Do Not Bound to a Single Position: Near-Optimal Multi-Positional Mismatch Attacks Against Kyber and Saber

Qian Guo[2], Erik Mårtensson[1,2]

[1] Selmer Center, Department of Informatics, University of Bergen, Norway

[2] Dept. of Electrical and Information Technology, Lund University, Sweden

Post-Quantum Cryptography

- Today cryptography depends on the assumption that either the integer factoring problem or the discrete logarithm problem is computationally infeasible.

Post-Quantum Cryptography

- Today cryptography depends on the assumption that either the integer factoring problem or the discrete logarithm problem is computationally infeasible.
- In the mid 90s Peter Shor showed that both problems can be solved in polynomial time on a large-scale quantum computer.

Post-Quantum Cryptography

- Today cryptography depends on the assumption that either the integer factoring problem or the discrete logarithm problem is computationally infeasible.
- In the mid 90s Peter Shor showed that both problems can be solved in polynomial time on a large-scale quantum computer.
- Post-quantum cryptography replaces these mathematical problems
  - **Lattice-based cryptography**
    - **Learning With Errors/Rounding (LW(E/R))**
    - Ring/**module LW(E/R)**
    - NTRU
  - Code-based, multivariate, hash-based, supersingular isogeny cryptography...

NIST Post-Quantum Cryptography Standardization

- First round (Dec. 2017): 59 PKE/KEM and 23 signature schemes

NIST Post-Quantum Cryptography Standardization

- First round (Dec. 2017): 59 PKE/KEM and 23 signature schemes
- Second round (Jan. 2019): 17 PKE/KEM and 9 signature schemes

NIST Post-Quantum Cryptography Standardization

- First round (Dec. 2017): 59 PKE/KEM and 23 signature schemes
- Second round (Jan. 2019): 17 PKE/KEM and 9 signature schemes
- Third round (Jul. 2020): 9 PKE/KEM and 6 signature schemes. 4 finalists for PKE/KEM
  - 3 lattice-based: **Kyber**, **Saber**, NTRU
  - 1 code-based: Classical McEliece

NIST Post-Quantum Cryptography Standardization

- First round (Dec. 2017): 59 PKE/KEM and 23 signature schemes
- Second round (Jan. 2019): 17 PKE/KEM and 9 signature schemes
- Third round (Jul. 2020): 9 PKE/KEM and 6 signature schemes. 4 finalists for PKE/KEM
  - 3 lattice-based: **Kyber**, **Saber**, NTRU
  - 1 code-based: Classical McEliece
- Fourth round (Jul. 2022): **Kyber** is selected for PKE/KEM!

Kyber/Saber Structure and Attack Model

- Kyber/Saber start by creating a Chosen Plaintext Attack (CPA) secure scheme.

Kyber/Saber Structure and Attack Model

- Kyber/Saber start by creating a Chosen Plaintext Attack (CPA) secure scheme.
- Then they make the scheme Chosen Ciphertext Attack (CCA) secure using the Fujisaki-Okamoto (FO) transform.

Kyber/Saber Structure and Attack Model

- Kyber/Saber start by creating a Chosen Plaintext Attack (CPA) secure scheme.
- Then they make the scheme Chosen Ciphertext Attack (CCA) secure using the Fujisaki-Okamoto (FO) transform.
- In this paper we study attacks on the CPA-secure version, when the secret key is re-used.
  - Resistance against these types of attacks is a desirable property according to the original NIST PQC call.
  - You shouldn't implement the schemes like this - but someone might still do it!
  - Mismatch attacks also have applications in side-channel attacks [SCZ+22, ...] and fault-injection attacks[XIU+21].

