Quantum Algorithms for Cryptanalysis and Post-Quantum Symmetric Cryptography

André Schrottenloher

June 23rd, 2022

Cryptography

Enable secure communications over insecure channels, at the lowest possible cost.

Asymmetric

- No shared secret
- Public-key schemes (RSA. . .), key-exchange protocols, signatures. . .

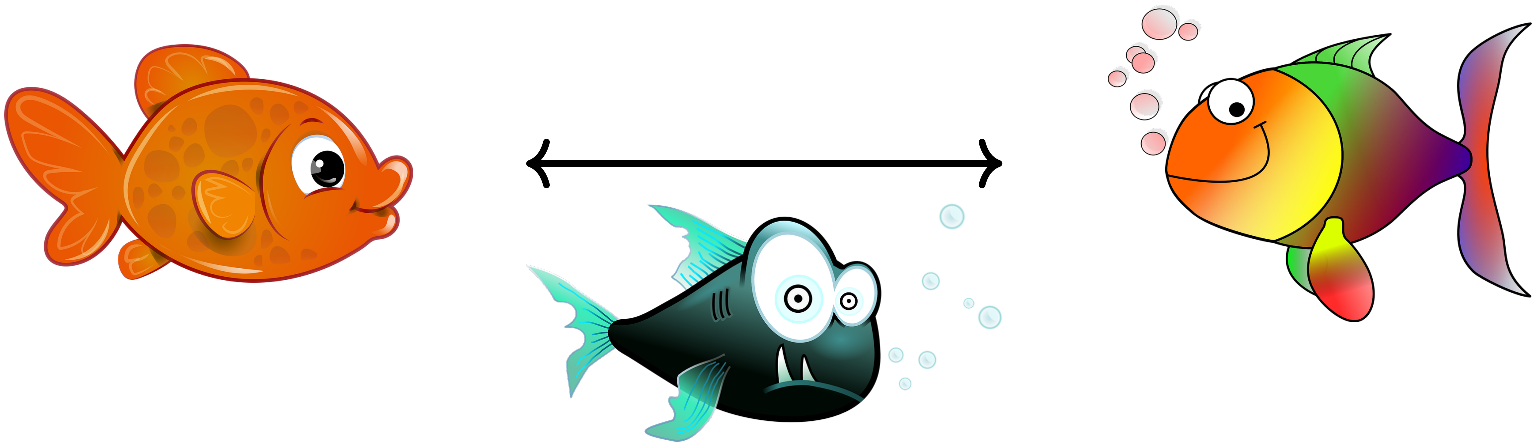
Symmetric

- **Shared secret**
- Block ciphers (AES. . .), stream ciphers, hash functions (SHA-3. . .). . .

Symmetric cryptography

Example:

- After having shared a **secret key** k, Alice and Bob communicate using an encryption scheme
- The algorithm is based on a block cipher $E_k : \{0,1\}^n \to \{0,1\}^n$ (the primitive)
- They agree on the standard **AES-128**: $|k| = 128, n = 128$

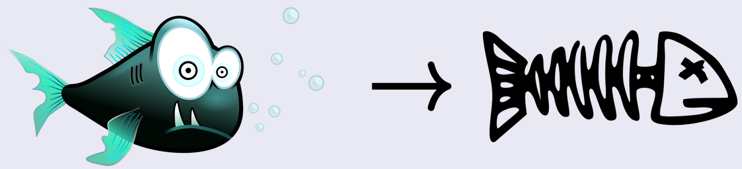


Security of primitives

The security of an **ideal** primitive is defined by **generic attacks**.

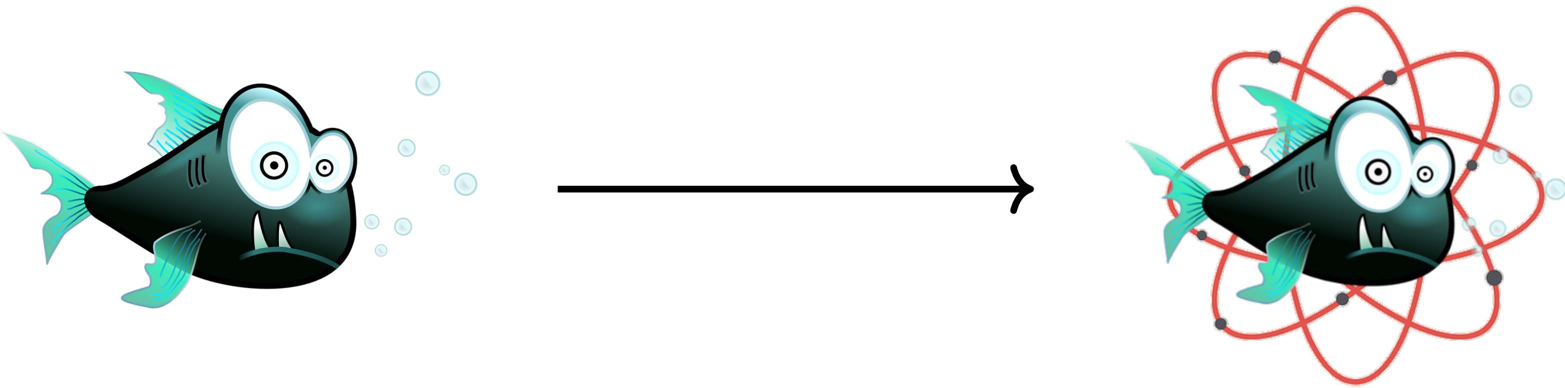
Generic key-recovery: "try all the keys"

- Given a few plaintext-ciphertext pairs, try all keys k and find the matching one. Costs $2^{|k|}$ encryptions.
- If $|k| = 128$: $2^{128} =$ approx. 10^{22} core-years
- "128 bits of security"



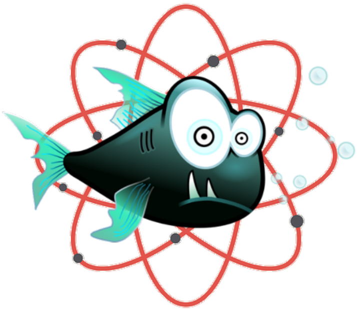
- But **AES is not ideal** and its security can only be **conjectured**
- **Cryptanalysis** is our **empirical measure of security**
- If we find a better attack than generic, the cipher is **broken** (the conjecture is false)

The adversary becomes quantum



- For long-term security, we need to take into account a **quantum adversary**
- By changing the notion of "computation", the status of our computational conjectures will **change**

The post-quantum world



What can this adversary do?

