Quantum Period Finding against Symmetric Primitives in Practice

Xavier Bonnetain Samuel Jaques

"Practical" quantum computing

An issue with quantum attacks

- We can't run them
- We often only have asymptotics

What we do

We propose complete quantum circuits for the offline Simon's cryptanalysis

Aims

- Get a better understanding of the attack
- Allow comparison with other quantum algorithms
- Study the limitations of the attack

Q#

- We wrote the components of the attack in Q#, a quantum programming language
- Simulates and tests X, CNOT, Toffoli, And, up to thousands of qubits
- Counts resource use with some rudimentary optimization
- The library is available: `https://github.com/sam-jaques/offline-quantum-period-finding`

Simon's algorithm

Simon's problem

Simon's problem

- $f : \{0,1\}^n \to \{0,1\}^n$
- $s \in \{0,1\}^n$
- $\forall x, y, f(y) = f(x) \Leftrightarrow x \oplus y \in \{0, s\}$
- f hides the period s
- Goal : find s, given oracle access to f.

Classical resolution

Find a collision, in $\Omega\left(2^{n/2}\right)$ samples.

Quantum resolution

Simon's algorithm, in $\mathcal{O}(n)$ quantum queries, $\mathcal{O}\left(n^3\right)$ classical operations

Simon's algorithm [Sim94]

Simon's problem

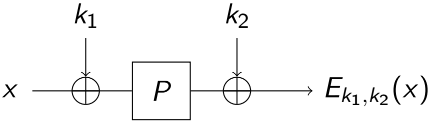
- $f : \{0,1\}^{n} \to \{0,1\}^{n}$, $s \in \{0,1\}^{n}$
- $\forall x, y, f(y) = f(x) \Leftrightarrow x \oplus y \in \{0, s\}$
- Goal : find s, given oracle access to f.

Simon's algorithm

- Superposition queries $\sum_x |x\rangle |f(x)\rangle$
- Sample y: $s \cdot y = 0$
- Repeat $O(n)$ times and solve the system

The Even-Mansour Cipher

Built from a random permutation $P : \{0,1\}^{n} \to \{0,1\}^{n}$.

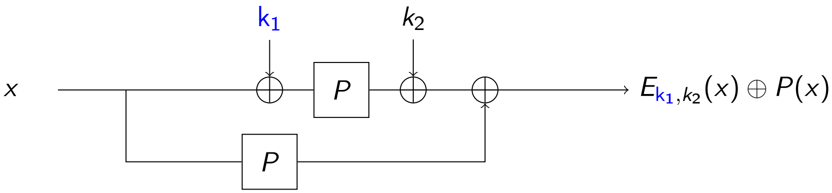


$$E_{k_1,k_2}(x) = k_2 \oplus P(x \oplus k_1)$$

Classical security

Any attack needs Time $\times$ Data $\geq 2^{n}$

Quantum attack [KM12]

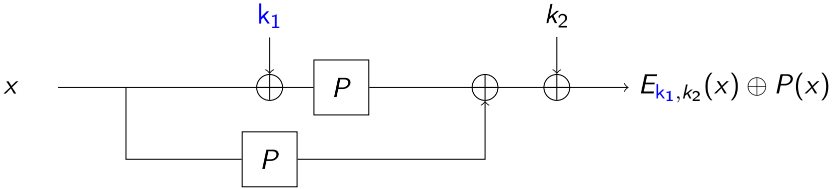


Quantum attack

$f(x) = E_{\mathsf{k}_1,k_2}(x) \oplus P(x)$ satisfies $f(x \oplus \mathsf{k}_1) = f(x)$.

Even-Mansour is broken in polynomial time, with quantum query access.

Quantum attack [KM12]

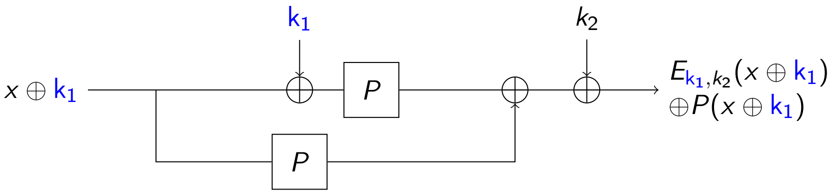


Quantum attack

$f(x) = E_{k_1,k_2}(x) \oplus P(x)$ satisfies $f(x \oplus k_1) = f(x)$.

