Implementation of a Higher-Order Masked Kyber KEM

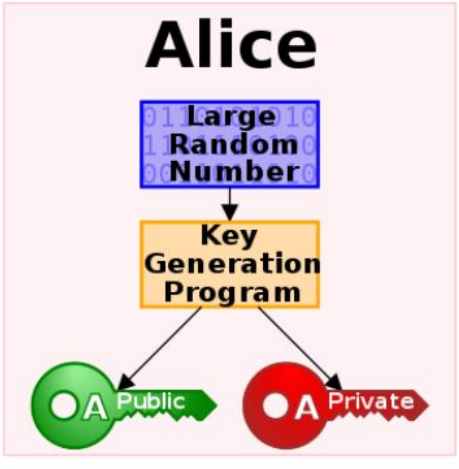
—

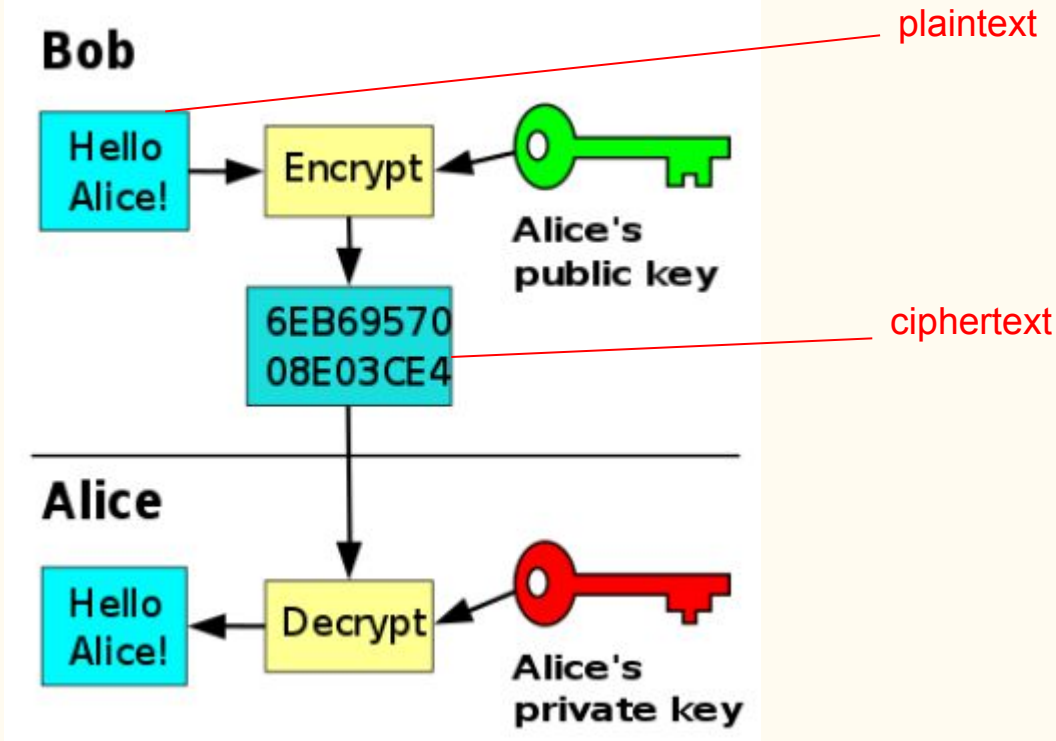
Group 8

JIAYI FENG, NILS PAULSRUD, VIKTOR OLSSON, YANNING JI, YUEHAN WANG

Content

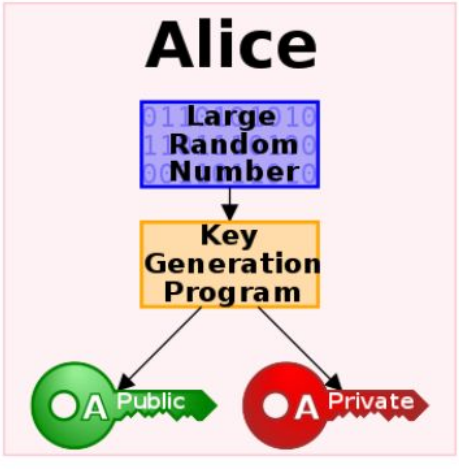
