HAL
open science

New results on symmetric quantum cryptanalysis (Keynote speaker)

María Naya-Plasencia

To cite this version:

María Naya-Plasencia. New results on symmetric quantum cryptanalysis (Keynote speaker). QUANTALGO Quantum Algorithms and Applications, Sep 2018, Paris, France. hal-01953994

HAL Id: hal-01953994

https://inria.hal.science/hal-01953994v1

Submitted on 14 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

New Results on Symmetric Quantum Cryptanalysis

María Naya-Plasencia

Inria, France

ERC project QUASYModo

QUANTALGO

Paris - 26 September 2018

Outline

- Introduction
  On Quantum-Safe Symmetric Cryptography

- Efficient Quantum Collision Search
  joint work with A. Chailloux and A. Schrottenloher
  [Asiacrypt17]

- Efficient Quantum k-XOR search
  joint work with L. Grassi and A. Schrottenloher
  [Asiacrypt18]

Symmetric Cryptography

Classical Cryptography

Enable secure communications even in the presence of malicious adversaries.

Asymmetric (e.g. RSA) *(no key exchange/computationally costly)*

Security based on well-known hard mathematical problems (e.g. factorization).

Symmetric (e.g. AES) *(key exchange needed/efficient)*

Ideal security defined by generic attacks ($2^{|K|}$).

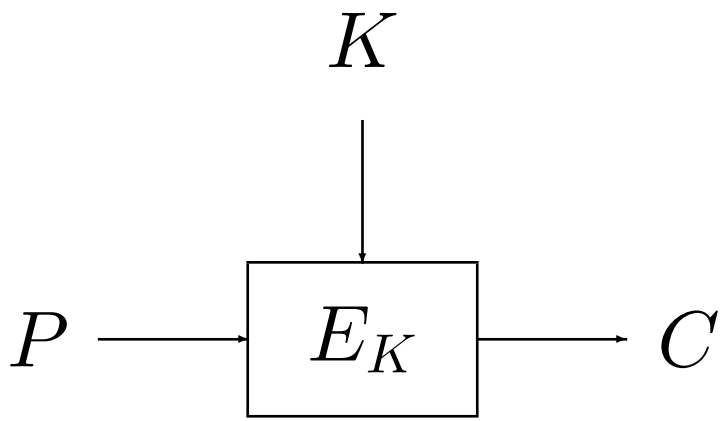
Need of continuous security evaluation (cryptanalysis).

$\Rightarrow$ Hybrid systems! (e.g. in SSH)

Symmetric primitives

- Block ciphers, (stream ciphers, hash functions..)

Message decomposed into blocks, each transformed by the same function E_K.



E_K is composed of a round transform repeated through several similar rounds.

Generic Attacks on Ciphers

- Security provided by an ideal block cipher defined by the best generic attack:
  exhaustive search for the key in $2^{|K|}$.

- Recovering the key from a secure cipher must be infeasible.

  $\Rightarrow$ typical key sizes $|K| = 128$ to 256 bits.

Cryptanalysis: Foundation of Confidence

Any attack better than the generic one
is considered a "break".

- Proofs on symmetric primitives need to make unrealistic assumptions.
- We are often left with an empirical measure of the security: cryptanalysis.
- Security redefinition when a new generic attack is found (e.g. accelerated key search with bicliques [BKR 12])

Current scenario

- Competitions (AES, SHA-3, eSTREAM, CAESAR).
- New needs: lightweight, FHE-friendly, easy-masking.
  $\Rightarrow$ Many good proposals/candidates.

- How to choose?

- How to be ahead of possible weaknesses?

- How to keep on trusting the chosen ones?

Cryptanalysis: Foundation of Confidence

When can we consider a primitive as secure?

- A primitive is secure as far as no attack on it is known.
- The more we analyze a primitive without finding any weaknesses, the more reliable it is.

Design new attacks + improvement of existing ones:

- essential to keep on trusting the primitives,
- or to stop using the insecure ones!

On weakened versions

If no attack is found on a given cipher, what can we say about its robustness, security margin?

The security of a cipher is not a 1-bit information:

- Round-reduced attacks.
- Analysis of components.

$\Rightarrow$ determine and adapt the security margin.

On high complexities

When considering large keys, sometimes attacks breaking the ciphers might have a very high complexity far from practical *e.g.*. 2^{120} for a key of 128 bits.