Asymmetric crypto

- **Shor's algorithm** breaks factorization and DL-based systems

Symmetric crypto

- **Grover's algorithm** accelerates exhaustive key-recovery to $\sqrt{2^{|k|}} = 2^{|k|/2}$
- So ideally, we should increase (double) the key sizes
- What else?

Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring", FOCS 1994

Quantum algorithms (feat. quantum search)

X a search space of size $2^{|k|}$, $f : X \to \{0,1\}$, find the single $x_0 \in X$ such that $f(x) = 1$.

Classical (exhaustive) search

$$\text{Repeat } 2^{|k|} \text{ times} \begin{cases} \text{Sample } x \in X \\ \text{Test if } f(x) = 1 \end{cases}$$

Quantum search (Grover's algorithm)

$$\text{Repeat } \mathcal{O}\left(\sqrt{2^{|k|}}\right) \text{ times} \begin{cases} \text{Sample } x \in X \to \text{quantumly} \\ \text{Test if } f(x) = 1 \to \text{quantumly} \end{cases}$$

$\implies$ we will treat it as a black box.

Grover, "A fast quantum mechanical algorithm for database search", STOC 96

Contributions

1. New algorithms for generic problems in cryptography

- Collisions and generalized collisions (k-XOR, k-SUM)

2. Quantum cryptanalysis of structured constructions

- New algorithmic tool: offline-Simon

3. Dedicated cryptanalysis

- Gimli, Spook (recent lightweight ciphers)
- Quantum security analysis of AES (spoiler: seems safe so far)

4. Design

- The Saturnin block cipher and algorithms (maximal security at a minimal cost)

Outline

Quantum Algorithms for the k-XOR Problem

k-XOR problem (with many solutions)

k-XOR

Let $H : \{0,1\}^n \rightarrow \{0,1\}^n$ be a random function, find $x_1, \ldots, x_k$ such that $H(x_1) \oplus \ldots \oplus H(x_k) = 0$.

- Usually H is a known, keyless function (a hash function, a list of data)
- We have **a quantum algorithm** for H (quantum oracle access)

The query complexity

Classical: $2^{n/k}$ (trivial)

Quantum: $2^{n/(k+1)}$ (not trivial) [Belovs & Spalek]

We will be interested in the **time complexity**, which is usually much higher.

- We focus on the exponent: α_k in $\widetilde{\mathcal{O}}(2^{\alpha_k n})$
- All the results apply with $+$ instead of $\oplus$ (k-SUM)

Potential applications

Subset-sum: given n integers $\bar{a} = a_0, \ldots a_{n-1}$ on poly(n) bits, find a binary $\bar{e}$ such that $\bar{a} \cdot \bar{e} = 0$ $\implies$ reduces to k-SUM

Parity check problem: find a low-weight multiple of a polynomial $\implies$ reduces to k-SUM

LPN: given samples $a, a \cdot s \oplus e$ with n-bit uniform random a and Bernoulli noise e, find s $\implies$ reduces to k-SUM

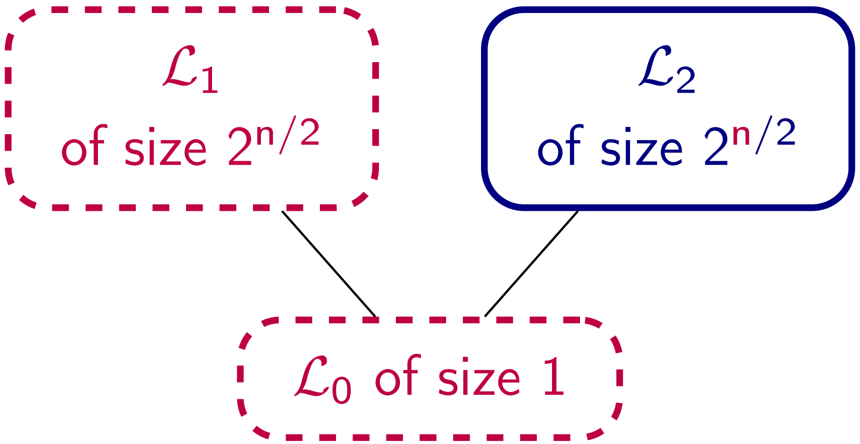
Multiple-encryption: given a few plaintext-ciphertext pairs $(x, E_{k_1} \circ \ldots \circ E_{k_r}(x))$, find the independent keys $k_1, \ldots k_r$ $\implies$ similar algorithms applicable

The **merging** algorithms used for k-SUM also appear in generic information set decoding, lattice sieving or subset-sum algorithms.