1 Introduction

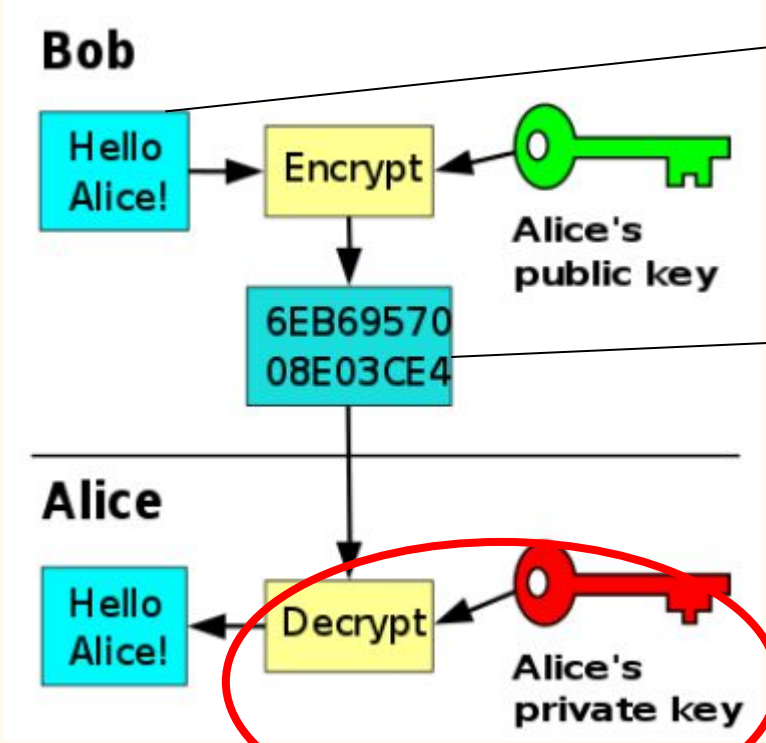




https://en.wikipedia.org/wiki/Public-key_cryptography

Side-channel attack





plaintext

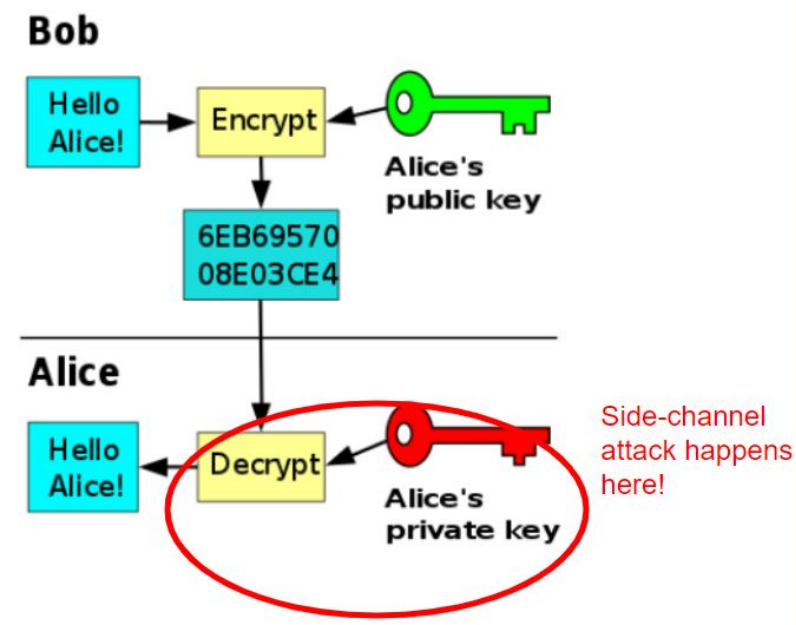
ciphertext

Side-channel attack happens here!