Kyber/Saber Structure and Attack Model

- Kyber/Saber start by creating a Chosen Plaintext Attack (CPA) secure scheme.
- Then they make the scheme Chosen Ciphertext Attack (CCA) secure using the Fujisaki-Okamoto (FO) transform.
- In this paper we study attacks on the CPA-secure version, when the secret key is re-used.
  - Resistance against these types of attacks is a desirable property according to the original NIST PQC call.
  - You shouldn't implement the schemes like this - but someone might still do it!
  - Mismatch attacks also have applications in side-channel attacks [SCZ+22, ...] and fault-injection attacks[XIU+21].
  - Finally, [QZC+21] gave a bound for the performance of this type of attack at Asiacrypt 2021 - we didn't believe the bound!

Some Notations

Given positive integers p, q, with $p < q$ and $x \in \mathbb{Z}_q$.

$$\textbf{Compress}_q(x, p) = \lceil x \cdot p/q \rfloor \mod^+ p,$$

where $\mod^+ p$ chooses a value in $(-p/2, p/2]$.

Some Notations

Given positive integers p, q, with $p < q$ and $x \in \mathbb{Z}_q$.

$$\textbf{Compress}_q(x, p) = \lceil x \cdot p/q \rfloor \mod {}^{+} p,$$

where $\mod {}^{+} p$ chooses a value in $(-p/2, p/2]$. Also,

$$\textbf{Decompress}_q(x, p) = \lceil x \cdot q/p \rfloor.$$

Finally, let $\mathbf{B}_\eta$ denote the central binomial distribution with parameter η.

Alice		**Bob**
1. Generate matrix $\mathbf{a} \in \mathcal{R}_q^{l \times l}$		
$\mathbf{s}_A, \mathbf{e}_A \leftarrow_{\$} \mathbf{B}_\eta^l$		2. $\mathbf{m} \leftarrow_{\$} \{0,1\}^{256}$
$\mathbf{P}_A \leftarrow \mathbf{a} \circ \mathbf{s}_A + \mathbf{e}_A$		Generate matrix $\mathbf{a} \in \mathcal{R}_q^{l \times l}$
Output: $(\mathbf{s}_A, \mathbf{P}_A)$	$\xrightarrow{\mathbf{P}_A}$	$\mathbf{s}_B \leftarrow_{\$} \mathbf{B}_\eta^l, \mathbf{e}_B \leftarrow_{\$} \mathbf{B}_{\eta'}^l, \mathbf{e}'_B \leftarrow_{\$} \mathbf{B}_{\eta'}$
		$\mathbf{P}_B \leftarrow \mathbf{a} \circ \mathbf{s}_B + \mathbf{e}_B$
		$\mathbf{v}_B \leftarrow \mathbf{P}_A^{\text{tr}} \circ \mathbf{s}_B + \mathbf{e}'_B + \mathbf{Decompress}_q(\mathbf{m}, 2)$
3. $\mathbf{u}_A \leftarrow \mathbf{Decompress}_q(\mathbf{c}_1, 2^{d_{\mathbf{P}_B}})$		$\mathbf{c}_1 \leftarrow \mathbf{Compress}_q(\mathbf{P}_B, 2^{d_{\mathbf{P}_B}})$
$\mathbf{v}_A \leftarrow \mathbf{Decompress}_q(\mathbf{c}_2, 2^{d_{\mathbf{v}_B}})$	$\xleftarrow{\mathbf{P}_B, \mathbf{c}_1, \mathbf{c}_2}$	$\mathbf{c}_2 \leftarrow \mathbf{Compress}_q(\mathbf{v}_B, 2^{d_{\mathbf{v}_B}})$
$\mathbf{m}' \leftarrow \mathbf{Compress}_q(\mathbf{v}_A - \mathbf{s}_A^{\text{tr}} \circ \mathbf{u}_A, 2)$		$K_B \leftarrow \mathbf{H}(\mathbf{m} \vert\vert (\mathbf{P}_B, (\mathbf{c}_1, \mathbf{c}_2)))$
$K_A \leftarrow \mathbf{H}(\mathbf{m}' \vert\vert (\mathbf{P}_B, (\mathbf{c}_1, \mathbf{c}_2)))$		

Figure: The CPA-secure version of Kyber

Mismatch Attack Idea

- Eve impersonates Bob and manipulates his public parameters $\mathbf{P}_B$, $\mathbf{c}_1$, $\mathbf{c}_2$ in a smart way.

