Vulnerable PQC against Side Channel Analysis - A Case Study on Kyber

Haocheng Ma[1], Shijian Pan[1], Ya Gao[1], Jiaji He[1], Yiqiang Zhao[1] and Yier Jin[2]

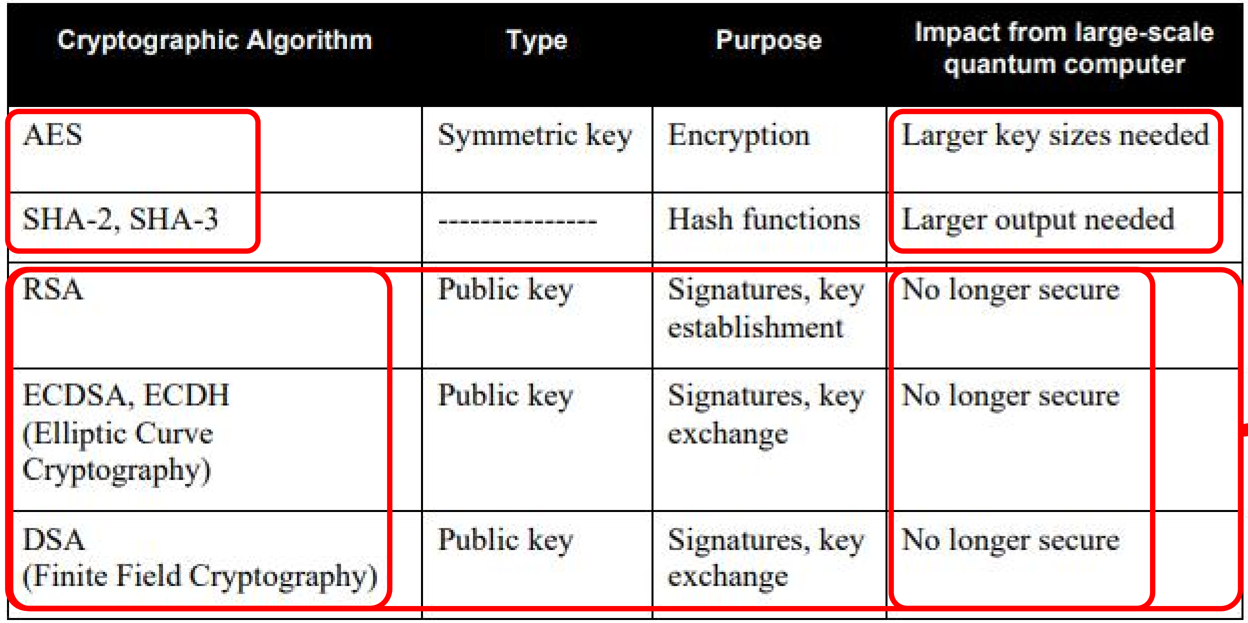
[1]Tianjin University, [2]University of Florida

Outline

- Motivation
- Security Analysis
  - Kyber Decryption
  - Vulnerable Regions
- Case Studies
  - Hardware Architectures
  - Experimental Results
- Conclusions

Motivation

- Impact of Quantum computing on common cryptographic algorithms
  - Symmetric cryptography needs larger key sizes
  - Public key cryptography (PKC) is no longer secure
  - Next-generation PKC, i.e., post-quantum cryptography (PQC)



Cryptographic Algorithm	Type	Purpose	Impact from large-scale quantum computer
AES	Symmetric key	Encryption	Larger key sizes needed
SHA-2, SHA-3	---------------	Hash functions	Larger output needed
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Public key	Signatures, key exchange	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key exchange	No longer secure

Security impact from large-scale quantum computer

I we

c

(Gro

Can be le

quantu])

New public key cryptosystems

secure against classical computers

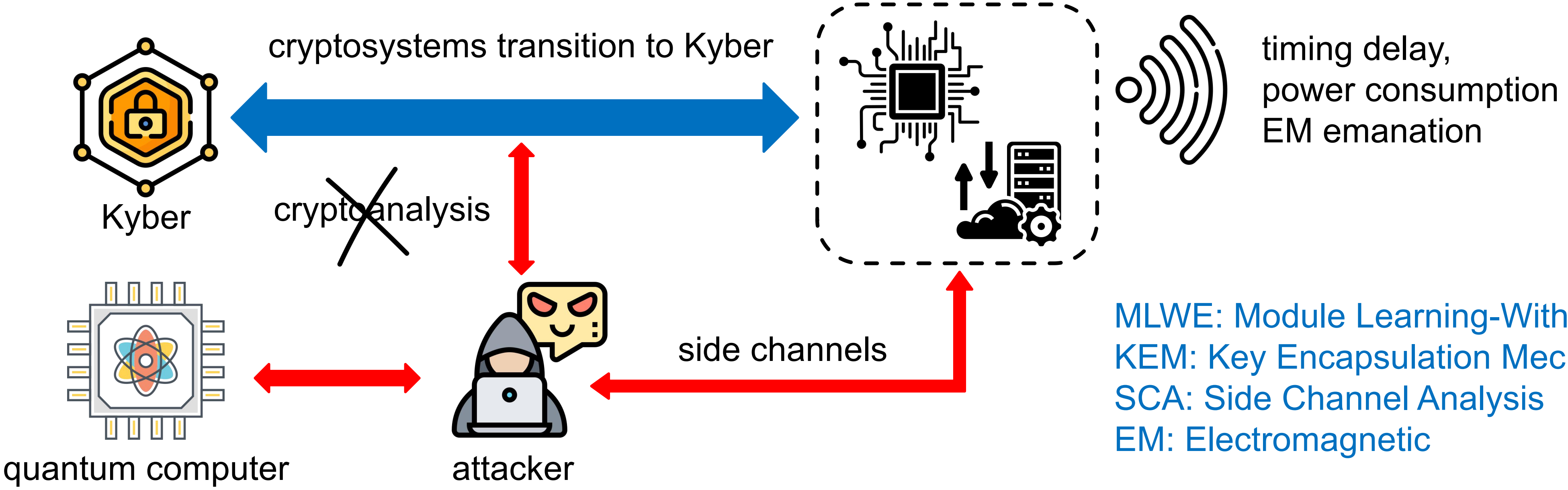
secure against quantum computers

[1] Grover, et al. A fast quantum mechanical algorithm for database search, 1996.

[2] Shor, et al. Algorithms for quantum computation: discrete logarithms and factoring, 1994.

