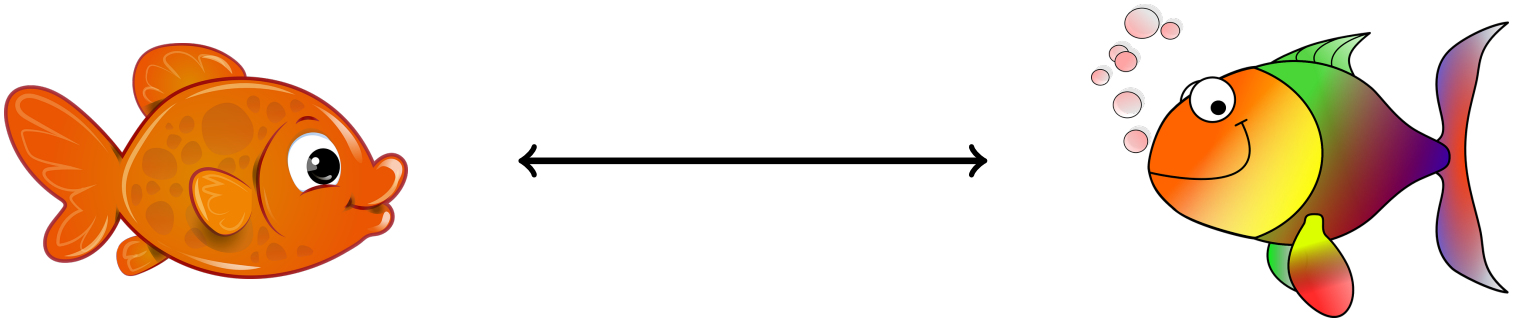
Hidden Structures in Quantum Cryptanalysis

Xavier Bonnetain

December 12, 2019

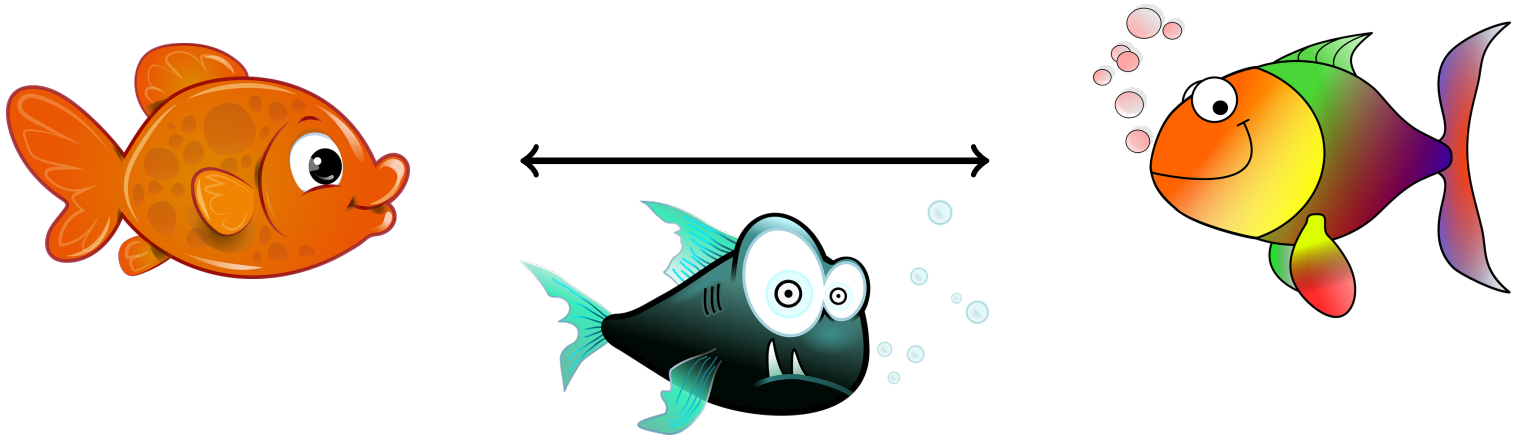
Cryptography

Aims at enabling secure communications



Cryptography

Aims at enabling secure communications

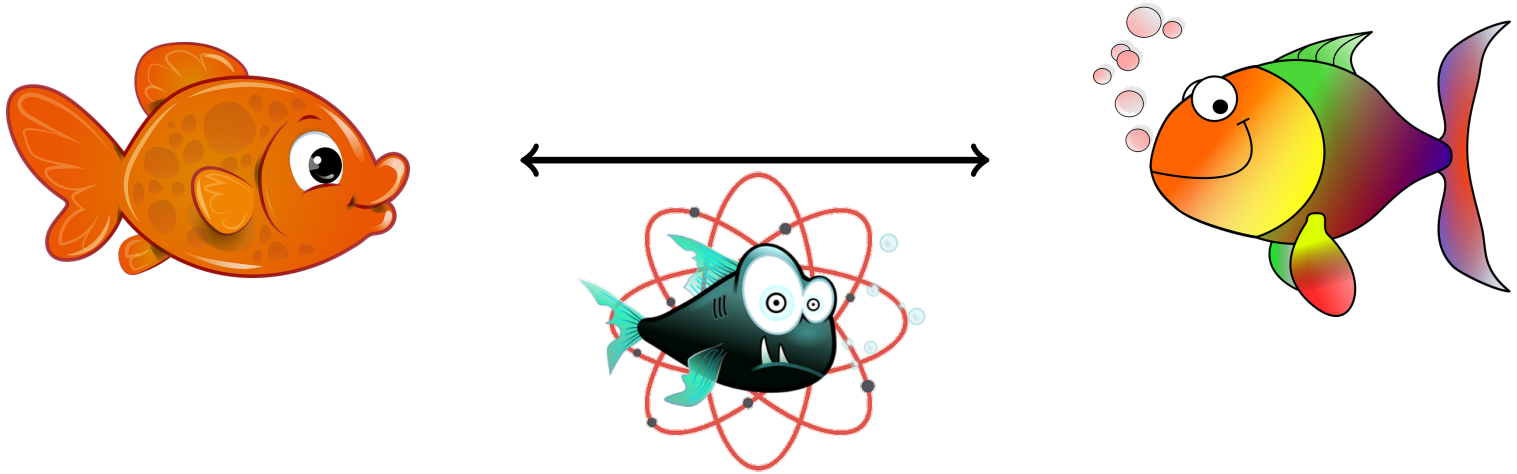


An adversary appears!

She wants to attack the communication.

Cryptography

Aims at enabling secure communications



A quantum adversary appears!

She wants to attack the communication.

Quantum computing

Classical computing

x	Input
$\downarrow$	
$a, b, \ldots$	Intermediate values
$\downarrow$	
y	Final result

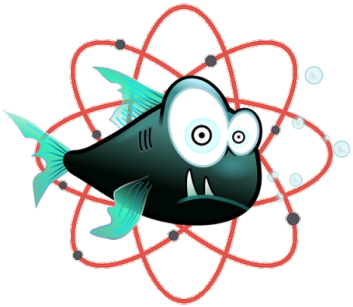
Quantum computing

$\lvert x\rangle$	Input
$\downarrow$	
$\lvert a, b, \ldots\rangle$	Intermediate state
$\downarrow$	
$\lvert y\rangle \longrightarrow y$	Final measurement

Differences

- More possibilities $|0\rangle$, $|1\rangle$, $|0\rangle - |1\rangle \ldots$
- Reversible computing
- New operators $H : |b\rangle \mapsto \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^b |1\rangle\right)$

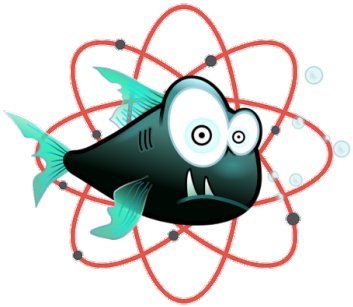
Quantum Adversary



First model: Classical queries (Q1)

- The adversary can perform quantum computing
- The adversary is restricted to classical queries : $x \mapsto f(x)$

Quantum Adversary



First model: Classical queries (Q1)

- The adversary can perform quantum computing
- The adversary is restricted to classical queries : $x \mapsto f(x)$

Second model: Quantum queries (Q2)

- The adversary can perform quantum computing
- Can do quantum queries : $\sum_x |x\rangle |0\rangle \mapsto \sum_x |x\rangle |f(x)\rangle$

Reference quantum algorithms

Shor's algorithm [Sho94]

Solves the factorization and discrete logarithm problems in polynomial time.

