Fault Attacks on CCA-secure Lattice KEMs

Peter Pessl[1], Lukas Prokop[2]

[1]Infineon Technologies
[2]Graz University of Technology

What's that all about?

- NIST's PQC standardization process: 3rd (and final?) round
  - 3 lattice-based KEMs (Kyber, NTRU, Saber)

What's that all about?

- NIST's PQC standardization process: 3rd (and final?) round
  - 3 lattice-based KEMs (Kyber, NTRU, Saber)
- Embedded devices and implementation security
  - finalists are a nice fit, but performance isn't everything
  - devices in the field, accessible to attackers: implementation attacks

Status: Implementation Security of Lattice KEMs

- Lattice KEMs and side-channels
  - various attacks published (DPA, algebraic, side-channel assisted CCA)
  - first protected (masked) implementations available

Status: Implementation Security of Lattice KEMs

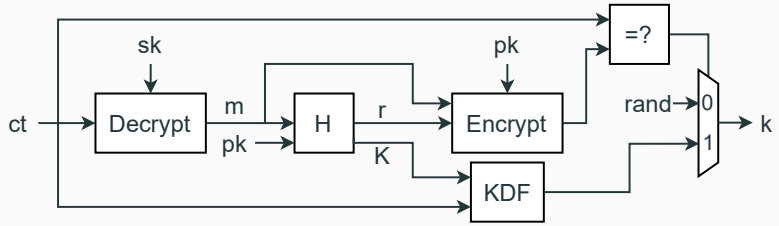
- Lattice KEMs and side-channels
  - various attacks published (DPA, algebraic, side-channel assisted CCA)
  - first protected (masked) implementations available
- Lattice KEMs and faults
  - very little published work on attacks
  - no public information on how to achieve protection

Kyber, Saber: High-Level Similarities

- Descendants of scheme by Lyubashevsky, Peikert, Rosen (LPR), Eurocrypt 2010
  - CPA-secure public-key encryption based on Ring-LWE

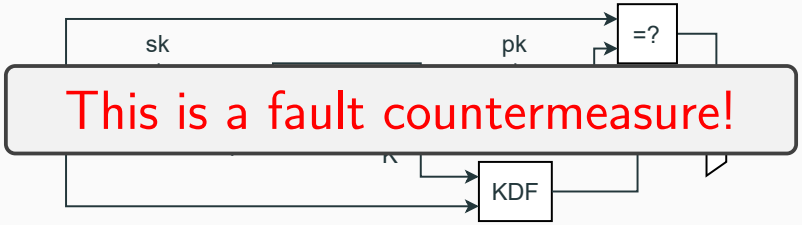
Kyber, Saber: High-Level Similarities

- Descendants of scheme by Lyubashevsky, Peikert, Rosen (LPR), Eurocrypt 2010
  - CPA-secure public-key encryption based on Ring-LWE
- Fujisaki-Okamoto transform: CPA-secure PKE $\rightarrow$ CCA-secure KEM
  - re-encrypt a plaintext after decryption, accept if ciphertexts match



Kyber, Saber: High-Level Similarities

- Descendants of scheme by Lyubashevsky, Peikert, Rosen (LPR), Eurocrypt 2010
  - CPA-secure public-key encryption based on Ring-LWE
- Fujisaki-Okamoto transform: CPA-secure PKE $\rightarrow$ CCA-secure KEM
  - re-encrypt a plaintext after decryption, accept if ciphertexts match



Our Attack

- Show that attacks are still possible and practical
  - FO is a fault deterrent, but not a countermeasure

Our Attack

- Show that attacks are still possible and practical
  - FO is a fault deterrent, but not a countermeasure
- Fault at a specific spot
  - observe: does the decapsulation fail or still return the correct value?
  - outcome carries information on the private key
  - gather information over many invocations (fault injections), solve for the key

Our Attack

- Show that attacks are still possible and practical
  - FO is a fault deterrent, but not a countermeasure
- Fault at a specific spot
  - observe: does the decapsulation fail or still return the correct value?
  - outcome carries information on the private key
  - gather information over many invocations (fault injections), solve for the key
- Attack Kyber and NewHope, minimum 6 500 faults
  - potential to extend to other schemes

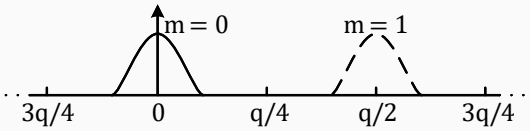
The LPR Encryption Scheme: "Noisy ElGamal"

