The Challenge of Side-Channel Countermeasures on Post-Quantum Crypto

Rina Zeitoun - rina.zeitoun@idemia.com
IDEMIA - Crypto & Security Labs

November 29 – December 1, 2022
Outline

1 〉 Context

2 〉 Side-channel Attacks on Lattice-based KEM

3 〉 Masking and Conversions Problematics

4 〉 The example of Kyber

5 〉 Conclusion
Context

IDEMIA: The leader in identity technologies

› Identity (3B ID docs, 5M biometric terminals).
› Payment (800M payment products - 2021).
› Telecoms (900M SIM cards - 2021).

Into the wild

› Our products are deployed in hostile environments.
› Attackers have physical access to the device.
› Must be resistant to side-channel/fault attacks.

☞ Security against side-channel attacks is mandatory.
## Side-Channel Attacks

<table>
<thead>
<tr>
<th>Main Powerful Attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>› Timing Attacks, Simple Power Analysis, Differential/Correlation Power/Electromagnetic Analysis, Template Attacks, Fault Attacks, etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Into Specifications of Selected NIST PQC Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>› Resistance to Timing Attacks is always addressed.</td>
</tr>
<tr>
<td>› All other attacks are mainly <strong>left for research</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Smartcards: In real life</th>
</tr>
</thead>
<tbody>
<tr>
<td>› Timing attacks are indeed important to consider.</td>
</tr>
<tr>
<td>› But <strong>all</strong> other classical side-channel attacks are definitely <strong>real threats</strong>!</td>
</tr>
<tr>
<td>› Main powerful attacks should <strong>systematically</strong> be studied in NIST submissions.</td>
</tr>
</tbody>
</table>
Outline

1 〉 Context

2 〉 Side-channel Attacks on Lattice-based KEM

3 〉 Masking and Conversions Problematics

4 〉 The example of Kyber

5 〉 Conclusion
Power/EM Attacks on Decapsulation based on FO Transform

[Diagram of the process with nodes labeled Decrypt, Encrypt, and Compare, and arrows indicating the flow of data from input 'c' to output 'K'.]

- Whole Decapsulation needs to be masked.
Power/EM Attacks on Decapsylation based on FO Transform

Whole Decapsulation needs to be masked

Template Attacks on Key Generation

Template attacks require detailed knowledge of target but can be a real threat!

Investigated in security certifications (Common Criteria and EMVco).
Outline

1 〡 Context

2 〡 Side-channel Attacks on Lattice-based KEM

3 〡 Masking and Conversions Problematics

4 〡 The example of Kyber

5 〡 Conclusion
## High-Order Masking Countermeasure

- Each sensitive variable $x$ is shared into $n$ variables: $x = x_1 \oplus x_2 \oplus \cdots \oplus x_n$
- Manipulate $x_1, x_2, \ldots, x_n$ independently
## High-Order Masking Countermeasure

- Each sensitive variable $x$ is shared into $n$ variables: $x = x_1 \oplus x_2 \oplus \cdots \oplus x_n$
- Manipulate $x_1, x_2, \ldots, x_n$ independently

### Computing with Boolean Masking

Given $x = x_1 \oplus \cdots \oplus x_n$ and $y = y_1 \oplus \cdots \oplus y_n$, how can we compute $x \oplus y$?
- Compute $x_1 \oplus y_1, \cdots, x_n \oplus y_n$
Masking Countermeasure

High-Order Masking Countermeasure

- Each sensitive variable $x$ is shared into $n$ variables: $x = x_1 \oplus x_2 \oplus \cdots \oplus x_n$
- Manipulate $x_1, x_2, \ldots, x_n$ independently

Computing with Boolean Masking

Given $x = x_1 \oplus \cdots \oplus x_n$ and $y = y_1 \oplus \cdots \oplus y_n$, how can we compute $x \oplus y$?
- Compute $x_1 \oplus y_1, \ldots, x_n \oplus y_n$

