A Systematic Approach and Analysis of Key Mismatch Attacks on Lattice-Based NIST Candidate KEMs

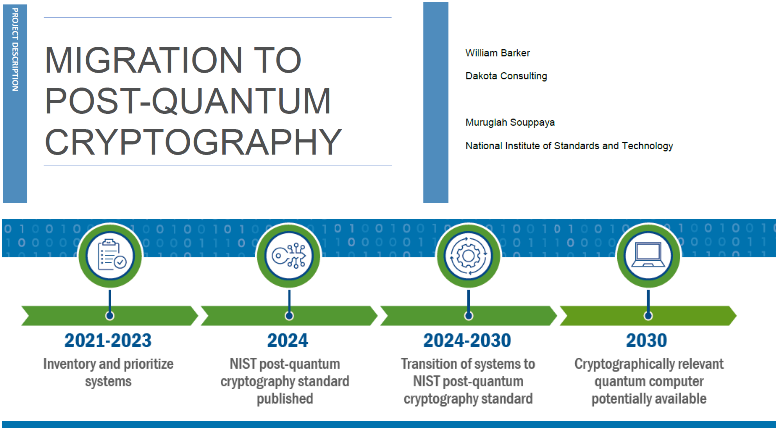
Xiaohan Zhang

School of Computer Science
China University of Geosciences (Wuhan)

Joint work with Yue Qin, Prof. Chi Cheng, Prof. Yanbin Pan,
Prof. Lei Hu and Prof. Jintai Ding

November 30, 2021

- NIST and Department of Homeland Security (DHS): a migration roadmap to PQC.

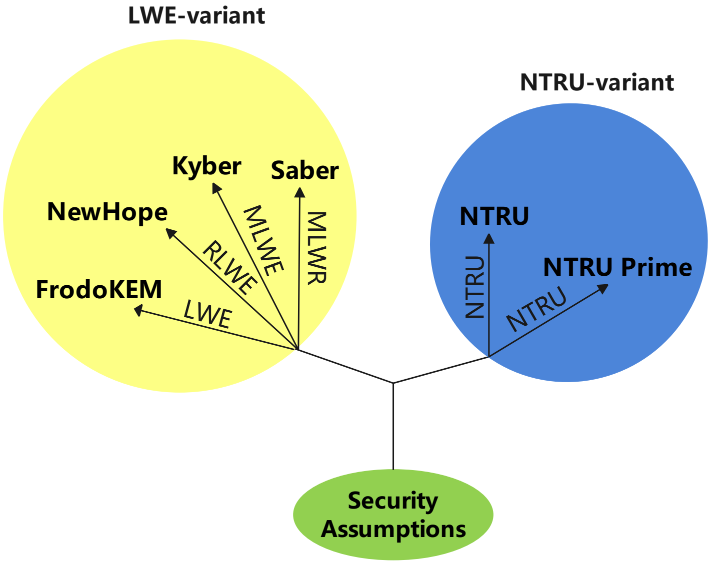


Third Round PQC Standardization

	Encryption/KEMs	**Signatures**	**Overall**
Lattice-based	**5**	2	7
Code-based	3	0	3
Isogeny-based	1	0	1
Multivariate-based	0	2	2
Symmetric-based	0	2	2
Total	9	6	15

- Lattice-based KEM finalists: KYBER, SABER, NTRU
- Lattice-based KEM alternates: FrodoKEM, NTRUprime

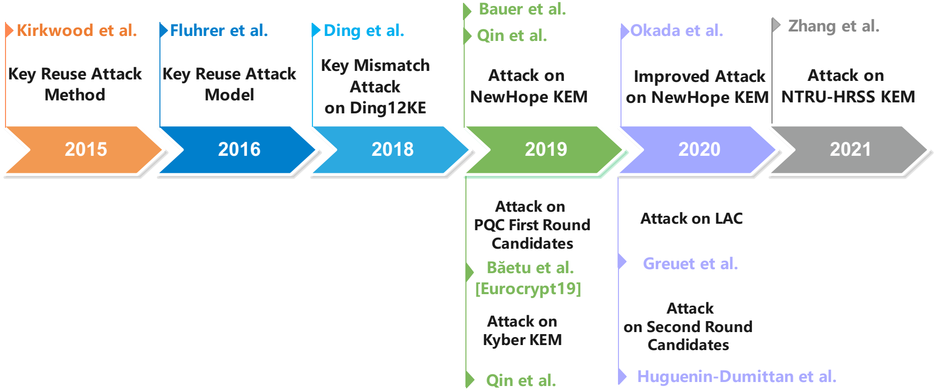
Security Assumption of Lattice-based KEMs



- Two flavours: IND-CPA and IND-CCA PKC.
- IND-CPA $\xrightarrow{FO\ transform}$ IND-CCA
- The IND-CPA version does not allow key-reuse but simpler or more efficient.
  - What will happen if a key is reused in the IND-CPA version?