Our enemy wants to get the private key!

https://en.wikipedia.org/wiki/Public-key_cryptography

Side-channel attack



Two situations when the private key is most likely to be revealed:

Situation 1. the private key is being loaded from cache to register. In this situation, the BUS can radiate power trace outwards

Situation 2. the private key is being involved in calculation. In this situation, the private key is added or multiplied bit by bit with a number.The “bit by bit” addition or multiplication operations will radiate different power trace outwards

Power trace analysis – one of the side-channel attack methods

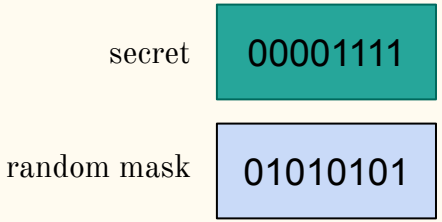
https://en.wikipedia.org/wiki/Public-key_cryptography

What is masking?

a simplest example:

What is masking?

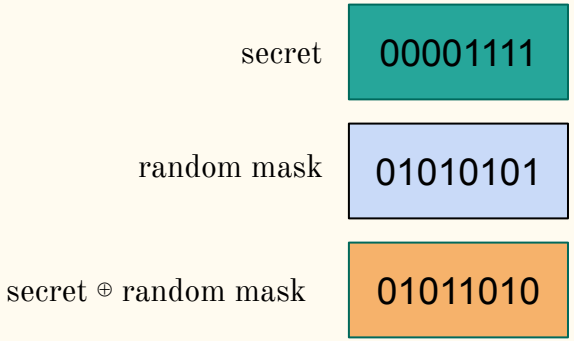
a simplest example:



Step 1: generate a random 0-1 string which has the same length as the secret. The string is called as “random mask”. e.g. 01010101

What is masking?

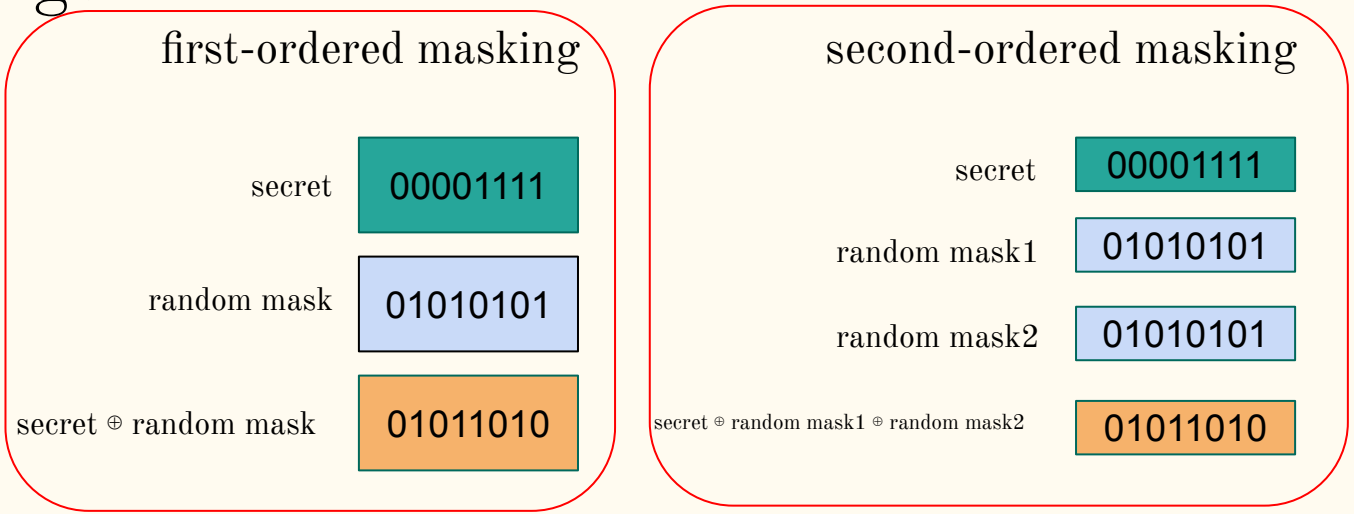
a simplest example:



Step 1: generate a random 0-1 string which has the same length as the secret. The string is called as “random mask”. e.g. 01010101

Step 2: XOR the secret 00001111 and the random mask 01010101, we obtain the masked secret: 00001111⊕01010101 = 01011010

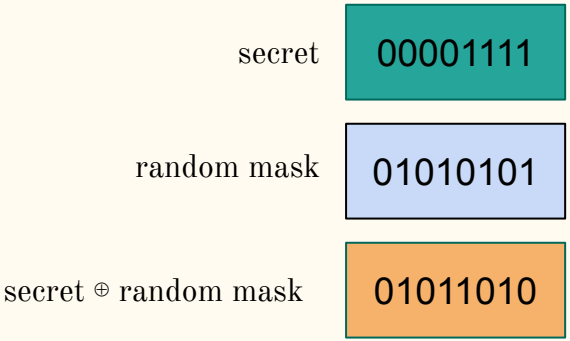
What is masking?



