SCIENCE
PASSION
TECHNOLOGY

Efficient SPA Countermeasures using Redundant Number Representation with Application to ML-KEM

Rishub Nagpal[1] Vedad Hadžić[2] Robert Primas[2] Stefan Mangard[1]

[1] Graz University of Technology | [2] Intel Labs

SAC 2025

› isec.tugraz.at

Problem Statement and Contributions

- **Implementation security of PQC against worst-case side-channel attacks such as SPA and SASCA**
- **Analyze Redundant Number Representation (RNR) as a countermeasure against SPA**
  - Mutual Information Analysis of RNR for arbitrary integer ring sizes.
  - Application of RNR to ML-KEM resulting in 62.8% overhead for the NTT and 0% overhead for the INTT.
  - Demonstrate countermeasure effectiveness in both against the strongest known SPA attack on ML-KEM.

Problem Statement and Contributions

- **Implementation security of PQC against worst-case side-channel attacks such as SPA and SASCA**
- **Analyze Redundant Number Representation (RNR) as a countermeasure against SPA**
  - **Mutual Information Analysis** of RNR for arbitrary integer ring sizes.
  - Application of RNR to ML-KEM resulting in **62.8% overhead for the NTT** and **0% overhead for the INTT**.
  - **Demonstrate countermeasure effectiveness** in both against the strongest known SPA attack on ML-KEM.

Problem Statement and Contributions

- Implementation security of PQC against worst-case side-channel attacks such as SPA and SASCA
- Analyze Redundant Number Representation (RNR) as a countermeasure against SPA
  - **Mutual Information Analysis** of RNR for arbitrary integer ring sizes.
  - Application of RNR to ML-KEM resulting in **62.8% overhead for the NTT and 0% overhead for the INTT.**
  - **Demonstrate countermeasure effectiveness** in both against the strongest known SPA attack on ML-KEM.

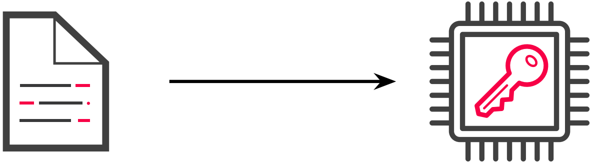
Problem Statement and Contributions

- Implementation security of PQC against worst-case side-channel attacks such as SPA and SASCA
- Analyze Redundant Number Representation (RNR) as a countermeasure against SPA
  - **Mutual Information Analysis** of RNR for arbitrary integer ring sizes.
  - Application of RNR to ML-KEM resulting in **62.8% overhead for the NTT** and **0% overhead for the INTT**.
  - **Demonstrate countermeasure effectiveness** in both against the strongest known SPA attack on ML-KEM.

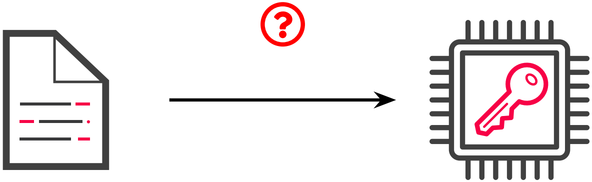
Problem Statement and Contributions

- **Implementation security of PQC against worst-case side-channel attacks such as SPA and SASCA**
- **Analyze Redundant Number Representation (RNR) as a countermeasure against SPA**
  - **Mutual Information Analysis** of RNR for arbitrary integer ring sizes.
  - Application of RNR to ML-KEM resulting in **62.8% overhead for the NTT** and **0% overhead for the INTT**.
  - **Demonstrate countermeasure effectiveness** in both against the strongest known SPA attack on ML-KEM.

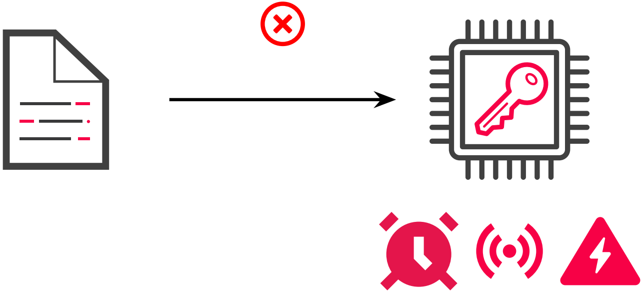
Motivation



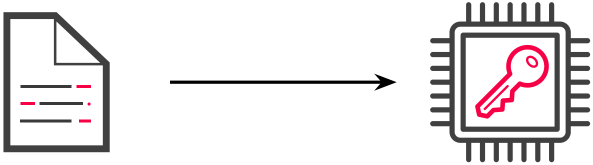
Motivation



Motivation



Motivation

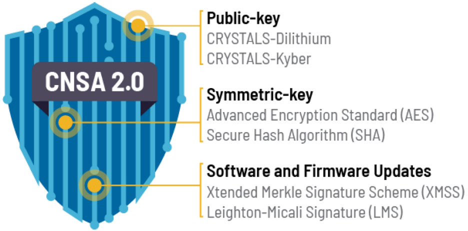


The Side-channel Problem

Cryptographic algorithms can be secure from a “black box” view, but insecure when implemented in the real-world due to physical effects.

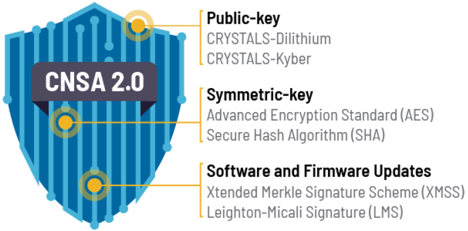
Motivation cont.

- Kyber and Dilithium are standardized by NIST as ML-KEM and ML-DSA.



Motivation cont.

- Kyber and Dilithium are standardized by NIST as ML-KEM and ML-DSA.



Side-channel attacks are still a problem despite quantum resistance...

Motivation cont.

- Chosen Ciphertext *k*-trace attack of Hamburg et al. [Ham+21]

Motivation cont.

- Chosen Ciphertext k-trace attack of Hamburg et al. [Ham+21]
  - One of the strongest known attacks on ML-KEM.

Motivation cont.

- Chosen Ciphertext *k*-trace attack of Hamburg et al. [Ham+21]
  - One of the strongest known attacks on ML-KEM.
  - Possible with only a few measurements via Soft-analytical Side-channel Analysis (SASCA).

Motivation cont.