Quantum Search [Gro96]

- Search space of size $2^{\mathbf{n}}$, quantum test circuit T_χ for χ
- Find $\mathbf{s}$ such that $\chi(\mathbf{s}) = 1$ in $2^{\mathbf{n}/2}$ iterations of T_χ

Quantum Collision search [BHT98]

- $f : \{0,1\}^{\mathbf{n}} \to \{0,1\}^{\mathbf{n}}$, find $x \neq y : f(x) = f(y)$
- Classically: $2^{\mathbf{n}/2}$ operations
- Brassard-Høyer-Tapp: $2^{\mathbf{n}/3}$ operations and $2^{\mathbf{n}/3}$ quantum memory

Outline

Simon's algorithm

Simon's problem

Simon's problem

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

Classical resolution

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

Quantum resolution

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

Simon's algorithm [Sim94]

Quantum circuit

- *Start from* $|0\rangle\,|0\rangle$

Simon's algorithm [Sim94]

Quantum circuit

- Start from $|0\rangle |0\rangle$
- *Apply H:* $\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle$

Simon's algorithm [Sim94]

Quantum circuit

- Start from $|0\rangle |0\rangle$
- Apply H: $\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle$
- *Apply* O_f*:* $\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle$

Simon's algorithm [Sim94]

Quantum circuit

- Start from $|0\rangle |0\rangle$
- Apply H: $\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle$
- Apply O_f: $\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle$
- *Measure the second register: get* $f(x_0)$ *and project to* $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus s\rangle)$

Simon's algorithm [Sim94]

Quantum circuit

- Start from $|0\rangle |0\rangle$
- Apply H: $\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle$
- Apply O_f: $\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle$
- Measure the second register: get $f(x_0)$ and project to $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus s\rangle)$
- *Reapply H:* $\frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} |y\rangle + (-1)^{(x_0 \oplus s) \cdot y} |y\rangle$

Simon's algorithm [Sim94]

Quantum circuit

- Start from $|0\rangle|0\rangle$
- Apply H: $\frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle|0\rangle$
- Apply O_f: $\frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle|f(x)\rangle$
- Measure the second register: get $f(x_0)$ and project to $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus s\rangle)$
- Reapply H: $\frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1}(-1)^{x_0\cdot y}|y\rangle + (-1)^{(x_0\oplus s)\cdot y}|y\rangle$
- *The state is* $\frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1}(-1)^{x_0\cdot y}(1+(-1)^{s\cdot y})|y\rangle$

The y_0 we measure must satisfy $1+(-1)^{s\cdot y_0} \neq 0 \Rightarrow y_0 \cdot s = 0$.

Simon's algorithm [Sim94]

Simon's problem

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

Simon's algorithm

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

Generalizations

- Works if $\forall x, y, f(y) = f(x) \Leftarrow x \oplus y \in \{0, s\}$
- Works with multiple periods

Simon-based cryptanalysis

General idea

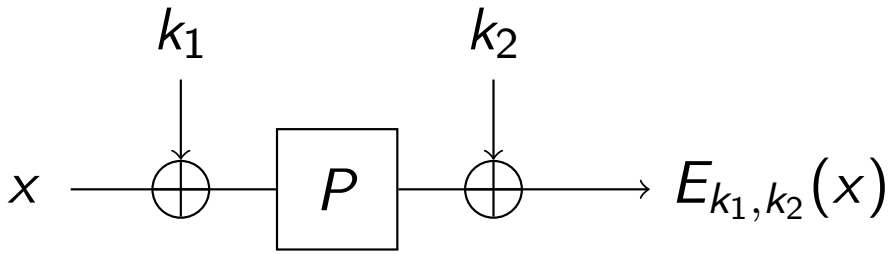
Create a periodic function from a cipher, whose period is a secret.

Characteristics

- Polynomial time, only $\mathcal{O}(n)$ queries
- Require quantum queries

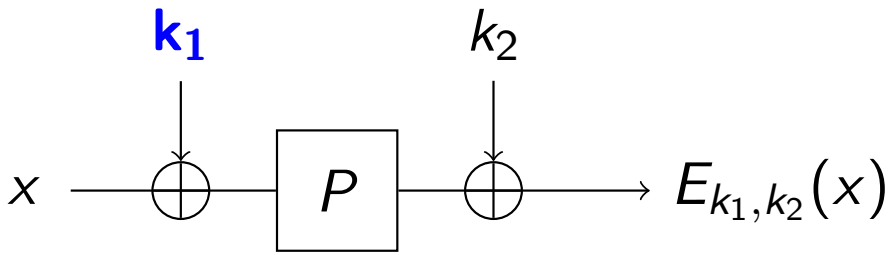
The Even-Mansour Cipher

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



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

Classical collision attack



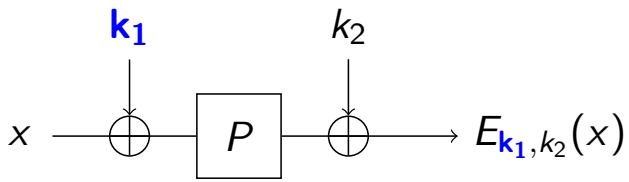
Let $f(x) = E_{\mathbf{k_1},k_2}(x) \oplus P(x) = k_2 \oplus P(x \oplus \mathbf{k_1}) \oplus P(x)$

It satisfies $f(x \oplus \mathbf{k_1}) = f(x)$: a collision of f yields $\mathbf{k_1}$ w.h.p.

Attack Cost

- Time $2^{\mathbf{n}/2}$, memory 1
- Data $2^{\mathbf{n}/2}$

Classical data-time tradeoff



Let $g(y) = P(y) \oplus P(y \oplus \mathbf{1})$, then

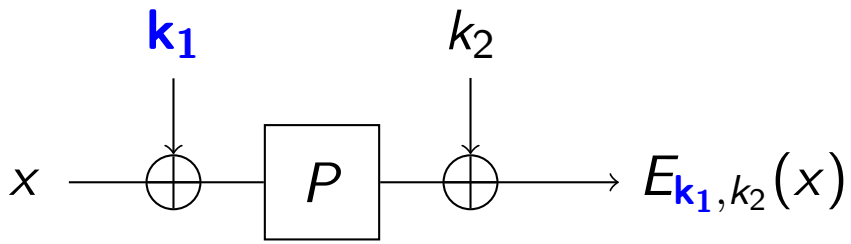
$$g(x \oplus \mathbf{k_1}) = E_{\mathbf{k_1},k_2}(x) \oplus E_{\mathbf{k_1},k_2}(x \oplus \mathbf{1})$$

Attack

- Collect D values of $E_{\mathbf{k_1},k_2}(x) \oplus E_{\mathbf{k_1},k_2}(x \oplus \mathbf{1})$
- Store a database $\mathcal{D}$ of size D
- Find y such that $g(y) \in \mathcal{D}$, in time $2^{\mathbf{n}}/D$
- With good probability $y = x \oplus \mathbf{k_1}$

Trade-off curve: $T \cdot D = 2^{\mathbf{n}}$, uses D memory.