1. For cryptographic assessment, it is important to evaluate key-reuse resilience of these candidates in misuse situation.
2. In many authentication key exchange protocols that use CPA version to improve efficiency, key reuse is essential.
3. Side-channel assisted chosen ciphertexts attacks can successfully attack against CCA-secure ones.

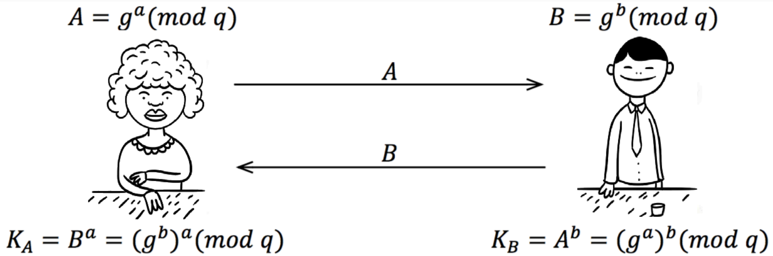
Key Mismatch Attacks: A Brief History



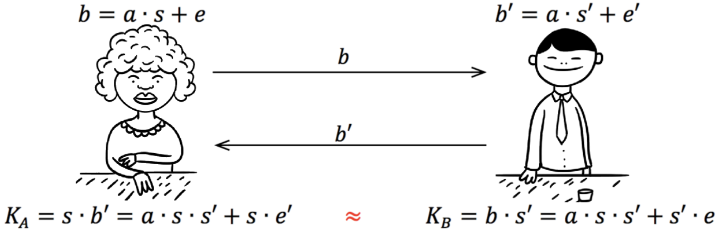
An Open Problem

- Can we find a unified method to evaluate the key reuse resilience (number of queries) of NIST candidates against key mismatch attacks?