- Chosen Ciphertext *k*-trace attack of Hamburg et al. [Ham+21]
  - One of the strongest known attacks on ML-KEM.
  - Possible with only a few measurements via Soft-analytical Side-channel Analysis (SASCA).
  - Even against CCA2-secure masked implementations…

k-trace attack of Hamburg et al. [Ham+21]

ML-KEM.PKE Decryption

Input: ciphertext $c = (c_1, c_2)$, $sk = \hat{s}$
Output: message $m \in \mathcal{R}_q$
1: $(u, v) = (\texttt{Decompress}(c_1), \texttt{Decompress}(c_2))$
2: **return** $m = v - \text{NTT}^{-1}(\hat{s}^{\mathsf{T}} \circ \text{NTT}(u))$

k-trace attack of Hamburg et al. [Ham+21]

ML-KEM.PKE Decryption

Input: ciphertext $c = (c_1, c_2)$, $sk = \hat{s}$
Output: message $m \in \mathcal{R}_q$
1: $(u, v) = (\texttt{Decompress}(c_1), \texttt{Decompress}(c_2))$
2: **return** $m = v - \mathrm{NTT}^{-1}(\underbrace{\hat{s}^{\mathrm{T}} \circ \mathrm{NTT}(u)}_{\text{sparse product}})$

k-trace attack of Hamburg et al. [Ham+21]

ML-KEM.PKE Decryption

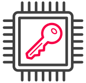
Input: ciphertext $c = (c_1, c_2), sk = \hat{s}$
Output: message $m \in \mathcal{R}_q$

1: $(u, v) = (\texttt{Decompress}(c_1), \texttt{Decompress}(c_2))$
2: **return** $m = v - \text{NTT}^{-1}(\underbrace{\hat{s}^{\mathsf{T}} \circ \text{NTT}(u)}_{\text{sparse product}})$

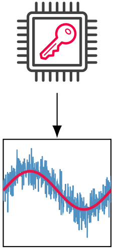
Chosen Ciphertext k-trace attack

Chosen ciphertexts enable divide-and-conquer recovery of $\hat{s}$ from the NTT^{-1} of the sparse product.

Background - Template Attacks

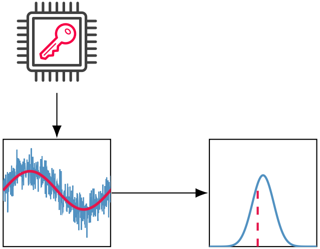


Background - Template Attacks



Noisy Measurement

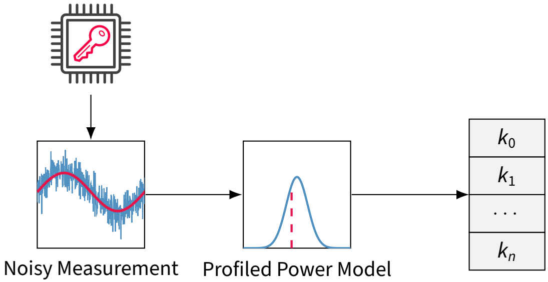
Background - Template Attacks



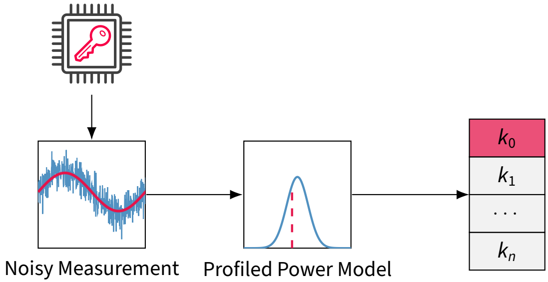
Noisy Measurement

Profiled Power Model

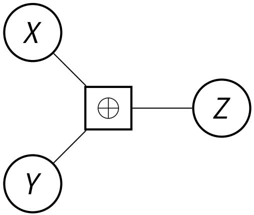
Background - Template Attacks



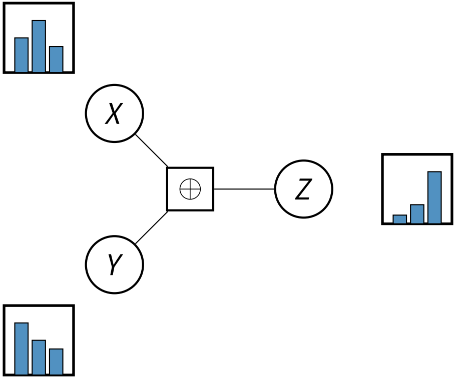
Background - Template Attacks



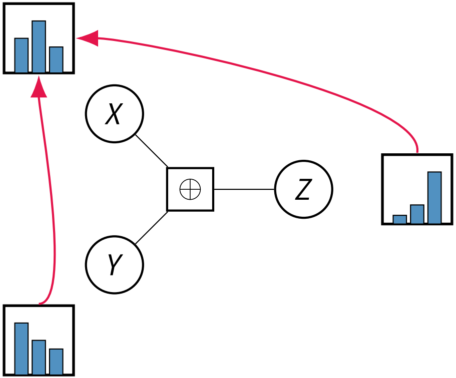
Background - SASCAs



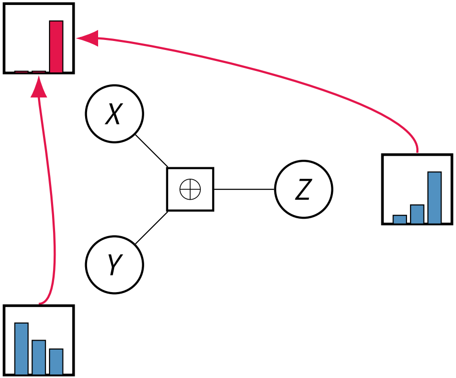
Background - SASCAs



Background - SASCAs



Background - SASCAs



Modeling the ML-KEM NTT^{-1}

Input: $\hat{f} \in \mathbb{Z}_q^{256}$
Output: $f \in \mathbb{Z}_q^{256}$

$f \leftarrow \hat{f}$
$k \leftarrow 127; j \leftarrow 0$
for $\text{len} \leftarrow 2; \text{len} \leq 128; \text{len} \leftarrow 2 \cdot \text{len}$ **do**
  for $\text{start} \leftarrow 0; \text{start} < 256; \text{start} \leftarrow j + \text{len}$ **do**
    for $j \leftarrow \text{start}; j < \text{start} + \text{len}; j++$ **do**
      $t \leftarrow f_j$
      $f_j \leftarrow \text{barrett\_reduce}(t + f_{j+\text{len}})$
      $f_{j+\text{len}} \leftarrow f_{j+\text{len}} - t$
      $f_{j+\text{len}} \leftarrow \text{montgomery\_reduce}(\zeta^k \cdot f_{j+\text{len}})$
    $k \leftarrow k - 1$