Quantum attack [KM12]



Quantum attack

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

Even-Mansour is broken in polynomial time in the Q2 model.

Simon-based cryptanalysis

- Distinguishers on Feistel constructions
- Multiple quantum slide attacks
- AEZ
- Multiple modes of operation
- Quantum related-key attacks
- ...

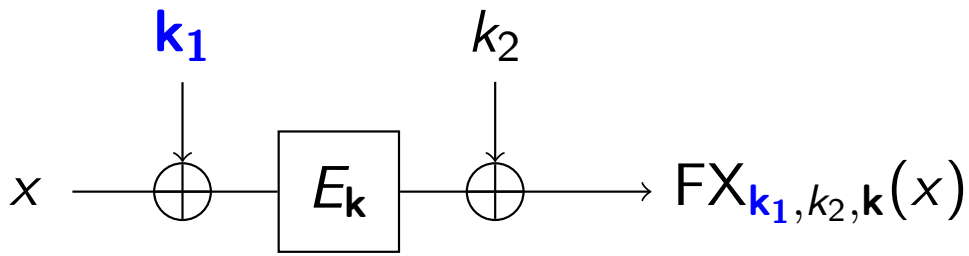
Simon-based cryptanalysis

- Distinguishers on Feistel constructions
- Multiple quantum slide attacks
- AEZ
- Multiple modes of operation
- Quantum related-key attacks
- ...

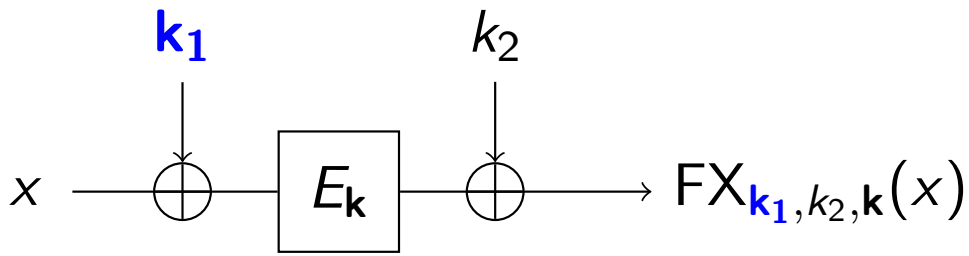
Require quantum queries

The Offline Simon's Algorithm

Example: FX construction



Example: FX construction



Quantum attack: "Grover-meet-Simon" [LM17]

- Quantum search for $\mathbf{k}$
- Checking: Kuwakado and Morii's attack works $\iff$ the guess of $\mathbf{k}$ is correct

Total time is $\underbrace{poly(\mathbf{n})}_{\text{Simon's algo}} \times \underbrace{2^{|\mathbf{k}|/2}}_{\text{Grover's iterates}}$.

Our remark on FX [BHNSS19]

The function:

$$f_z(x) = \mathrm{FX}_{\mathbf{k_1},k_2,\mathbf{k}}(x) \oplus E_z(x)$$

has $f_z(x \oplus \mathbf{k_1}) = f_z(x)$ **if** $z = \mathbf{k}$ **(the good one)**.
f_z is a sum:

$$f_z(x) = \underbrace{\mathrm{FX}_{\mathbf{k_1},k_2,\mathbf{k}}(x)}_{\substack{\textbf{Independent of z: online function } f}} \oplus \underbrace{E_z(x)}_{\substack{\textbf{Grover search space: offline function } g}}$$

For one query to f_z

- Do one quantum query to $\mathrm{FX}_{\mathbf{k_1},k_2,\mathbf{k}}(x)$ (fixed!)
- Add $E_z(x)$ (only depends on public information)

A new test algorithm

1. Begin with $\mathcal{O}(\mathbf{n})$ states of the form $\sum_{x\in\{0,1\}^{\mathbf{n}}} |x\rangle\,|f(x)\rangle$
2. Make queries to g and build: $\sum_{x\in\{0,1\}^{\mathbf{n}}} |x\rangle\,|(f\oplus g)(x)\rangle$
3. With Simon's algorithm, obtain a single output bit: whether $f\oplus g$ has a period or not
4. **Revert the computations**, query g again, put the "sample states" back to

$$\sum_{x\in\{0,1\}^{\mathbf{n}}} |x\rangle\,|f(x)\rangle$$

This emulates a reversible quantum circuit that tests for the periodicity of $f\oplus g$, *with only preprocessed queries to* f.

Our Q2 attack on FX

The queries to $\mathrm{FX}_{\mathbf{k_1},k_2,\mathbf{k}}(x)$ are made beforehand.

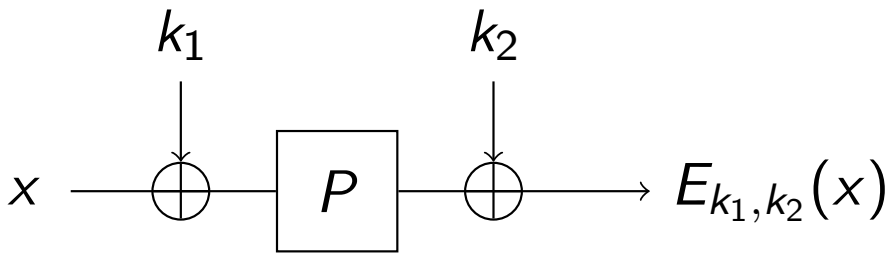
Test function

- Fetch the sample states $\sum_{x\in\{0,1\}^{\mathbf{n}}} |x\rangle\,|\mathrm{FX}_{\mathbf{k_1},k_2,\mathbf{k}}(x)\rangle$
- Create the Simon states $\sum_{x\in\{0,1\}^{\mathbf{n}}} |x\rangle\,|\mathrm{FX}_{\mathbf{k_1},k_2,\mathbf{k}}(x)\oplus E_z(x)\rangle$
- Test if there is a period
- **Revert** the operations and get back the sample states

Quantum search cost

- Time unchanged
- Queries reduced from $\mathcal{O}\left(\mathbf{n}2^{|\mathbf{k}|/2}\right)$ to $\mathcal{O}\left(\mathbf{n}\right)$
- Needs $\mathcal{O}\left(\mathbf{n}^2\right)$ Qubits

