Fast First-Order Masked NTTRU

14^{th} International Workshop on Constructive Side-Channel Analysis and Secure Design

Daniel Heinz[1,2] and Gabi Dreo Rodosek[1]

[1]Research Institute CODE, Universität der Bundeswehr München, 85577 Neubiberg, Germany

[2]Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany

April, 4th 2023

What is NTTRU?

- NIST Standardization process in final round:
  - Kyber
  - Saber
  - NTRU
- Kyber was standardized but NTRU remains important
  - OpenSSH
  - Google: NTRU in ALTS protocol
- Bottleneck: Polynomial multiplication
- Kyber's solution: Number Theoretic Transform (NTT)

NTT

- Discrete version of the Fast Fourier Transform (FFT)
- Requires specific parameter set for efficiency
- Allows fast (pointwise) polynomial multiplication
- Reason: Isomorphism by the CRT when $f = gh$ (g, h relatively prime):

$$\mathbb{Z}_q[X]/(f) \cong \mathbb{Z}_q[X]/(g) \times \mathbb{Z}_q[X]/(h) \tag{1}$$

- Bottleneck of NTT computation: Cooley-Tukey Algorithm

NTTRU

- Version of NTRU using NTT by Lyubashevsky and Seiler[1]
- Parameter set: $q = 7681$, $n = 768$
- Re-Encryption step (FO-Transform)

Algorithm 1: NTTRU.Decrypt($\hat{c}$,$\hat{f}$)

$\hat{m} \leftarrow \hat{c} \circ \hat{f}$

return $m := INTT(\hat{m}) \bmod {}^{\pm}3$

Kyber	NTRU-HRSS Dec.	NTTRU Dec.
102 029	65 042	7878

Table: Cycle Counts on an Intel Skylake i7-6600U CPU[1]

Side-Channels

- Embedded devices are in danger of being attacked by power analysis or fault attacks
- NTTRU: potentially used on embedded devices
- How to protect the secret key against EM or power analysis attacks (DPA)?

$\Rightarrow$ Masking

- Provable security in the t-probing model (i.e. resistance against probing t wires at the same time)
- Two types of Masking:
  - Arithmetic Masking: $x = x_1 + x_2 (\bmod q)$
  - Boolean Masking: $x = x_1 \oplus x_2$

Masking NTTRU

Contributions:

- We present a fully first-order masked version of NTTRU
- We present a fully first-order masked version of SHA512 which is part of NTRU
- We present table-based approaches for a first-order masked $\bmod 3$ operation and a sampler
- We propose a faster alternative by using the SHA3 standard
- Evaluation with respect to speed and first-order side-channel security

Concept

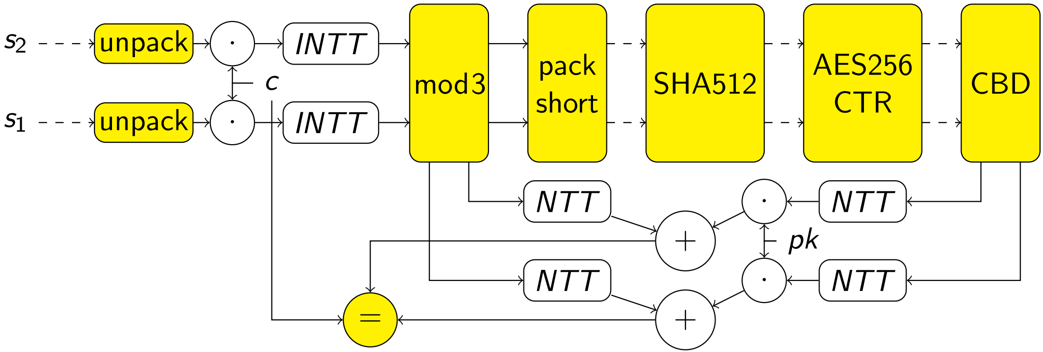
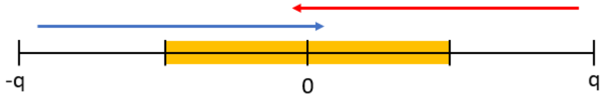


Figure: Masked decapsulation of NTTRU. Boolean shared data paths in dashed lines. Arithmetically shared data paths in solid lines. Non-linear functions in yellow.

Masked Mod3

- Representative $x \equiv c \mod q$ is important when reducing $x \mod 3$
- Idea: Remove the masking $\mod q$ to ensure linearity.
- Unmasked Output of INTT is in $[-(q-1), q-1]$



$\Rightarrow$ Reduce to correct representative shared in $\mathbb{Z}$

- Sign of the unmasked output: A2B conversion

Masked SHA512

- Boolean and Arithmetic operations combined
- First-Order: Conversions instead of Boolean Adders
- Masked control flow:

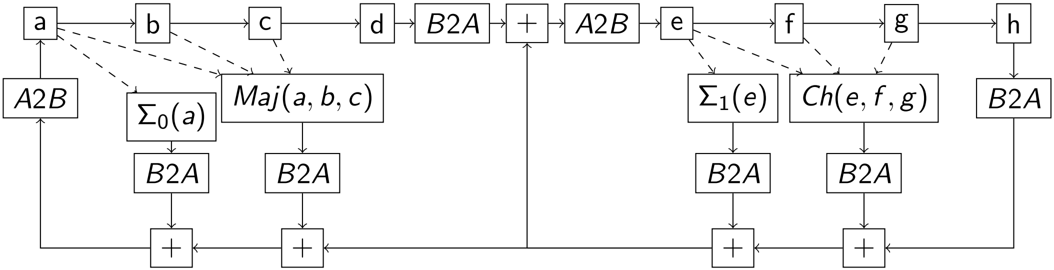


Figure: Masked SHA512 Compression function with conversions in place.