Mismatch Attack Idea

- Eve impersonates Bob and manipulates his public parameters $\mathbf{P}_B$, $\mathbf{c}_1$, $\mathbf{c}_2$ in a smart way.
- By observing whether Bob's key K_B matches Alice's key K_A she learns (up to) a bit of information about the secret $\mathbf{s}_A$.
  - Eve essentially asks a yes/no question about the contents of $\mathbf{s}_A$ - with some restrictions.

Mismatch Attack Idea

- Eve impersonates Bob and manipulates his public parameters $\mathbf{P}_B$, $\mathbf{c}_1$, $\mathbf{c}_2$ in a smart way.
- By observing whether Bob's key K_B matches Alice's key K_A she learns (up to) a bit of information about the secret $\mathbf{s}_A$.
  - Eve essentially asks a yes/no question about the contents of $\mathbf{s}_A$ - with some restrictions.
- By repeating the process enough times Eve learns the entire secret $\mathbf{s}_A$.

Mismatch Attack Idea Detailed for Kyber1024

- $\mathbf{m} = [1, 0, \ldots, 0]$.
- $\mathbf{P}_B = [\lceil \frac{q}{32} \rfloor, 0, \ldots, 0]$
- $\mathbf{c}_1 = \textbf{Compress}_q(\mathbf{P}_B, 2^{d_{\mathbf{P}_B}})$
- $\mathbf{c}_2 = [h, 0, \ldots, 0]$

1

Mismatch Attack Idea Detailed for Kyber1024

- $\mathbf{m} = [1, 0, \ldots, 0]$.
- $\mathbf{P}_B = [\lceil \frac{q}{32} \rfloor, 0, \ldots, 0]$
- $\mathbf{c}_1 = \mathbf{Compress}_q(\mathbf{P}_B, 2^{d_{\mathbf{P}_B}})$
- $\mathbf{c}_2 = [h, 0, \ldots, 0]$

Alice' and Bob's keys match if and only if $\mathbf{m}'[0]$ and $\mathbf{m}[0] = 1$ match[1].

$$\begin{aligned}\mathbf{m}'[0] =& \mathbf{Compress}_q((\mathbf{v}_A - \mathbf{s}_A^{\mathrm{tr}}\mathbf{u}_A)[0], 2) \\ =& \mathbf{Compress}_q(\mathbf{v}_A[0] - (\mathbf{s}_A^{\mathrm{tr}}\mathbf{u}_A)[0], 2) \\ =& \left\lceil \frac{2}{q}\left(\left\lceil \frac{q}{32} h \right\rfloor - \mathbf{s}_A[0]\left\lceil \frac{q}{32} \right\rfloor\right)\right\rfloor \mod 2.\end{aligned}$$

[1] Minor tweaks make it possible for Eve to find $\mathbf{s}_A[i]$, for $i \neq 0$.

Selecting h for Mismatch Attacks on Kyber1024

Table: $\mathbf{m}'[0]$ as a function of $\mathbf{s}_A[0]$ for different values of h for Kyber1024.

	$\mathbf{s}_A[0]$				
h	-2	-1	0	1	2
7	1	0	0	0	0
8	1	1	0	0	0
9	1	1	1	0	0
10	1	1	1	1	0
22	0	1	1	1	1
23	0	0	1	1	1
24	0	0	0	1	1
25	0	0	0	0	1

Mismatch Attack on Kyber1024 [QZC+21]

	s_0				
	-2	-1	0	1	2
$16 \cdot P(s_0)$	1	4	6	4	1

Mismatch Attack on Kyber1024 [QZC+21]

s_0	-2	-1	0	1	2
$16 \cdot P(s_0)$	1	4	6	4	1

Mismatch Attack on Kyber1024 [QZC+21]

s_0	-2	-1	0	1	2
$16 \cdot P(s_0)$	1	4	6	4	1

Mismatch Attack on Kyber1024 [QZC+21]

s_0	-2	-1	0	1	2
$16 \cdot P(s_0)$	1	4	6	4	1

Our Mismatch Attacks in Two Dimensions

Allow the values of $\mathbf{m}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{P}_B$ to be non-zero for index $i = 0$ and/or $i = 128$.
Alice' and Bob's keys match if and only if $\mathbf{m}'[i]$ and $\mathbf{m}[i]$ match for $i = 0$ and $i = 128$.[2]