Back to the Even-Mansour cipher

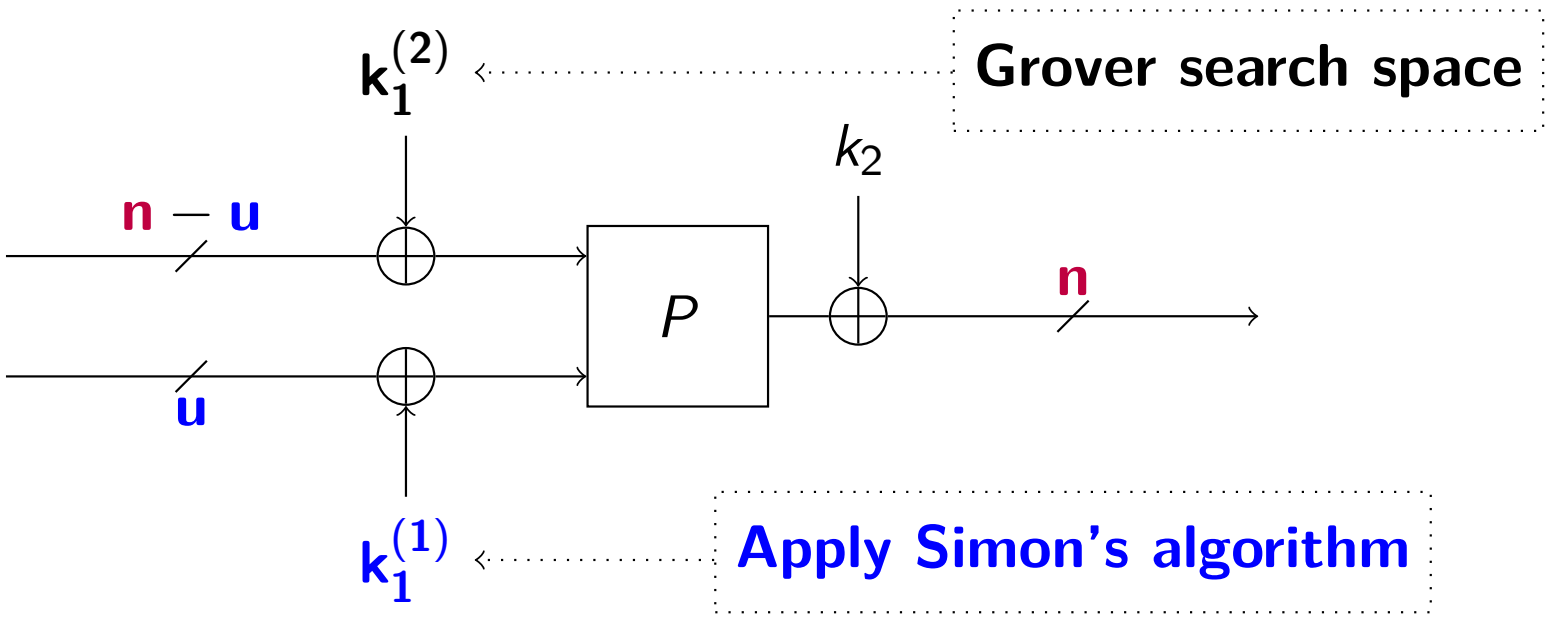


Producing the sample states with Q1 queries is possible. . . in time $2^{\mathbf{n}}$, with the whole codebook.

$\Longrightarrow$ not an attack.

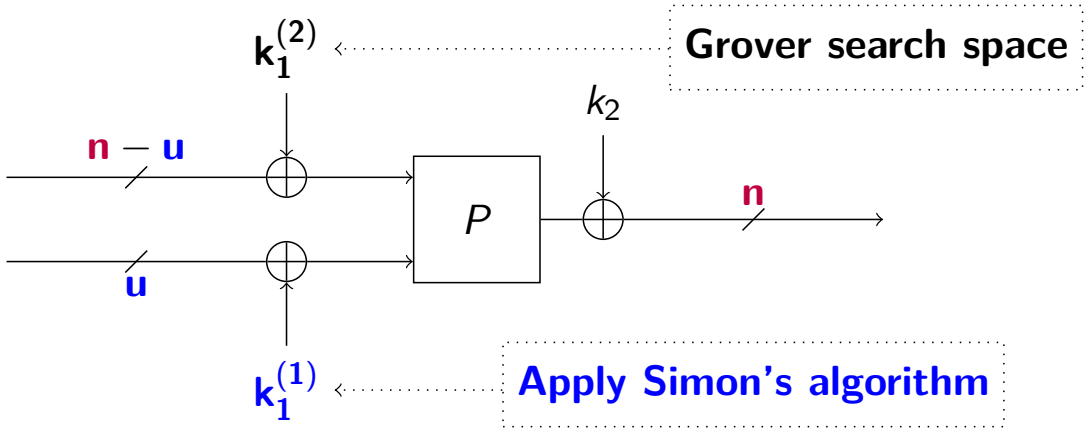
Q1 attack on Even-Mansour

We separate k_1 in two parts.



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

Q1 attack on Even-Mansour (ctd.)



$f(x) = E_{k_1,k_2}(x\|0^{\mathbf{n}-\mathbf{u}}) \oplus P(x\|\mathbf{k}_1^{(2)})$ has period $\mathbf{k}_1^{(1)}$

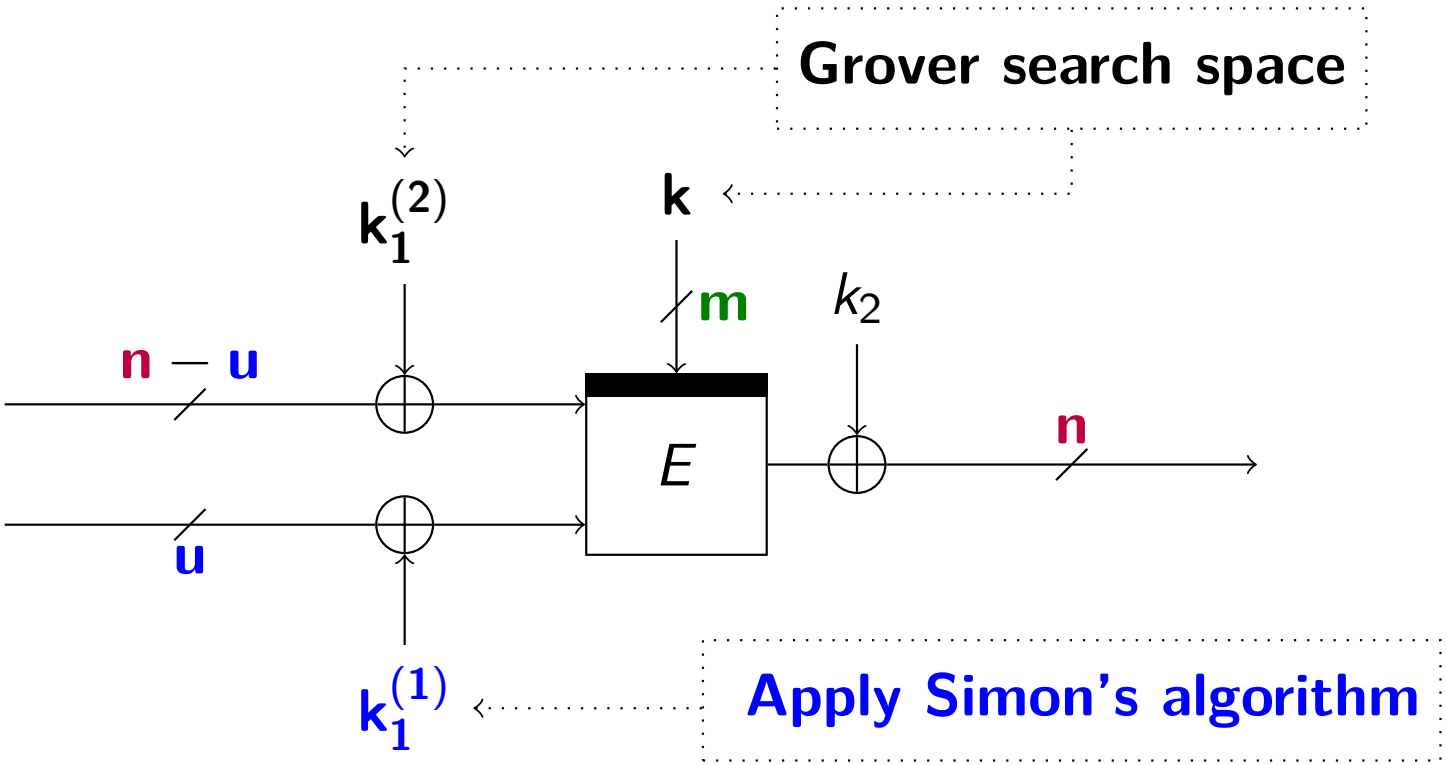
1. Produce the sample states $\sum_x |x\rangle |E_{k_1,k_2}(x\|0^{\mathbf{n}-\mathbf{u}})\rangle$
2. Search the good $\mathbf{k}_1^{(2)}$ ($\mathbf{n}-\mathbf{u}$ bits)

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

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

Balances when Data = Time = $2^{\mathbf{n}/3}$

Q1 attack on the FX construction



We do the same, with more guesses in Grover's algorithm:
Data = Time = $2^{(n+m)/3}$.