Diffie-Hellman Key Exchange

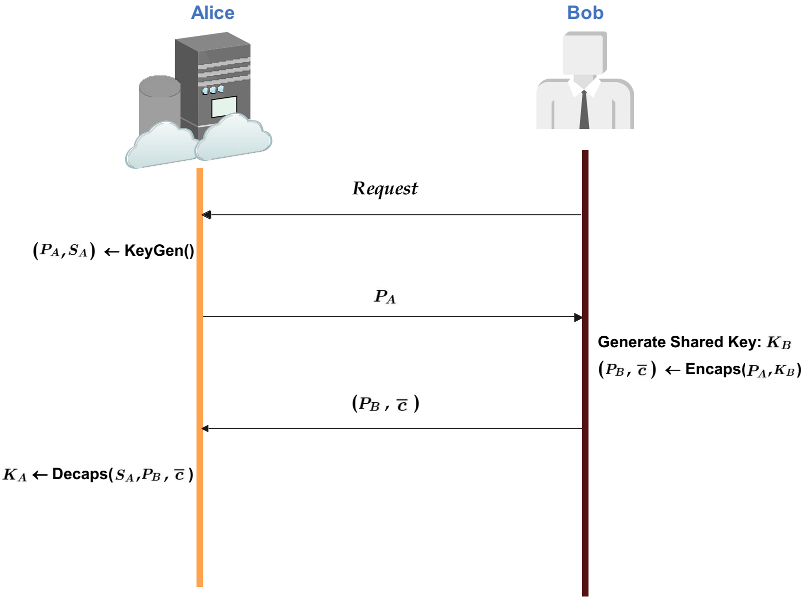


Lattice-based Diffie-Hellman-like Key exchange

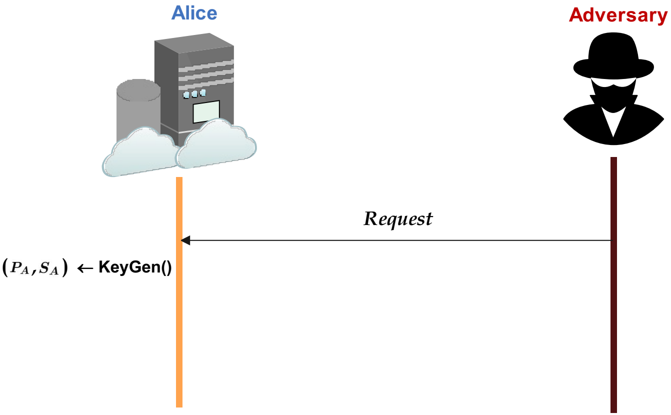


- The biggest challenge: How to make the approximate K_A and K_B equal?
- Solution: send additional information

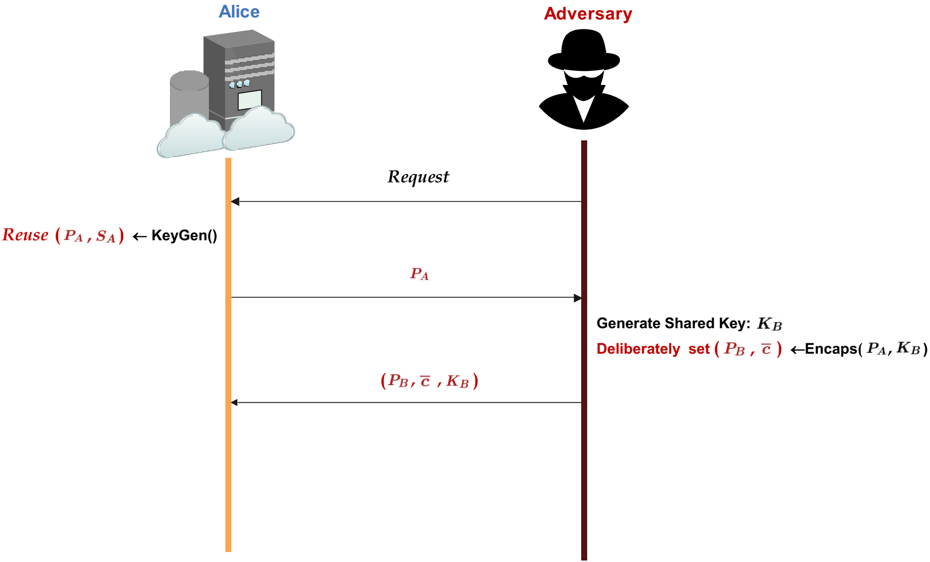
The Meta CPA-secure KEM



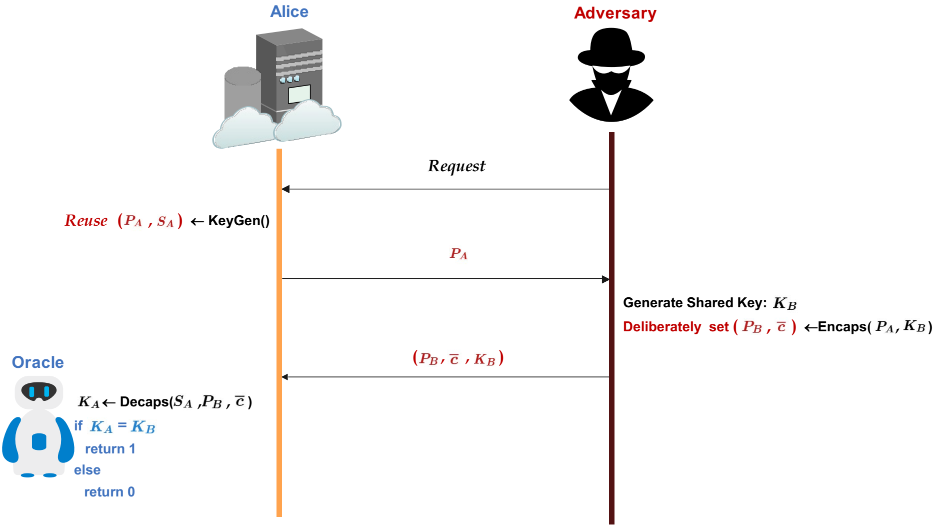
Model of Key Mismatch Attack – Part 1



Model of Key Mismatch Attack – Part 2



Model of Key Mismatch Attack – Part 3



- Alice's public-secret key pair is reused.
- The adversary $\mathcal{A}$ can recover Alice's secret key by knowing whether the shared two keys match or not.
  - the shared two keys $K_A = K_B \rightarrow$ Match
  - the shared two keys $K_A \neq K_B \rightarrow$ Mismatch

A Response to the Open Problem

- Can we find a unified method to evaluate the key reuse resilience of NIST candidates against key mismatch attacks?