Modeling the ML-KEM NTT^{-1}

Input: $\hat{f} \in \mathbb{Z}_q^{256}$

Output: $f \in \mathbb{Z}_q^{256}$

$f \leftarrow \hat{f}$

$k \leftarrow 127; j \leftarrow 0$

for $\text{len} \leftarrow 2; \text{len} \leq 128; \text{len} \leftarrow 2 \cdot \text{len}$ **do** ▷ *7 layers*

  for $\text{start} \leftarrow 0; \text{start} < 256; \text{start} \leftarrow j + \text{len}$ **do**

    for $j \leftarrow \text{start}; j < \text{start} + \text{len}; j{+}{+}$ **do**

      $t \leftarrow f_j$

      $f_j \leftarrow \text{barrett\_reduce}(t + f_{j+\text{len}})$

      $f_{j+\text{len}} \leftarrow f_{j+\text{len}} - t$

      $f_{j+\text{len}} \leftarrow \text{montgomery\_reduce}(\zeta^k \cdot f_{j+\text{len}})$

    $k \leftarrow k - 1$

Modeling the ML-KEM NTT^{-1}

Input: $\hat{f} \in \mathbb{Z}_q^{256}$
Output: $f \in \mathbb{Z}_q^{256}$

$f \leftarrow \hat{f}$
$k \leftarrow 127; j \leftarrow 0$
for $\text{len} \leftarrow 2; \text{len} \leq 128; \text{len} \leftarrow 2 \cdot \text{len}$ **do** ▷ *7 layers*
  for $\text{start} \leftarrow 0; \text{start} < 256; \text{start} \leftarrow j + \text{len}$ **do** ▷ *256 coeffs.*
    for $j \leftarrow \text{start}; j < \text{start} + \text{len}; j{+}{+}$ **do**
      $t \leftarrow f_j$
      $f_j \leftarrow \text{barrett\_reduce}(t + f_{j+\text{len}})$
      $f_{j+\text{len}} \leftarrow f_{j+\text{len}} - t$
      $f_{j+\text{len}} \leftarrow \text{montgomery\_reduce}(\zeta^k \cdot f_{j+\text{len}})$
    $k \leftarrow k - 1$

Modeling the ML-KEM NTT^{-1}

Input: $\hat{f} \in \mathbb{Z}_q^{256}$

Output: $f \in \mathbb{Z}_q^{256}$

$f \leftarrow \hat{f}$

$k \leftarrow 127; j \leftarrow 0$

for $\text{len} \leftarrow 2; \text{len} \leq 128; \text{len} \leftarrow 2 \cdot \text{len}$ **do** ▷ *7 layers*

  for $\text{start} \leftarrow 0; \text{start} < 256; \text{start} \leftarrow j + \text{len}$ **do** ▷ *256 coeffs.*

    for $j \leftarrow \text{start}; j < \text{start} + \text{len}; j{+}{+}$ **do** ▷ *Select coeff. pairs*

      $t \leftarrow f_j$

      $f_j \leftarrow \text{barrett\_reduce}(t + f_{j+\text{len}})$

      $f_{j+\text{len}} \leftarrow f_{j+\text{len}} - t$

      $f_{j+\text{len}} \leftarrow \text{montgomery\_reduce}(\zeta^k \cdot f_{j+\text{len}})$

    $k \leftarrow k - 1$

Modeling the ML-KEM NTT^{-1}

Input: $\hat{f} \in \mathbb{Z}_q^{256}$

Output: $f \in \mathbb{Z}_q^{256}$

$f \leftarrow \hat{f}$

$k \leftarrow 127; j \leftarrow 0$

for $\text{len} \leftarrow 2; \text{len} \leq 128; \text{len} \leftarrow 2 \cdot \text{len}$ **do** ▷ *7 layers*

  for $\text{start} \leftarrow 0; \text{start} < 256; \text{start} \leftarrow j + \text{len}$ **do** ▷ *256 coeffs.*

    for $j \leftarrow \text{start}; j < \text{start} + \text{len}; j{+}{+}$ **do** ▷ *Select coeff. pairs*

      $t \leftarrow f_j$ ▷ *GS-Butterfly*

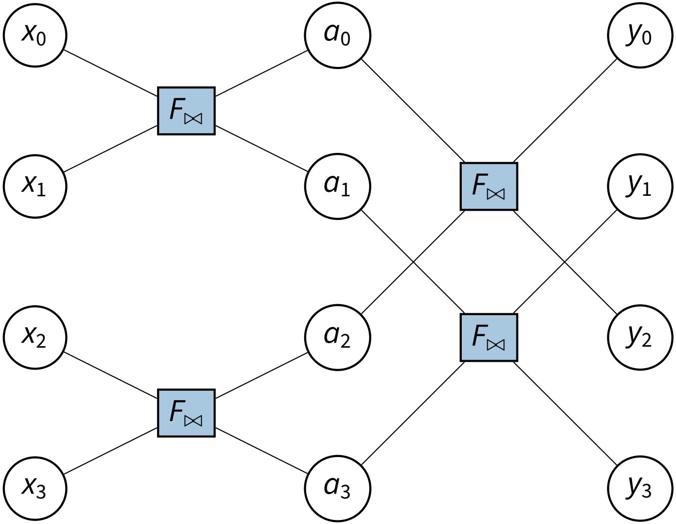
      $f_j \leftarrow \text{barrett\_reduce}(t + f_{j+\text{len}})$

      $f_{j+\text{len}} \leftarrow f_{j+\text{len}} - t$

      $f_{j+\text{len}} \leftarrow \text{montgomery\_reduce}(\zeta^k \cdot f_{j+\text{len}})$