Still dangerous because:

- Weak properties not expected by the designers.
- Experience shows us that attacks only get better.
- Other existing ciphers without the "ugly" properties.

- When determining the security margin: find the highest number of rounds reached.

Post-Quantum Symmetric Cryptography

Post-Quantum Cryptography

Adversaries have access to **quantum computers.**

Asymmetric (e.g. RSA):

Shor's algorithm: Factorization in polynomial time

$\Rightarrow$ current systems not secure!

Solutions: lattice-based, code-based cryptography...

Symmetric (e.g. AES):

Grover's algorithm: Exhaustive search from $2^{|K|}$ to $2^{|K|/2}$.

Double the key length for equivalent ideal security.

We don't know much about cryptanalysis of current ciphers when having quantum computing available.

Post-Quantum Cryptography

Problem for present existing long-term secrets.
$\Rightarrow$ start using quantum-safe primitives NOW.

Important tasks:

- Conceive the cryptanalysis algorithms for evaluating the security of symmetric primitives in the P-Q world.

- Use them to evaluate and design symmetric primitives for the P-Q world.

Quantum Symmetric Cryptanalysis

Some recent results on Q-symmetric cryptanalysis:

3-R Feistel [Kuwakado-Morii10], Even-Mansour [Kuwakado-Morii12], Mitm [Kaplan14], Related-Key [Roetteler-Steinwandt15], Diff-lin [Kaplan-Leurent-Leverrier-NP16], Simon on modes/slides [Kaplan-Leurent-Leverrier-NP16], FX [Leander-May17], parallel multi-preim. [Banegas-Bernstein17], Multicollision [Hosoyamada-Sasaki-Xagawa17], AEZ [Bonnetain17], DS-MITM [Hosoyamada-Sasaki18], Modular additons [Bonnetain-NP18]...

Quantum Symmetric Cryptanalysis

Two main models used:

- Q1:
  classical queries and access to a quantum computer.

- Q2:
  +superposition queries to a quantum cryptog. oracle.

Very powerful, BUT...

Q2: Superposition Model

Many good reasons to study security in this scenario:

- Simple
- Non-trivial: Many constructions still seem resistant: AES, SALSA20, NMAC, HMAC...
- Inclusive of all intermediate scenarios

Defined and used in: [Zhandry12], [Boneh-Zhandry13], [Damgård-Funder-Nielsen-Salvail13], [Mossayebi-Schack16], [Song-Yun17], Simon's attacks, FX, AEZ...

An attack in this model $\Rightarrow$ might not be safe to implement the primitive in a quantum computer.

On Quantum attacks

- Compare to best generic attack,
- generic attack is accelerated, so
- broken classical primitive might be unbroken in a quantum setting.

Collision Search

w. A. Chailloux & A. Schrottenloher

Collision Search Problem

Given a random function $H : \{0,1\}^n \to \{0,1\}^n$, find $x, y \in \{0,1\}^n$ with $x \neq y$ such that $H(x) = H(y)$.

Many applications: *i.e.* generic attacks on hash functions.

(Multi-preimage search can be seen as a particular case).

Best known algorithms

	Time	Queries	Memory
Pollard's rho	$2^{n/2}$	$2^{n/2}$	$poly(n)$
Parallelization (2^s)	$2^{n/2-s}$	$2^{n/2}$	2^s

	Time	Queries	Qubits
Grover	$2^{n/2}$	$2^{n/2}$	$poly(n)$
BHT	$2^{2n/3}*$	$2^{n/3}$	$poly(n)*$
Ambainis	$2^{n/3}$	$2^{n/3}$	$2^{n/3}$

Considered Model

- The same one as in all the previous quantum algorithms BUT we limit the amout of quantum memory available to a small amount $poly(n)$.

- Available small quantum computers seems like the most plausible scenario.

- We are interested in the theoretical algorithm and we did not take into account implementation aspects.

Starting Point: BHT Algorithm

- Optimal number of queries,
- $poly(n)$ qbits,
- But time?

BHT: Summarized procedure

- Build a list L of size $2^{n/3}$ elements (classic memory),

- Exhaustive search for finding one element that collides: With AA, the number of iterations is $(\frac{2^n}{2^{n/3}})^{1/2} = 2^{n/3}$.