Even-Mansour is broken in polynomial time, with quantum query access.

Quantum attack [KM12]

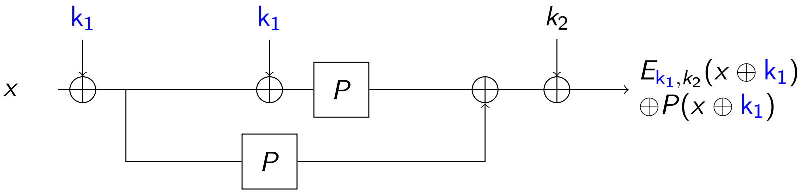


Quantum attack

$f(x) = E_{k_1,k_2}(x) \oplus P(x)$ satisfies $f(x \oplus k_1) = f(x)$.

Even-Mansour is broken in polynomial time, with quantum query access.

Quantum attack [KM12]

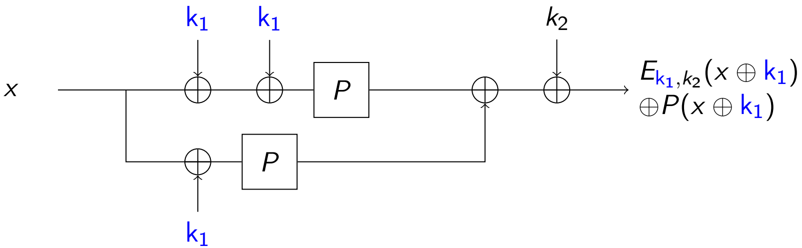


Quantum attack

$f(x) = E_{k_1,k_2}(x) \oplus P(x)$ satisfies $f(x \oplus k_1) = f(x)$.

Even-Mansour is broken in polynomial time, with quantum query access.

Quantum attack [KM12]

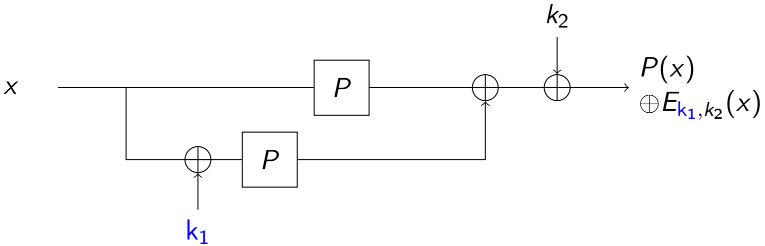


Quantum attack

$f(x) = E_{k_1,k_2}(x) \oplus P(x)$ satisfies $f(x \oplus k_1) = f(x)$.

Even-Mansour is broken in polynomial time, with quantum query access.

Quantum attack [KM12]



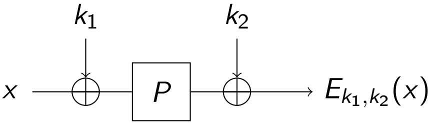
Quantum attack

$f(x) = E_{k_1,k_2}(x) \oplus P(x)$ satisfies $f(x \oplus k_1) = f(x)$.

Even-Mansour is broken in polynomial time, with quantum query access.

The Offline Simon's Algorithm

Removing the quantum queries

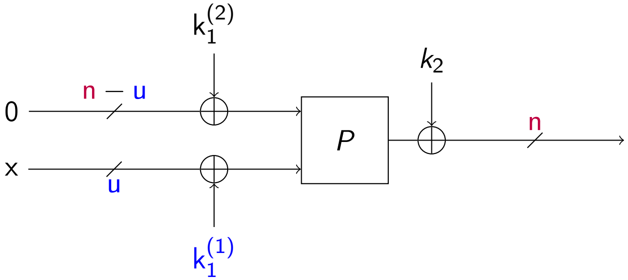


Producing the sample states with Q1 queries is possible. . . in time 2^n, with the whole codebook.

$\implies$ not an attack.

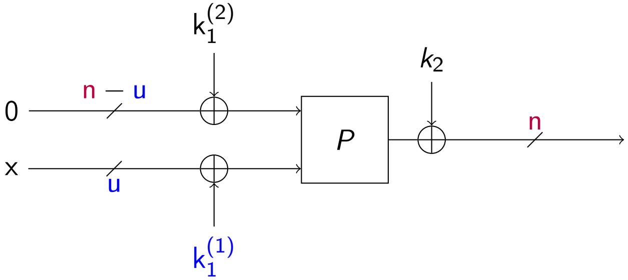
Q1 attack on Even-Mansour

We separate k_1 in two parts.