Motivation

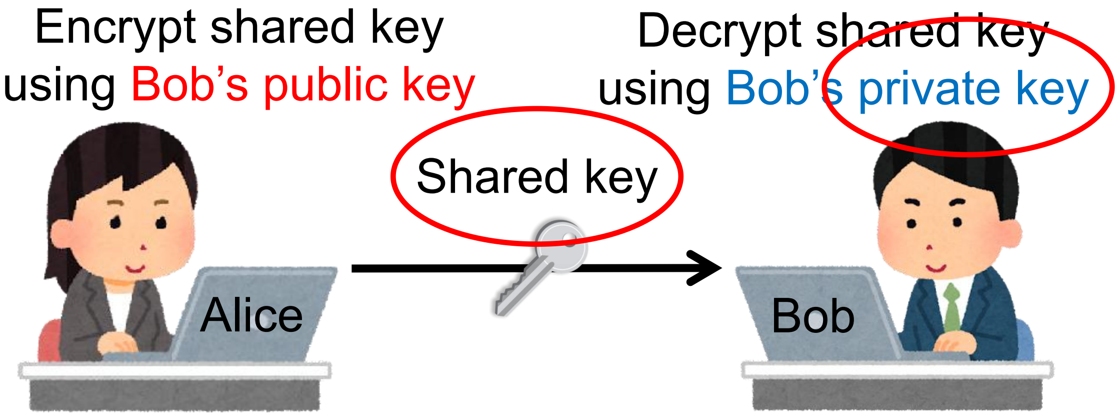
- CRYSTALS-Kyber
  - A KEM that stands out from the NIST standardization project
  - Strong security based on the MLWE problem
  - Excellent performance on most applications
  - Security against SCA attacks?



MLWE: Module Learning-With-Errors
KEM: Key Encapsulation Mechanism
SCA: Side Channel Analysis
EM: Electromagnetic

Motivation

- SCA on Kyber is under intensive research
  - Key recovery - long-term private key
  - Message recovery - ephemeral session key
  - Most of recent works focus on software implementations
  - Security evaluation of Kyber hardware with different architectures



Encrypt shared key using Bob's public key

Decrypt shared key using Bob's private key

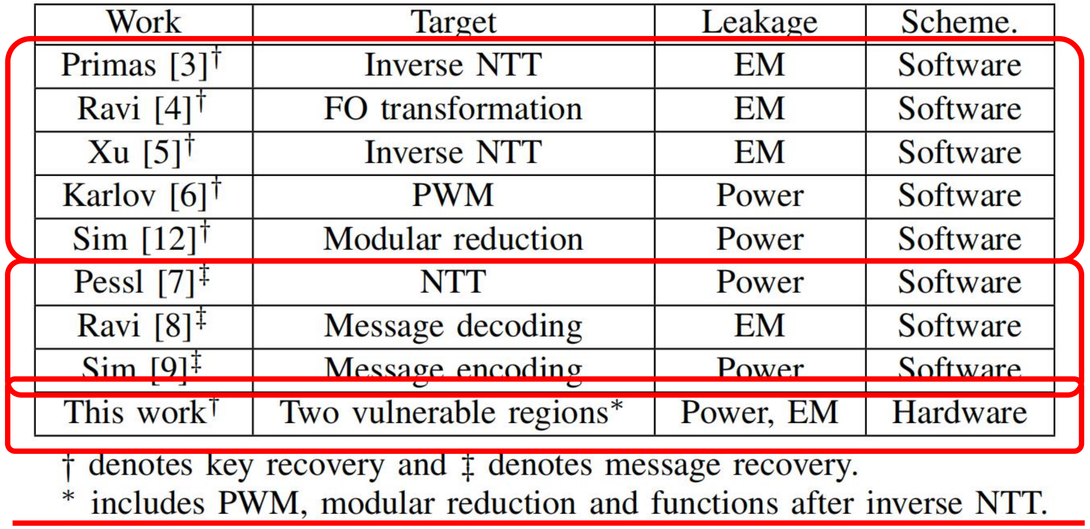
Shared key

Alice

Bob

Kyber ensures the security of the key exchange using the public and private key pair

TABLE I: Comparison with existing works.



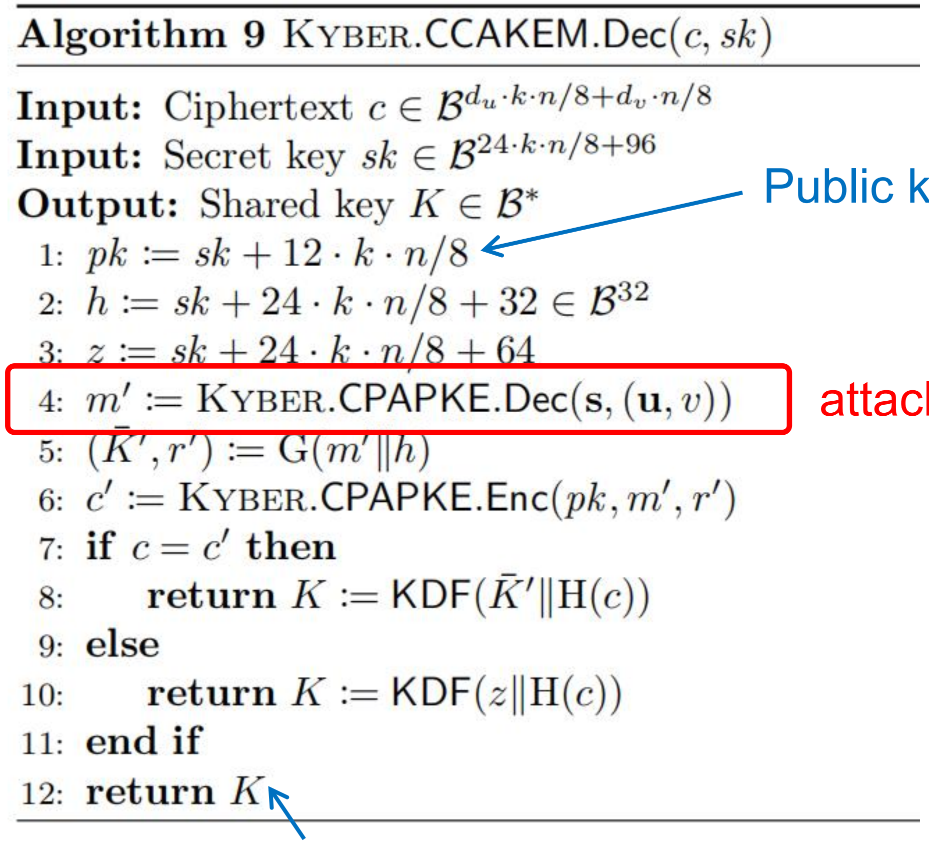
Work	Target	Leakage	Scheme.
Primas [3]†	Inverse NTT	EM	Software
Ravi [4]†	FO transformation	EM	Software
Xu [5]†	Inverse NTT	EM	Software
Karlov [6]†	PWM	Power	Software
Sim [12]†	Modular reduction	Power	Software
Pessl [7]‡	NTT	Power	Software
Ravi [8]‡	Message decoding	EM	Software
Sim [9]‡	Message encoding	Power	Software
This work†	Two vulnerable regions*	Power, EM	Hardware

† denotes key recovery and ‡ denotes message recovery.