2-XOR: collision search

Classical setting (naive)

1. Store $2^{n/2}$ queries $(x, H(x))$ in a list $\mathcal{L}_2$
2. Enumerate a list $\mathcal{L}_1$, looking for a collision with $\mathcal{L}_2$

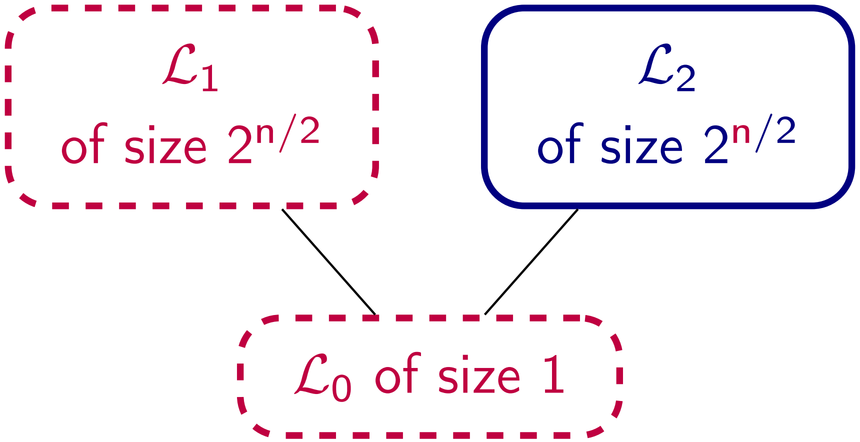


Brassard, Høyer and Tapp, "Quantum Cryptanalysis of Hash and Claw-Free Functions", LATIN 98

2-XOR: collision search

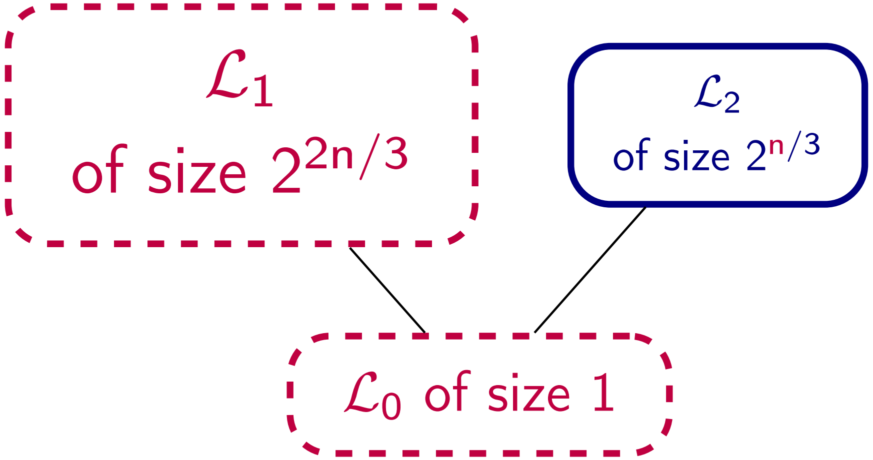
Classical setting (naive)

1. Store $2^{n/2}$ queries $(x, H(x))$ in a list $\mathcal{L}_2$
2. Enumerate a list $\mathcal{L}_1$, looking for a collision with $\mathcal{L}_2$



Quantum setting (BHT)

1. Store $2^{n/3}$ queries $(x, H(x))$ in a list $\mathcal{L}_2$
2. (Quantum) search in $\mathcal{L}_1$ for a collision with $\mathcal{L}_2$



Brassard, Høyer and Tapp, "Quantum Cryptanalysis of Hash and Claw-Free Functions", LATIN 98

Merging with $k = 4$

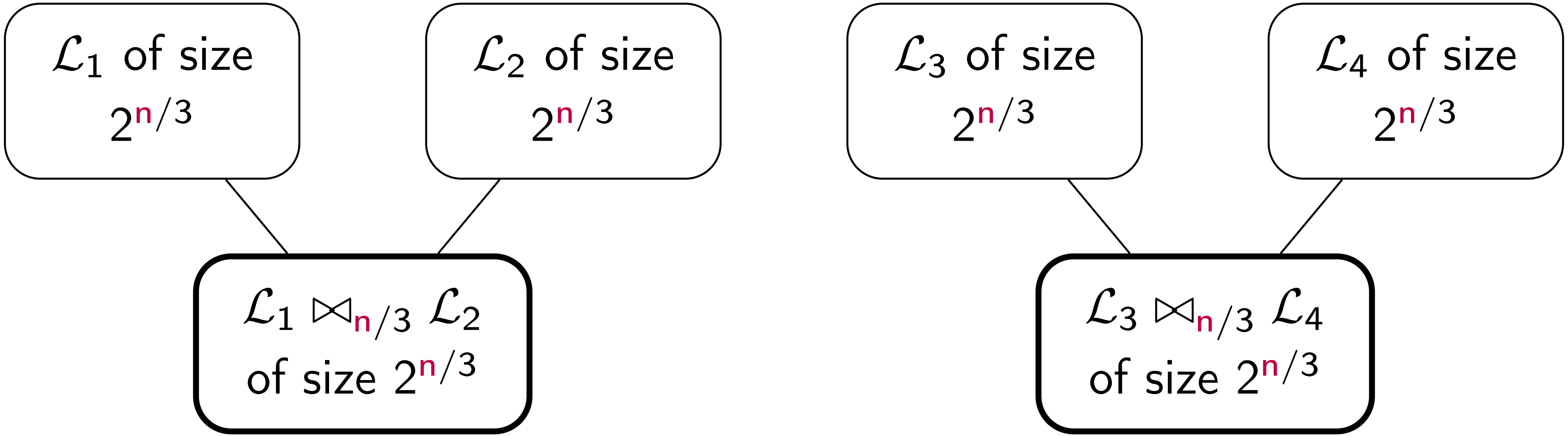
1. Make 4 lists of $2^{n/3}$ queries $(x, H(x))$

$\mathcal{L}_1$ of size $2^{n/3}$	$\mathcal{L}_2$ of size $2^{n/3}$	$\mathcal{L}_3$ of size $2^{n/3}$	$\mathcal{L}_4$ of size $2^{n/3}$

