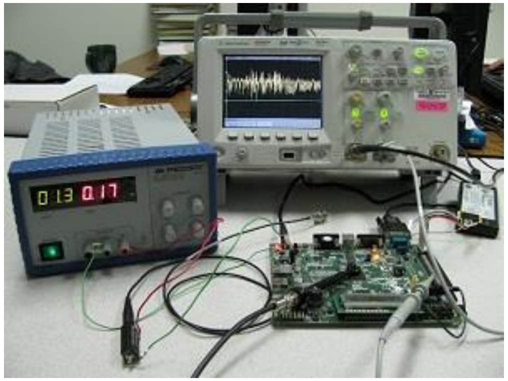
Towards Leakage-Resistant Post-Quantum CCA-Secure Public Key Encryption



C. Hoffmann, B. Libert, C. Momin, **T. Peters, F.-X. Standaert**

UCLouvain (Belgium), CNRS (France), Shandong U. (China)

4th NIST PQC Standardization Conference

November 29, 2022

Outline

- Introduction/motivation
  - Parallel with symmetric crypto
  - Challenge for PQ crypto

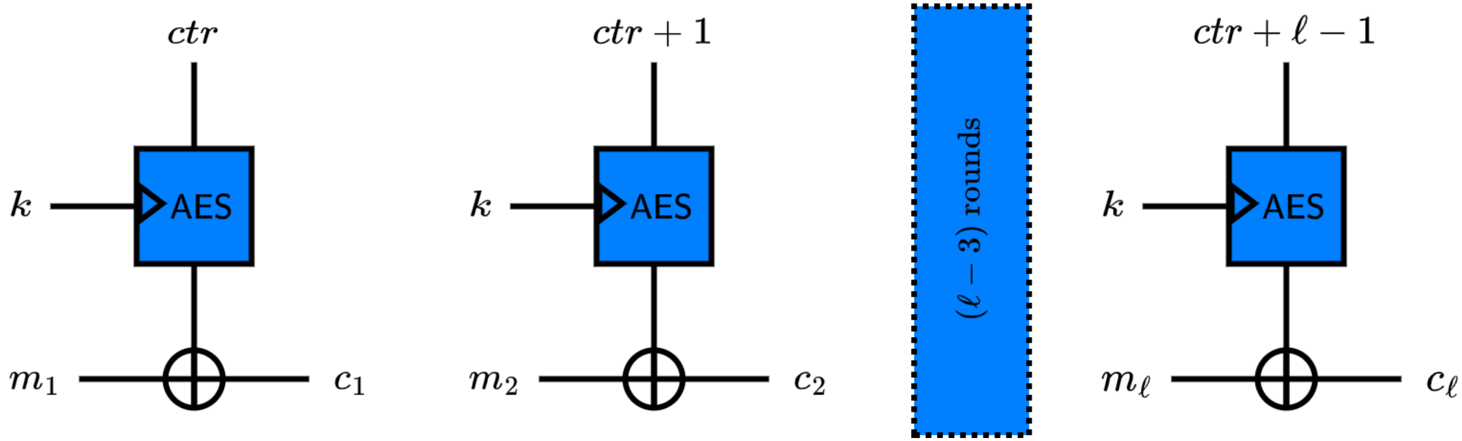
- POLKA's main design tweaks
  - Rigidity without FO-transform
  - Dummy ciphertext ($\Rightarrow$ leveled implementations)
  - Hard physical learning problems

- Conclusions & open problems

Outline

- Introduction/motivation
  - Parallel with symmetric crypto
  - Challenge for PQ crypto
- POLKA's main design tweaks
  - Rigidity without FO-transform
  - Dummy ciphertext ($\Rightarrow$ leveled implementations)
  - Hard physical learning problems
- Conclusions & open problems