* includes PWM, modular reduction and functions after inverse NTT.

Security Analysis

- Kyber decryption



```
Algorithm 9 KYBER.CCAKEM.Dec$(c, sk)$
Input: Ciphertext $c \in \mathcal{B}^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$
Input: Secret key $sk \in \mathcal{B}^{24 \cdot k \cdot n/8 + 96}$
Output: Shared key $K \in \mathcal{B}^*$
$pk := sk + 12 \cdot k \cdot n/8$
$h := sk + 24 \cdot k \cdot n/8 + 32 \in \mathcal{B}^{32}$
$z := sk + 24 \cdot k \cdot n/8 + 64$
$m' := \text{KYBER.CPAPKE.Dec}(\mathbf{s}, (\mathbf{u}, v))$
$(\bar{K}', r') := \text{G}(m' \| h)$
$c' := \text{KYBER.CPAPKE.Enc}(pk, m', r')$
if $c = c'$ then
    return $K := \text{KDF}(\bar{K}' \| \text{H}(c))$
else
    return $K := \text{KDF}(z \| \text{H}(c))$
end if
return $K$
```

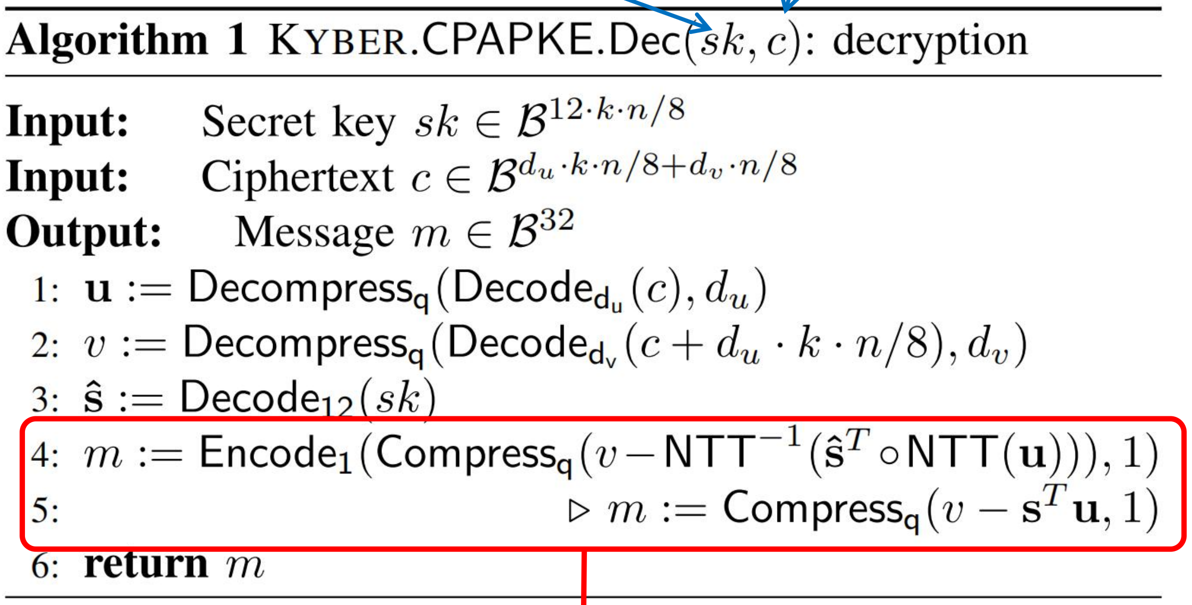
Public key

attack point

Session key

Long-term secret key

Ciphertext



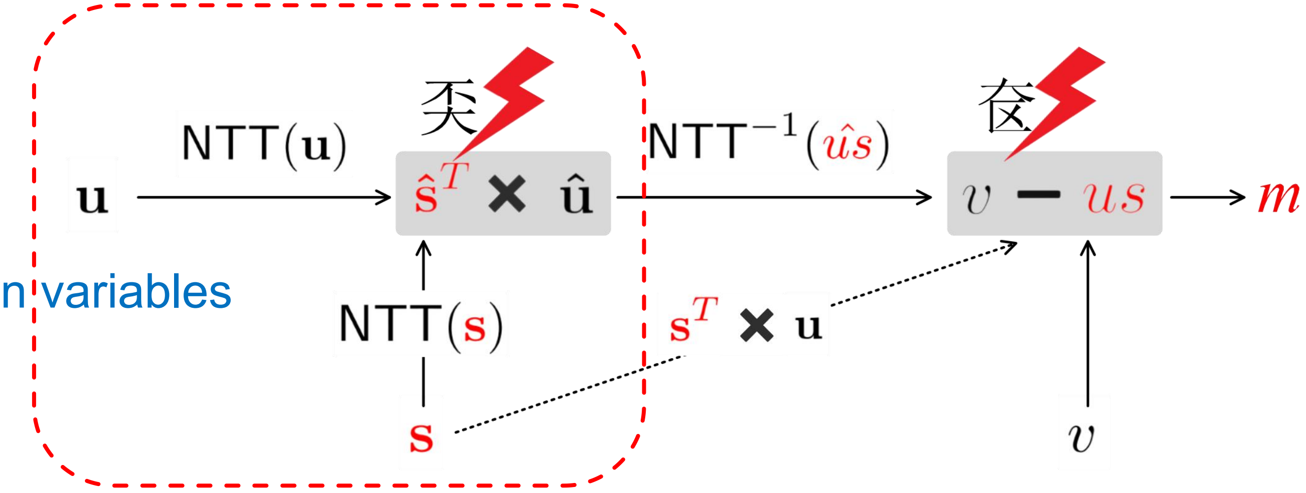
```
Algorithm 1 KYBER.CPAPKE.Dec$(sk, c)$: decryption
Input: Secret key $sk \in \mathcal{B}^{12 \cdot k \cdot n/8}$
Input: Ciphertext $c \in \mathcal{B}^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$
Output: Message $m \in \mathcal{B}^{32}$
$\mathbf{u} := \text{Decompress}_q(\text{Decode}_{d_u}(c), d_u)$
$v := \text{Decompress}_q(\text{Decode}_{d_v}(c + d_u \cdot k \cdot n/8), d_v)$
$\hat{\mathbf{s}} := \text{Decode}_{12}(sk)$
$m := \text{Encode}_1(\text{Compress}_q(v - \text{NTT}^{-1}(\hat{\mathbf{s}}^T \circ \text{NTT}(\mathbf{u}))), 1)$
                                        ▷ $m := \text{Compress}_q(v - \mathbf{s}^T \mathbf{u}, 1)$
return $m$
```