Summary

The offline approach

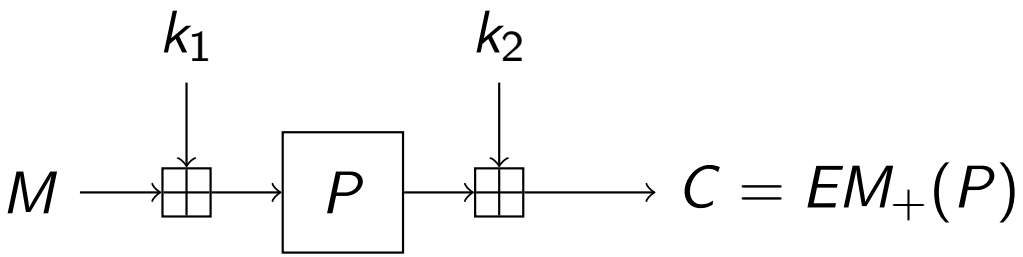
We reuse the quantum queries for each iteration of Simon's algorithm when the periodic function allows it.

Consequences

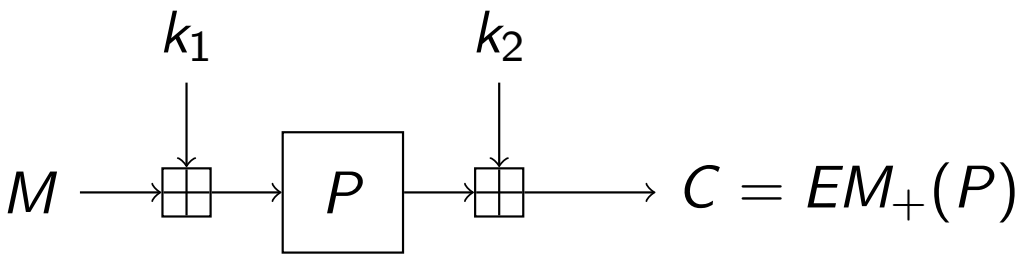
- Drastically reduces the number of quantum queries.
- Allows to convert a Q2 attack into a Q1 attack.
- Provides the best known Q1 attacks

Hidden shift algorithms

Avoiding Simon's attacks [AR17]



Avoiding Simon's attacks [AR17]



Properties

- $f(x) = EM_+(x) - EM_+(x+1) \quad g(x) = P(x) - P(x+1)$
- $f(x) = g(x + k_1)$

Security

- No (known) polynomial algorithm
- Hidden shift algorithm!

Hidden shift problem

Hidden shift problem

- $f, g : \mathcal{G} \to X$ injective
- $s \in \mathcal{G}$
- $f(x) = g(x + s)$
- Goal : find s, given oracle access to f and g.

Classical resolution

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

Easy cases

- $\mathcal{G} = (\mathbb{Z}/(2))^n$: Simon's algorithm
- $f = g$: Shor's algorithm

Labeled qubits (in $\mathbb{Z}/(2^n)$)

- *Start with* $|0\rangle\,|0\rangle\,|0\rangle$

Labeled qubits (in $\mathbb{Z}/(2^n)$)

- Start with $|0\rangle |0\rangle |0\rangle$
- *Apply H* : $\sum_{b=0}^{1}, \sum_{x=0}^{2^n-1} |b\rangle |x\rangle |0\rangle$

Labeled qubits (in $\mathbb{Z}/(2^n)$)

- Start with $|0\rangle |0\rangle |0\rangle$
- Apply $H : \sum_{b=0}^{1}, \sum_{x=0}^{2^n-1} |b\rangle |x\rangle |0\rangle$
- *Apply the quantum oracles*

$$\sum_x |0\rangle |x\rangle |f(x)\rangle + |1\rangle |x\rangle |g(x)\rangle$$

Labeled qubits (in $\mathbb{Z}/(2^n)$)

- Start with $|0\rangle |0\rangle |0\rangle$
- Apply $H : \sum_{b=0}^{1}, \sum_{x=0}^{2^n-1} |b\rangle |x\rangle |0\rangle$
- Apply the quantum oracles

$$\sum_x |0\rangle |x\rangle |f(x)\rangle + |1\rangle |x\rangle |g(x)\rangle$$

- *Measure in the last register* $y = f(x_0) = g(x_0 + s)$

$$|0\rangle |x_0\rangle + |1\rangle |x_0 + s\rangle$$

Labeled qubits (in $\mathbb{Z}/(2^n)$)

- Start with $|0\rangle |0\rangle |0\rangle$
- Apply $H : \sum_{b=0}^{1}, \sum_{x=0}^{2^n-1} |b\rangle |x\rangle |0\rangle$
- Apply the quantum oracles

$$\sum_x |0\rangle |x\rangle |f(x)\rangle + |1\rangle |x\rangle |g(x)\rangle$$

- Measure in the last register $y = f(x_0) = g(x_0 + s)$

$$|0\rangle |x_0\rangle + |1\rangle |x_0 + s\rangle$$

- *Apply a quantum Fourier Transform*

$$\sum_\ell \exp\left(2i\pi \frac{x_0 \ell}{2^n}\right) |0\rangle |\ell\rangle + \exp\left(2i\pi \frac{(x_0 + s)\ell}{2^n}\right) |1\rangle |\ell\rangle$$

Labeled qubits (in $\mathbb{Z}/(2^n)$)

- Start with $|0\rangle|0\rangle|0\rangle$
- Apply H : $\sum_{b=0}^{1}, \sum_{x=0}^{2^n-1} |b\rangle|x\rangle|0\rangle$
- Apply the quantum oracles

$$\sum_x |0\rangle|x\rangle|f(x)\rangle + |1\rangle|x\rangle|g(x)\rangle$$

- Measure in the last register $y = f(x_0) = g(x_0 + s)$

$$|0\rangle|x_0\rangle + |1\rangle|x_0 + s\rangle$$

- Apply a quantum Fourier Transform

$$\sum_\ell \exp\left(2i\pi\frac{x_0\ell}{2^n}\right)|0\rangle|\ell\rangle + \exp\left(2i\pi\frac{(x_0+s)\ell}{2^n}\right)|1\rangle|\ell\rangle$$

- *Measure* ℓ

$$|\psi_\ell\rangle = |0\rangle + \exp\left(2i\pi\frac{s\ell}{2^n}\right)|1\rangle$$

Combining qubits

Labeled qubits

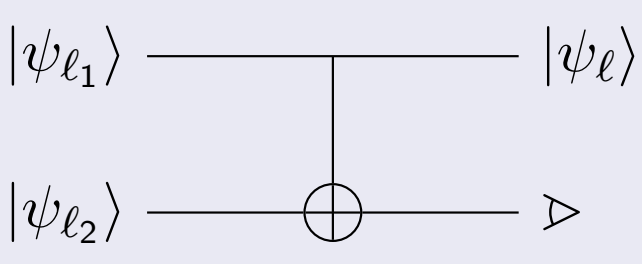
- $|\psi_\ell\rangle = |0\rangle + \exp\left(2i\pi\frac{s\ell}{2^n}\right)|1\rangle$
- $|\psi_{2^n-1}\rangle = |0\rangle + (-1)^s|1\rangle$

Combining qubits

Labeled qubits

- $|\psi_\ell\rangle = |0\rangle + \exp\left(2i\pi\frac{s\ell}{2^n}\right)|1\rangle$
- $|\psi_{2^n-1}\rangle = |0\rangle + (-1)^s|1\rangle$

Combination: CNOT [Kup05]