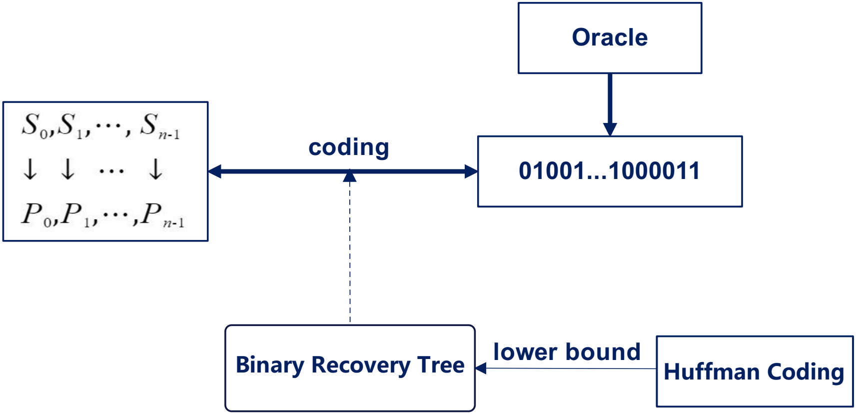
- Can we find a unified method to evaluate the key reuse resilience of NIST candidates against key mismatch attacks?

✔ YES!

Notations

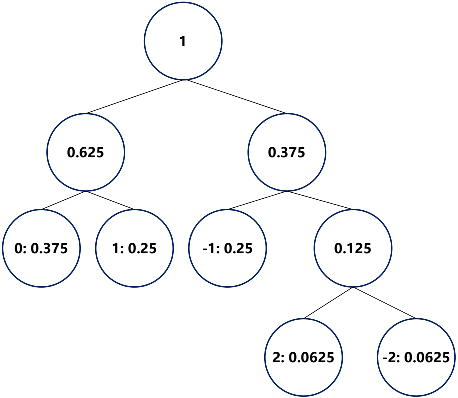
- $\mathcal{A}$ recovers Alice's secret key $\boldsymbol{S}_A$ one coefficient block by one coefficient block.
- Let $\boldsymbol{S} = \{\boldsymbol{S}_0, \boldsymbol{S}_1, \cdots, \boldsymbol{S}_{n-1}\}$ be the set of all possible values for one coefficient block.
- $\{P_0, P_1, \cdots, P_{n-1}\}$ is the corresponding probability set, where $P_0 \geq P_1 \geq \cdots \geq P_{n-1}$, $\sum_{i=0}^{n-1} P_i = 1$.

- Average #queries: $E(\boldsymbol{S}) = \sum_{i=0}^{n-1} P_i \cdot \text{depth}_T(\boldsymbol{S}_i)$.
- How to recover $\boldsymbol{S}_A$ with the fewest number of queries?
  $\Rightarrow$ Transfer it into a binary variable-length coding problem
- Basic idea: Using Huffman Coding to get min $E(\boldsymbol{S})$.



Huffman Coding

- **Rule:** Combine two symbols with the lowest probabilities in each step.
- $\boldsymbol{S} = \{0, \pm 1, \pm 2\}$, the probability $= \{0.375, 0.25, 0.0625\}$.