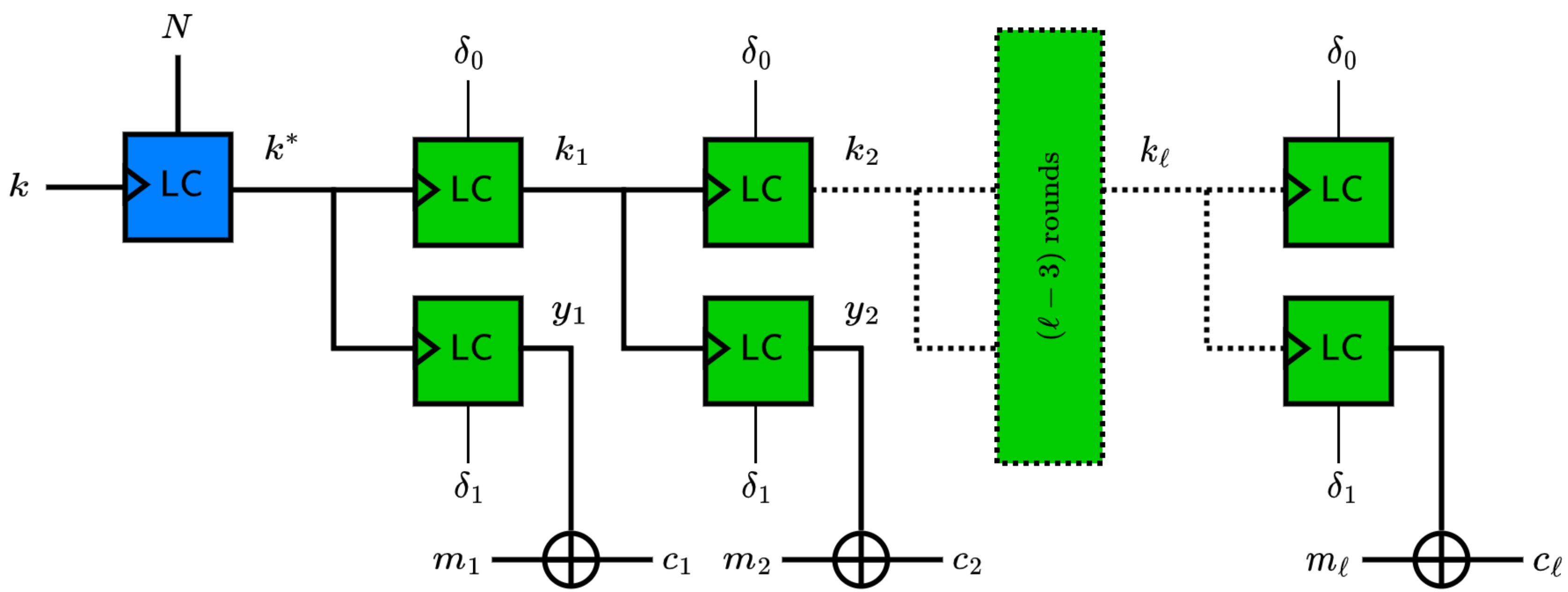
Symmetric crypto (20 years ago)



- CTR mode: uniform protection against DPA
- AES: expensive countermeasures (e.g., masking)

(Vocabulary: *DPA = key/state recovery attack with many side-channel traces*)