Wagner, "A Generalized Birthday Problem", CRYPTO 2002

Merging with $k = 4$

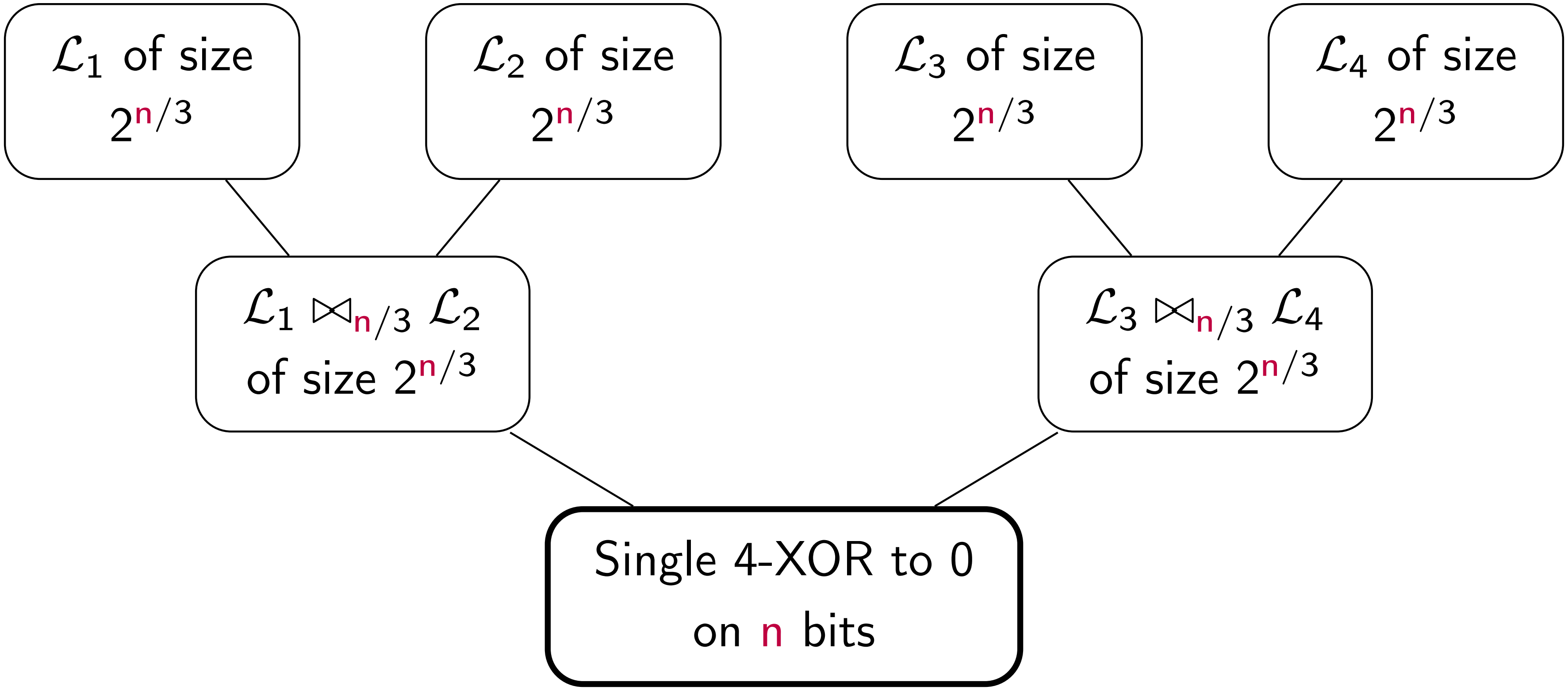
1. Make 4 lists of $2^{n/3}$ queries $(x, H(x))$
2. **Merge** into 2 lists of **pairs** (x, y) with $n/3$ zeroes in the sum $H(x) \oplus H(y)$



Wagner, "A Generalized Birthday Problem", CRYPTO 2002

Merging with $k = 4$

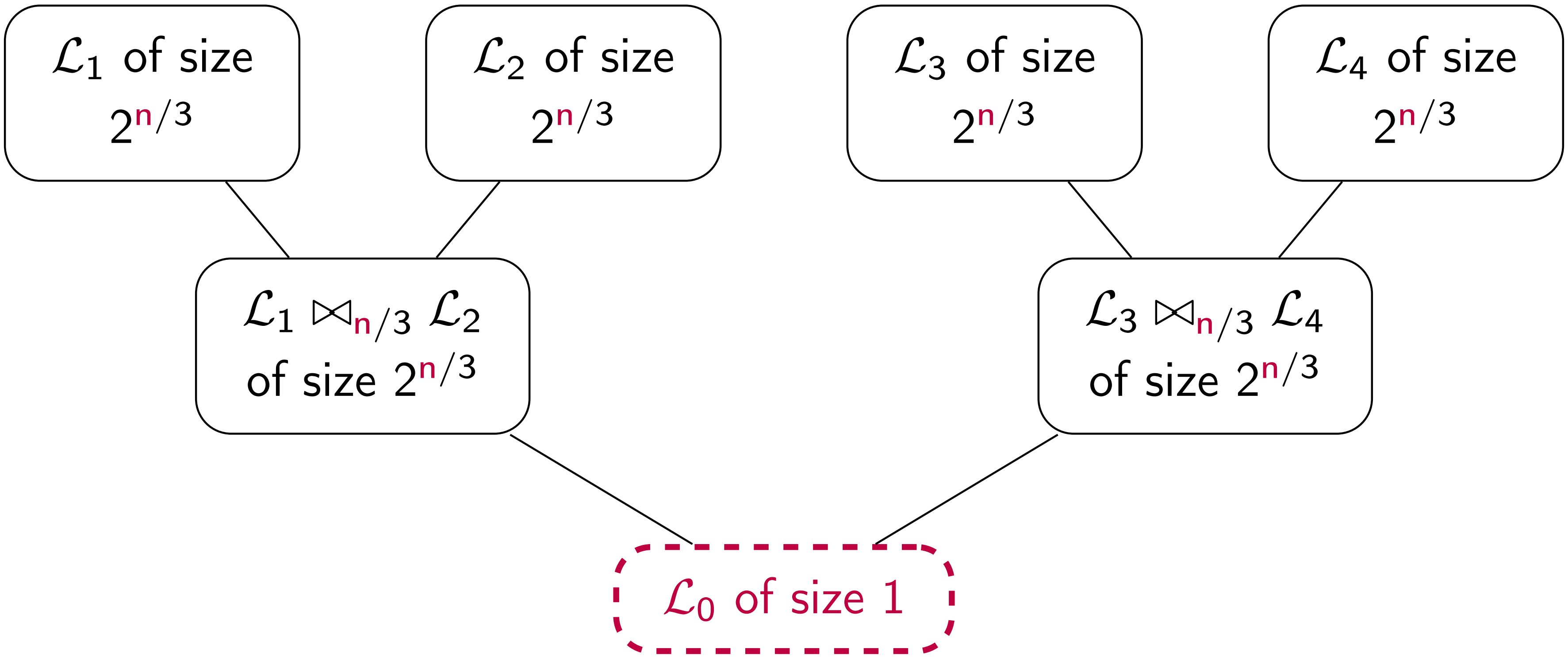
1. Make 4 lists of $2^{n/3}$ queries $(x, H(x))$
2. **Merge** into 2 lists of **pairs** with $n/3$ zeroes in the sum
3. Merge into 1 list of 4-tuples with $n/3 + 2n/3 = n$ zeroes (4-XOR to zero)