vulnerable regions of Kyber decryption

Security Analysis

- First vulnerable region - NTT domain
  - Point-wise multiplication (PWM) function, Modular reduction function
  - Perform CPA attacks with random ciphertext
  - Convert recovered polynomials into the time domain by inverse NTT function

first vulnerable region



known variables

secret key

The i-th PWM function and modular reduction

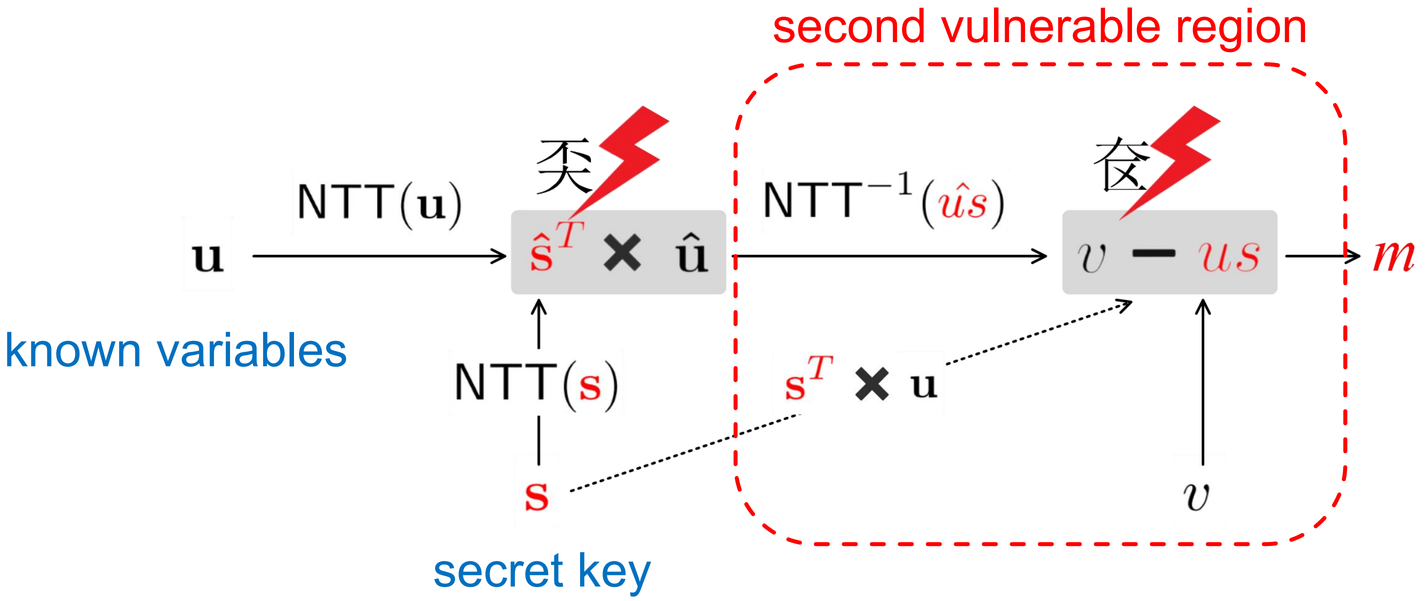
$$L(\text{point1}_i) = \alpha \times \mathrm{HD}((\hat{\mathbf{s}}^T \circ \hat{\mathbf{u}})[i]) + \mathcal{N}$$

$$L(\text{point2}_i) = \alpha \times \mathrm{HD}((\hat{\mathbf{s}}^T \circ \hat{\mathbf{u}} \bmod q)[i]) + \mathcal{N}$$

HD denotes the Hamming distance model
α denotes the scaling factor and
N represents a Gaussian noise term

Security Analysis

- Second vulnerable region - time domain
  - Inverse NTT function, subtractions and Compress function
  - Property of polynomial multiplication in Kyber
  - Manipulate the product result by chosen ciphertexts