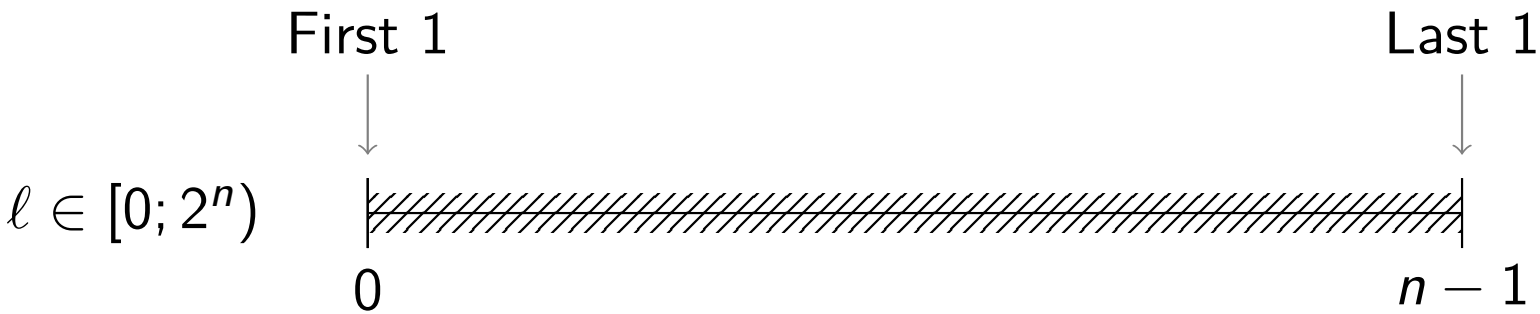


$$(\ell_1, \ell_2) \mapsto \ell_1 \pm \ell_2 \mod 2^n$$

Kuperberg's first algorithm [Kup05]

Look for partial collisions

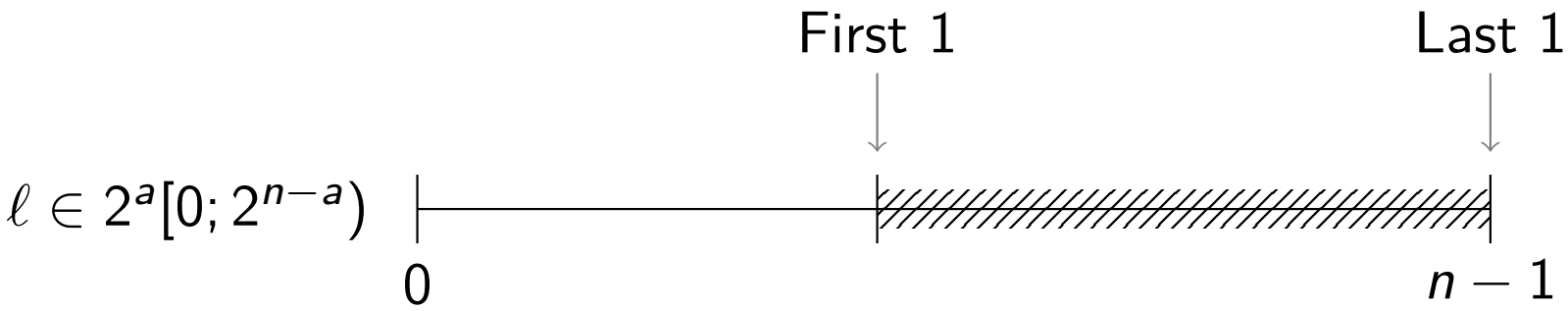
Find ℓ_1, ℓ_2: $\ell_1 \pm \ell_2 = 2^a \ell'$



Kuperberg's first algorithm [Kup05]

Look for partial collisions

Find ℓ_1, ℓ_2: $\ell_1 \pm \ell_2 = 2^a \ell'$



Kuperberg's first algorithm [Kup05]

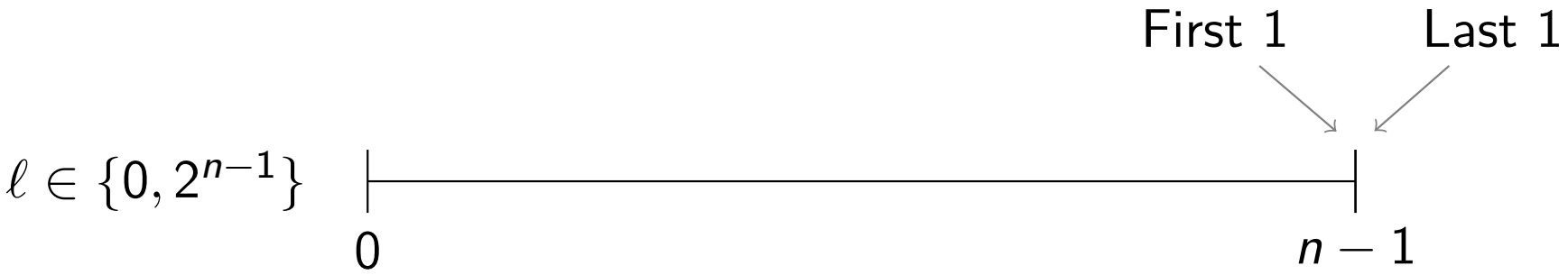
Look for partial collisions

Find ℓ_1, ℓ_2: $\ell_1 \pm \ell_2 = 2^a \ell'$

Kuperberg's first algorithm [Kup05]

Look for partial collisions

Find ℓ_1, ℓ_2: $\ell_1 \pm \ell_2 = 2^a \ell'$



Summary

Algorithms principles

- Produce random $|\psi_\ell\rangle = |0\rangle + \exp\left(2i\pi\frac{s\ell}{2^n}\right)|1\rangle$
- Combine them to converge to $|\psi_{2^{n-1}}\rangle = |0\rangle + (-1)^s|1\rangle$

Complexity

Asymptotic complexity $\widetilde{\mathcal{O}}\left(2^{\sqrt{2\log_2(3)n}}\right)$ quantum time and memory

New results

Algorithm Improvements

More efficient recovery of the other bits of s, gain $\mathcal{O}(\mathbf{n})$.

Simulations

Heuristic complexity of $0.7 \times 2^{1.8\sqrt{n}}$ quantum time and memory.

Simon-meets-Kuperberg

More efficient algorithm for $(\mathbb{Z}/(2^w))^p$

Regev's variant [Reg04]

Same principles, different combination method.

- *Start with* $|\psi_{\ell_1}\rangle, \ldots |\psi_{\ell_k}\rangle$

$$\bigotimes_j |\psi_{\ell_j}\rangle = \sum_{b_j \in \{0,1\}} \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle$$

Regev's variant [Reg04]

Same principles, different combination method.

- Start with $|\psi_{\ell_1}\rangle, \ldots |\psi_{\ell_k}\rangle$

$$\bigotimes_j |\psi_{\ell_j}\rangle = \sum_{b_j \in \{0,1\}} \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle$$

- *Compute* $\sum_j \ell_j b_j \mod 2^m$

$$\sum_{b_j \in \{0,1\}} \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle \left|\sum \ell_j b_j \mod 2^m\right\rangle$$

Regev's variant [Reg04]

Same principles, different combination method.