[2]Minor tweaks make it possible for Eve to find $\mathbf{s}_A[i]$ and $\mathbf{s}_A[i+128]$, for $i \neq 0$.

Our Mismatch Attacks in Two Dimensions

Allow the values of $\mathbf{m}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{P}_B$ to be non-zero for index $i = 0$ and/or $i = 128$.
Alice' and Bob's keys match if and only if $\mathbf{m}'[i]$ and $\mathbf{m}[i]$ match for $i = 0$ and $i = 128$.[2]

- $\mathbf{m}[0] = 1$ and/or $\mathbf{m}[128] = 1$.
- $\mathbf{P}_B[0] = b_1 \lceil \frac{q}{32} \rfloor, \mathbf{P}_B[128] = b_2 \lceil \frac{q}{32} \rfloor, b_1, b_2 \in \{-1, 0, 1\}$.
- $\mathbf{c}_1 = \textbf{Compress}_q(\mathbf{P}_B, 2^{d_{\mathbf{P}_B}})$
- $\mathbf{c}_2[0] = h_1, \mathbf{c}_2[128] = h_2$

[2] Minor tweaks make it possible for Eve to find $\mathbf{s}_A[i]$ and $\mathbf{s}_A[i + 128]$, for $i \neq 0$.

Our Mismatch Attacks in Two Dim. Cont.

$$
\begin{aligned}
\mathbf{m}'[0] &= \mathbf{Compress}_q(\mathbf{v}_A[0] - (\mathbf{s}_A^{\mathrm{tr}}\mathbf{u}_A)[0], 2) \\
&= \left\lceil \frac{2}{q}\left(\left\lceil \frac{q}{32} h_1 \right\rfloor - \left(\mathbf{s}_A[0] b_1 \left\lceil \frac{q}{32} \right\rfloor - \mathbf{s}_A[128] b_2 \left\lceil \frac{q}{32} \right\rfloor\right)\right)\right\rfloor \mod 2, \\
\mathbf{m}'[128] &= \mathbf{Compress}_q(\mathbf{v}_A[128] - (\mathbf{s}_A^{\mathrm{tr}}\mathbf{u}_A)[128], 2) \\
&= \left\lceil \frac{2}{q}\left(\left\lceil \frac{q}{32} h_2 \right\rfloor - \left(\mathbf{s}_A[0] b_2 \left\lceil \frac{q}{32} \right\rfloor + \mathbf{s}_A[128] b_1 \left\lceil \frac{q}{32} \right\rfloor\right)\right)\right\rfloor \mod 2.
\end{aligned}
$$

Planar Splits

$\mathbf{m}'[0]$	s_0 = -2	-1	0	1	2
-2	1	1	1	0	0
-1	1	1	1	0	0
s_{128} 0	1	1	1	0	0
1	1	1	1	0	0
2	1	1	1	0	0

(a) A vertical split.

$\mathbf{m}'[0]$	s_0 = -2	-1	0	1	2
-2	0	0	0	0	0
-1	0	0	0	0	0
s_{128} 0	0	0	0	0	0
1	1	1	1	1	1
2	1	1	1	1	1

(b) A horizontal split.

Rectangular Split

	$\mathbf{m}'[0]$	s_0 = -2	-1	0	1	2
	-2	0	0	0	1	1
	-1	0	0	0	1	1
s_{128}	0	0	0	0	1	1
	1	0	0	0	1	1
	2	0	0	0	1	1

(a) The vertical cut.