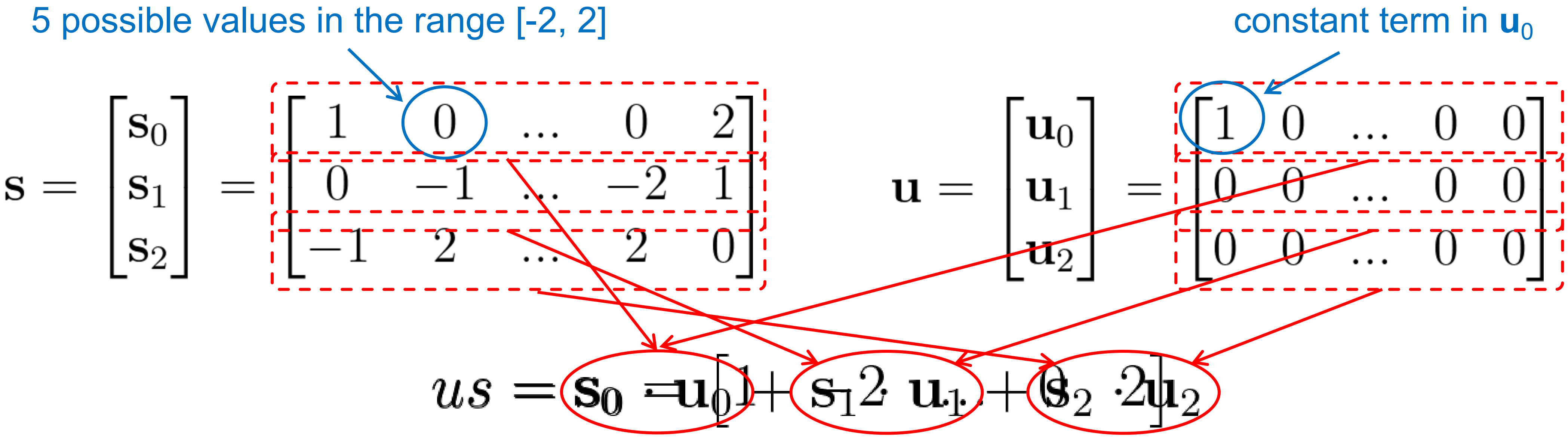

second vulnerable region
$\mathrm{NTT}(\mathbf{u})$
$\mathbf{u}$
$\hat{\mathbf{s}}^T \times \hat{\mathbf{u}}$
$\mathrm{NTT}^{-1}(\hat{us})$
$v - us$
m
known variables
$\mathrm{NTT}(\mathbf{s})$
$\mathbf{s}^T \times \mathbf{u}$
$\mathbf{s}$
v
secret key

$$\mathrm{NTT}^{-1}(\hat{\mathbf{s}}^T \circ \hat{\mathbf{u}}) := \mathbf{s}^T\mathbf{u} \bmod q$$

- the inverse NTT function brings the result back to the time domain
- the classical polynomial multiplication can obtain equal values

Security Analysis

- Second vulnerable region - time domain
  - Each sub-key has 5 possible values in the range [-2, 2]
  - Set the first element of **u** as constant, other elements are kept as 0
  - The product *us* is equal to multiplication between $\mathbf{s}_0$ and constant


5 possible values in the range [-2, 2]
constant term in $\mathbf{u}_0$

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 & 2 \\ 0 & -1 & \dots & -2 & 1 \\ -1 & 2 & \dots & 2 & 0 \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

$$us = \mathbf{s}_0 \mathbf{u}_0 + \mathbf{s}_1 \mathbf{u}_1 + \mathbf{s}_2 \mathbf{u}_2$$

Simple example of the chosen ciphertext

Security Analysis

- Second vulnerable region - time domain
  - Subsets U and V provide all possible coefficient values for **u** and v
  - 1024 possible chosen ciphertexts for the inverse NTT function
  - A larger space of chosen ciphertexts for subsequent functions (e.g., v - *us*)

1024 possible coefficient values for **u**

$$U = \{\lceil (3329/2^{10}) \cdot i \rfloor : i = 0, 1, \ldots, 1023\}$$

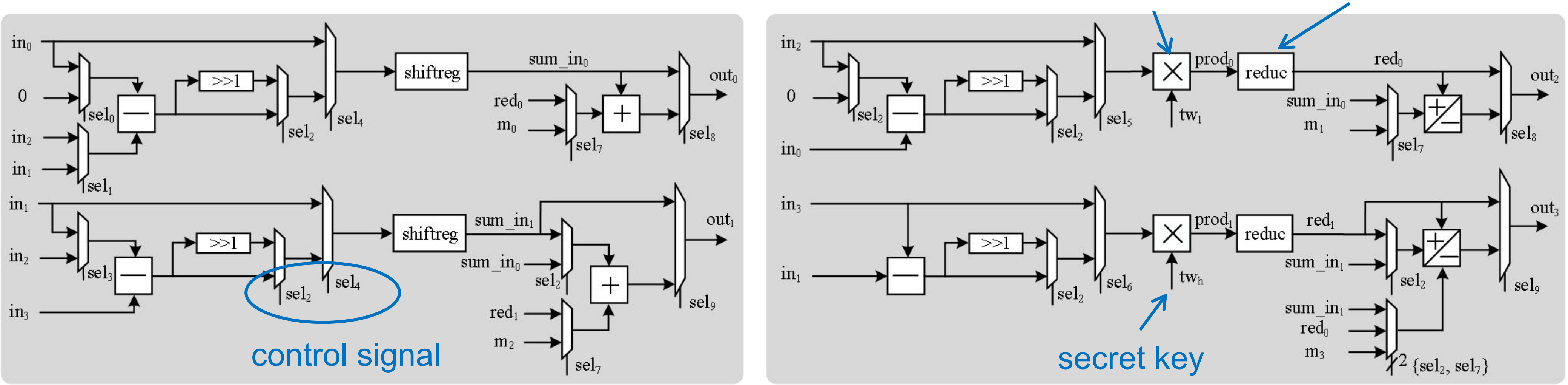
8 possible coefficient values for v

$$V = \{\lceil (3329/2^{4}) \cdot i \rfloor : i = 0, 1, \ldots, 7\}$$

Compression cause the data loss of restored **u** and v

Case Studies

- Direct implementation of Kyber
  - Realize and control most functions by two sets of butterfly units
  - Use various levels of parallel computations to achieve high performance
  - Retain operations that are less relevant for decryption



PWM

Modular reduction

control signal

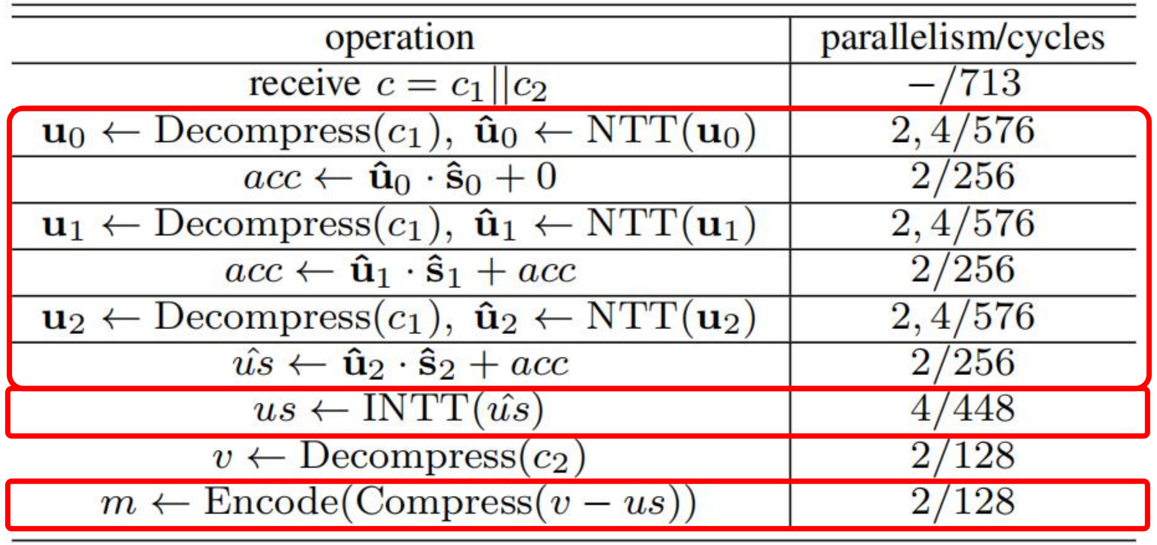
secret key

Structure of two sets of butterfly units, with two input data pairs and corresponding output pairs

Case Studies

- Direct implementation of Kyber
  - Realize and control most functions by two sets of butterfly units
  - Use various levels of parallel computations to achieve high performance
  - Retain operations that are less relevant for decryption

TABLE II: Detailed operations in Kyber-HDL design [10].