    $k \leftarrow k - 1$

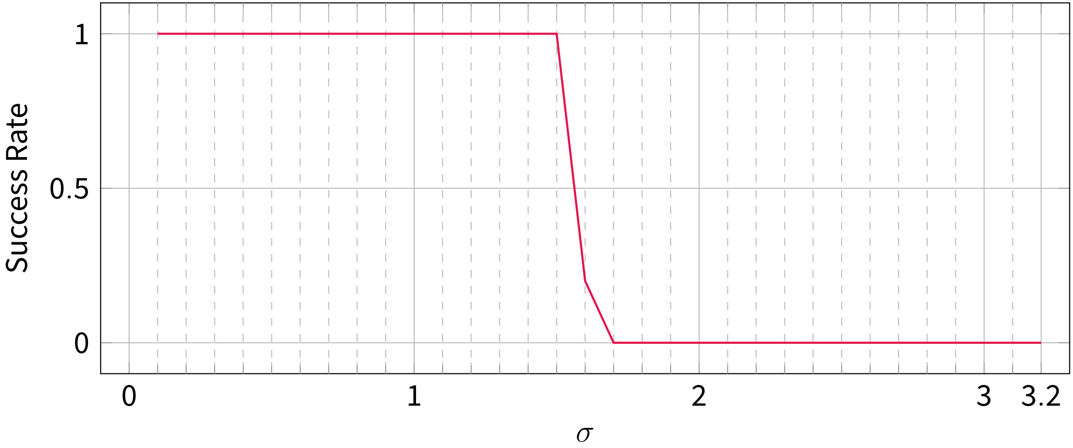
Modeling the ML-KEM NTT^{-1}



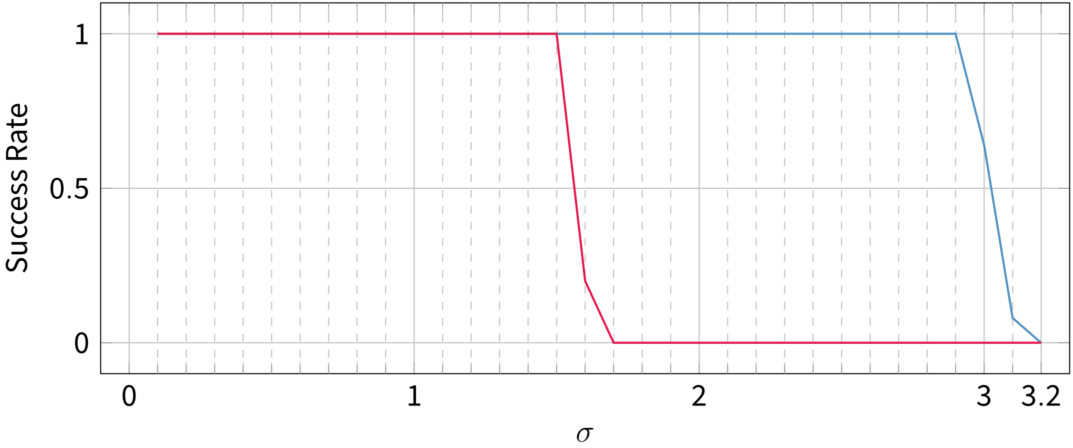
Modeling the ML-KEM NTT^{-1}

$$F_{\bowtie}(x_0, x_1, y_0, y_1) = \begin{cases} 1 & y_0 = x_0 + \zeta x_1 \bmod q \wedge \\ & y_1 = x_0 - \zeta x_1 \bmod q \\ 0 & \text{otherwise} \end{cases}$$

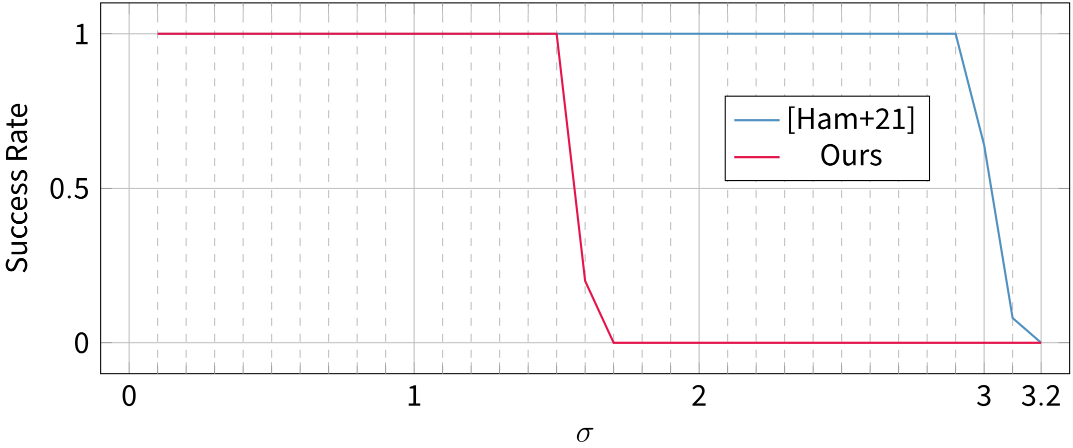
Simulated *k*-trace attack on the ML-KEM NTT^{-1}



Simulated k-trace attack on the ML-KEM NTT^{-1}



Simulated k-trace attack on the ML-KEM NTT^{-1}



What's Wrong? - A Closer Look at ML-KEM

- Operates in a Polynomial ring with coefficients in $\mathbb{Z}_q$; $q = 3329$

What's Wrong? - A Closer Look at ML-KEM

- Operates in a Polynomial ring with coefficients in $\mathbb{Z}_q$; $q = 3329$
- $\log_2 q \approx 11.7$-bits ➔ stored in 16-bit machine representations.

What's Wrong? - A Closer Look at ML-KEM

- Operates in a Polynomial ring with coefficients in $\mathbb{Z}_q$; $q = 3329$
- $\log_2 q \approx 11.7$-bits ➔ stored in 16-bit machine representations.
- Efficient implementations represent the integers in the signed range $\left[\left\lfloor \frac{-q}{2} \right\rfloor, \left\lceil \frac{q}{2} \right\rceil\right)$

What’s Wrong? - A Closer Look at ML-KEM

- The signed representation of the Polynomial ring makes Barrett and Montgomery reductions more efficient.

What’s Wrong? - A Closer Look at ML-KEM

- The signed representation of the Polynomial ring makes Barrett and Montgomery reductions more efficient.
- Great performance optimization! Less instructions, enables lazy reductions etc.