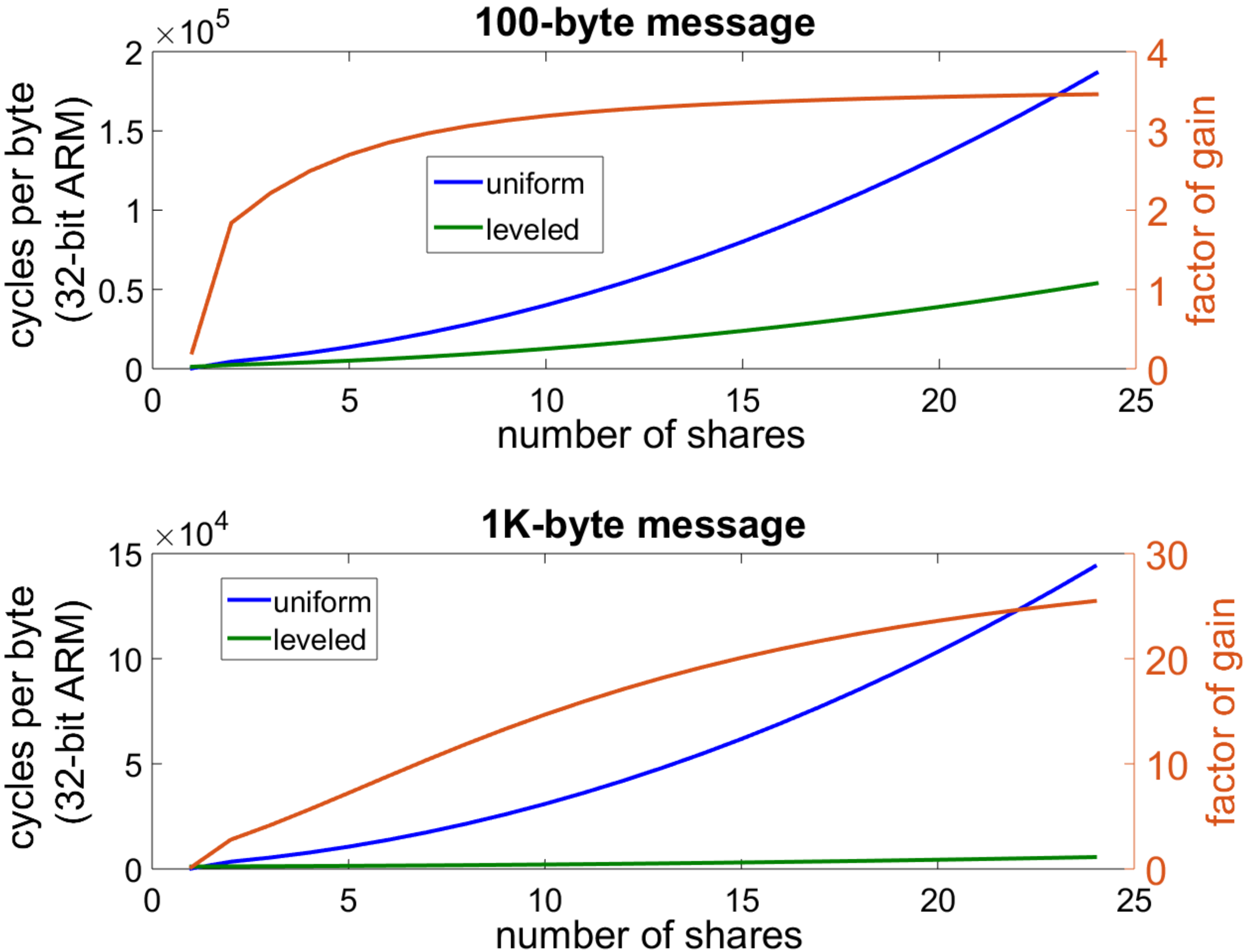
Symmetric crypto (nowadays)



- Leakage-resistant mode of operation
  - Leveled implementations (mixing expensive DPA protections for a few blocks a cheaper SPA protections)
- Lightweight (easier to protect) block ciphers

(Vocabulary: *SPA = key/state recovery attack with a few side-channel traces*)

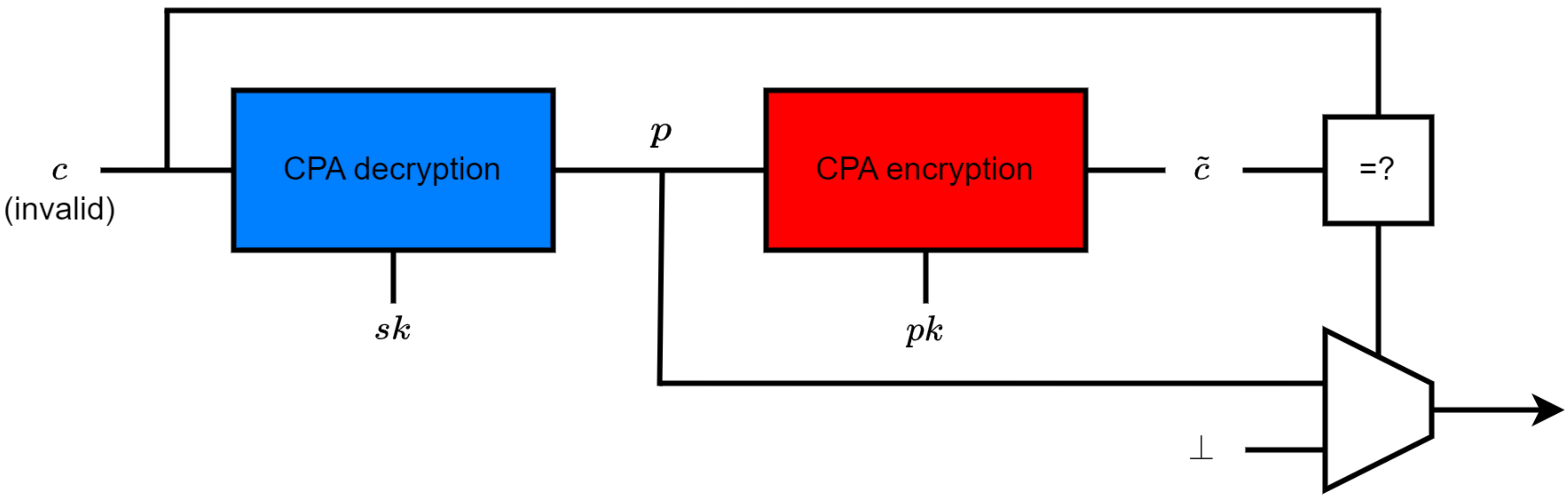
Impact can be massive!