- Start with $|\psi_{\ell_1}\rangle, \ldots |\psi_{\ell_k}\rangle$

$$\bigotimes_j |\psi_{\ell_j}\rangle = \sum_{b_j \in \{0,1\}} \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle$$

- Compute $\sum_j \ell_j b_j \mod 2^m$

$$\sum_{b_j \in \{0,1\}} \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle \left|\sum \ell_j b_j \mod 2^m\right\rangle$$

- *Measure a value V*

$$\sum_{b_j \in \{0,1\}, \sum_j \ell_j b_j \mod 2^m = V} \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle$$

Regev's variant [Reg04]

- *Measure a value V*

$$\sum_{b_j \in \{0,1\}, \sum_j \ell_j b_j \mod 2^m = V} \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle$$

Regev's variant [Reg04]

- Measure a value V

$$\sum_{b_j \in \{0,1\}, \sum_j \ell_j b_j \mod 2^m = V} \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle$$

- *Find two solutions* $(b_1, \ldots, b_k), (b'_1, \ldots, b'_k)$ *of*

$$\sum_j \ell_j b_j \mod 2^m = V$$

Regev's variant [Reg04]

- Measure a value V

$$\sum_{b_j \in \{0,1\}, \sum_j \ell_j b_j \mod 2^m = V} \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle$$

- Find two solutions $(b_1, \ldots, b_k), (b'_1, \ldots, b'_k)$ of

$$\sum_j \ell_j b_j \mod 2^m = V$$

- *Project on them*

Regev's variant [Reg04]

- Measure a value V

$$\sum_{b_j \in \{0,1\}, \sum_j \ell_j b_j \mod 2^m = V} \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle$$

- Find two solutions $(b_1, \ldots, b_k), (b'_1, \ldots, b'_k)$ of

$$\sum_j \ell_j b_j \mod 2^m = V$$

- Project on them
- *Map* $(b_1, \ldots, b_k)$ *to* 0, $(b'_1, \ldots, b'_k)$ *to 1*

$$|0\rangle + \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b'_j - \sum \ell_j b_j\right)\right) |1\rangle$$

Regev's variant [Reg04]

- Measure a value V

$$\sum_{b_j \in \{0,1\}, \sum_j \ell_j b_j \mod 2^m = V} \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle$$

- Find two solutions $(b_1, \ldots, b_k), (b'_1, \ldots, b'_k)$ of

$$\sum_j \ell_j b_j \mod 2^m = V$$

- Project on them
- Map $(b_1, \ldots, b_k)$ to 0, $(b'_1, \ldots, b'_k)$ to 1

$$|0\rangle + \exp\left(2i\pi \frac{s}{2^n}\left(\sum \ell_j b'_j - \sum \ell_j b_j\right)\right) |1\rangle$$

- *New labeled qubit* $|\psi_\ell\rangle$, *with* $\ell = \sum \ell_j b'_j - \sum \ell_j b_j$, $2^m | \ell$

Pipeline of routines

Routine

- Input: $|\psi_{\ell_1}\rangle, \ldots |\psi_{\ell_k}\rangle$, m
- Output: $|\psi_\ell\rangle$, $2^m|\ell$
- Needs to compute the solutions of $\sum_j \ell_j b_j \mod 2^m = V$

Algorithm cost [CJS14]

- $\widetilde{\mathcal{O}}\left(2^{\sqrt{2\mathbf{n}\log_2(\mathbf{n})}}\right)$ quantum time and queries
- $\mathcal{O}(\mathbf{n})$ quantum memory

New tradeoffs

Algorithm cost

- Number of queries: increases with $\mathbf{n}/k$
- Classical cost: increases with k

$$L(\alpha, c) = 2^{(c+o(1))\mathbf{n}^{\alpha}\log(\mathbf{n})^{1-\alpha}}$$

Subset-sum algorithm of dimension k in 2^{bk}

Tradeoffs

- Minimal classical cost: $L(1/2, \sqrt{b})$
- Tradeoff classical $L(1-\alpha, *)$, quantum $L(\alpha, *)$ $(\alpha < 1/2)$
- Minimal quantum cost: $\mathcal{O}\left(\mathbf{n}^{\alpha+1}\right)$, with classical $\widetilde{\mathcal{O}}\left(2^{b\mathbf{n}/\alpha}\right)$ $(\alpha \geq 1)$

Summary

Hidden shift problem

$f, g : \mathcal{G} \to X$, $f(x) = g(x + s)$

Many tradeoffs

- Most time-efficient: $2^{\sqrt{2\mathbf{n}}}$ [Kup13]
- Can trade between classical and quantum time, classical memory, quantum queries
- Even a small quantum computer reduces the cost

Special cases

- $\mathcal{G} = (\mathbb{Z}/(2))^n$: Simon's algorithm
- Intermediate case $\mathcal{G} = (\mathbb{Z}/(2^w))^p$
- $f = g$: Shor's algorithm

Hidden shift Cryptanalysis

General idea

Create from a cipher two functions that fullfill $f(x) = g(x + s)$.

Applications

- Ciphers with modular addition instead of xors
- Polynomial MACs
- Offline variant possible

Numerical estimates for Even-Mansour-like constructions

- $\mathbf{n} = 128$, 16 bits of security
- $\mathbf{n} = 1600$, 56 bits of security

Key exchanges

Diffie-Hellman Key Exchange

Protocol

$$G, g \in G$$

(α) Alice

Bob (β)

Diffie-Hellman Key Exchange

Protocol

$$G, g \in G$$

$$(\alpha) \text{ Alice} \xrightarrow{[\alpha]g} \text{Bob } (\beta)$$

Diffie-Hellman Key Exchange

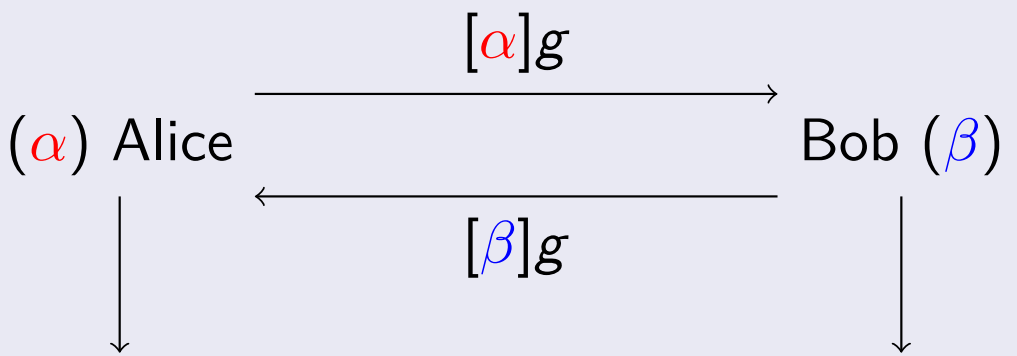
Protocol

$$G, g \in G$$

Diffie-Hellman Key Exchange

Protocol

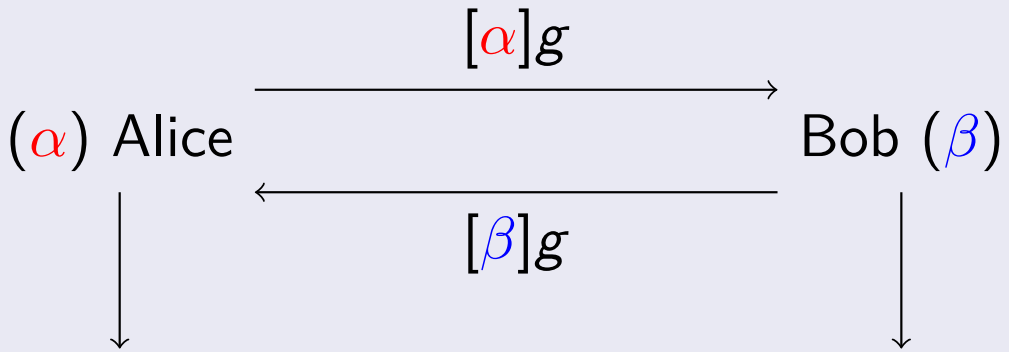
$$G, g \in G$$



Diffie-Hellman Key Exchange

Protocol

$$G, g \in G$$



Security

$$(g, [\alpha]g) \to \alpha$$

Quantum Security of Diffie-Hellman

Discrete log

$$(g, [\alpha]g) \to \alpha$$

Hidden subgroup

- $f(x, y) = [x]g + [y][\alpha]g$
- $[\alpha]g = [1]([\alpha]g)$
- $f(x, y) = f(x + \alpha, y - 1)$

Broken with Shor's algorithm.