What's Wrong? - A Closer Look at ML-KEM

- The signed representation of the Polynomial ring makes Barrett and Montgomery reductions more efficient.
- Great performance optimization! Less instructions, enables lazy reductions etc.
- Sounds great, but…

What’s Wrong? - A Closer Look at ML-KEM

- The signed representation of the Polynomial ring makes Barrett and Montgomery reductions more efficient.
- Great performance optimization! Less instructions, enables lazy reductions etc.
- Sounds great, but…

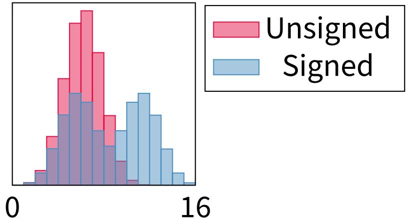
What’s Wrong? - A Closer Look at ML-KEM

- The signed representation of the Polynomial ring makes Barrett and Montgomery reductions more efficient.
- Great performance optimization! Less instructions, enables lazy reductions etc.
- Sounds great, but…

Side-channel Distinguisher [TMS24]

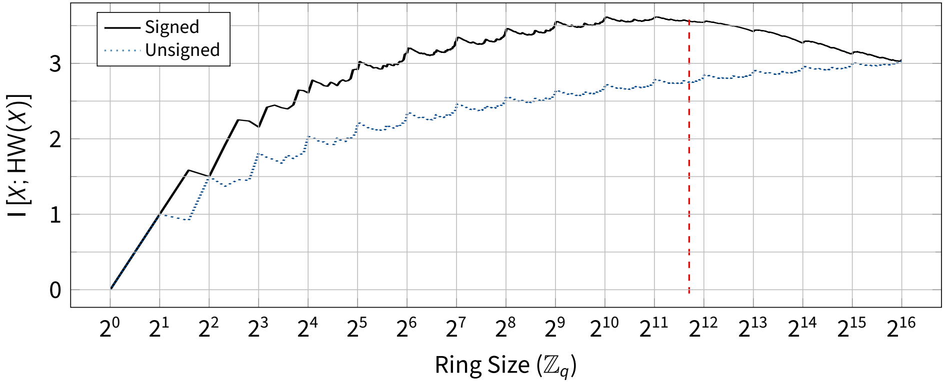
Small integer ranges (relative to the machine-word size) will have a large Hamming weight disparity between positive and negative numbers.

Hamming Weight Distributions of $\mathbb{Z}_q$



Hamming weights of $\mathbb{Z}_q$

Mutual Information Analysis of Machine Representations



Mutual Information Analyis of the ML-KEM NTT

1. Input polynomial: $\boldsymbol{X} = (X_i)_{i=0}^{255}$ where $X_i \in \mathbb{Z}_q$.

Mutual Information Analyis of the ML-KEM NTT

1. Input polynomial: $\boldsymbol{X} = (X_i)_{i=0}^{255}$ where $X_i \in \mathbb{Z}_q$.
2. Total Entropy: $\mathrm{H}[\boldsymbol{X}] = \sum_{\boldsymbol{X}} \mathrm{H}[X_i] \approx 2995.4 \, \mathrm{bits}$.

Mutual Information Analyis of the ML-KEM NTT

1. Input polynomial: $\boldsymbol{X} = (X_i)_{i=0}^{255}$ where $X_i \in \mathbb{Z}_q$.
2. Total Entropy: $\mathrm{H}[\boldsymbol{X}] = \sum_{\boldsymbol{X}} \mathrm{H}[X_i] \approx 2995.4 \text{ bits}$.
3. Adversary must learn: $\mathrm{H}[\boldsymbol{X}] - \sum_{\boldsymbol{X}} \mathrm{I}[X_i; \mathrm{W}(X_i)]$

Mutual Information Analyis of the ML-KEM NTT

1. Input polynomial: $\boldsymbol{X} = (X_i)_{i=0}^{255}$ where $X_i \in \mathbb{Z}_q$.
2. Total Entropy: $\mathrm{H}[\boldsymbol{X}] = \sum_{\boldsymbol{X}} \mathrm{H}[X_i] \approx 2995.4\,\mathrm{bits}$.
3. Adversary must learn: $\mathrm{H}[\boldsymbol{X}] - \sum_{\boldsymbol{X}} \mathrm{I}[X_i; \mathrm{W}(X_i)]$

Signed

- $\approx 2083.8\,\mathrm{bits}$.

Mutual Information Analyis of the ML-KEM NTT

1. Input polynomial: $\boldsymbol{X} = (X_i)_{i=0}^{255}$ where $X_i \in \mathbb{Z}_q$.
2. Total Entropy: $\mathrm{H}[\boldsymbol{X}] = \sum_{\boldsymbol{X}} \mathrm{H}[X_i] \approx 2995.4\,\mathrm{bits}$.
3. Adversary must learn: $\mathrm{H}[\boldsymbol{X}] - \sum_{\boldsymbol{X}} \mathrm{I}[X_i; \mathrm{W}(X_i)]$

± Signed

- $\approx 2083.8\,\mathrm{bits}$.

+ Unsigned

- $\approx 2289.9\,\mathrm{bits}$.

Mutual Information Analyis of the ML-KEM NTT

1. Input polynomial: $\boldsymbol{X} = (X_i)_{i=0}^{255}$ where $X_i \in \mathbb{Z}_q$.
2. Total Entropy: $\mathrm{H}[\boldsymbol{X}] = \sum_{\boldsymbol{x}} \mathrm{H}[X_i] \approx 2995.4\,\mathrm{bits}$.
3. Adversary must learn: $\mathrm{H}[\boldsymbol{X}] - \sum_{\boldsymbol{x}} \mathrm{I}[X_i; \mathrm{W}(X_i)]$