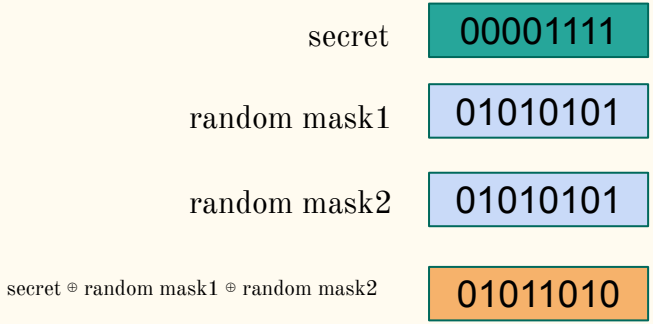
Step 1: generate a random 0-1 string which has the same length as the secret. The string is called as “random mask”. e.g. 01010101

Step 2: XOR the secret 00001111 and the random mask 01010101, we obtain the masked secret: 00001111⊕01010101 = 01011010

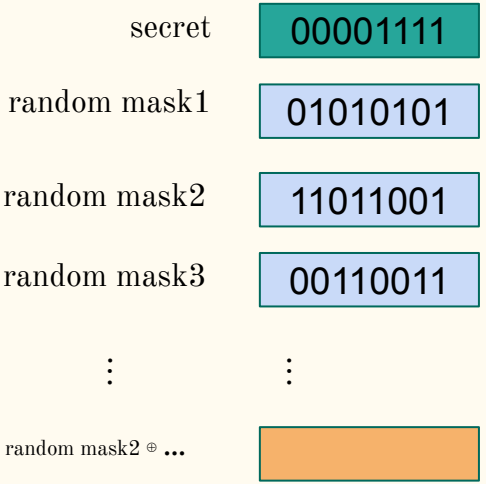
first-ordered masking



second-ordered masking



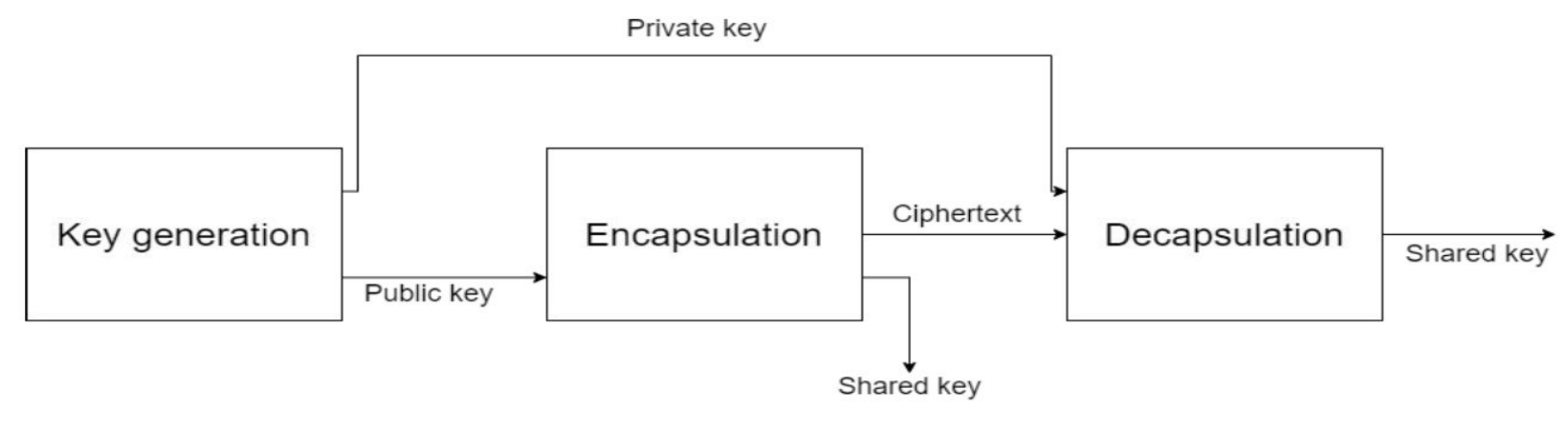
higher-ordered masking



Public Key Crypto