Wagner, "A Generalized Birthday Problem", CRYPTO 2002

Depth-first traversal of Wagner's tree

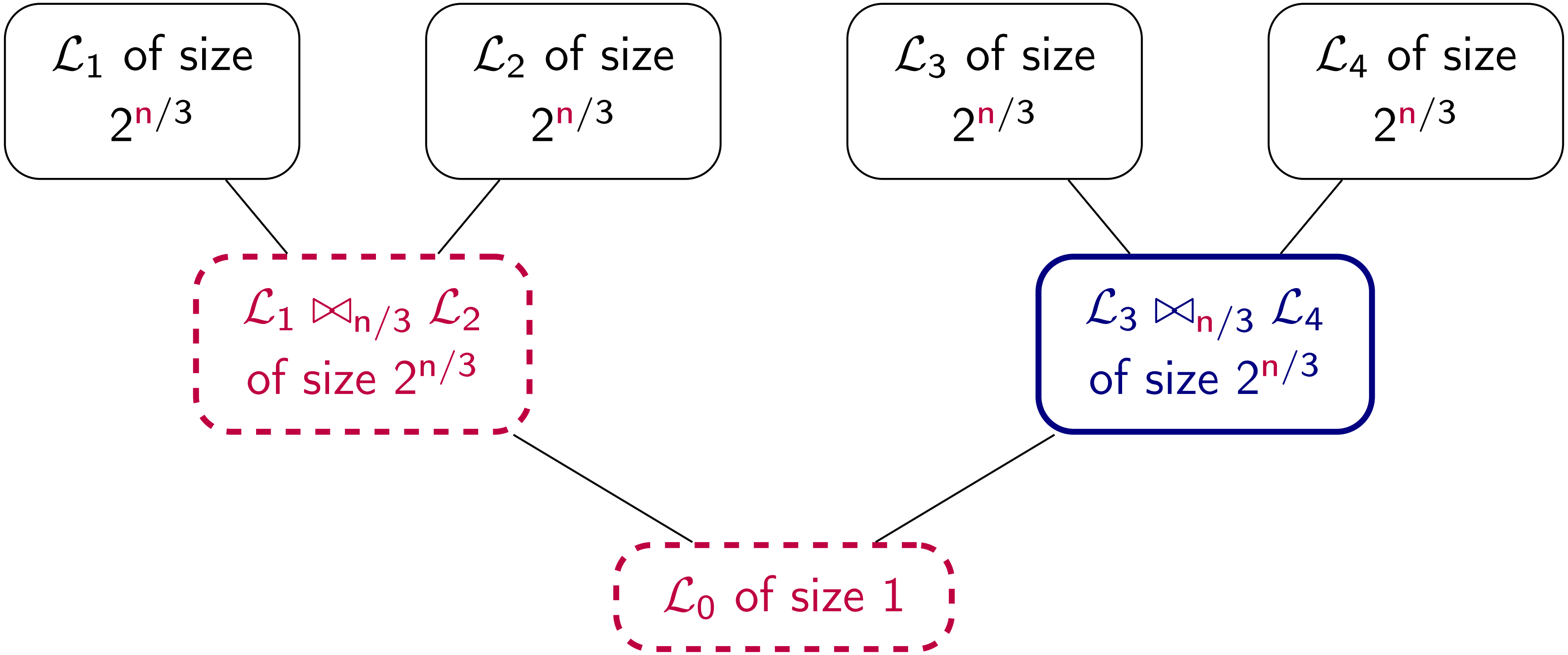
We **search** an element of $\mathcal{L}_0$



Depth-first traversal of Wagner's tree

We **search** an element of $\mathcal{L}_0$

$\Longrightarrow$ We **search** an element of $\mathcal{L}_1 \bowtie \mathcal{L}_2$ that collides with $\mathcal{L}_3 \bowtie \mathcal{L}_4$