Outline

- Introduction/motivation
  - Parallel with symmetric crypto
  - Challenge for PQ crypto
- POLKA's main design tweaks
  - Rigidity without FO-transform
  - Dummy ciphertext ($\Rightarrow$ leveled implementations)
  - Hard physical learning problems
- Conclusions & open problems

PQ encryption (FO-calypse)



- Decryption & re-encryption before the test
- Allows "state comparison" attacks
  - Just distinguishing L(p) from L(cp) leaks about sk
- Even more expensive to prevent than DPA
  - Factor of overheads: 6, 16, 50 for 2, 4 & 8 shares!

Research challenge

top-down: from abstract models to implementations

Can we design quantum-safe CCA-secure encryption schemes that are (much) cheaper to protect against leakage?

(& a bit less efficient if leakage is not a concern)

bottom-up: from heuristic tweaks to formal proofs

Needs humility: completely connecting top-down and bottom-up approaches remains a challenge after 20 years in symmetric crypto

Heuristic tweak proposals

1. Remove the state comparison attack path
   ≈ Avoid the FO-transform and leverage rigidity
2. Enable leveled implementations
   ≈ Use dummy ciphertexts & ephemeral secrets so that not all intermediate computations are DPA targets
3. Make the remaining DPA targets easier to mask
   ≈ leverage key-homomorphic computations and the (admittedly provocative) hard physical learning problems

- Focusing on key security (leakage-resilience) and not message security (leakage-resistance)

Outline

- Introduction/motivation
  - Parallel with symmetric crypto
  - Challenge for PQ crypto
- POLKA's main design tweaks
  - Rigidity without FO-transform
  - Dummy ciphertext ($\Rightarrow$ leveled implementations)
  - Hard physical learning problems
- Conclusions & open problems

POst-quantum **L**eakage-resilient
public **K**ey encryption **A**lgorithm

- CCA-secure in the QROM (w/o leakage)
- Hybrid encryption with an LPR-like KEM

$$c_1 = a \cdot r + e_1$$
$$c_2 = b \cdot r + e_2$$

small $r, e_1, e_2 \leftarrow D$

Then, $K = \mathrm{H}(r, e_1, e_2)$ and $c_0 = \mathrm{AEnc}_K(m)$

- Rigidity w/o FO + explicit rejection
- Partially randomized decapsulation

Base scheme + rigidity

- KeyGen(pp): $a \leftarrow R_q = F_q[x]/(x^n + 1)$
  $b = p \cdot (a \cdot s + e) \in R_q^*$ *medium*
- Encrypt$_{a,b}$(m): $c_1 = a \cdot r + e_1$
  $c_2 = b \cdot r + e_2$
  small $r, e_1, e_2 \leftarrow D$

  Then, $K = \mathrm{H}(r, e_1, e_2)$ and $c_0 = \mathrm{AEnc}_K(m)$
- Decrypt$_s$(c): $c_2 - p \cdot c_1 \cdot s = p \cdot (er - e_1 s) + e_2$