  Testing the membership with L for the superposition of states costs $2^{n/3}$ with n qbits:

$$\text{Time: } 2^{n/3} + 2^{n/3}(1 + 2^{n/3}) \approx 2^{2n/3}$$

Can we improve this?

Lets build the list L with distinguished points
e.g. $H(x_i) = 0^u || z$, for $z \in \{0,1\}^{n-u}$.

The cost of building the list is bigger: $2^{n/3+u/2}$.
The setup of AA is bigger: $2^{u/2}$
The membership test stays the same: $|L| = 2^{n/3}$
BUT The number of iterations is smaller: $2^{n/3-u/2}$

Time: $2^{n/3+u/2} + 2^{n/3-u/2}(2^{u/2} + 2^{n/3}) \approx 2^{2n/3-u/2} + 2^{n/3+u/2}$

With optimal parameters

The cost will be optimized for a certain size of L: $2^v \neq 2^{n/3}$.

Time: $2^{v+u/2} + 2^{\frac{n-v-u}{2}}(2^{u/2} + 2^v)$

For $v = n/5$, $u = 2n/5$: Time: $\tilde{\mathrm{O}}(2^{2n/5})$

For multiple preimage search, the algorithm is similar, but we only keep in L the distinguished points amongst the already given ones.

Comparison

	Time	Queries	Qubits	Classic Memory
Pollard	$2^{n/2}$	$2^{n/2}$	0	$poly(n)$
Grover	$2^{n/2}$	$2^{n/2}$	$poly(n)$	0
BHT	$2^{2n/3}$	$2^{n/3}$	$poly(n)$	$2^{n/3}$
Ambainis	$2^{n/3}$	$2^{n/3}$	$2^{n/3}$	0
New algorithm	$2^{2n/5}$	$2^{2n/5}$	$poly(n)$	$2^{n/5}$