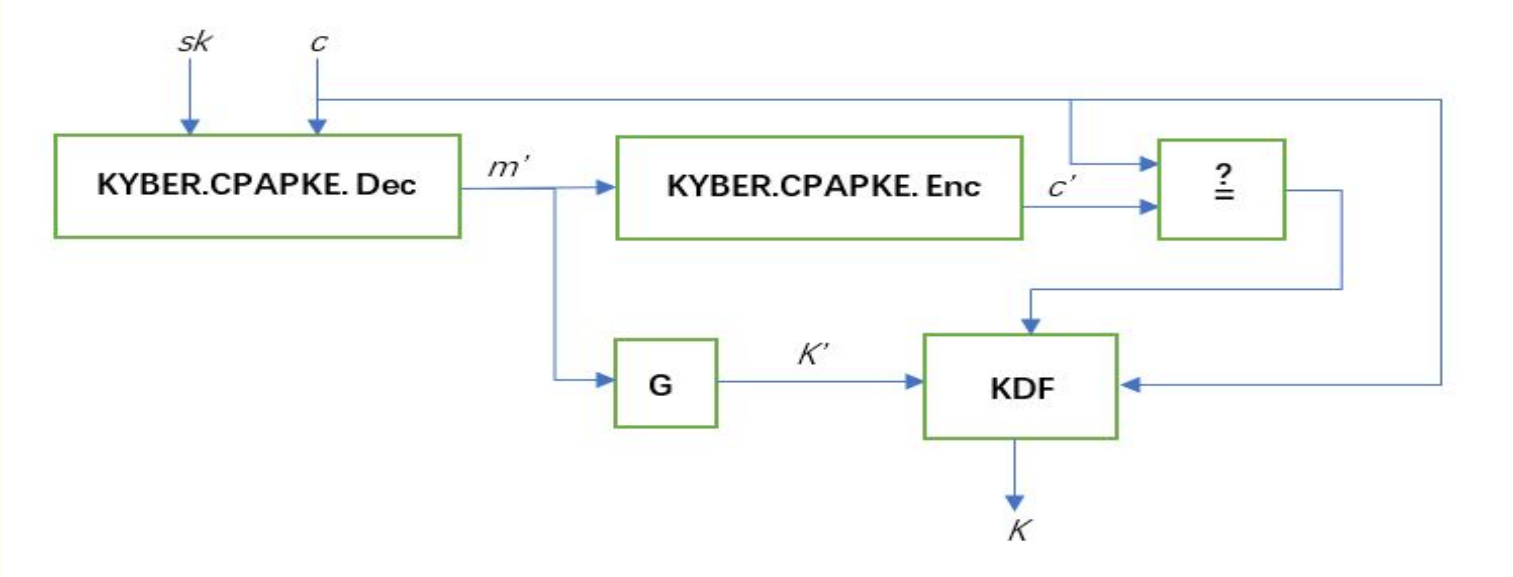
- Kyber: Lattice-based LWE round 3 finalist KEM of NIST PQC project
- PQC, alternative of PKC
- PKC(RSA) in classic computer
- PKC is broken easier by large integer factorization mentioned above
- Stronger countermeasure, higher order masking, for Kyber is needed
- Bit vs Qubit, Bloch sphere
- Shor’s algorithm
- Factorization brings collapse of conventional PKC

2 Overview of the implementation of Kyber



Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020.