Shor's algorithm [Sho94]

Shor's problem

- $f : \mathbb{Z} \to \mathbb{Z}$
- $r \in \mathbb{Z}$
- $\forall x, f(x) = f(x + r)$
- f hides the period r
- $r < 2^n$
- Goal : find r, given oracle access to f.

Shor's circuit ($f(x) = f(x + r)$)

Quantum circuit

- *Start from* $|0\rangle\,|0\rangle$

Measure biaised toward ℓ such that $\frac{r\ell}{2^n}$ is close to an integer.

Shor's circuit ($f(x) = f(x + r)$)

Quantum circuit

- Start from $|0\rangle |0\rangle$
- *Apply H* : $\sum_x |x\rangle |0\rangle$

Measure biaised toward ℓ such that $\frac{r\ell}{2^n}$ is close to an integer.

Shor's circuit ($f(x) = f(x + r)$)

Quantum circuit

- Start from $|0\rangle |0\rangle$
- Apply H : $\sum_x |x\rangle |0\rangle$
- *Apply the oracle* $\sum_x |x\rangle |f(x)\rangle$

Measure biaised toward ℓ such that $\frac{r\ell}{2^n}$ is close to an integer.

Shor's circuit ($f(x) = f(x + r)$)

Quantum circuit

- Start from $|0\rangle |0\rangle$
- Apply H : $\sum_x |x\rangle |0\rangle$
- Apply the oracle $\sum_x |x\rangle |f(x)\rangle$
- *Measure the second register* $\sum_{j=0}^{\alpha} |x_0 + jr\rangle$

Measure biaised toward ℓ such that $\frac{r\ell}{2^n}$ is close to an integer.

Shor's circuit ($f(x) = f(x+r)$)

Quantum circuit

- Start from $|0\rangle|0\rangle$
- Apply H : $\sum_x |x\rangle|0\rangle$
- Apply the oracle $\sum_x |x\rangle|f(x)\rangle$
- Measure the second register $\sum_{j=0}^{\alpha} |x_0 + jr\rangle$
- *Apply a QFT* $\sum_{j=0}^{\alpha} \sum_{\ell} \exp\left(2i\pi \frac{(x_0+jr)\ell}{2^n}\right) |\ell\rangle$

Measure biaised toward ℓ such that $\frac{r\ell}{2^n}$ is close to an integer.

Shor's circuit ($f(x) = f(x+r)$)

Quantum circuit

- Start from $|0\rangle|0\rangle$
- Apply H : $\sum_x |x\rangle|0\rangle$
- Apply the oracle $\sum_x |x\rangle|f(x)\rangle$
- Measure the second register $\sum_{j=0}^{\alpha} |x_0 + jr\rangle$
- Apply a QFT $\sum_{j=0}^{\alpha} \sum_\ell \exp\left(2i\pi \frac{(x_0+jr)\ell}{2^n}\right)|\ell\rangle$
- *Amplitude of ℓ :*

$$\sum_{j=0}^{\alpha} \exp\left(2i\pi \frac{jr\ell}{2^n}\right)$$

Measure biaised toward ℓ such that $\frac{r\ell}{2^n}$ is close to an integer.

Shor's algorithm

Shor's problem

Given f such that $f(x) = f(x + r)$, find r.

Summary

- Sample biaised values
- Recover r in $O(1)$ queries
- Generalizes to any abelian group

Applications

Breaks discrete log and factoring

A Dillema

The problem

- Commutation allows key exchange
- Commutative groups are too easy

A Dillema

The problem

- Commutation allows key exchange
- Commutative groups are too easy

Commutative group action

- G a commutative group
- X a set
- $* : G \times X \to X$
- $g * (g' * x) = (gg') * x$

Hard homogeneous space [Cou96]

- Given g, x, computing $g * x$ is easy
- Given x, $g * x$, computing g is hard

Group action Diffie-Hellman

Protocol

$$G, X, x \in X$$

(α) Alice

Bob (β)

Examples of schemes

- Ordinary isogeny-based schemes
- CSIDH
- But not SIKE

Group action Diffie-Hellman

Protocol

$$G, X, x \in X$$

$$(\alpha)\ \text{Alice} \xrightarrow{\alpha * x} \text{Bob}\ (\beta)$$

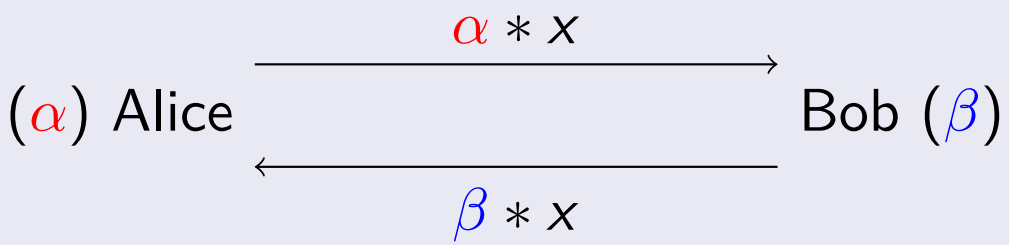
Examples of schemes

- Ordinary isogeny-based schemes
- CSIDH
- But not SIKE

Group action Diffie-Hellman

Protocol

$$G, X, x \in X$$



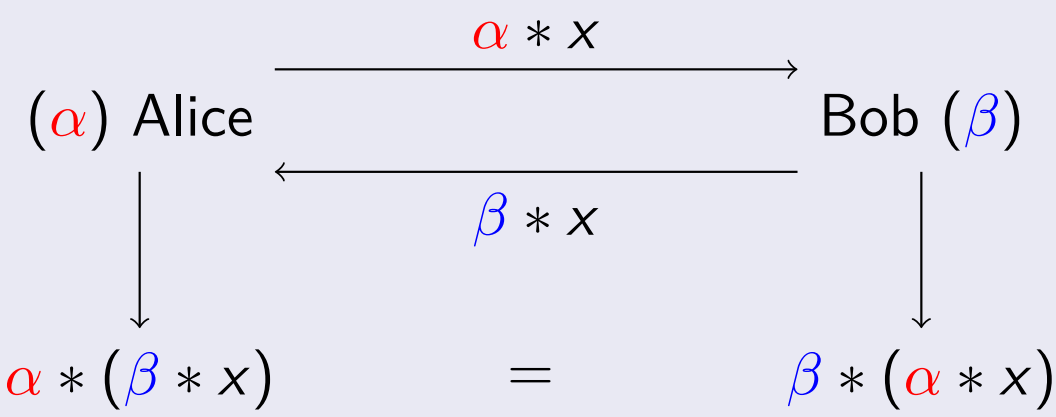
Examples of schemes

- Ordinary isogeny-based schemes
- CSIDH
- But not SIKE

Group action Diffie-Hellman

Protocol

$$G, X, x \in X$$



Examples of schemes

- Ordinary isogeny-based schemes
- CSIDH
- But not SIKE

Attacking the commutative group action

Inverting the group action

$$(x, \alpha * x) \to \alpha$$

A Hidden Shift!

- $f(z) = z * x$
- $g(z) = z * (\alpha * x)$
- $f(z + \alpha) = g(z)$

Numerical estimates for CSIDH

Claimed security

For $p \sim 512$ bits: $\simeq 2^{81}$ quantum operations, 2^{128} classical operations

A tradeoff

Attack in 2^{67} quantum operations, 2^{86} classical memory and operations

Safe instances

$p \sim 2000$ bits: 2^{80} quantum operations, 2^{78} classical operations, 2^{49} classical memory

Conclusion

New quantum algorithms

- The offline approach
- Improvements and tradeoffs for hidden shift algorithms

Many cryptanalyses

- Many instances of hidden structure scenarios
- Hidden structures are exploitable *even with classical queries*