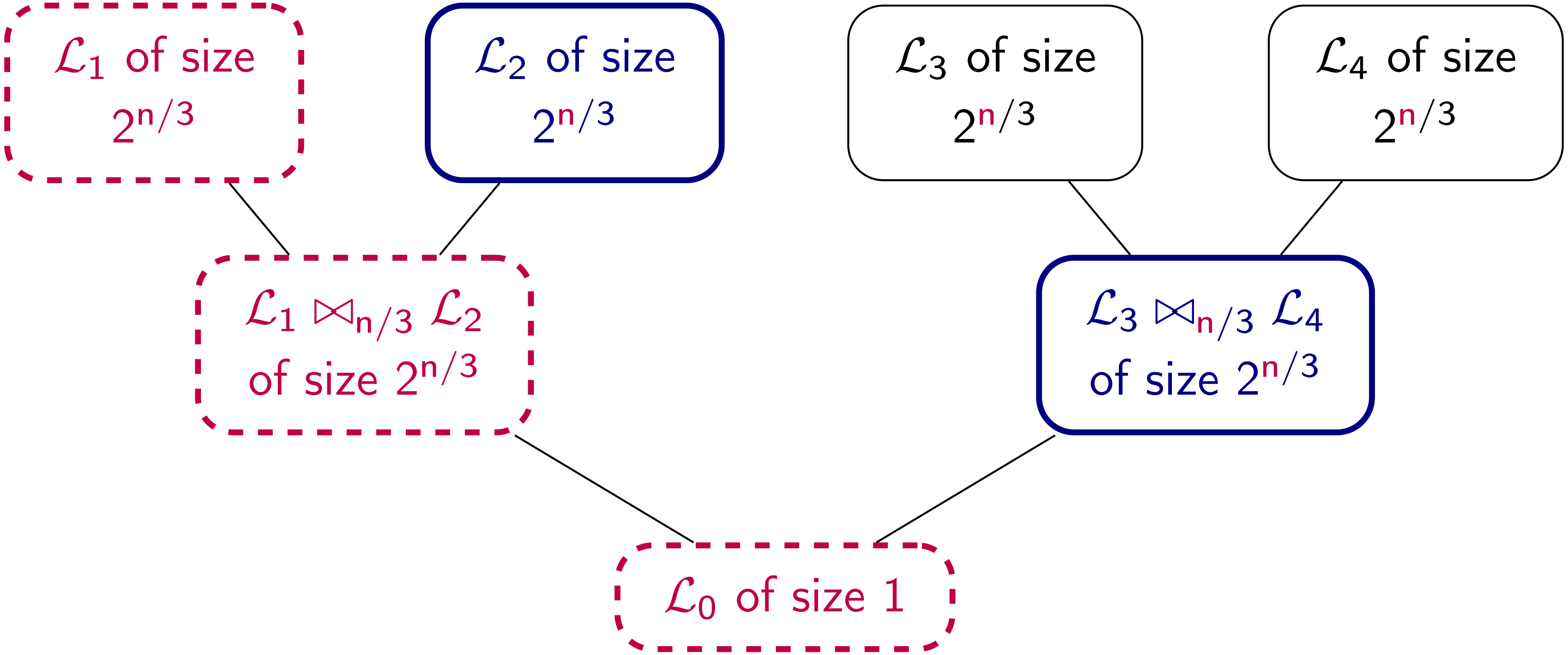


Depth-first traversal of Wagner's tree

Search an element of $\mathcal{L}_0$

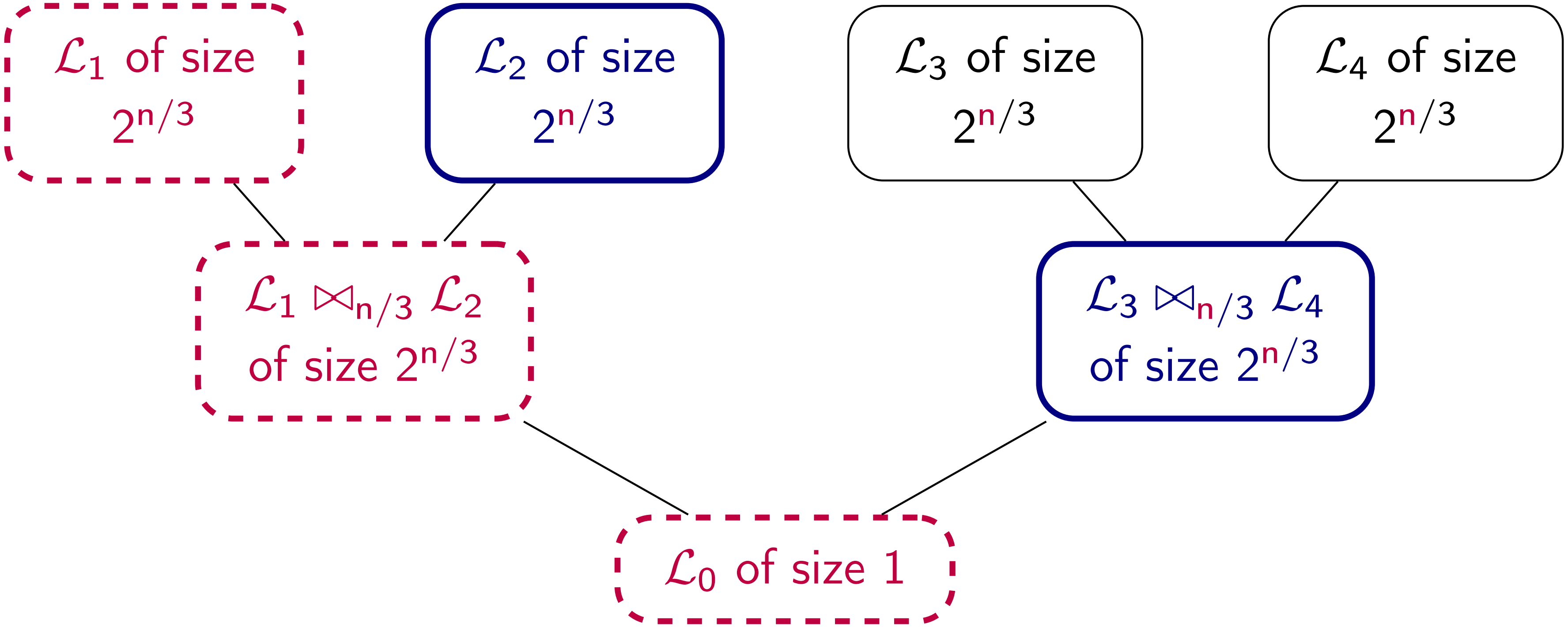
$\Longrightarrow$ **Search** an element of $\mathcal{L}_1 \bowtie \mathcal{L}_2$ that collides with $\mathcal{L}_3 \bowtie \mathcal{L}_4$