Arithmetic Masking Countermeasure

Generate arithmetic sharings s.t. $x = x_1 + \cdots + x_n \mod 2^k$ and $y = y_1 + \cdots + y_n \mod 2^k$
- Compute $x_1 + y_1 \mod 2^k, \ldots, x_n + y_n \mod 2^k$
Arithmetic and Boolean Masking

Masks Conversions

- Need to convert between arithmetic and Boolean masking.
- Efficient classical masks conversions exist ([Gou01],[CGV14],[CGTV15],[BCZ18], etc.)
Arithmetic and Boolean Masking

Masks Conversions

- Need to convert between arithmetic and Boolean masking.
- Efficient classical masks conversions exist ([Gou01],[CGV14],[CGTV15],[BCZ18], etc.)

Difference with previous schemes

- Symmetric schemes: \(k\)-bit Boolean \(\leftrightarrow\) arithmetic modulo \(2^k\); usually \(k = 32\)
- Post-Quantum schemes: \(k\)-bit Boolean \(\leftrightarrow\) arithmetic modulo \(q\); arbitrary \(k, q\)
New Problematics with Post-Quantum Crypto

Arbitrary Masks Conversions

- Generic conversions suitable for PQ schemes exist ([BBE+18]: generalization of [CGTV15])
- Downside: Can be too costly in practice
### New Problematics with Post-Quantum Crypto

#### Arbitrary Masks Conversions

- Generic conversions suitable for PQ schemes exist ([BBE+18]: generalization of [CGTV15])
- Downside: Can be too costly in practice

<table>
<thead>
<tr>
<th>Other Problematics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure polynomials comparison (Kyber, Dilithium)</td>
</tr>
<tr>
<td>Secure computation of compression: $\lceil (2^d/q) \cdot x \rceil \mod 2^d$ (Kyber)</td>
</tr>
<tr>
<td>Secure generation of a random in a given interval (Dilithium)</td>
</tr>
<tr>
<td>Secure Euclidean division (NTRU, Dilithium)</td>
</tr>
<tr>
<td>etc.</td>
</tr>
</tbody>
</table>
New Problematics with Post-Quantum Crypto

Arbitrary Masks Conversions

▷ Generic conversions suitable for PQ schemes exist ([BBE+18]: generalization of [CGTV15])
▷ Downside: Can be too costly in practice

Other Problematics

▷ Secure polynomials comparison (Kyber, Dilithium)
▷ Secure computation of compression: $\lceil (2^d/q) \cdot x \rceil \mod 2^d$ (Kyber)
▷ Secure generation of a random in a given interval (Dilithium)
▷ Secure Euclidean division (NTRU, Dilithium)
▷ etc.

☞ Need specific solution for each problem
Outline

1  ›  Context

2  ›  Side-channel Attacks on Lattice-based KEM

3  ›  Masking and Conversions Problematics

4  ›  The example of Kyber

5  ›  Conclusion
Encryption Problematic: Securely compute $\lceil q/2 \rceil \cdot m$ (prime $q = 3329$)

We have $m = m_1 \oplus \cdots \oplus m_n$ where $m_i$ are 1-bit long.

Compute $y_1 + \cdots + y_n \mod q = 1665 \cdot (m_1 \oplus \cdots \oplus m_n)$. 

Encryption Solution

Convert 1-bit Boolean sharing $m_1, \cdots, m_n$ into arithmetic modulo $q$.

Use $[BBE+18]$ with complexity $O(n^2 \cdot \log \log q)$.

Use $[SPOG19]$ or $[CGMZ21a]$ with complexity $O(n^2)$. 

Centered Binomial Distribution (CBD)

Similar problematic and solution to securely compute $e = \text{HW}(x) - \text{HW}(y)$ in CBD.
Kyber Encryption

Encryption Problematic: Securely compute $\left\lfloor \frac{q}{2} \right\rfloor \cdot m$ (prime $q = 3329$)

 › We have $m = m_1 \oplus \cdots \oplus m_n$ where $m_i$ are 1-bit long.
 › Compute $y_1 + \cdots + y_n \mod q = 1665 \cdot (m_1 \oplus \cdots \oplus m_n)$.