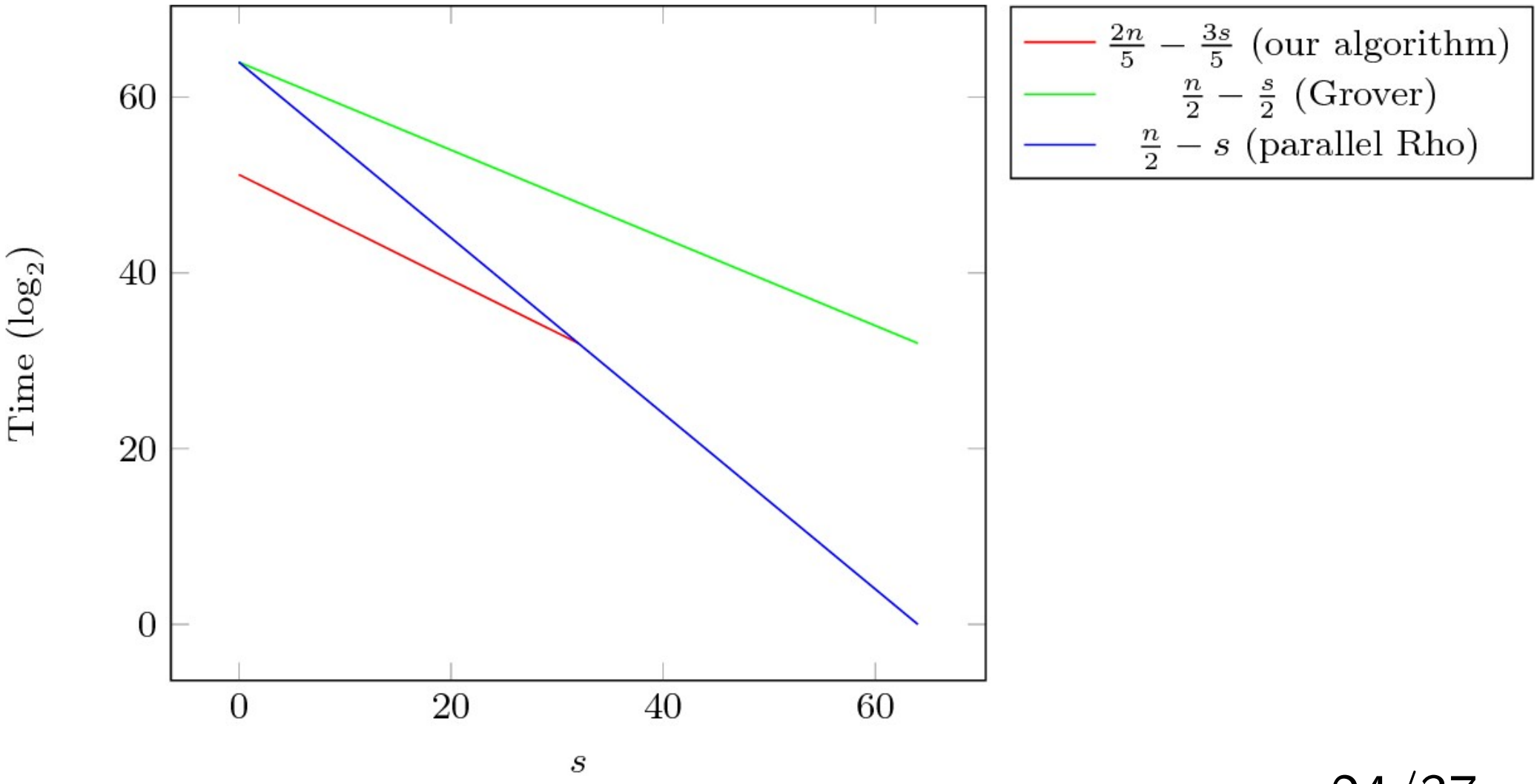
Parallelization

With 2^s n-qbit registers and "external" parallelization we can achieve:

Time: $2^{v+u/2-s} + 2^{\frac{n-v-u}{2}-s/2}(2^{u/2} + 2^v)$

Our theoretical algorithm seems more efficient than classical parallelization/Beal up to $s = n/4$

Comparison example: n=128



Example of Applications (1)

- 1. Hash functions: Collision and Multi-preimages time from $2^{n/2}$ to $2^{2n/5}$ and $2^{3n/7}$ (Q1).
  Ex.- time and queries for $n = 128$:
  rho$= 2^{64}$, ours$= 2^{51.2}$ (with less than 1GB classical)

- 2. Multi-user setting: Recover Ctxt, from same Ptxt, 2^t different keys: apply multi-preimage algorithm (Q1). Depending on the value of t different gain.

Example of Applications (2)

- 3. Operation modes: Collision attacks on CBC: 2^t Ctxt, find one preimage $\Rightarrow$ Ptxt. (Q2). If frequent rekeying (Q1).
- 4. Bricks for Cryptanalysis: Collision, multi-preimage search: often bricks of more technical cryptanalysis: improve the steps.

Conclusion 1

New efficient collision search algorithm with small quantum memory.

Many applications in symmetric cryptograhy.

Open question: is it possible to meet the optimal $2^{n/3}$ in time with small quantum memory? (Quantum random walks, quantum learning graphs...?)

Quantum Efficient Algorithms for the k-XOR Problem

w. L. Grassi & A. Schrottenloher

k-XOR problem with random functions

Given query access to a random function $H : \{0,1\}^n \to \{0,1\}^n$, find $x_1, \ldots, x_k$ such that $H(x_1) \oplus \ldots \oplus H(x_k) = 0$.

For us, equivalent to the case with k different random functions.

Many applications (with k-SUM, similar algorithms apply), ex.: attacks on FSB, XLS, SWIFFT; correlation attacks.