Masked Sampler

- Centered Binomial Distribution Sampling $[-1, 1]$
- Input: 4 bits $\Rightarrow$ 16 possibilities
- **Idea**: Extend the unmasked table-based approach to first-order
- Generate the table with random but fixed input mask($\in [0, 15]$) and output mask($\in [0, q-1]$)
- Get the required entry from the table
- **Drawback**: Requires re-generation of the table

Keccak instead of SHA2

- SHA2 requires Boolean and arithmetic shares during every round
- A2B, B2A Conversions are especially expensive in higher-orders
- Boolean shared Adders require many masked `AND` gates
- Keccak: No masked additions $\bmod 2^{64}$ required
- Proposal: SHA2-512 $\rightarrow$ SHA3-512, AES256-CTR $\rightarrow$ SHAKE256
- No security reduction

Performance Evaluation

- Platform: ARM Cortex-M4 on STM32F407G-DISC1 board
- 24 MHz, 192kB RAM

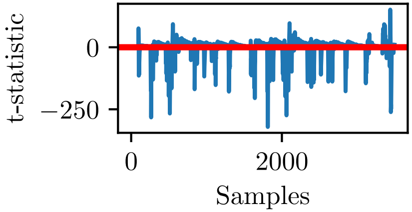
Table: CCA2-secure decapsulation cycle counts for different masked lattice-based schemes.

Scheme	**CPU**	**Cycles** $\times 10^3$ **Masked**	**Cycles** $\times 10^3$ **Unmasked**
Saber[2]	Cortex M4	2833	774
Kyber768[3]	Cortex M4	2978	783
NTRU[4]	Cortex M3	32 472	10 508
NTTRU (This work)	Cortex M4	9448	796
NTTRU-SHA3 (This work)	Cortex M4	3119	

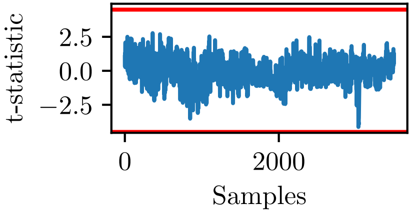
Side-Channel Evaluation

- TVLA methodology: Fixed vs random [5]
- Compute t-statistic

$$t = \frac{\mu_0 - \mu_1}{\sqrt{\frac{s_0^2}{n_0} + \frac{s_1^2}{n_1}}} \tag{2}$$



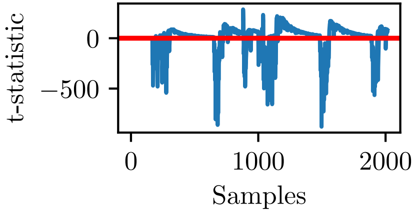
(a) RNG disabled (100 traces)



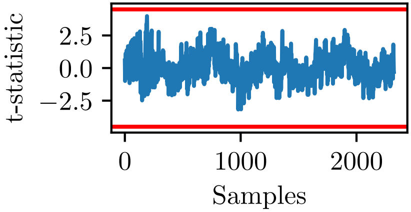
(b) RNG enabled (20 000 traces)

Figure: t-statistic of the masked modulus conversion. Red lines indicate the threshold of 4.5.

Side-Channel Evaluation



(a) RNG disabled (100 traces)



(b) RNG enabled (20 000 traces)

Figure: *t*-statistic of the masked modulus conversion. Red lines indicate the threshold of 4.5.

Conclusion and Outlook

- NTTRU is a competitive candidate among lattice-based schemes in a first-order masked setting
- Using Keccak the masked performance overhead is comparable to Kyber and Saber (around 300%)
- Future work: Improve the performance of linear parts in NTTRU
- **However**: ML-based attacks on Kyber and Saber
- Future work: Analyze the resistance of masked implementations against such attacks

Vadim Lyubashevsky and Gregor Seiler.
NTTRU: truly fast NTRU using NTT.
IACR Trans. Cryptogr. Hardw. Embed. Syst., 2019(3):180–201, 2019.

Michiel Van Beirendonck, Jan-Pieter D'Anvers, Angshuman Karmakar, Josep Balasch, and Ingrid Verbauwhede.
A side-channel-resistant implementation of SABER.
ACM J. Emerg. Technol. Comput. Syst., 17(2):10:1–10:26, 2021.

Daniel Heinz, Matthias J. Kannwischer, Georg Land, Thomas Pöppelmann, Peter Schwabe, and Daan Sprenkels.
First-order masked kyber on ARM cortex-m4.
IACR Cryptol. ePrint Arch., page 58, 2022.

Jean-Sébastien Coron, François Gérard, Matthias Trannoy, and Rina Zeitoun.
High-order masking of NTRU.
IACR Cryptol. ePrint Arch., page 1188, 2022.

Tobias Schneider and Amir Moradi.
Leakage assessment methodology - extended version.

J. Cryptogr. Eng., 6(2):85–99, 2016.