Theorem 1

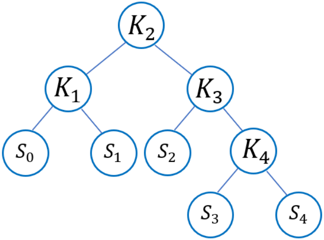
Let $\boldsymbol{S} = \{\boldsymbol{S}_0, \boldsymbol{S}_1, \cdots, \boldsymbol{S}_{n-1}\}$, its corresponding probabilities $\{P_0, P_1, \cdots, P_{n-1}\}$. And set $H(\boldsymbol{S})$ the Shannon entropy for $\boldsymbol{S}$, then we have

$$H(\boldsymbol{S}) \leq \min E(\boldsymbol{S}) < H(\boldsymbol{S}) + 1.$$

Example: Lower Bound for Kyber1024

- In Kyber1024, $\mathbf{S}_A$ is sampled from centered binomial distribution, and $\mathbf{S}_A[i] \in [-2, 2]$.
- min $E(\boldsymbol{S}) = 2.125$, $H(\boldsymbol{S}) = 2.03$, consistent with Theorem 1.
- Lower bound: 2176

l_{rs}	rs	$\mathbf{S}_i$	Probability				
2	11	0	0.375	0.375	0.375	0.625	1
2	10	1	0.25	0.25	0.375	0.375	
2	01	-1	0.25	0.25	0.25		
3	001	2	0.0625	0.125			
3	000	-2	0.0625				



Lower bounds for key mismatch attacks on lattice-based NIST KEMs

Schemes	$\mathbf{s}_A$ & $\mathbf{e}$ Ranges	Encode Decode	Comp Decomp	Unknowns	E(#Queries) Bounds
Newhope512	[-8,8]	✓	✓	512	1568
Newhope1024				1024	3127
Kyber512	[-3,3]	/	✓	512	1216
Kyber768	[-2,2]			768	1632
Kyber1024				1024	2176
LightSaber	[-5,5]	/	✓	512	1412
Saber	[-4,4]			768	1986
FireSaber	[-3,3]			1024	2432
Frodo640	[-12,12]	/	✓	5120	18,227
Frodo976	[-10,10]			7808	25,796
Frodo1344	[-6,6]			10,752	27,973
NTRU hps4096821	[-1,1]	/	/	821	1369
NTRU hrss701				701	1183
NTRU Prime sntrup857				857	1574
NTRU Prime ntrulpr857				857	1553

Overview

	Huguenin-Dumittan et al.'s	Lower Bounds	Gap
LightSaber	2048	1412	31.05%
Frodo640	65536	18227	72.19%

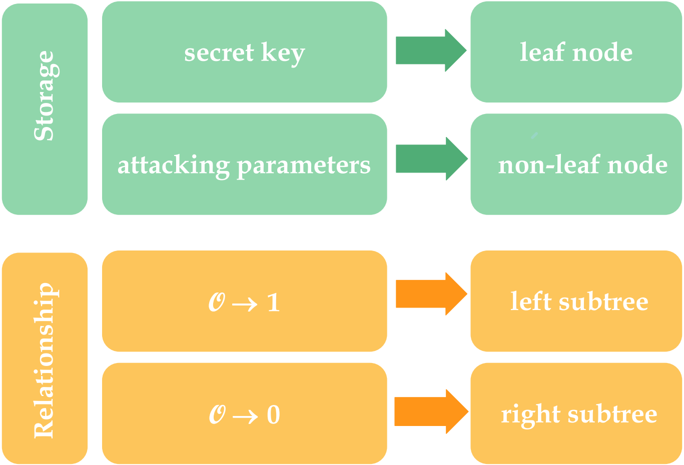
- A huge gap in terms of # queries between existing attacks and lower bounds
- Huffman Tree guides us to improve these attacks

On the basis of Huffman Tree

1. **Pre-computation phase:** $\mathcal{A}$ selects proper parameters and constructs a corresponding Binary Recovery Tree (BRT) T in consistent with the Huffman tree.
2. **Recovery phase:** $\mathcal{A}$ determines the secret key according to the precomputed binary tree T.

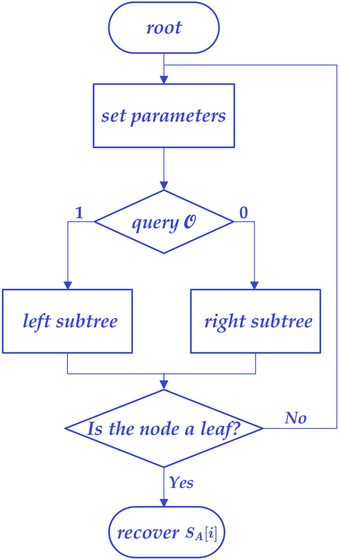
Description of Pre-computation phase

How to construct the BRT T ?