  $\Rightarrow$ Extract e_2, then r, and then e_1
  Check if they are small, else abort
  $K = \mathrm{H}(r, e_1, e_2)$ and $c_0 = \mathrm{ADec}_K(c_0)$

- KeyGen(pp): $a \leftarrow R_q = F_q[x]/(x^n + 1)$
  $b = p \cdot (a \cdot s + e) \in R_q^*$ *medium*
- Encr … $e_1, e_2 \leftarrow D$

  Then
- Decr … $e_1 s) + e_2$

If extracted r, e_1, e_2 are small,
computing $a \cdot r + e_1$ and $b \cdot r + e_2$
would lead to c_1 and c_2

But we do not have to do it!
($\neq$ FO-transform)

$\Rightarrow$ Extract e_2, then r, and then e_1
Check if they are small, else abort
$K = \mathrm{H}(r, e_1, e_2)$ and $c_0 = \mathrm{ADec}_K(c_0)$

Outline

- Introduction/motivation
  - Parallel with symmetric crypto
  - Challenge for PQ crypto
- POLKA's main design tweaks
  - Rigidity without FO-transform
  - Dummy ciphertext (⇒ leveled implementations)
  - Hard physical learning problems
- Conclusions & open problems

Decrypt with dummy ciphertext (I)

- Encapsulation is almost homomorphic

$$c_1 = a \cdot r + e_1$$
$$c_2 = b \cdot r + e_2$$

small $r, e_1, e_2 \leftarrow D$

$$c_1' = a \cdot r' + e_1'$$
$$c_2' = b \cdot r' + e_2'$$

small $r', e_1', e_2' \leftarrow D$

$+$

$$\bar{c}_1 = a \cdot \bar{r} + \bar{e}_1$$
$$\bar{c}_2 = b \cdot \bar{r} + \bar{e}_2$$

$$\bar{r} = r + r'$$
$$\bar{e}_1 = e_1 + e_1'$$
$$\bar{e}_2 = e_2 + e_2'$$

- Partial randomized decapsulation
  - Compute c_1' and c_2', *add-then-decrypt* $\bar{c}_1$ and $\bar{c}_2$

- Decrypt$_s$(c): $c_1{}' = a \cdot r' + e_1{}'$ $\quad r', e_1{}', e_2{}' \leftarrow D$

  $c_2{}' = b \cdot r' + e_2{}'$

  Compute $\bar{c}_1 = c_1 + c_1'$ and $\bar{c}_2 = c_2 + c_2{}'$

  $$\bar{c}_2 - p \cdot \bar{c}_1 \cdot s = p \cdot (e\bar{r} - \bar{e}_1 s) + \bar{e}_2$$

$\Rightarrow$ Extract $\bar{e}_2$, then $\bar{r}$, and then $\bar{e}_1$

abort if not « $2 \times$ small »

recover $r = \bar{r} - r'$, $e_1 = \bar{e}_1 - e_1'$ and $e_2 = \bar{e}_2 - e_2{}'$

abort if not small, $K = \mathrm{H}(r, e_1, e_2)$ and $c_0 = \mathrm{ADec}_K(c_0)$

- Decryption failure issue
  - Not an issue as long as $sk = s$ is small

Decrypt with dummy ciphertext (II)

- Decrypt$_s$(c): $c_1{}' = a \cdot r' + e_1{}'$
  $c_2{}' = b \cdot r' + e_2{}'$ $r', e_1{}', e_2{}' \leftarrow D$

Com

⇒ Extrac

abort i

recove $- e_2{}'$

abort if not small, $K' = H(r', e_1{}', e_2{}')$ and $c_0 = \mathrm{ADec}_K(c_0)$

Most of the sensitive computations manipulate ephemeral (sometimes randomized) secrets