	$\mathbf{m}'[128]$	s_0 = -2	-1	0	1	2
	-2	1	1	1	1	1
	-1	1	1	1	1	1
s_{128}	0	1	1	1	1	1
	1	0	0	0	0	0
	2	0	0	0	0	0

(b) The horizontal cut.

	m'	s_0 = -2	-1	0	1	2
	-2	0	0	0	1	1
	-1	0	0	0	1	1
s_{128}	0	0	0	0	1	1
	1	0	0	0	0	0
	2	0	0	0	0	0

(c) The rectangular result.

Figure: The cuts with respect to $\mathbf{m}'[0]$, $\mathbf{m}'[128]$ and $m' = \mathbf{m}'[0] \& \mathbf{m}'[128]$.

Triangular Splits

$\mathbf{m}'[0]$ \ s_0	-2	-1	0	1	2
-2	0	1	1	1	1
-1	0	0	1	1	1
s_{128} 0	0	0	0	1	1
1	0	0	0	0	1
2	0	0	0	0	0

(a) A triangular cut of the secret values, originating from the upper right corner.

$\mathbf{m}'[0]$ \ s_0	-2	-1	0	1	2
-2	1	1	1	1	1
-1	1	1	1	1	1
s_{128} 0	1	1	1	1	0
1	1	1	1	0	0
2	1	1	0	0	0

(b) A triangular cut of the secret values, originating from the upper left corner.

Intersecting Triangular Splits

	$\mathbf{m}'[0]$ \ s_0	-2	-1	0	1	2
	-2	1	1	1	1	1
	-1	1	1	1	1	1
s_{128}	0	1	1	1	1	0
	1	1	1	1	0	0
	2	1	1	0	0	0

(a) First triangular cut

	$\mathbf{m}'[128]$ \ s_0	-2	-1	0	1	2
	-2	0	1	1	1	1
	-1	0	0	1	1	1
s_{128}	0	0	0	0	1	1
	1	0	0	0	0	1
	2	0	0	0	0	0

(b) Second triangular cut

	m' \ s_0	-2	-1	0	1	2
	-2	0	1	1	1	1
	-1	0	0	1	1	1
s_{128}	0	0	0	0	1	0
	1	0	0	0	0	0
	2	0	0	0	0	0

(c) The intersection

Figure: The cuts with respect to $\mathbf{m}'[0]$, $\mathbf{m}'[128]$ and $m' = \mathbf{m}'[0]\&\mathbf{m}'[128]$.

Mismatch Attack on Kyber1024 in Two Dim.

$256 \cdot P(s_0, s_{128})$		s_0: -2	-1	0	1	2
	-2	1	4	6	4	1
	-1	4	16	24	16	4
s_{128}	0	6	24	36	24	6
	1	4	16	24	16	4
	2	1	4	6	4	1

Mismatch Attack on Kyber1024 in Two Dim.

$256 \cdot P(s_0, s_{128})$	s_0: -2	-1	0	1	2
-2	1	4	6	4	1
-1	4	16	24	16	4
s_{128} 0	6	24	36	24	6
1	4	16	24	16	4
2	1	4	6	4	1

Mismatch Attack on Kyber1024 in Two Dim.

$256 \cdot P(s_0, s_{128})$	s_0 = -2	-1	0	1	2
s_{128} = -2	1	4	6	4	1
-1	4	16	24	16	4
0	6	24	36	24	6
1	4	16	24	16	4
2	1	4	6	4	1

Mismatch Attack on Kyber1024 in Two Dim.