1. Use all possible secret keys as leaf nodes.
2. Non-leaf nodes store the parameters that the adversary use to access Oracle.
3. For each non-leaf node, if the Oracle returns 1, it corresponds to the left subtree of the current node, otherwise it corresponds to its right subtree.

How to use the BRT T to recover the secret key?



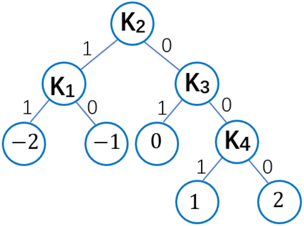
Description of Recovery phase

1. The adversary $\mathcal{A}$ starts from the root of T, and selects the parameter in this node to access Oracle.
2. If Oracle returns 1, $\mathcal{A}$ will continue to access the left subtree of the current node, otherwise he will access the right subtree.
3. If the current node is a leaf node, $\mathcal{A}$ can determine the secret key.

1. The pre-computation phase

   1. $\mathcal{A}$ sets $\mathbf{m}$ as (1,0, $\cdots$, 0).
   2. Then he sets $\mathbf{P}_B = \mathbf{0}$ and $\mathbf{P}_B[0] = \lceil \frac{q}{32} \rfloor$.
   3. After that, $\mathcal{A}$ sets $\mathbf{c}_2 = \mathbf{0}$ and $\mathbf{c}_2[0] = h$.

	State 1	State 2	State 3	State 4
h	8	9	10	7
$\mathcal{O} \to 0$	State 2	State 3	$\mathbf{S}_A[0] = 2$	$\mathbf{S}_A[0] = -1$
$\mathcal{O} \to 1$	State 4	$\mathbf{S}_A[0] = 0$	$\mathbf{S}_A[0] = 1$	$\mathbf{S}_A[0] = -2$



Improved key mismatch attacks on Kyber KEM

	Existing Attacks	Improved Attacks	Lower bounds	Success rate
Kyber1024	2475	2368	2176	100%
Kyber768	1855	1777	1632	100%
Kyber512	1401 (Round 2)	1311	1216	100%

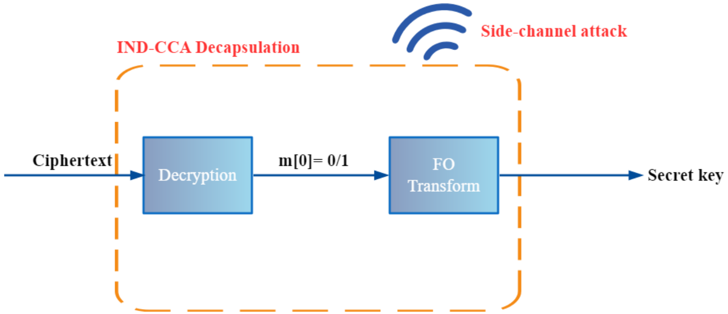
Main idea: Construct a Nearly Optimal Binary Search Tree T'.

- T' should satisfy:
  1. For each non-leaf node, the probability of left subtree and right subtree should be as equal as possible.
  2. If the Oracle returns 1, it corresponds to its left subtree, otherwise it corresponds to its right subtree.

	Okada et. al's	Vacek et. al's	Our improved attacks	Lower bounds
NewHope1024	233,803	3197	3180	3127
NewHope512	\	\	1660	1568
Success rate	97.4%	100%	100%	\

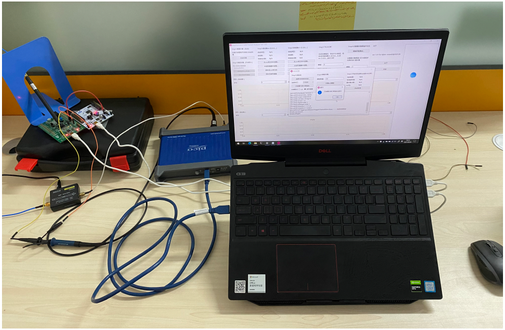
- The gap between our improved attacks and the lower bounds is 1.69% and 5.86%, respectively

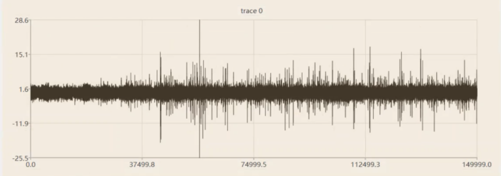
- At CHES 2020, Ravi et al. proposed a generic side-channel attack on CCA-secure KEMs.