DPA attack paths replaced by SPA attack paths

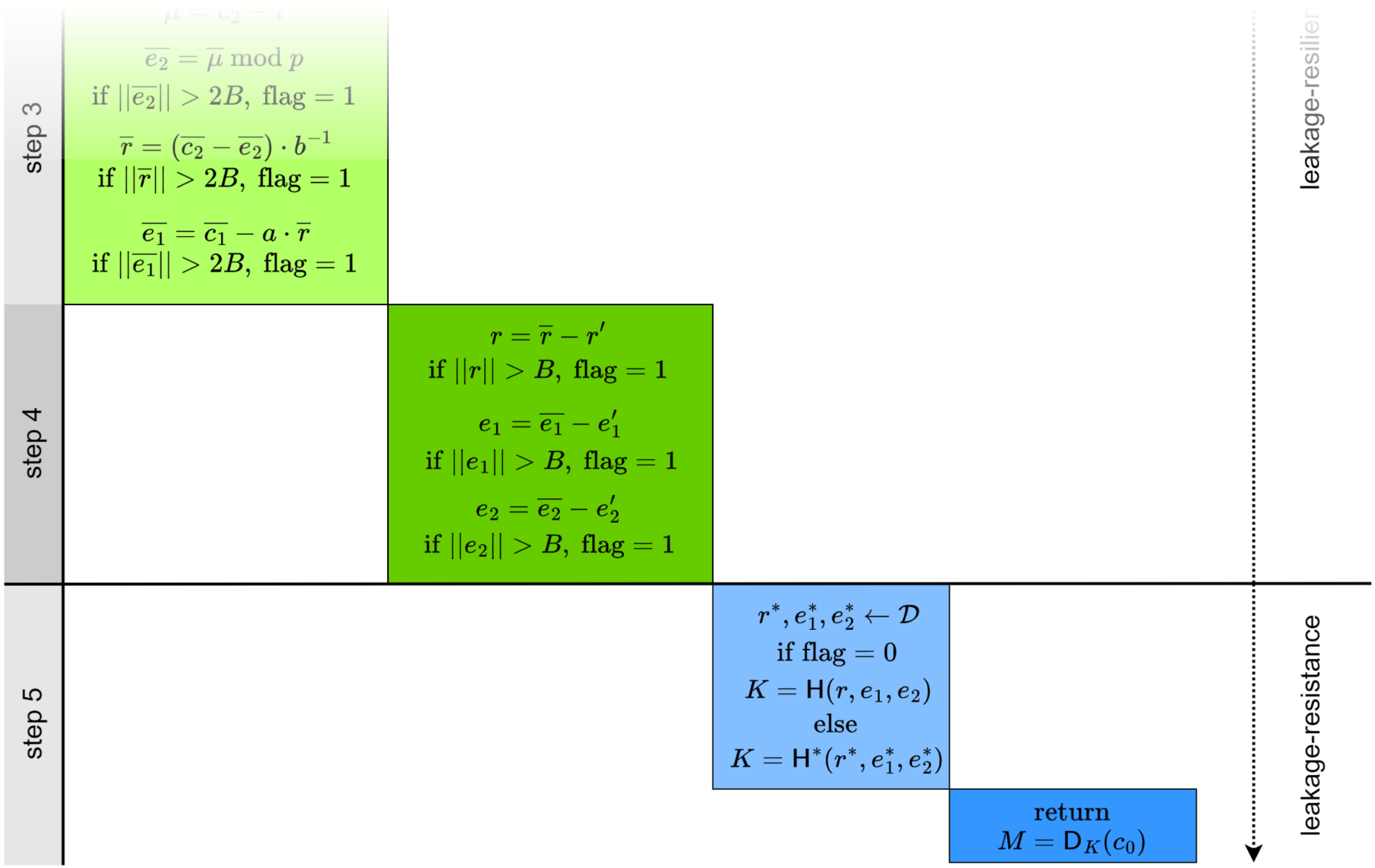
- Decryption failure issue
  - Not an issue as long as $sk = s$ is small

Leveling POLKA.dec (I)