KYBER.Decapsulation



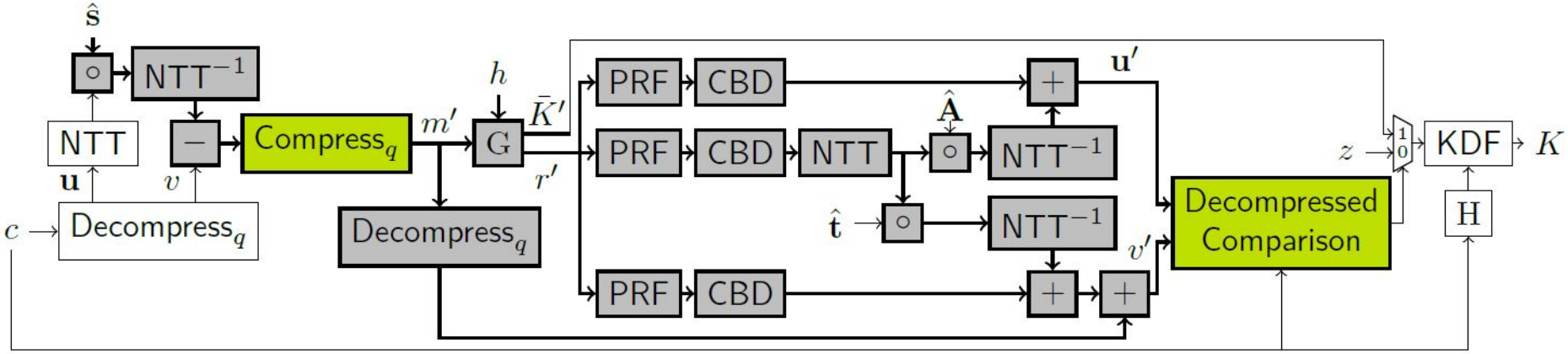
3 Masking Kyber decapsulation

- Second order arithmetic masking
- Prime modulus ($q = 3329$)
- X = secret message
- $X1$, $X2$, $X3$ = shares
- $X1$, $X2$ = random values in range $\{0, q-1\}$
- $X3 = (X - X1 - X2) \bmod (q)$

$$X \equiv (X1 + X2 + X3) \bmod (q)$$

[1] Joppe W. Bos, Marc Gourjon, Joost Renes, Tobias Schneider and Christine van Vredendaal. “Masking Kyber: First- and Higher-Order Implementations”, IACR Transactions on Cryptographic Hardware and Embedded Systems, Ruhr-Universität Bochum, 2021, Issue 4, pp. 173-214.

Kyber Decapsulation



s = secret key c = ciphertext K = session key (output)

Parts that require masking:

[1] Joppe W. Bos, Marc Gourjon, Joost Renes, Tobias Schneider and Christine van Vredendaal. "Masking Kyber: First- and Higher-Order Implementations", IACR Transactions on Cryptographic Hardware and Embedded Systems, Ruhr-Universität Bochum, 2021, Issue 4, pp. 173-214.

Details about masked parts

- Polynomial operation
- G and PRF
- Compress
- Comparison
- Decompress
- CBD

Symmetric Primitives: G and PRF

- G(SHA3-512), PRF(SHAKE256): KECCAK
- Linear mapping $\lambda = \pi(\rho(\theta))$ followed by nonlinear χ:
  - $x_i \leftarrow x_i + (x_{i+1} + 1)x_i + 2$
- Second-order masking of χ [1]:
  - $a_i \leftarrow b_i + (b_{i+1} + 1)b_{i+2} + b_{i+1}c_{i+2} + c_{i+1}b_{i+2}$
  - $b_i \leftarrow c_i + (c_{i+1} + 1)c_{i+2} + c_{i+1}a_{i+2} + a_{i+1}c_{i+2}$
  - $c_i \leftarrow a_i + (a_{i+1} + 1)a_{i+2} + a_{i+1}b_{i+2} + b_{i+1}a_{i+2}$

[1] Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche. “Building power analysis resistant implementations of Keccak”, Round 3 finalist of the Cryptographic Hash Algorithm Competition of NIST, 2010

Compress

- $\text{Compress}_q(x,d)$ takes $x \in Z_q$ and outputs an integer of d-bit
- For $d = 1$,

$$\text{Compress}_q(x,1) = \lceil (2/q) \cdot x \rfloor \bmod 2 = \begin{cases} 1 & \text{if } \frac{q}{4} < x < \frac{3q}{4}, \\ 0 & \text{otherwise.} \end{cases}$$

