https://lukas-prokop.at/talks/seminar-masking-ntru/>

IAIK – Graz University of Technology