The 3-XOR problem

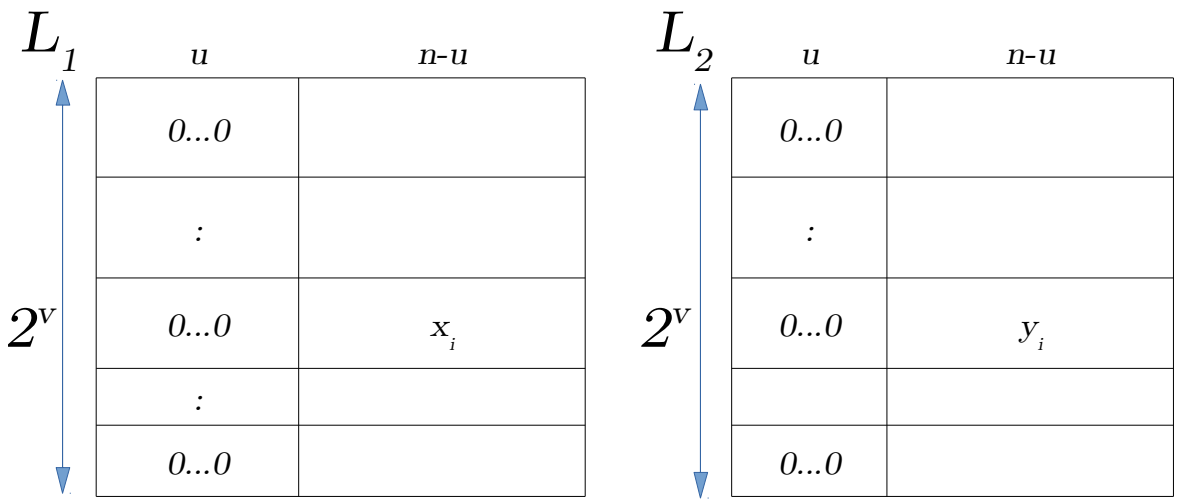
Find 3 elements that XOR to 0: not much better than collision in classical setting.

Classically, no exponential acceleration, only logarithmic factors:

Complexity of about $2^{n/2}$ with out this factors.

3-XOR: Low Quantum Memory Algorithm

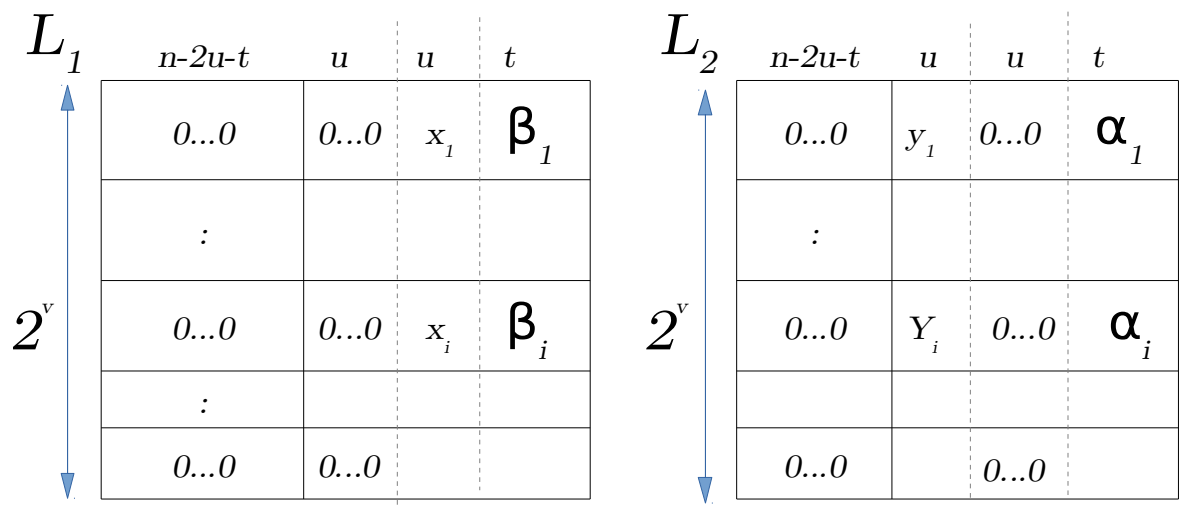
- 1st approach, distinguished point: $2^v = 2^{n/8}$, $T = 2^{3n/8}$



- Intuition: With a memory of $2^v + 2^v$ we obtain 2^{2v} potential collisions.

3-XOR: Low Quantum Memory Algorithm

- 1st approach, distinguished point: $2^v = 2^{n/8}$, $T = 2^{3n/8}$
- 2nd approach, techniques linked to "list merging":



Improved time$= 2^{5n/14}$, with $2^v = 2^{n/7}$.

- More efficient than collision, contrary to classical!