KEYGEN()	ENCRYPT($pk = (a, b), m$)	DECRYPT($c = (u, v), sk = s$)
$s, e \in \mathcal{R}_q \leftarrow \chi^n$	$r, e_1, e_2 \in \mathcal{R}_q \leftarrow \chi^n$	$m' = v - us$
$a \in \mathcal{R}_q \leftarrow \mathcal{U}_q^n$	$u = ar + e_1$	**return** DECODE(m')
$b = as + e$	$v = br + e_2 + m \cdot \lfloor q/2 \rfloor$	
return $(pk = (a, b), sk = s)$	**return** $c = (u, v)$	

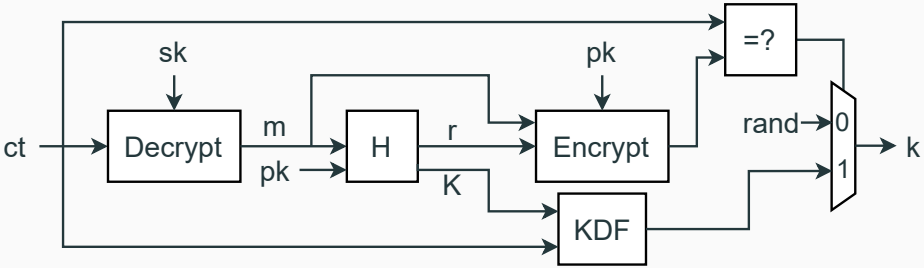
- Elements in $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$
- Narrow error distribution χ
- Encode each message bit onto one coefficient by multiplying with $\lfloor q/2 \rfloor$

Correctness and Decoding

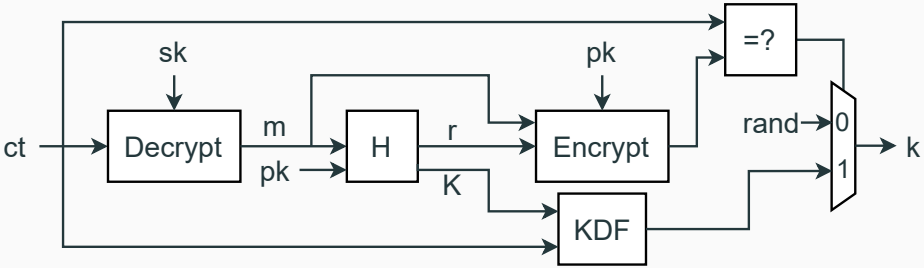
- Backsubstitution: $m' = v - us = \ldots = m \cdot \lfloor \frac{q}{2} \rfloor + re - e_1 s + e_2 \approx m \cdot \lfloor \frac{q}{2} \rfloor$
- Recover m from m': Decoder



Attacking an FO-KEM

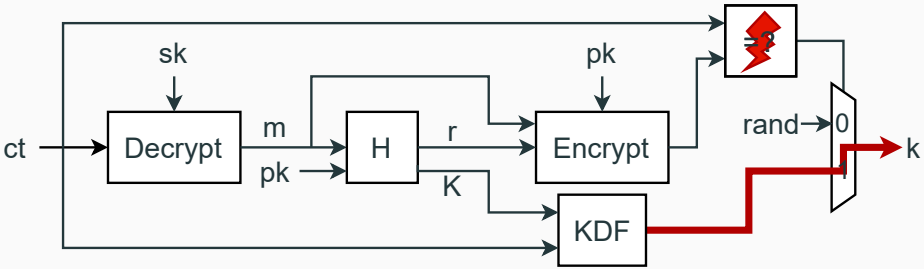


Attacking an FO-KEM



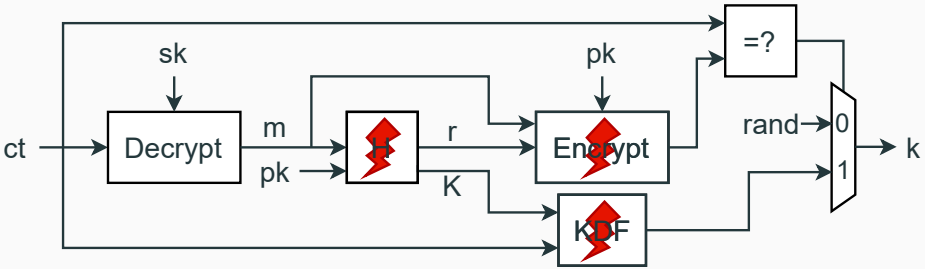
Where can we inject a fault?

Attacking an FO-KEM



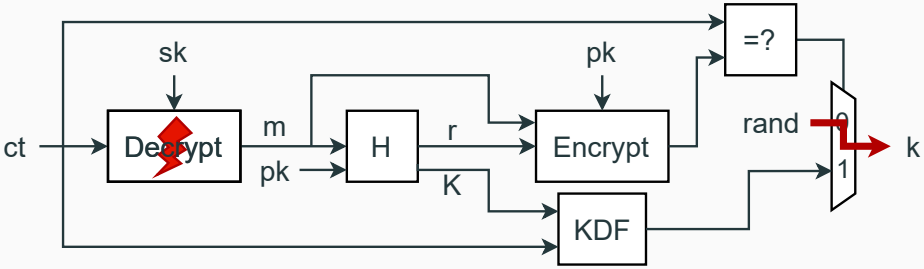
Fault the check (re-enable CCA)? Already known...