Q1 attack on Even-Mansour

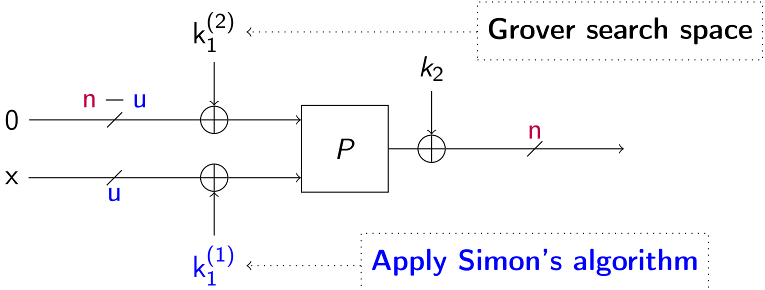
We separate k_1 in two parts.



Define $f(x) = E_{k_1,k_2}(x \| 0^{n-u}) \oplus P(x \| k_1^{(2)})$.

Q1 attack on Even-Mansour

We separate k_1 in two parts.



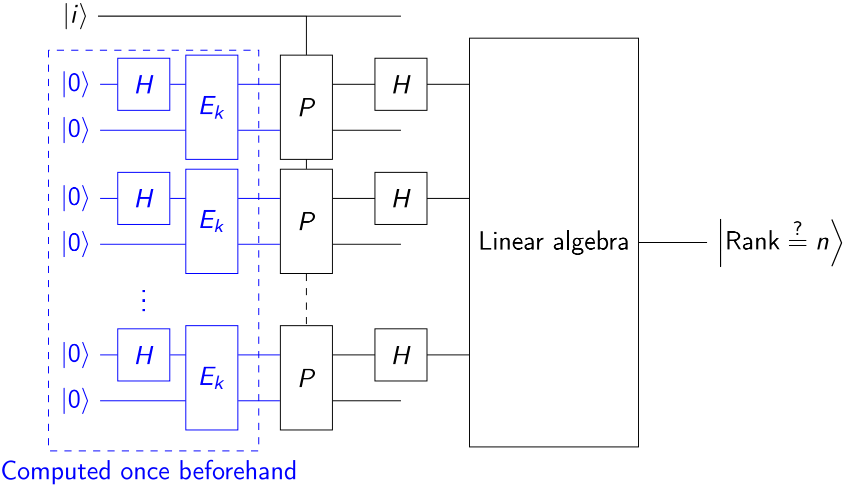
Define $f(x) = E_{k_1,k_2}(x \| 0^{n-u}) \oplus P(x \| k_1^{(2)})$.

Data: 2^u **Memory:** $\mathcal{O}(nu)$

Time: $\mathcal{O}\left(2^u + 2^{(n-u)/2}\right)$

Quantum circuits

Shape of the circuit



Concrete query estimates [Bon21]

For an Even-Mansour with an n-bit state, it is enough to have:

- $n+9$ queries to the periodic function
- 11 bits of output for the periodic function

Attack-specific optimizations

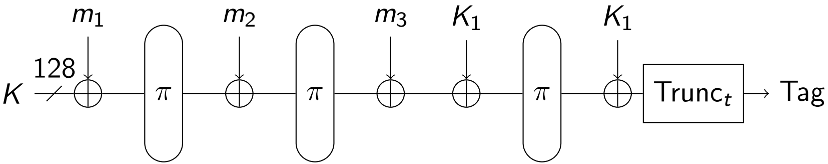
Attack properties

- Part of the input is fixed
- We only need 11 bits of output
- only relevant property is the periodicity

Optimizations

- Compute only once the shared part
- Remove useless parts of the last rounds
- Remove the linear functions of the last rounds

Primitive: Chaskey



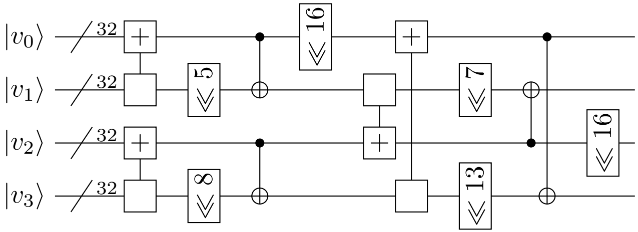
- Lightweight MAC, ISO standard
- At most 2^{48} message blocks with the same key.