Signed

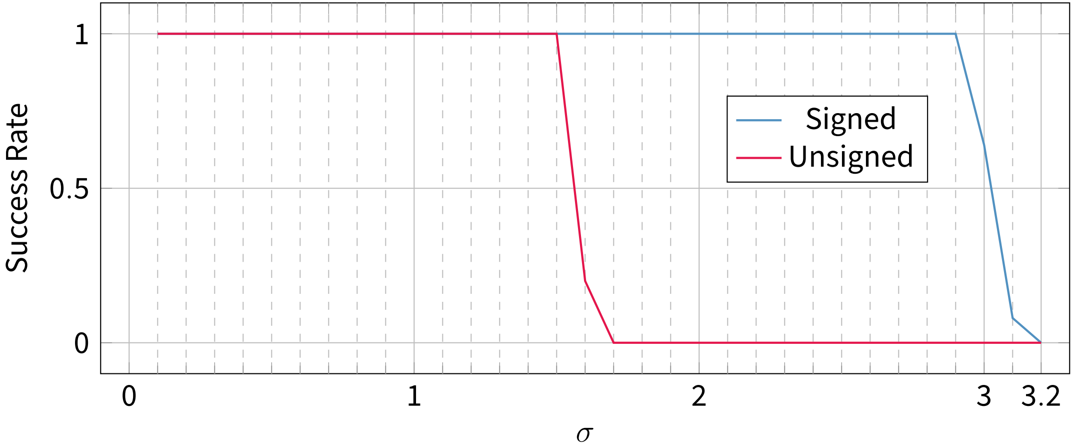
- $\approx 2083.8\,\mathrm{bits}$.

Unsigned

- $\approx 2289.9\,\mathrm{bits}$.

Adversary learns $\approx$ 206.092 bits just from signed representation!

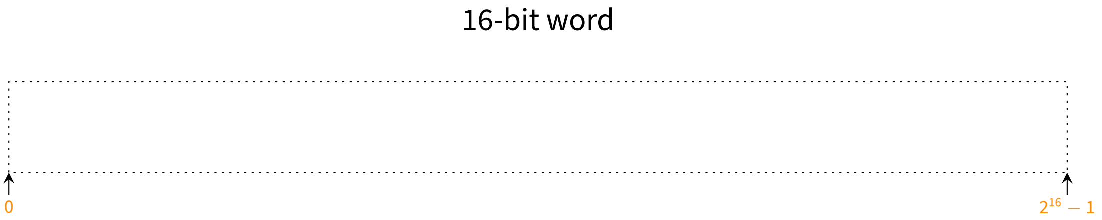
k-trace Attack on ML-KEM



Redundant Number Representation

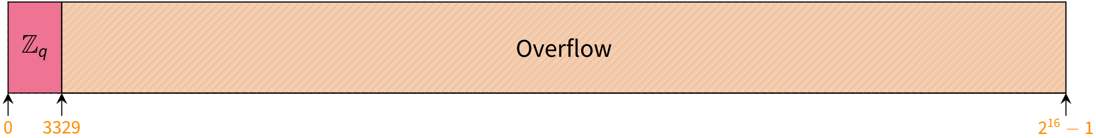
Application to ML-KEM

Redundant Number Representation (RNR)

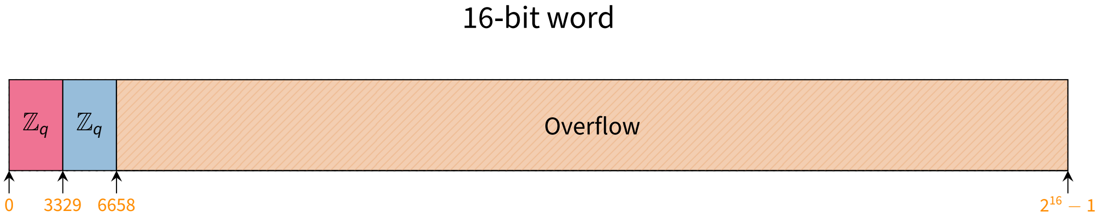


Redundant Number Representation (RNR)

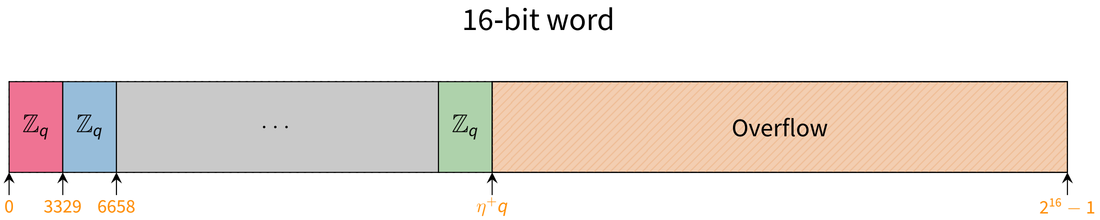
16-bit word



Redundant Number Representation (RNR)



Redundant Number Representation (RNR)



Applying RNR on ML-KEM

- Encode $x \in \mathbb{Z}_q$ to $x' \in \mathbb{Z}_{\eta q}$ where $x' = x + Kq$ where K is sampled uniformly from $[0, \eta)$.

Applying RNR on ML-KEM

- Encode $x \in \mathbb{Z}_q$ to $x' \in \mathbb{Z}_{\eta q}$ where $x' = x + Kq$ where K is sampled uniformly from $[0, \eta)$.
- The algorithm operates on η redundant encodings of $\mathbb{Z}_q$

Applying RNR on ML-KEM

- Encode $x \in \mathbb{Z}_q$ to $x' \in \mathbb{Z}_{\eta q}$ where $x' = x + Kq$ where K is sampled uniformly from $[0, \eta)$.
- The algorithm operates on η redundant encodings of $\mathbb{Z}_q$
- Ex: $0 \in \mathbb{Z}_q \equiv \{0, 3329, 6658, \ldots, (\eta - 1)q\}$

Applying RNR on ML-KEM

- Encode $x \in \mathbb{Z}_q$ to $x' \in \mathbb{Z}_{\eta q}$ where $x' = x + Kq$ where K is sampled uniformly from $[0, \eta)$.
- The algorithm operates on η redundant encodings of $\mathbb{Z}_q$
- Ex: $0 \in \mathbb{Z}_q \equiv \{0, 3329, 6658, \ldots, (\eta - 1)q\}$
- Works for signed representations too!

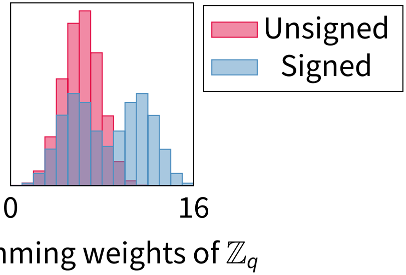
Applying RNR on ML-KEM

- Encode $x \in \mathbb{Z}_q$ to $x' \in \mathbb{Z}_{\eta q}$ where $x' = x + Kq$ where K is sampled uniformly from $[0, \eta)$.
- The algorithm operates on η redundant encodings of $\mathbb{Z}_q$
- Ex: $0 \in \mathbb{Z}_q \equiv \{0, 3329, 6658, \ldots, (\eta - 1)q\}$
- Works for signed representations too!