Encryption Solution

Convert 1-bit Boolean sharing $m_1, \cdots, m_n$ into arithmetic modulo $q$

 › Use [BBE+18] with complexity $O(n^2 \cdot \log \log q)$.
 › Use [SPOG19] or [CGMZ21a] with complexity $O(n^2)$. 

The Challenge of Side-Channel Countermeasures on Post-Quantum Crypto

The example of Kyber
Kyber Encryption

Encryption Problematic: Securely compute $\lceil q/2 \rceil \cdot m$ (prime $q = 3329$)

- We have $m = m_1 \oplus \cdots \oplus m_n$ where $m_i$ are 1-bit long.
- Compute $y_1 + \cdots + y_n \mod q = 1665 \cdot (m_1 \oplus \cdots \oplus m_n)$.

Encryption Solution

Convert 1-bit Boolean sharing $m_1, \cdots, m_n$ into arithmetic modulo $q$

- Use [BBE+18] with complexity $O(n^2 \cdot \log \log q)$.
- Use [SPOG19] or [CGMZ21a] with complexity $O(n^2)$.

Centered Binomial Distribution (CBD)

- Similar problematic and solution to securely compute $e = HW(x) - HW(y)$ in CBD.
Kyber Decryption

Decryption Problematic: Securely compute \( m = \lceil (2/q) \cdot x \rceil \mod 2 \)

- We have \( x = x_1 + \cdots + x_n \mod q \).
- Compute \( m_1 \oplus \cdots \oplus m_n = \lceil (2/q) \cdot (x_1 + \cdots + x_n) \rceil \mod 2 \).
Kyber Decryption

Decryption Problematic: Securely compute \( m = \lceil (2/q) \cdot x \rceil \mod 2 \)

- We have \( x = x_1 + \cdots + x_n \mod q \).
- Compute \( m_1 \oplus \cdots \oplus m_n = \lceil (2/q) \cdot (x_1 + \cdots + x_n) \rceil \mod 2 \).

Decryption Solution

- [BGR+21]: Convert \( A \mod q \leftrightarrow B \) (BBE+18) + 4 secure AND. Complexity \( \mathcal{O}(n^2 \cdot \log q) \).
- [CGMZ21a]: Switch \( A \mod q \leftrightarrow A \mod 2^k \); convert \( A \mod 2^k \leftrightarrow B \). Complexity \( \mathcal{O}(n^2 \cdot \log n) \).
Compress\(_{q,d}(x)\) Problematic: Securely compute \([\frac{2^d}{q} \cdot x] \mod 2^d\)

- We have \(x = x_1 + \cdots + x_n \mod q\) and \(d = 4, 5, 10\) or \(11\).
- Compute \(y_1 \oplus \cdots \oplus y_n = [(\frac{2^d}{q} \cdot (x_1 + \cdots + x_n))] \mod 2^d\).
Kyber other problematics: Compress & Compare

**Compress\_q,d(x) Problematic:** Securely compute \( \lceil (2^d/q) \cdot x \rceil \mod 2^d \)

- We have \( x = x_1 + \cdots + x_n \mod q \) and \( d = 4, 5, 10 \) or 11.
- Compute \( y_1 \oplus \cdots \oplus y_n = \lceil (2^d/q) \cdot (x_1 + \cdots + x_n) \rceil \mod 2^d \).