ARX construction

- Addition: Easily in-place; cheap circuits are well-studied
- Rotation: Done "in-software" by re-labelling qubits
- Xor: Just CNOT gates

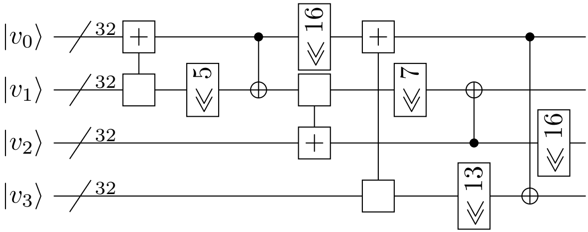
Chaskey Circuits

First round:



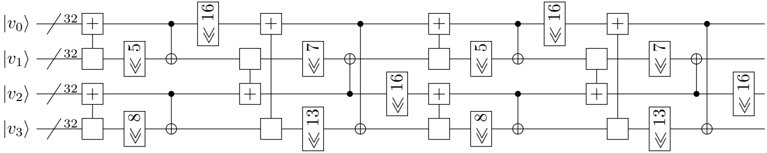
Chaskey Circuits

First round:



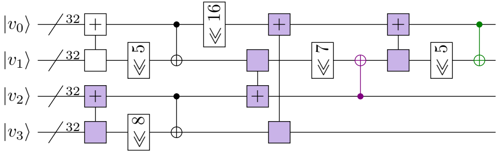
Chaskey Circuits

Last 2 rounds:



Chaskey Circuits

Last 2 rounds:



Linear Algebra

Circuit to find the rank of an $m \times n$ binary matrix, with $m > n$:

- Compute a triangular basis and reduce the input vectors in-place.
- Depth: $O((m+n)\lg n)$
- Gates: $mn^2 + mn$ Toffoli gates
- Qubits: mn as input, plus $m + \frac{n(3n-1)}{2}$ extra qubits

Chaskey Summary

Target	Bitlength	Offline Queries	Operations All	Operations T	Depth All	Depth T	Qubits	Note
Chaskey-8	128	48	64.9	64.4	56.0	53.9	14.5	limited queries
Chaskey-12	128	48	65.1	64.5	56.4	54.1	14.5	
Chaskey-8	128	50	64.3	64.0	55.5	54.4	14.5	unlimited queries
Chaskey-12	128	51	64.5	64.2	55.9	55.2	14.5	
Chaskey-8	128	1	80.3	77.5	79.0	75.4	8.6	Exhaustive search
Chaskey-12	128	1	80.8	78.0	79.6	75.9	8.6	

All figures in log base 2 except bitlength.

Conclusion

Summary

- The attack is competitive against exhaustive search
- Requires large amount of data
- Reasonable amount of qubits (~ 10000)
- Not a near-term quantum attack
- Requires a specific structure

Limitations of the attack

Optimizations are limited with an increased number of rounds:

- $\sim 25\%$ gain with Chaskey-8 (8 rounds), $\sim 1\%$ gain with Elephant (80 rounds)

Limitations of the attack

Optimizations are limited with an increased number of rounds:

- $\sim$ 25% gain with Chaskey-8 (8 rounds), $\sim$ 1% gain with Elephant (80 rounds)

Many approaches can limit the impact of the attack:

- Data limitation (ex. Chaskey, Elephant)
- Large state size (ex. Elephant)
- Avoid the required structure (ex. PRINCEv2)

Thanks for listening!

Target	Bitlength	Offline Queries	Operations All	Operations T	Depth All	Depth T	Qubits	Source
RSA	2048	–	–	31	31	–	12.6	[GE19]
Chaskey-8	128	48	64.9	64.4	56.0	53.9	14.5	ours
Chaskey-12	128	48	65.1	64.5	56.4	54.1	14.5	
PRINCE	64	48	65.0	64.5	55.2	53.8	14.0	
Elephant	160	47	84.1	82.5	72.6	70.4	14.8	
	176	47	92.5	90.9	80.8	78.5	15.1	
	200	69	93.6	91.7	83.7	79.3	16.4	
AES	128	1	82.3	80.4	74.7	71.6	10.7	[DP20]

All figures in log base 2 except bitlength.