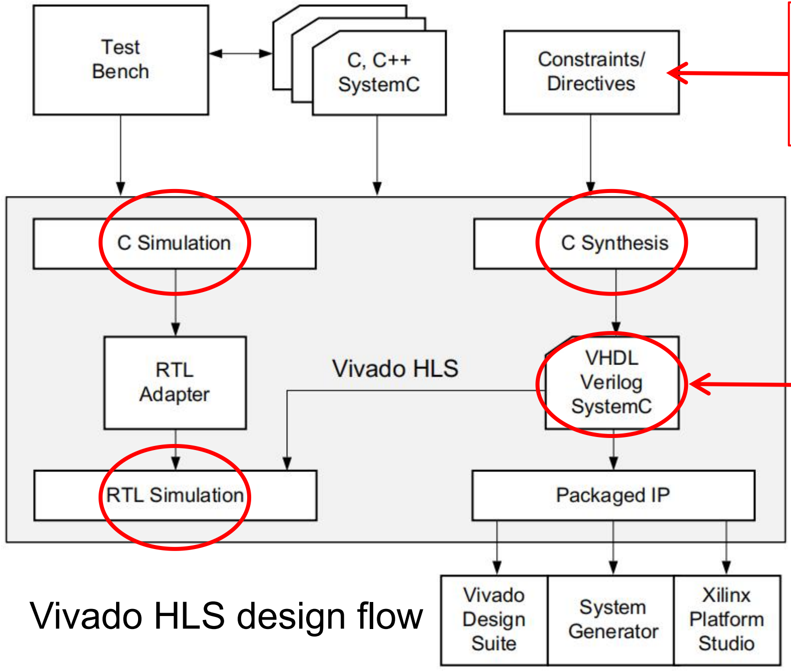


operation	parallelism/cycles
receive $c = c_1 \| c_2$	$-/713$
$\mathbf{u}_0 \leftarrow \text{Decompress}(c_1),\ \hat{\mathbf{u}}_0 \leftarrow \text{NTT}(\mathbf{u}_0)$	$2, 4/576$
$acc \leftarrow \hat{\mathbf{u}}_0 \cdot \hat{\mathbf{s}}_0 + 0$	$2/256$
$\mathbf{u}_1 \leftarrow \text{Decompress}(c_1),\ \hat{\mathbf{u}}_1 \leftarrow \text{NTT}(\mathbf{u}_1)$	$2, 4/576$
$acc \leftarrow \hat{\mathbf{u}}_1 \cdot \hat{\mathbf{s}}_1 + acc$	$2/256$
$\mathbf{u}_2 \leftarrow \text{Decompress}(c_1),\ \hat{\mathbf{u}}_2 \leftarrow \text{NTT}(\mathbf{u}_2)$	$2, 4/576$
$\hat{us} \leftarrow \hat{\mathbf{u}}_2 \cdot \hat{\mathbf{s}}_2 + acc$	$2/256$
$us \leftarrow \text{INTT}(\hat{us})$	$4/448$
$v \leftarrow \text{Decompress}(c_2)$	$2/128$
$m \leftarrow \text{Encode}(\text{Compress}(v - us))$	$2/128$

- Multiplications are computed in parallelism level of 2 by two multipliers
- Modular reduction is applied to these products in a pipelined manner
- Inverse NTT function transforms 4 elements of products in parallel
- Following functions have degrees of parallelism 2

Case Studies

- Kyber Implementation through HLS
  - C simulation, C synthesis and RTL simulation
  - Combinations of directives provide relative optimum speed and cost trade-offs
  - The degree of parallelism ranges from 1 to 3 during PWM executions



Vivado HLS design flow

- set the initiation interval of the pipeline as 8
- limit the instance of multiplication to less than 4