- Their side-channel attack mainly consists of two stages:
  1. **pre-processing stage:** generate template for each class
     - $\Gamma_0 \Leftrightarrow$ failure of KEM.CCA.Dec()
     - $\Gamma_1 \Leftrightarrow$ success of KEM.CCA.Dec()
  2. **template-matching stage:** collect wave $\mathcal{W}$ and distinguish which class $\mathcal{W}$ belongs to.
- The same as our proposed key mismatch attack aforementioned

- E.g. TVLA analyzer for Kyber512 (Template Matching)





Improved side-channel attacks against IND-CCA KEMs

	Ravi et. al's	Our improved attacks
Kyber512	2560	1311
NewHope512	6945	1660
NewHope1024	26624	3180

- On Kyber512, we reduce E(#Queries) by 48.79%.
- Similarly, we reduce E(#Queries) for NewHope512 and NewHope1024 by 76.1% and 88.06%, respectively.

Experimental Setup

- **Environment:** A computer with two 3 GHz Intel Xeon E5-2620 CPUs and a 64 GB RAM.
- Our code is available at https://github.com/AHaQY/Key-Mismatch-Attack-on-NIST-KEMs.

Experimental Results

Schemes	E(#Queries)			
	Lower Bounds	**Our improved attacks**		*Existing*
		Theory	Experiments	
Kyber512	1216	**1312**	1311	*1401* (Round 2)
Kyber768	1632	**1774**	1777	*1855*
Kyber1024	2176	**2365**	2368	*2475*
LightSaber	1412	**1460**	1476	*2048*
Saber	1986	**2091**	2095	-
FireSaber	2432	**2642**	2622	-
Frodo640	18,227	**18,329**	18,360	*65, 536*
Frodo976	25,796	**26,000**	26,078	-
Frodo1344	27,973	**29,353**	29,378	-
NewHope512	1568	**1660**	1660	-
NewHope1024	3127	**3180**	3180	*3197*
NTRU hps2048509	846	-	1012	-
NTRU hps2048761	1125	-	1348	-
NTRU hps4096821	1365	-	1634	-
NTRU hrss701	1183	-	1844	-

- For Frodo640 and LightSaber, E(#Queries) is reduced by 71.99% and 27.93%.

Conclusion

1. Lower bounds for all the lattice-based KEMs
2. Our BRT method to further optimize the key mismatch attacks
3. Optimizing side-channel attacks against IND-CCA secure KEMs.

References

- Kirkwood et al. "Failure is not an option: Standardization issues for post-quantum key agreement." Workshop on Cybersecurity in a Post-Quantum World (2015).
- Fluhrer et al. "Cryptanalysis of ring-LWE based key exchange with key share reuse." Cryptology ePrint Archive (2016).
- Ding et al. "Complete attack on RLWE key exchange with reused keys, without signal leakage." Australasian conference on information security and privacy. Springer (2018).
- Bauer et al. "Assessment of the key-reuse resilience of NewHope." Cryptographers' track at the RSA conference. Springer (2019).
- Qin et al. "A complete and optimized key mismatch attack on NIST candidate NewHope." European symposium on research in computer security. Springer (2019).
- Băetu et al. "Misuse attacks on post-quantum cryptosystems." Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer (2019).
- Qin et al. "An efficient key mismatch attack on the NIST second round candidate Kyber." Cryptology ePrint Archive (2019).

References (contd.)

- Okada et al. "Improving key mismatch attack on NewHope with fewer queries." Australasian Conference on Information Security and Privacy. Springer (2020).
- Greuet et al. "Attack on LAC key exchange in misuse situation." International Conference on Cryptology and Network Security. Springer (2020).
- Huguenin-Dumittan et al. "Classical misuse attacks on NIST round 2 PQC." International Conference on Applied Cryptography and Network Security. Springer (2020).
- Zhang et al. "Small Leaks Sink a Great Ship: An Evaluation of Key Reuse Resilience of PQC Third Round Finalist NTRU-HRSS." International Conference on Information and Communications Security. Springer (2021).
- Ravi et al. "Generic Side-channel attacks on CCA-secure lattice-based PKE and KEMs." IACR Trans. Cryptogr. Hardw. Embed. Syst. 2020.3 (2020): 307-335.

Thanks & Questions?