3-XOR: High Quantum Memory Algorithm

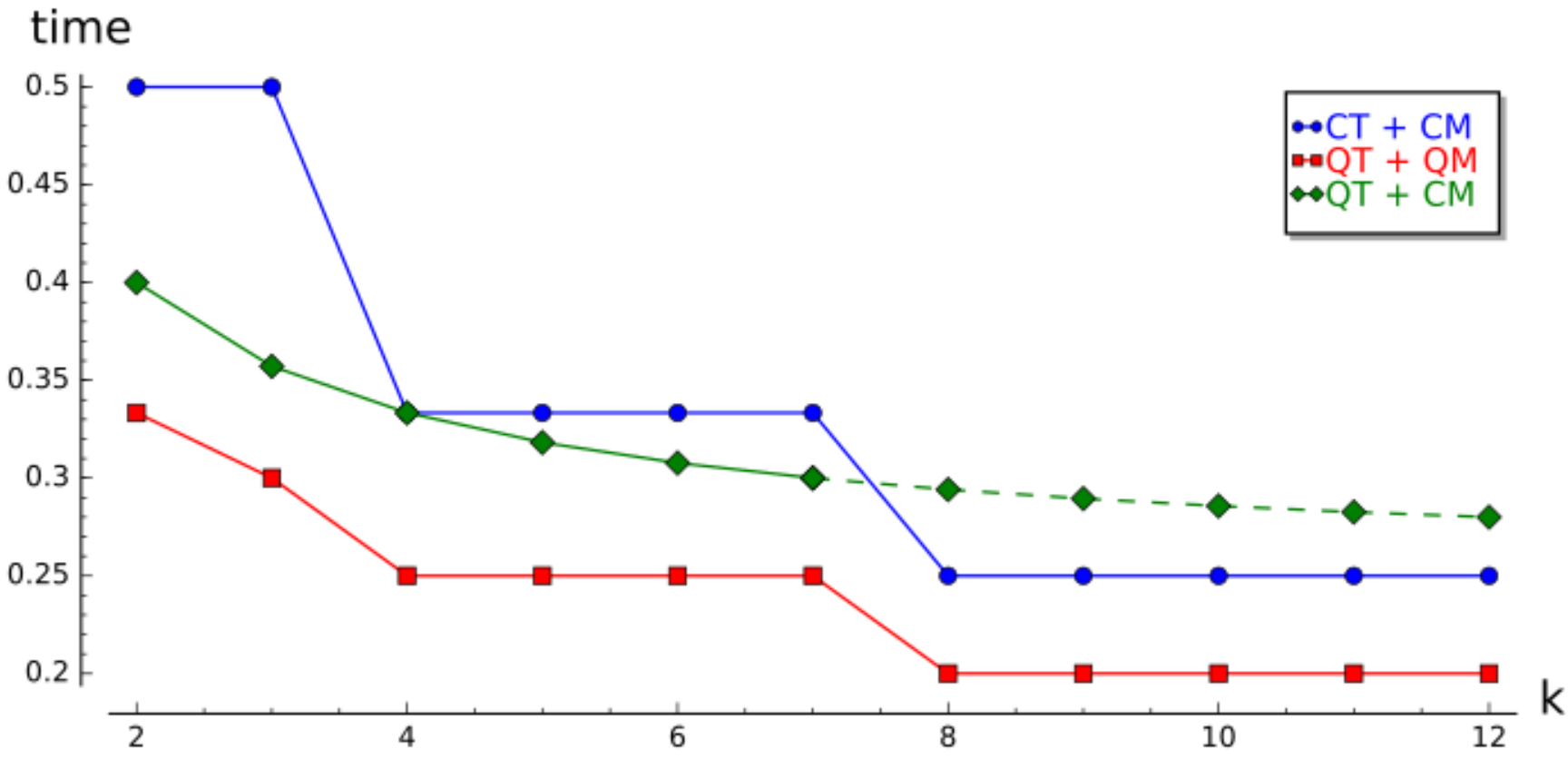
- Same technique as before, but no need for the positions to '0' in both lists.

- Complexity of:
  $2^{v+u/2} + 2^{\frac{n-2v}{2}}(2^{v-u})$.

- This becomes optimal for
  QM$= 2^{n/5}$ and Time$= 2^{3n/10}$.