- re
- re
  st
- Th
  0

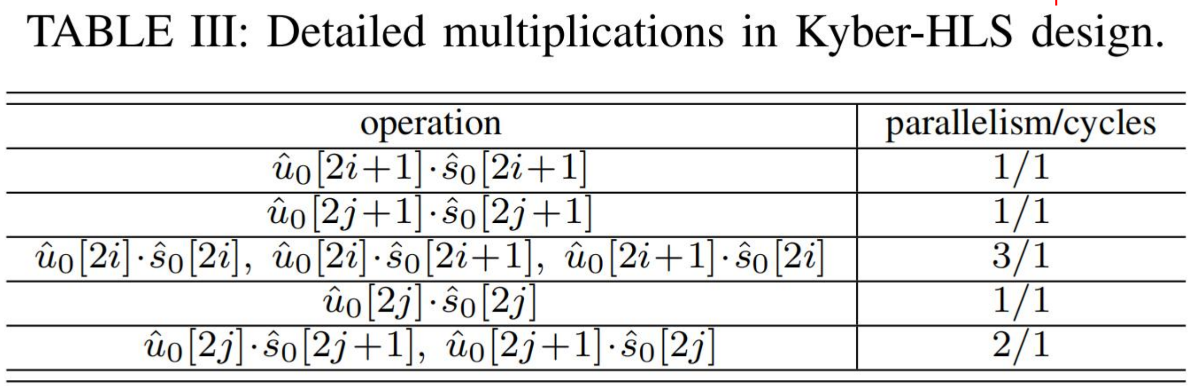


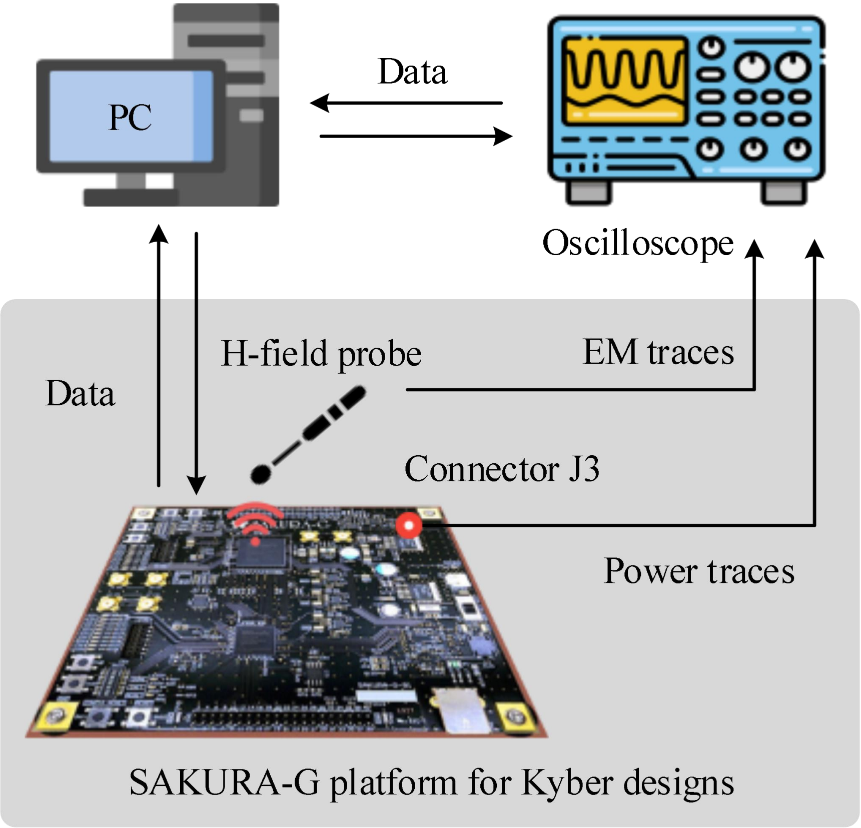
TABLE III: Detailed multiplications in Kyber-HLS design.

operation	parallelism/cycles
$\hat{u}_0[2i+1]\cdot\hat{s}_0[2i+1]$	1/1
$\hat{u}_0[2j+1]\cdot\hat{s}_0[2j+1]$	1/1
$\hat{u}_0[2i]\cdot\hat{s}_0[2i]$, $\hat{u}_0[2i]\cdot\hat{s}_0[2i+1]$, $\hat{u}_0[2i+1]\cdot\hat{s}_0[2i]$	3/1
$\hat{u}_0[2j]\cdot\hat{s}_0[2j]$	1/1
$\hat{u}_0[2j]\cdot\hat{s}_0[2j+1]$, $\hat{u}_0[2j+1]\cdot\hat{s}_0[2j]$	2/1

Multiplications between 4 elements occupy 5 clock cycles

Case Studies

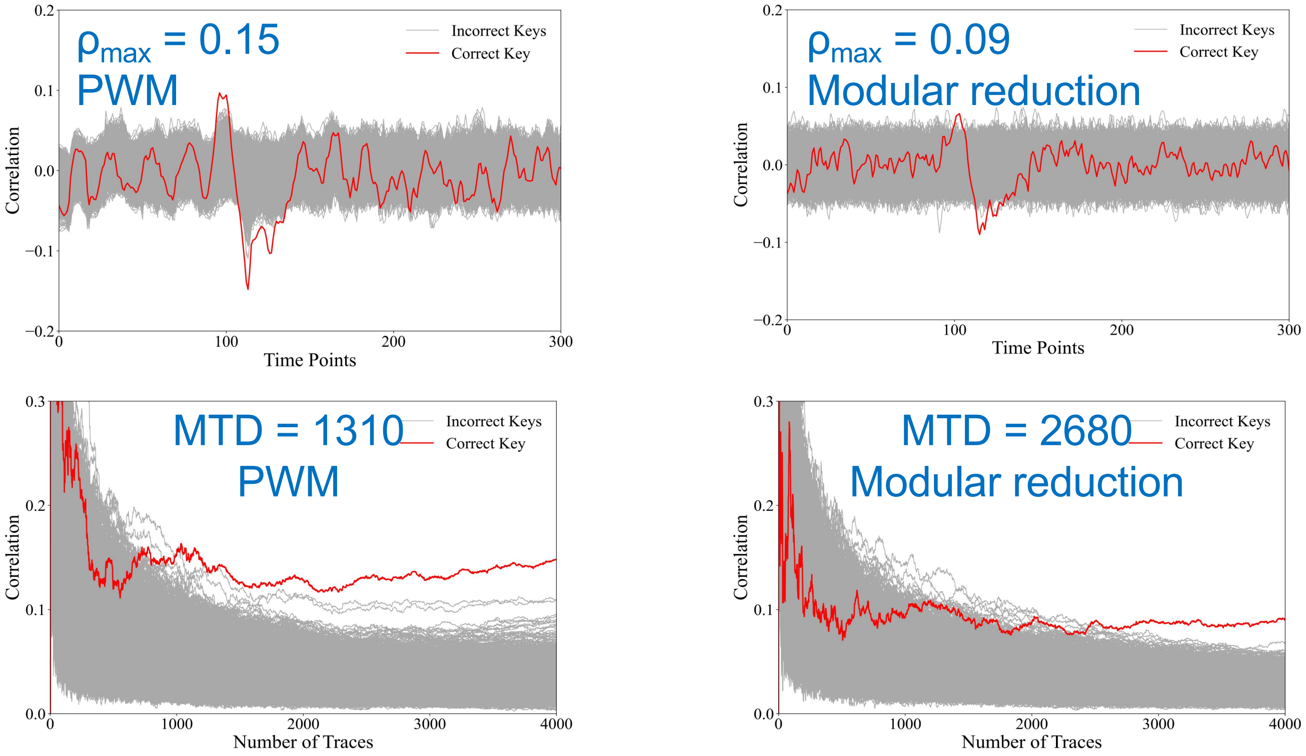
- Experimental setup
  - Target platform: SAKURA-G board
    - Spartan-6 XC6SLX75 FPGA
    - Spartan-6 XC6SLX9 FPGA
    - 20 MHz Clock frequency
  - Traces measurement
    - Langer RF-U 5-2 probe → EM
    - SMA connector J3 → Power
    - 2.5 GSa/s sampling rate
    - 500 MHz bandwidth



The overview of the experiment setup

Case Studies

- Result on Kyber-HDL
  - First vulnerable region - PWM and modular reduction functions
  - Recover the correct subkey within 1310 ~ 2680 traces