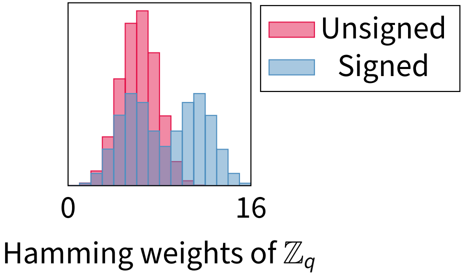
Outcome

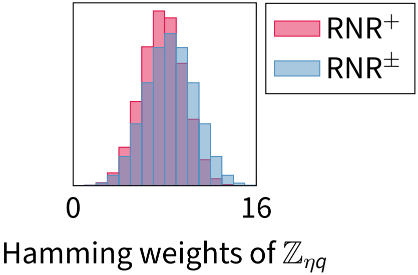
η encodings means upto η unique Hamming weights for a given x - **Makes SPA harder!**

Hamming weight distriubtions of RNR

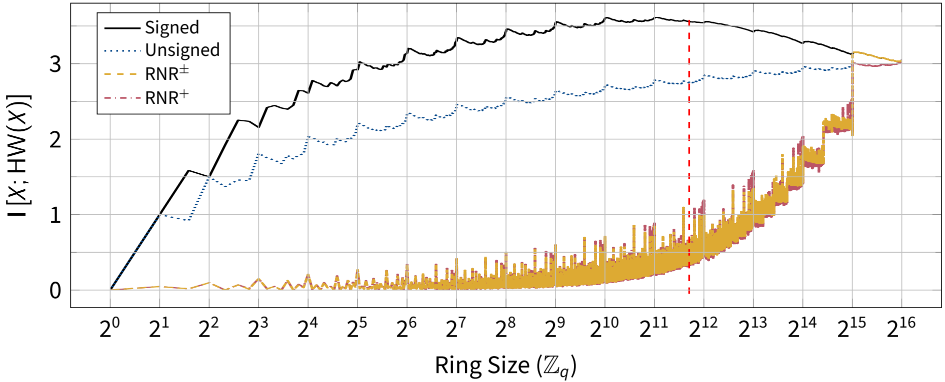


Hamming weight distriubtions of RNR

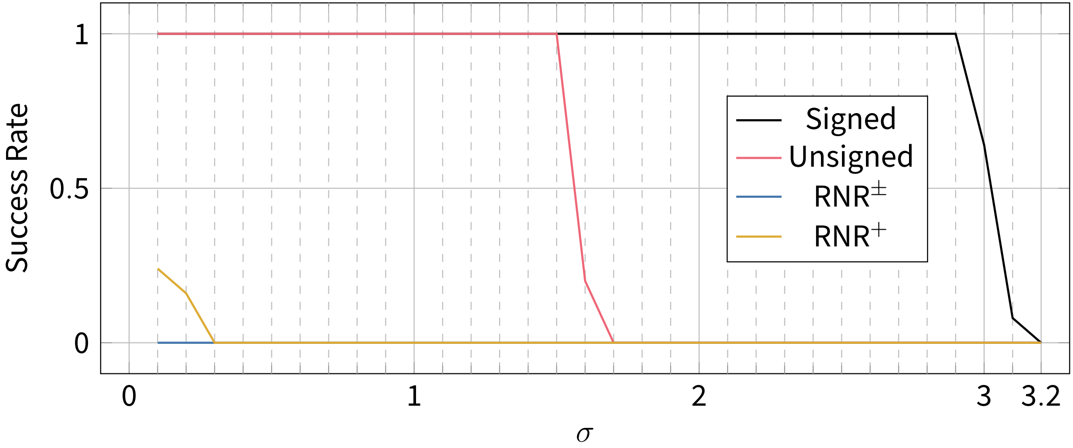




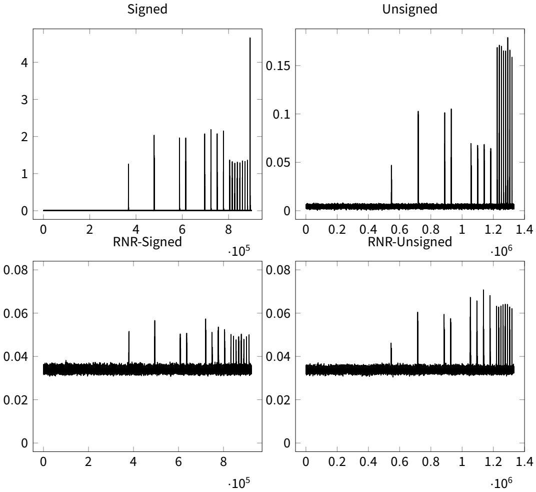
Redundant Number Representation



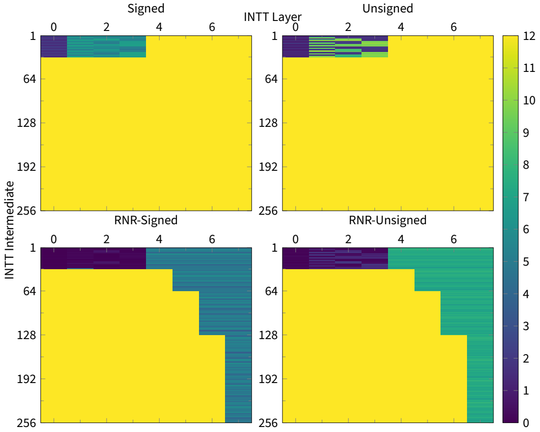
Simulated *k*-trace attack on the ML-KEM RNR-NTT^{-1}