Attacking an FO-KEM



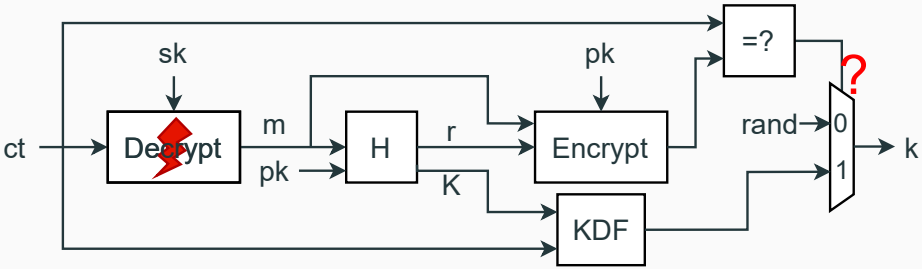
Processing only known data (if attacker generated ct)

Attacking an FO-KEM



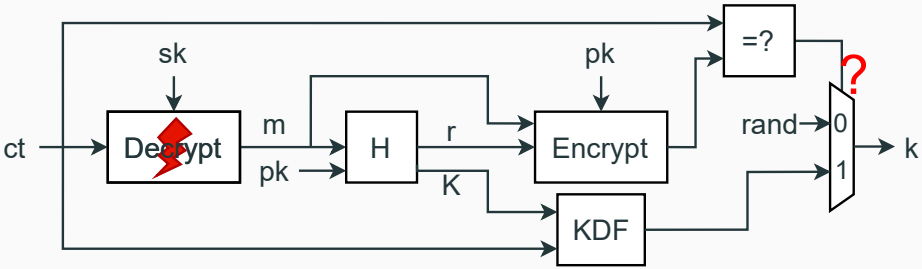
Problem: if m changed, then k always random

Attacking an FO-KEM



only recoverable information: is k correct or random?

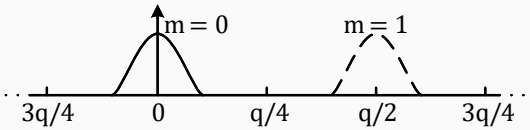
Attacking an FO-KEM



correct k vs. random $k \rightarrow$ information on key

(In)Effective Faults in the Decoder

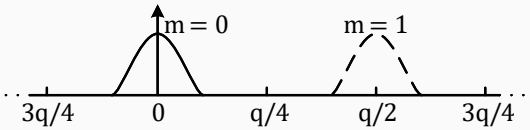
- Decoder: recover message m from "noisy" m', for each coefficient
  - input: $[0, q)$, output: $\{0, 1\}$
- Assumption: attacker runs encaps. and sends ct (knows k and intermediates)
  - $m' = m \cdot \left\lfloor \frac{q}{2} \right\rfloor + re - e_1 s + e_2$



(In)Effective Faults in the Decoder

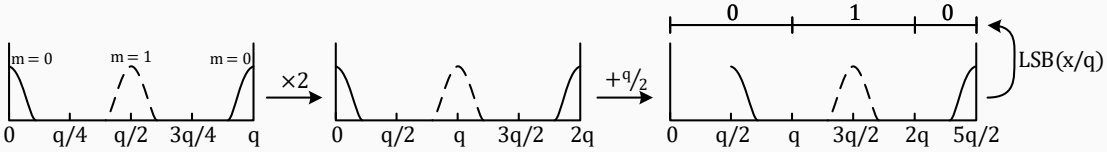
- Decoder: recover message m from "noisy" m', for each coefficient
  - input: $[0, q)$, output: $\{0, 1\}$

- Assumption: attacker runs encaps. and sends ct (knows k and intermediates)
  - $m' = m \cdot \left\lfloor \frac{q}{2} \right\rfloor + re - e_1 s + e_2 \rightarrow$ linear in the key (e, s)



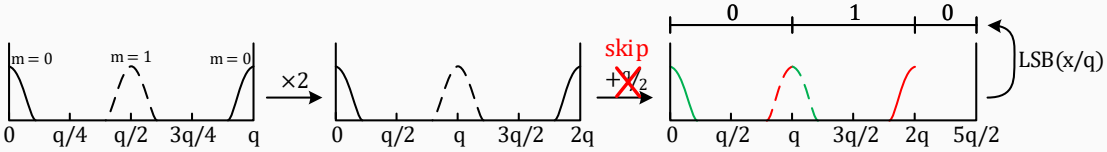
Faulting the Decoder

```
uint16_t t = (((a << 1) + KYBER_Q/2) / KYBER_Q) & 1;
```