	SPA	avg-SPA	UP-DPA	DPA
step 1	$r', e'_1, e'_2 \leftarrow \mathcal{D}$ $c'_1 = a \cdot r' + e'_1$ $c'_2 = a \cdot r' + e'_2$ $\overline{c_1} = c_1 + c'_1$ $\overline{c_2} = c_2 + c'_2$			
step 2			$t = (p \cdot \overline{c_1}) \cdot s$	
step 3	$\overline{\mu} = \overline{c_2} - t$ $\overline{e_2} = \overline{\mu} \bmod p$ $\text{if } \lvert\lvert\overline{e_2}\rvert\rvert > 2B,\ \text{flag} = 1$ $\overline{r} = (\overline{c_2} - \overline{e_2}) \cdot b^{-1}$ $\text{if } \lvert\lvert\overline{r}\rvert\rvert > 2B,\ \text{flag} = 1$ $\overline{e_1} = \overline{c_1} - a \cdot \overline{r}$ $\text{if } \lvert\lvert\overline{e_1}\rvert\rvert > 2B,\ \text{flag} = 1$			
step 4		$r = \overline{r} - r'$ $\text{if } \lvert\lvert r\rvert\rvert > B,\ \text{flag} = 1$ $e_1 = \overline{e_1} - e'_1$		

leakage-resilience

Leveling POLKA.Dec (II)



What remains (leakage-resilience)

- DPA against $t = \overbrace{(p \cdot \overline{c_1})}^{\text{unknown ephemeral value } r} \cdot s$ } long-term secret

\+ Key-homomorphic (masked with linear overheads)

$$s = s_1 + s_2 + \cdots + s_s \Rightarrow t = \sum_{i=1}^{d} (r \cdot s_i)$$

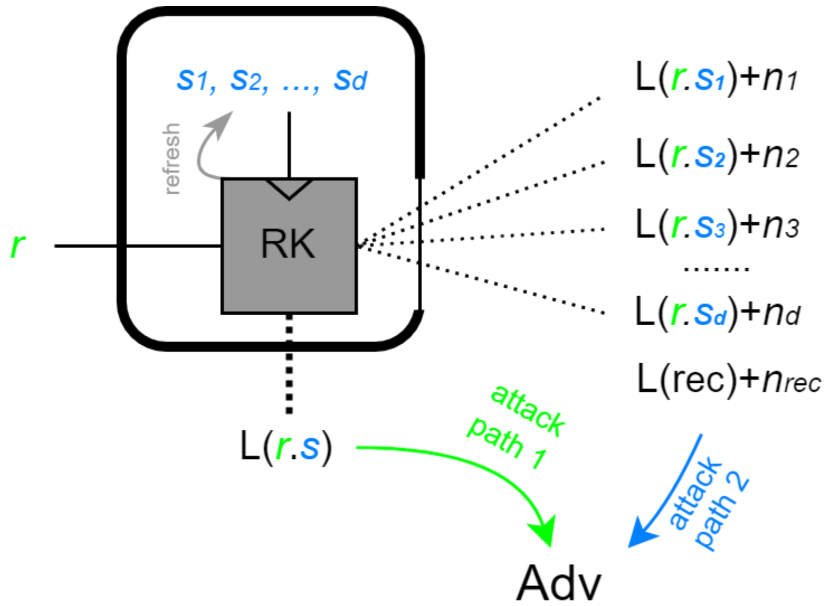
– Norm computations are not key-homomorphic

Outline

- Introduction/motivation
  - Parallel with symmetric crypto
  - Challenge for PQ crypto
- POLKA's main design tweaks
  - Rigidity without FO-transform
  - Dummy ciphertext ($\Rightarrow$ leveled implementations)
  - Hard physical learning problems
- Conclusions & open problems

Leveraging physical assumptions?

Very similar to fresh re-keying in symmetric crypto



- Attack path 1: hard physical learning problem
  - Assumed to be hard if L is noisy or algebraically incompatible with *r.s* (formalized as the ring Learning With Physical Rounding problem)

$\Rightarrow$ It may be sound to unmask $t = (p \cdot \overline{c_1}) \cdot s$

Outline

- Introduction/motivation
  - Parallel with symmetric crypto
  - Challenge for PQ crypto

- POLKA's main design tweaks
  - Rigidity without FO-transform
  - Dummy ciphertext ($\Rightarrow$ leveled implementations)
  - Hard physical learning problems