- equivalent to [1]

$$\text{Compress}_q^{s}(x) := \begin{cases} 0 & \text{if } x + \lfloor \frac{q}{4} \rfloor < \frac{q}{2}, \\ 1 & \text{otherwise.} \end{cases}$$

- $\lfloor \frac{q}{2} \rfloor = 1664 = (0110\ 1000\ 0000)_{\text{bin}}$,

$$\text{Compress}_q^{s}(x) = \overline{x}_{11} \oplus (\neg \overline{x}_{11} \cdot \overline{x}_{10} \cdot \overline{x}_9 \cdot (\overline{x}_8 \oplus (\neg \overline{x}_8 \cdot \overline{x}_7))). \quad (1)$$

- where $\overline{x} = x + \lfloor \frac{q}{4} \rfloor$

[1] Joppe W. Bos, Marc Gourjon, Joost Renes, Tobias Schneider and Christine van Vredendaal. “Masking Kyber: First- and Higher-Order Implementations”, IACR Transactions on Cryptographic Hardware and Embedded Systems, Ruhr-Universität Bochum, 2021, Issue 4, pp. 173-214.

Comparison

- Kyber: compare ciphertext C with *encrypt(decrypt(C))*
  - Compression for 3 shares is time-consuming
- "Masking Kyber: First- and Higher-Order Implementation" [1]:
  - Decompressed comparison:
    - Compare *decompress(C)* with $encrypt_{without\_compress}(decrypt(C))$
  - Requirement:
    - Sum of three shares $\in Z_q$
    - A2B with negative numbers
  - Needs further study

[1] Joppe W. Bos, Marc Gourjon, Joost Renes, Tobias Schneider and Christine van Vredendaal. "Masking Kyber: First- and Higher-Order Implementations", IACR Transactions on Cryptographic Hardware and Embedded Systems, Ruhr-Universität Bochum, 2021, Issue 4, pp. 173-214.

Decompress

$$x' = \mathsf{Decompress}_q\big(\mathsf{Compress}_q(x, d), d\big)$$

- Needs to satisfy the condition:

$$|x' - x \bmod^{\pm} q| \leq B_q := \left\lceil \frac{q}{2^{d+1}} \right\rfloor$$

- Unmasked, this is satisfied by:

$$\mathsf{Decompress}_q(x, d) = \lceil (q/2^d) \cdot x \rfloor$$

- Masked version is only needed for $\mathrm{d} = 1$

Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020.

Decompress

- Binary to arithmetic conversion
  - From binary shares: $m = m1, m2, m3 \in \{0, 1\}$
  - To arithmetic shares: $A1, A2, A3 \in \{0, q\text{-}1\}$
  - Where: $(A1 + A2 + A3) \bmod (q) = m1 \oplus m2 \oplus m3 \in \{0, 1\}$

$$Decompress_q(m, 1) = \begin{cases} (q+1)/2 & \text{if } m1 \oplus m2 \oplus m3 = 1 \\ 0 & \text{if } m1 \oplus m2 \oplus m3 = 0 \end{cases}$$

CBD

- Centered Binomial Distribution (CBD)
- Sample noise
- Random bytes ⟶ Coefficients ~ CBD
- Example input: 00011101
- Sum input bits ($a_0 = 1$, $b_0 = 2$, $a_1 = 1$, $b_1 = 0$)
- Calculate $f_i = a_i - b_i$ ($f_0 = -1$, $f_1 = 1$)
- Uses bitwise operations (XOR, AND)

4 Results

- Preliminary implementation of second-order masked Kyber
  - Standalone parts work well
  - All parts not integrated yet

5 Future work

Future work:

- Implement sampler using secure bitwise operations
- Implement comparison as described in [1]
- Optimization

[1] Joppe W. Bos, Marc Gourjon, Joost Renes, Tobias Schneider and Christine van Vredendaal. “Masking Kyber: First- and Higher-Order Implementations”, IACR Transactions on Cryptographic Hardware and Embedded Systems, Ruhr-Universität Bochum, 2021, Issue 4, pp. 173-214.

6 Lessons learned

We chose to work agile

- Create backlog, prioritization, selecting tasks for sprint, etc
- Get a good overview of project as early as possible
- Feedback from supervisors

Questions?