$\Longrightarrow$ **Search** an element of $\mathcal{L}_1$ that yields an element of $\mathcal{L}_1 \bowtie \mathcal{L}_2$ that collides with $\mathcal{L}_3 \bowtie \mathcal{L}_4$



4-XOR example

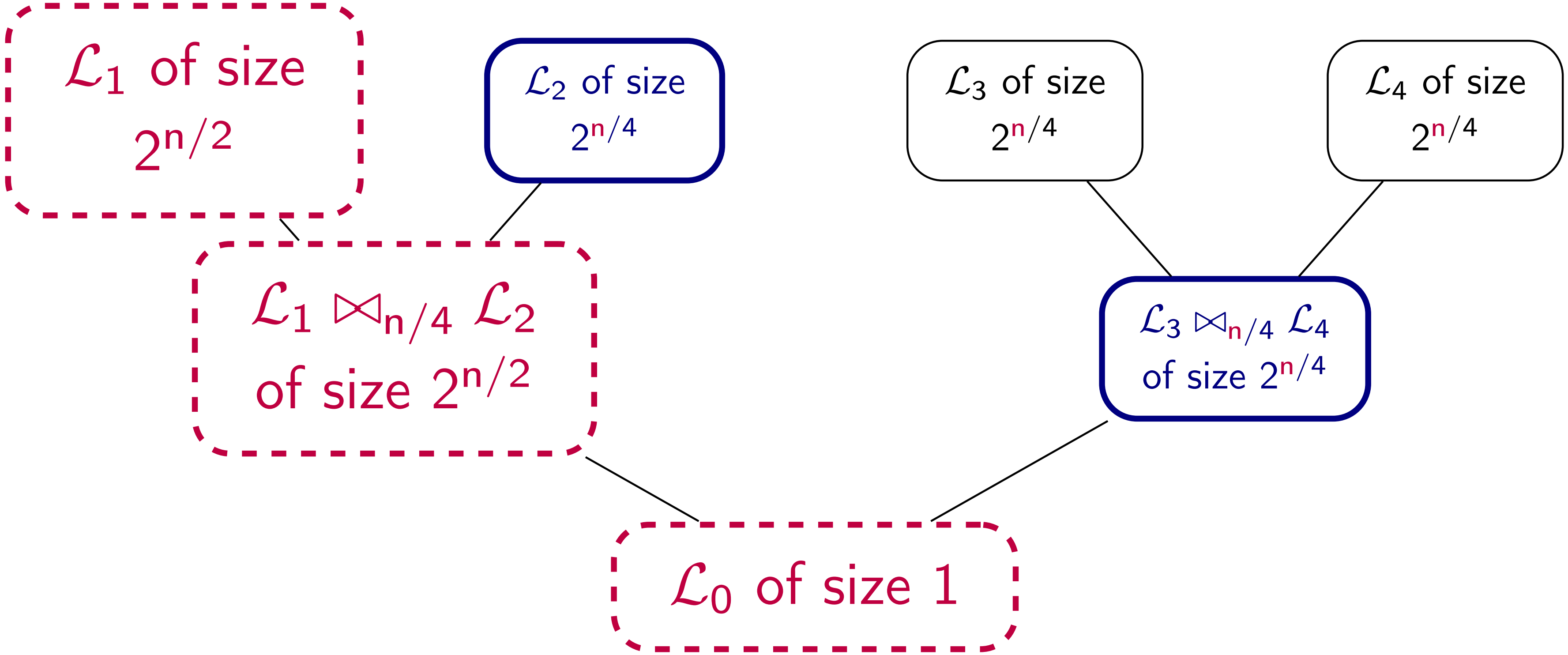
- Time $2^{n/6}$ for the **search**
- Time $2^{n/3}$ for the **intermediate lists**



Naya-Plasencia, S., "Optimal Merging in Quantum k-XOR and k-SUM Algorithms", EUROCRYPT 2020

4-XOR example

- Time $2^{n/4}$ for the **search**
- Time $2^{n/4}$ for the **intermediate lists**



$\Longrightarrow$ Similar results follow for all k

Naya-Plasencia, S., "Optimal Merging in Quantum k-XOR and k-SUM Algorithms", EUROCRYPT 2020

Single-solution k-XOR

k-XOR

Let $H : \{0,1\}^{n/k} \to \{0,1\}^n$ be a random function, find $x_1, \ldots, x_k$ such that $H(x_1) \oplus \ldots \oplus H(x_k) = 0$.

Classical:

- Time $2^{n/2}$ for a generic k (like a collision search)
- Advanced algorithms can reduce the memory using merging trees

Quantum:

- Time decreases with k, down to $2^{2n/7}$ (**not** like a collision search)
- Merging trees reduce the memory **and the time** complexity

Case Study: Quantum Security of AES

Key-recovery attacks on AES

- A 128-bit block cipher based on an SPN structure
- 20 years of cryptanalysis

Classical (key-recovery) attacks:

$$\text{time} < 2^{|k|}$$

- AES-128: **7/10-round** Impossible Differential
- AES-256: **9/14-round** Demirci-Selçuk-MITM

Quantum (key-recovery) attacks:

$$\text{time} < 2^{|k|/2}$$

- AES-128: **6/10-round** quantum Square
- AES-256: **8/14-round** quantum DS-MITM

Bonnetain, Naya-Plasencia, S., "Quantum Security Analysis of AES", ToSC 2019

Key-recovery attacks (ctd.)

So far all attacks on AES follow a “quantization” strategy:

1. start from a classical attack
2. use Grover search to accelerate the parts that we can

- A classical attack cannot be always “quantized”.
- The 7-round DS-MITM attack from [DFJ13] on AES-128 uses a table of size 2^{80}. Creating this table exceeds the 2^{64} quantum time limit.

Derbez, Fouque, Jean, “Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting”, EUROCRYPT 2013

Security of AES

So far AES-256 remains a good cipher for post-quantum applications.