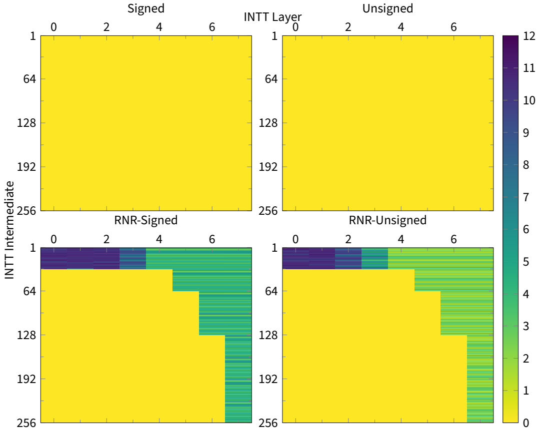
k-trace Attack on a ARM Cortex-M4 - SNR



k-trace Attack on a ARM Cortex-M4 - PI Estimate



k-trace Attack on a ARM Cortex-M4 - SASCA Result



Implementation and Performance Results

Implementation	KCycles ($\cdot 10^3$)			
	-O0	Overhead	-O3	Overhead
Signed-NTT	127.02		26.48	
Unsigned-NTT	158.00		36.75	
$RNR^{\pm}$-NTT	196.01	42.7%	50.70	62.8%
RNR^{+}-NTT	260.52	49.0%	84.74	79.0%
Signed-NTT^{-1}	202.04		42.61	
Unsigned-NTT^{-1}	270.39		64.91	
$RNR^{\pm}$-NTT^{-1}	203.19	0.6%	42.61	0%
RNR^{+}-NTT^{-1}	305.59	12.2%	91.15	27.7%

Comparison to Shuffling [Rav+20]

Countermeasures	Shuffle Algo.	KCycles ($\times 10^3$)			
		Count	Overhead (%)	Shuffle	Rand.
Kyber NTT					
Unprotected	NA	31.0	-	-	-
Coarse-Full-Shuffled	Knuth-Yates	87.2	**181.1**	16.6 (**19%**)	38.4 (**44.1%**)
Coarse-In-Group-Shuffle	Knuth-Yates	84.4	**172.2**	17.1 (**20.3%**)	32.4 (**38.4%**)
Basic-Fine-Shuffled	Arith. cswap	76.7	**147.4**	35.1 (**45.7%**)	9.5 (**12.4%**)
Bitwise-Fine-Shuffle	Arith. cswap	142.6	**356**	100.1 (**70.2%**)	9.5 (**6.7%**)
Kyber INTT					
Unprotected	NA	50.6	-	-	-
Coarse-Full-Shuffled	Knuth-Yates	113.3	**123.8**	16.6 (**14.6%**)	38.4 (**33.9%**)
Coarse-In-Group-Shuffled	Knuth-Yates	101.2	**99.9**	16 (**15.8%**)	33 (**32.6%**)
Basic-Fine-Shuffled	Arith. cswap	101.8	**101.1**	40.9 (**40.1%**)	9.5 (**9.4%**)
Bitwise-Fine-Shuffled	Arith. cswap	172.4	**240.8**	102.2 (**59.3%**)	9.6 (**5.5%**)

Conclusion

- Even small performance optimizations can have unforseen and impactful consequences.
- RNR is sufficient at preventing the strongest known SPA attack against ML-KEM.
- Can be achieved with a low performance impact and simple to implement!

rishub.nagpal@tugraz.at

https://github.com/rishubn/rnr-kyber-spa

Acknowledgments

This research was funded in whole or in part by the Austrian Science Fund (FWF) (FWF SFB project SPyCoDe 10.55776/F85) and the Austrian Research Promotion Agency (FFG) via the AWARE project (FFG grant number 891092).

Bibliography

[Ham+21] Mike Hamburg et al. **Chosen Ciphertext k-Trace Attacks on Masked CCA2 Secure Kyber.** IACR Trans. Cryptogr. Hardw. Embed. Syst. 2021.4 (2021), pp. 88–113. DOI: 10.46586/TCHES.V2021.I4.88-113. URL: https://doi.org/10.46586/tches.v2021.i4.88-113.

[Rav+20] Prasanna Ravi et al. **On Configurable SCA Countermeasures Against Single Trace Attacks for the NTT - A Performance Evaluation Study over Kyber and Dilithium on the ARM Cortex-M4.** Security, Privacy, and Applied Cryptography Engineering - 10th International Conference, SPACE 2020, Kolkata, India, December 17-21, 2020, Proceedings. Ed. by Lejla Batina, Stjepan Picek, and Mainack Mondal. Vol. 12586. Lecture Notes in Computer Science. Springer, 2020, pp. 123–146. DOI: 10.1007/978-3-030-66626-2_7. URL: https://doi.org/10.1007/978-3-030-66626-2%5C_7.

[TMS24] Tolun Tosun, Amir Moradi, and Erkay Savas. **Exploiting the Central Reduction in Lattice-Based Cryptography.** IEEE Access 12 (2024), pp. 166814–166833. DOI: 10.1109/ACCESS.2024.3494593. URL: https://doi.org/10.1109/ACCESS.2024.3494593.

Backup Slides

Derivation of η

$$\begin{aligned} \eta^+ q + \left(\frac{\eta^+ q^2}{2^{16}} + \eta^+ q\right) + q &< 2^{16} \\ \eta^+ \cdot \left(2q + \frac{q^2}{2^{16}}\right) &< 2^{16} - q \\ \eta^+ &< \frac{2^{32} - 2^{16} q}{2^{17} q + q^2} < 10 \end{aligned} \tag{1}$$

Modeling the ML-KEM NTT

Number Theoretic Transform

An algorithm analogous to the Discrete Fourier Transform (DFT) which allows one to compute the product of two polynomials efficiently.

- In ML-KEM:

Modeling the ML-KEM NTT

Number Theoretic Transform

An algorithm analogous to the Discrete Fourier Transform (DFT) which allows one to compute the product of two polynomials efficiently.

- In ML-KEM:
  - Factors degree-256 polynomials with small 128 degree-2 polynomials

$$(x^{256} + 1) = \prod_{i=0}^{127} \left(x^2 - \zeta^{2i+1}\right),$$

where ζ^n is the n-th root-of-unity.

Modeling the ML-KEM NTT

$$\mathrm{NTT}(a) = \hat{a} = \hat{a}_0 + \hat{a}_1 x + \ldots \hat{a}_{255} x^{255},$$

$$\hat{a}_i = \sum_{j=0}^{127} a_{2j} \zeta^{(2i+1)j} \quad \text{and} \quad \hat{a}_{2i+1} = \sum_{j=0}^{127} a_{2j+1} \zeta^{(2i+1)j}.$$

Multiplication of polynomials: $\mathrm{NTT}^{-1}(\mathrm{NTT}(f) \circ \mathrm{NTT}(g))$.