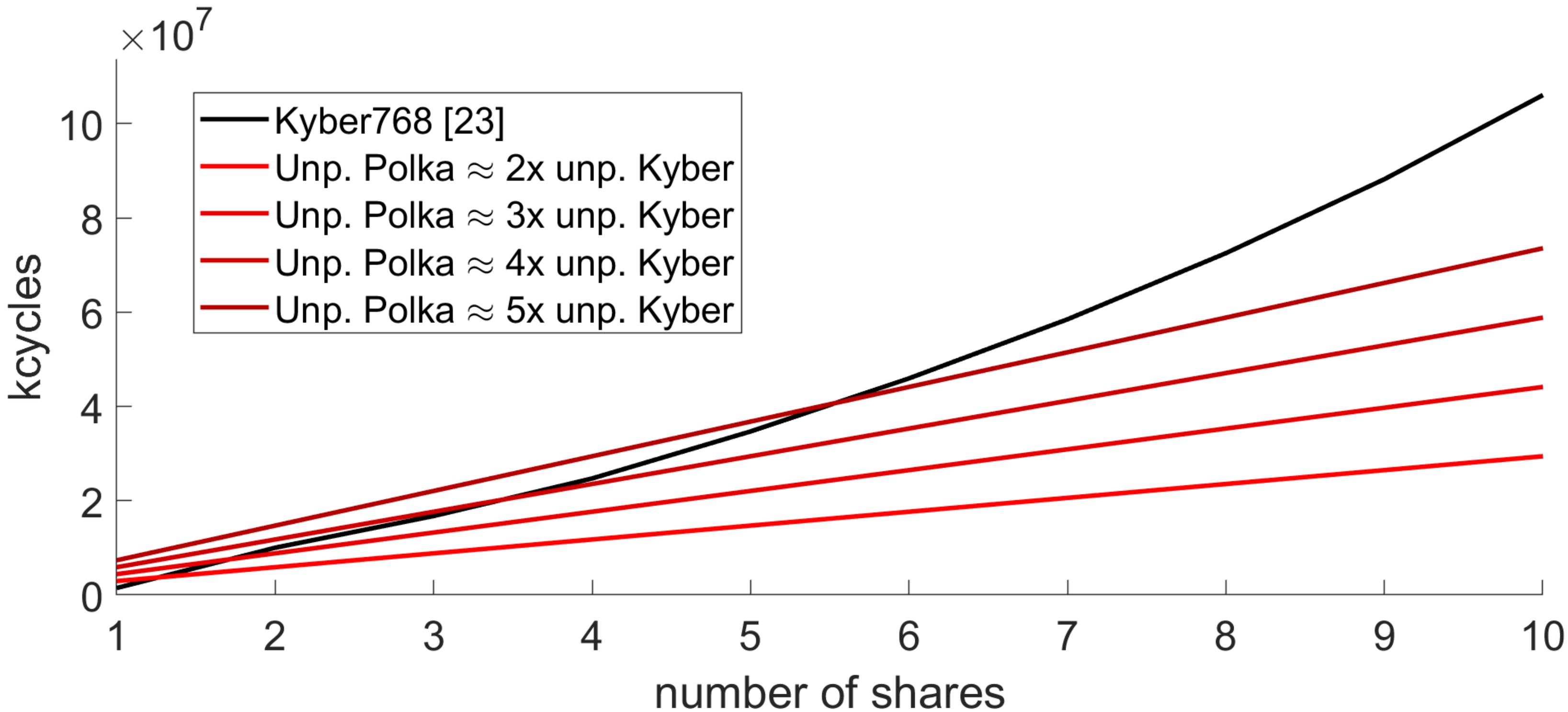
- Conclusions & open problems

Conclusion

- Food for thought (there is a lot to gain)
- Decent instances (e.g., 16-bit $q, n = 1024$)
- Many important open questions
  - Concrete comparison (e.g., masked Kyber)
    - Challenge: masked Kyber's security?
  - Formalization & reductions
    - Challenge: finer-grain than symmetric crypto
  - Assessing hard physical learning problems?
  - From leakage-resilience to leakage-resistance

(+ other tweaks in the paper: key-homomorphic one-time MAC, implicit vs. explicit rejection)

THANKS



×10^7
Kyber768 [23]
Unp. Polka ≈ 2x unp. Kyber
Unp. Polka ≈ 3x unp. Kyber
Unp. Polka ≈ 4x unp. Kyber
Unp. Polka ≈ 5x unp. Kyber
kcycles
number of shares
0
2
4
6
8
10
1
2
3
4
5
6
7
8
9
10