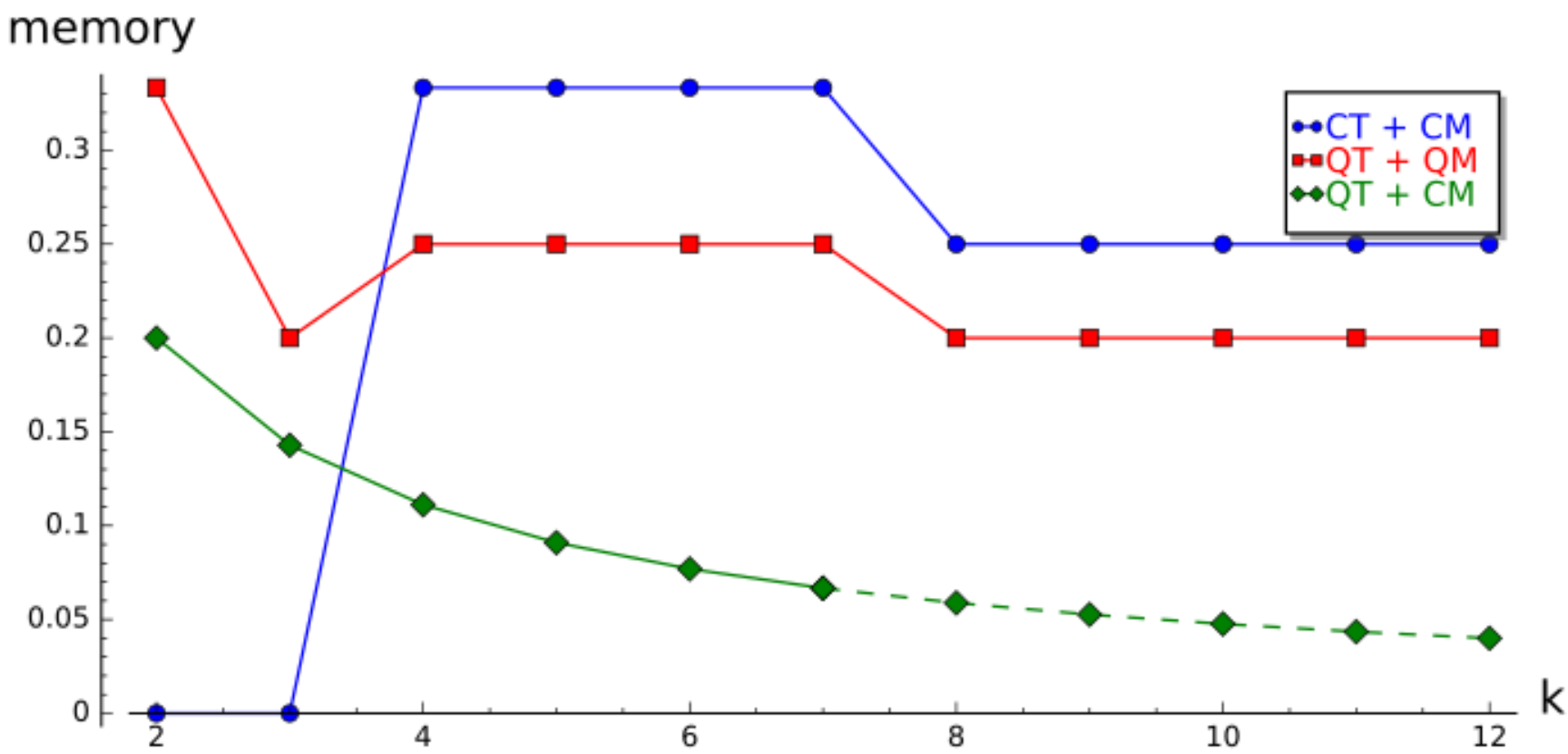
The k-XOR algorithms

Similar algorithms can be applied to other values of k



The k-XOR algorithms

Similar algorithms can be applied to other values of k



Conclusion 2

- We have shown that quantum 3-xor problem is exponentially easier that the quantum collision problem (in both settings), contrary to classical.
- The complexity of solving the 3-xor problem with allowed quantum memory beats the lower bound for quantum collision of $2^{n/3}$
- For generic k, low quantum memory improves Wagner up to $k = 8$, and allowed quantum memory for all k.

Final Conclusion

Open problems

- Optimal collision time $2^{n/3}$?.
- Algebraic attacks
- Boomerang attacks
- FSE Stevens: Quantum cryptanalysis of SHA-2?
- AES quantum evaluation- on going work.
- Generic key-length extensions?
- What about state size? ...

Lots of things to do !