$256 \cdot P(s_0, s_{128})$	s_0: -2	-1	0	1	2
-2	1	4	6	4	1
-1	4	16	24	16	4
s_{128} 0	6	24	36	24	6
1	4	16	24	16	4
2	1	4	6	4	1

Mismatch Attack on Kyber1024 in Two Dim.

	s_0				
$256 \cdot P(s_0, s_{128})$	-2	-1	0	1	2
-2	1	4	6	4	1
-1	4	16	24	16	4
s_{128} 0	6	24	36	24	6
1	4	16	24	16	4
2	1	4	6	4	1

Results and Comparisons

	Kyber512	Kyber768	Kyber1024	LightSaber	Saber	FireSaber
[QZC+21]	1312	1776	2368	1460	2091	2624
Huffman Bound 1	1216	1632	2176	1412	1986	2432
Our Result 1	**1205.3**	**1588.5**	**2118**	-	-	**2410.6**
Our Result 2	1217.7	1599	2132	**1410.2**	**1984.9**	2435.4
Huffman Bound 2	1202.1	1575	2100	1395.9	1970.0	2404.3
Huffman Bound 3	1199.9	1569.8	2093.0	1391.7	1962.3	2399.7
Shannon Bound	1195	1560	2079	1386	1954	2389

Mismatch Attack Plus Lattice Reduction[3]

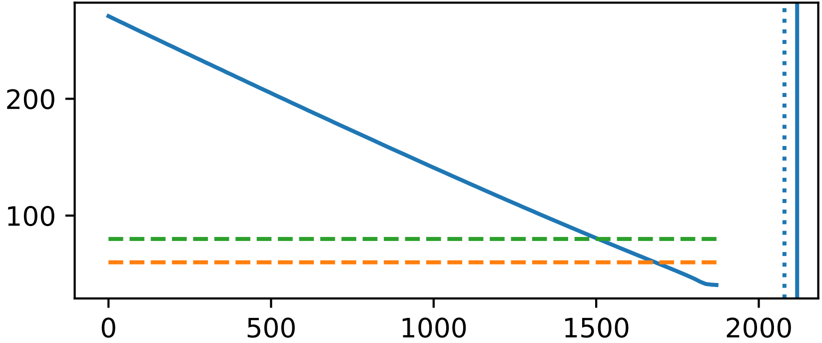

Figure: Complexity to break Kyber1024 as a function of # mismatch attacks queries.

[3] Studied concurrently and independently in `https://eprint.iacr.org/2022/1064`.

Recent Large Improvement

- In a very recent work[4] our method got improved - by a lot!

[4] `https://eprint.iacr.org/2023/887`

Recent Large Improvement

- In a very recent work[4] our method got improved - by a lot!
- Instead of gaining up to 1 bit per query, the authors can get up to p bits per query, at a computational cost of $\mathcal{O}(2^p)$.

[4] https://eprint.iacr.org/2023/887

Recent Large Improvement

- In a very recent work[4] our method got improved - by a lot!
- Instead of gaining up to 1 bit per query, the authors can get up to p bits per query, at a computational cost of $\mathcal{O}(2^p)$.
- At a very modest computational cost they reduce the query complexity by around 95 %!

[4] https://eprint.iacr.org/2023/887

Recent Large Improvement

- In a very recent work[4] our method got improved - by a lot!
- Instead of gaining up to 1 bit per query, the authors can get up to p bits per query, at a computational cost of $\mathcal{O}(2^p)$.
- At a very modest computational cost they reduce the query complexity by around 95 %!
- The main reviewer complaint about our paper was its incremental improvement - interestingly it inspired a method for a huge improvement!

[4] https://eprint.iacr.org/2023/887

Recent Large Improvement

- In a very recent work[4] our method got improved - by a lot!
- Instead of gaining up to 1 bit per query, the authors can get up to p bits per query, at a computational cost of $\mathcal{O}(2^p)$.
- At a very modest computational cost they reduce the query complexity by around 95 %!
- The main reviewer complaint about our paper was its incremental improvement - interestingly it inspired a method for a huge improvement!
- Their attack is similar to (and applies to) parallel PC oracle attacks [GPDA+23,TUX23]

[4] `https://eprint.iacr.org/2023/887`

Open Questions

- Can the recent improvement of our work[5] be further improved?

[5] https://eprint.iacr.org/2023/887

Open Questions

- Can the recent improvement of our work[5] be further improved?
- What can be achieved for other lattice-based schemes like NewHope, Frodo, etc.?

[5] https://eprint.iacr.org/2023/887