- With some limitations, e.g., (quantum) birthday bound security levels for a 128-bit state size.
- A bigger block size would be helpful. . . can it also be a lightweight cipher?

Saturnin

Context

Saturnin:

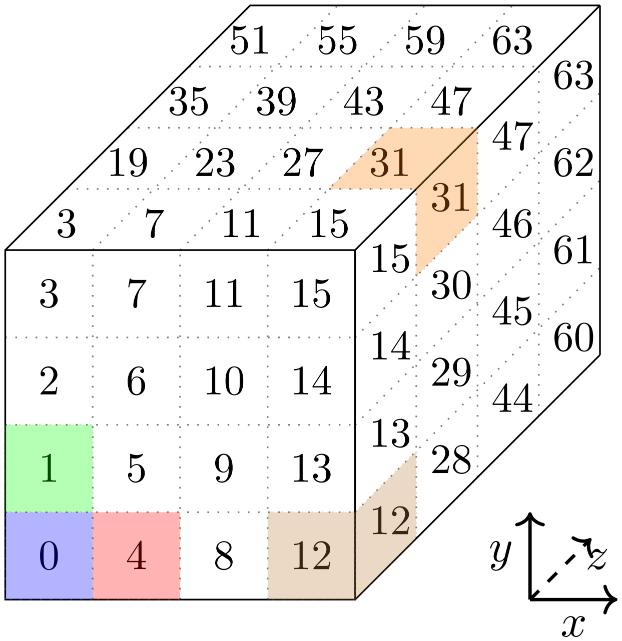
- (was) one of the second-round candidates in the current NIST "lightweight crypto standardization process"
- the only one with a 256-bit block cipher and (superposition) quantum security claims

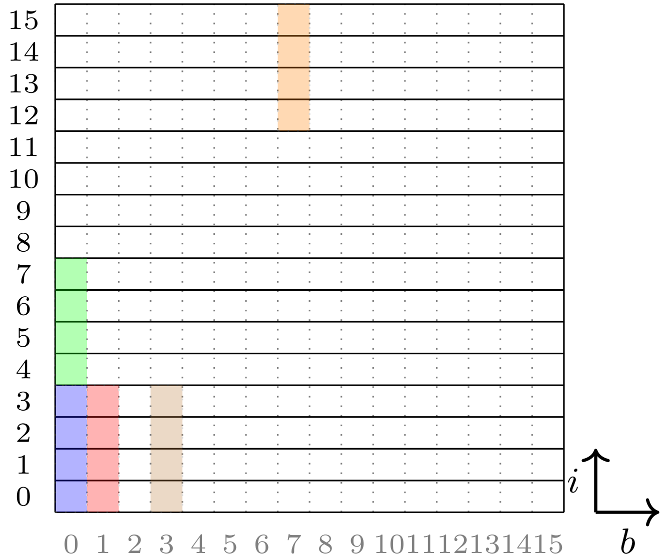
5	4	3	1	2
Saturnin:	a suite of	lightweight	symmetric algorithms for	post-quantum security

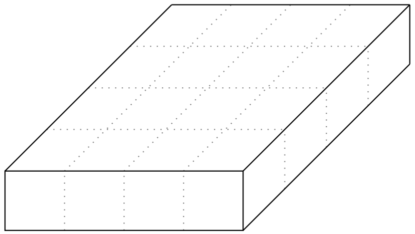
1. **we wanted to build a block cipher**
2. **. . . post-quantum:** 256-bit keys **and blocks**, quantum security claims
3. **. . . lightweight:** performs well on all platforms
4. **with quantum-secure modes of operation** for AEAD / Hashing
5. **and a good name**

Canteaut, Duval, Leurent, Naya-Plasencia, Perrin, Pornin, S., "Saturnin: a suite of lightweight symmetric algorithms for post-quantum security", ToSC S1, 2020

The state







$4 \times 4 \times 4$ cube of 4-bit nibbles

16 registers of 16 bits

16 values of 16 bits (the columns)

Operations are easier to describe

Good for implementations

Looks like a scaled-up version of AES

The round function

One round of Saturnin

- **S-Box layer**
- **Nibble permutation** SR and its inverse
- **Linear MixColumns**
- Every two rounds: **Sub-key addition** (and round constants)

Two rounds of Saturnin

Similar to a single round of AES in the AES-like representation.

- AES-128 has **10 rounds**: Saturnin has **20 rounds**.
- AES has very simple security arguments: Saturnin also.
- AES has 20 years of cryptanalysis: Saturnin benefits from it.

Modes

Saturnin-Short: AE for small messages

- Single 256-bit encryption of message and nonce

Saturnin-CTR-Cascade: all-purpose AEAD

- Encrypt-then-MAC using CTR for encryption and a Cascade MAC

Saturnin-Hash: hashing

- Merkle-Damgård with the MMO mode, using a 16 Super-round version (a.k.a. Faturnin)

Modes (ctd.)

- Saturnin-CTR-Cascade is a rate-2 AEAD (2 encryptions per block)
- (Fully) quantum-secure rate-1 AEAD **from a block cipher**, **in the standard model**, is an open question

- With the QCB mode, we can achieve rate-1 AEAD with a **related-key** quantum-secure block cipher (e.g. Faturnin)
- With a standard-secure block cipher, this is still an open question.

Bhaumik, Bonnetain, Chailloux, Leurent, Naya-Plasencia, S., Seurin, "QCB: Efficient Quantum-Secure Authenticated Encryption", ASIACRYPT 2021

Conclusion

Conclusion

A quantum adversary can:

- Use new generic algorithms
- Leverage existing classical attacks to reduce the actual bit-security (not only the generic level)
- (Sometimes) use new quantum attacks

- Symmetric cryptography holds well against quantum adversaries.
- However, the post-quantum security of our primitives / constructions should not be taken for granted, but clearly analyzed.
- Fortunately, quantum security does not come at the expense of lightness.

Thank you!