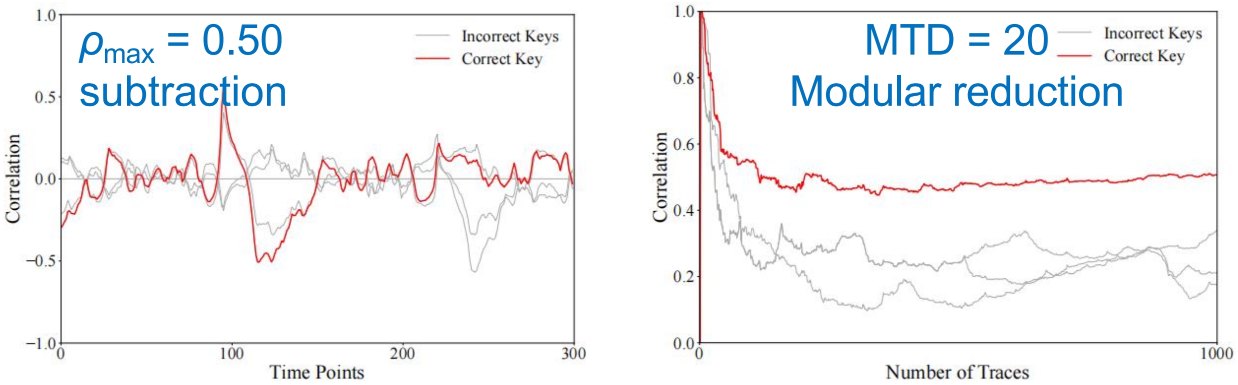


Attack results of Kyber-HDL in the first vulnerable region

The correlation peak at modular reduction is relatively small. This is because the residue is the truncation of the product output, which decreases trace discrepancy caused by data transitions.

Case Studies

- Result on Kyber-HDL
  - Second vulnerable region - functions after inverse NTT function
  - Inverse NTT function is secure with parallelism level of 4
  - A full key recovery from subtraction needs around 60 traces



$\rho_{\max}$ = 0.50
subtraction

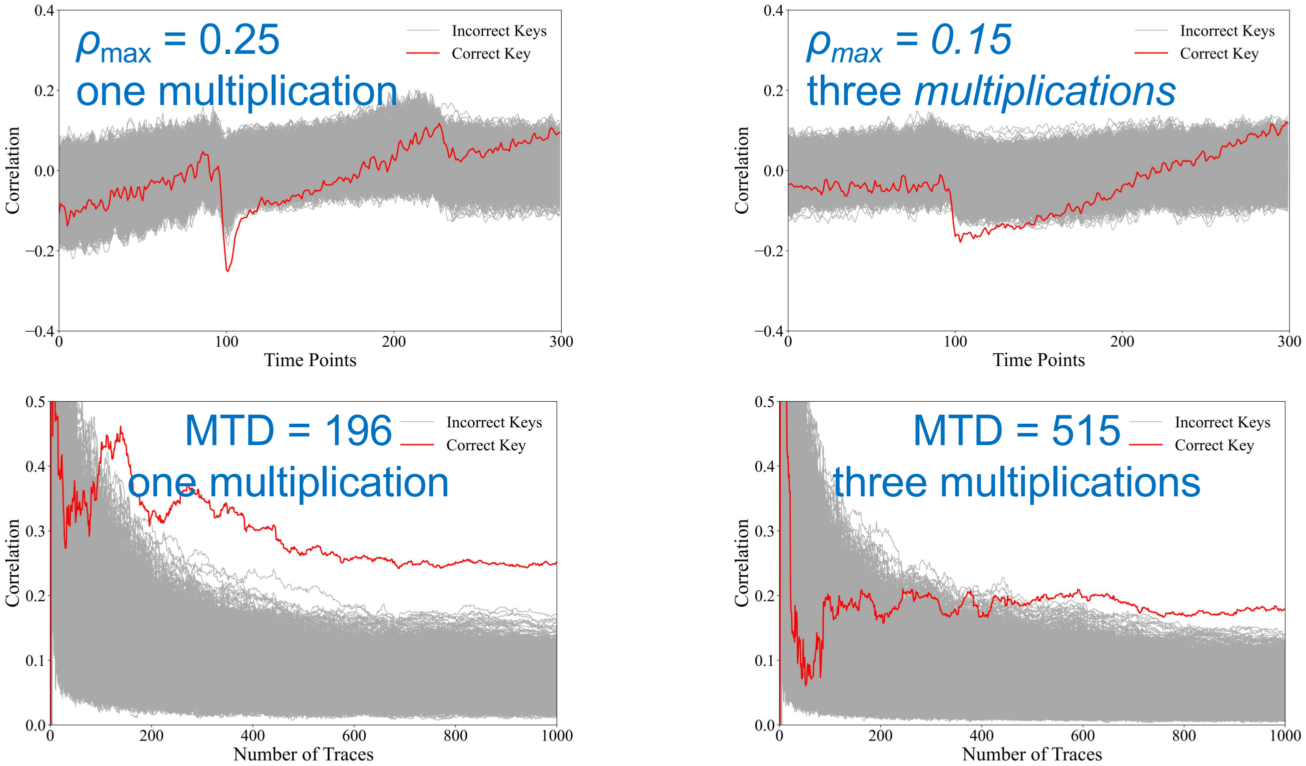
MTD = 20
Modular reduction

Attack results of Kyber-HDL in the second vulnerable region

When the degree of parallelism keeps constant, Kyber-HDL is more vulnerable in the time domain relative to the NTT domain. This is because the property of polynomial multiplication reduces the search space of the secret key.

Case Studies

- Result on Kyber-HLS
  - First vulnerable region - PWM function
  - Recover the correct subkey within 196 ~ 515 traces



Attack results of Kyber-HLS in the first vulnerable region

The security of Kyber-HLS is comparatively low with Kyber-HDL. The principal reason is function pipelining and other irrelevant operations of Kyber-HDL, which obfuscate the data flow of PWM to some extent.

As parallel architectures result in lower SNR of leakage, Kyber-HLS with higher parallelism has increased security than the lower degree of parallelism.

Conclusions

- This paper evaluates the side-channel security of Kyber's hardware implementations with different architectures.
- We make a comprehensive analysis of their decryption procedure, including two vulnerable regions and multiple points of interest.
- Although hardware designs improve security, they still leak sufficient information for SCA attacks and call for countermeasures about the NTT domain and time domain.

Thank You!

Any Questions?