**Comparison Problematic**

- We are given \( \ell \) coefficients \( x^{(\ell)} = x_1^{(\ell)} \oplus \cdots \oplus x_n^{(\ell)} \) and \( y^{(\ell)} \).
- Compare \( x_1^{(\ell)} \oplus \cdots \oplus x_n^{(\ell)} \) and \( y^{(\ell)} \) without revealing which coefficients fail.
Kyber other problematics: Compress & Compare

<table>
<thead>
<tr>
<th>Compress(_{q,d})(x) Problematic: Securely compute (\lceil (2^d/q) \cdot x \rceil \mod 2^d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>➜ We have (x = x_1 + \cdots + x_n \mod q) and (d = 4, 5, 10) or 11.</td>
</tr>
<tr>
<td>➜ Compute (y_1 \oplus \cdots \oplus y_n = \lceil (2^d/q) \cdot (x_1 + \cdots + x_n) \rceil \mod 2^d).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comparison Problematic</th>
</tr>
</thead>
<tbody>
<tr>
<td>➜ We are given (\ell) coefficients (x^{(\ell)} = x_1^{(\ell)} \oplus \cdots \oplus x_n^{(\ell)}) and (y^{(\ell)}).</td>
</tr>
<tr>
<td>➜ Compare (x_1^{(\ell)} \oplus \cdots \oplus x_n^{(\ell)}) and (y^{(\ell)}) without revealing which coefficients fail.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>➜ [FBR+21]: First order only; compare (H(x_1^{(\ell)} \oplus y^{(\ell)})) with (H(x_2^{(\ell)})).</td>
</tr>
<tr>
<td>➜ [BGR+21]: Do not compress: compare uncompressed coefficients with public ones.</td>
</tr>
<tr>
<td>➜ [CGMZ21b]: Hybrid approach with new compression and comparison methods.</td>
</tr>
</tbody>
</table>
Fully masked implementation of Kyber [CGMZ21a/b]

Kyber768 Decapsulation on ARM Cortex-M3 for given security order:

- For security order \( t > 3 \), required RAM too large for ARM Cortex-M3 target device.
Outline

1 〉 Context

2 〉 Side-channel Attacks on Lattice-based KEM

3 〉 Masking and Conversions Problematics

4 〉 The example of Kyber

5 〉 Conclusion
Conclusion

Smartcards:
   › Real need to secure implementations against all SCA.

Standard specifications:
   › Resistance against timing attacks studied in standardized PQ algorithms.
   › Other Side-Channel Attacks (Power/EM DPA, templates, fault) mainly left for research.

Attacks in practice:
   › Many practical Side-Channel Attacks published.

Countermeasures:
   › New challenges for PQ crypto countermeasures.
   › Not trivial and imply large overhead (can be unacceptable for many products).

Going Forward:
   › Encourage designers to study classical SCA at an early stage ("Masking friendly" PQ crypto).
Bibliography

[**Gou01**] A Sound Method for Switching between Boolean and Arithmetic Masking. Goubin, CHES’01.


[**BCZ18**] Improved High-Order Conversion From Boolean to Arithmetic. Bettale, Coron, Zeitoun, CHES’18.

[**BBE+18**] Masking the GLP Lattice-Based Signature Scheme at Any Order. Barthe, Belaïd, Espitau, Fouque, Grégoire, Rossi, Tibouchi, EUROCRYPT’18.

[**SPOG19**] Efficiently Masking Binomial Sampling at Arbitrary Orders for Lattice-Based Crypto. Schneider, Paglia, Oder, Güneysu, PKC’19.

[**CGMZ21a**] High-order Table-based Conversion Alg. and Masking Lattice-based Encryption. Coron, Gérard, Montoya, Zeitoun, CHES’22.


[**FBR+21**] Masked Accelerators and Instruction Set Extensions for PQ. Fritzmann, Beirendonck, Roy, Karl, Schamberger, Verbauwhede, Sigl, CHES’22.

[**PPM17**] Single-trace side-channel attacks on masked lattice-based encryption. Primas, Pessl, Mangard, CHES’17.

[**XPRO21**] Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of Kyber. Xu, Pemberton, Roy, Oswald, IEEE’21.


Thank you for your attention!
rina.zeitoun@idemia.com