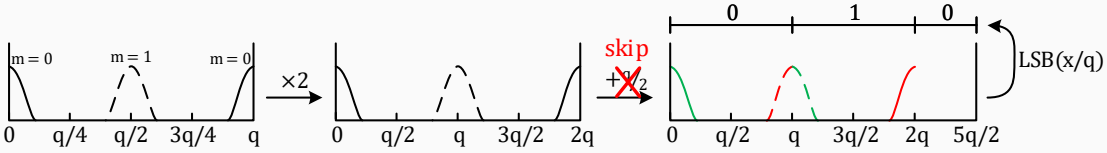
Faulting the Decoder

```
uint16_t t = (((a << 1) + KYBER_Q/2) / KYBER_Q) & 1;
```



Faulting the Decoder

```
uint16_t t = (((a << 1) + KYBER_Q/2) / KYBER_Q) & 1;
```



- Fault decoding of coefficient i:

$$(re - e_1 s + e_2)[i] \begin{cases} \geq 0 & \text{if fault was ineffective (correct } k\text{)} \\ < 0 & \text{if fault was effective (random } k\text{)} \end{cases}$$

Solving for the Key

- Gather inequalities
  - send many ct, fault one coeff. per call

$$\begin{pmatrix} r_1 & -e_{1,1} \\ r_2 & -e_{1,2} \\ & \vdots \\ r_m & -e_{1,m} \end{pmatrix} \begin{pmatrix} e \\ s \end{pmatrix} \begin{bmatrix} < \\ \geq \\ \vdots \\ < \end{bmatrix} \begin{pmatrix} -e_{2,1} \\ -e_{2,2} \\ \vdots \\ -e_{2,m} \end{pmatrix}$$

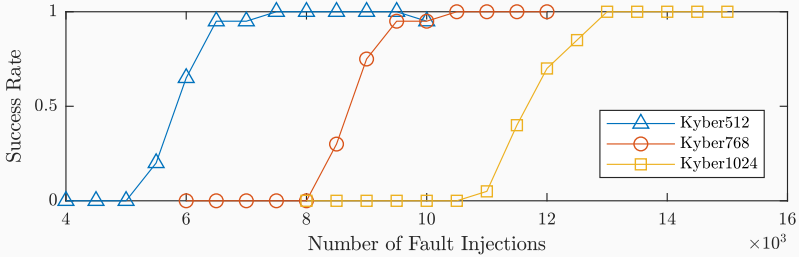
Solving for the Key

- Gather inequalities
  - send many ct, fault one coeff. per call
- Similarity to linear decoding
  - use technique similar to *belief propagation*

$$\begin{pmatrix} r_1 & -e_{1,1} \\ r_2 & -e_{1,2} \\ & \vdots \\ r_m & -e_{1,m} \end{pmatrix} \begin{pmatrix} e \\ s \end{pmatrix} \begin{bmatrix} < \\ \geq \\ \vdots \\ < \end{bmatrix} \begin{pmatrix} -e_{2,1} \\ -e_{2,2} \\ \vdots \\ -e_{2,m} \end{pmatrix}$$

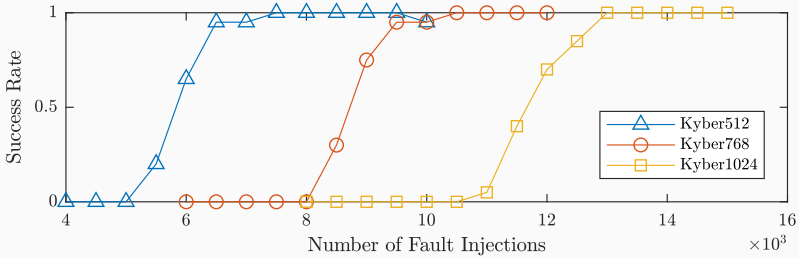
Results

- Attack implemented for Kyber and NewHope
  - + a masked decoder implementation
  - success rate via simulations



Results

- Attack implemented for Kyber and NewHope
  - + a masked decoder implementation
  - success rate via simulations
- Verified attack by attacking a microcontroller
  - ARM Cortex M4 running Kyber512, attacked with clock glitches



Conclusion

- Demonstrated practical attacks
  - FO might be a fault deterrent, but attacks are still possible!
  - Kyber + NewHope, but similar attacks for other schemes likely exist
  - also other faulting positions, other techniques, etc.

https://github.com/latticekemfaults/latticekemfaults/

Thank you!