Quantum Attacks on Symmetric Cryptography

Gregor Leander (joint work with Alex May)

MMC 2017

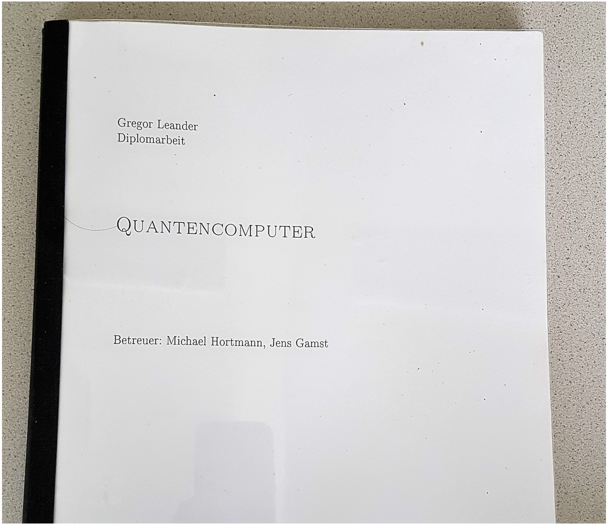
Horst Görtz Institute for IT-Security

Outline

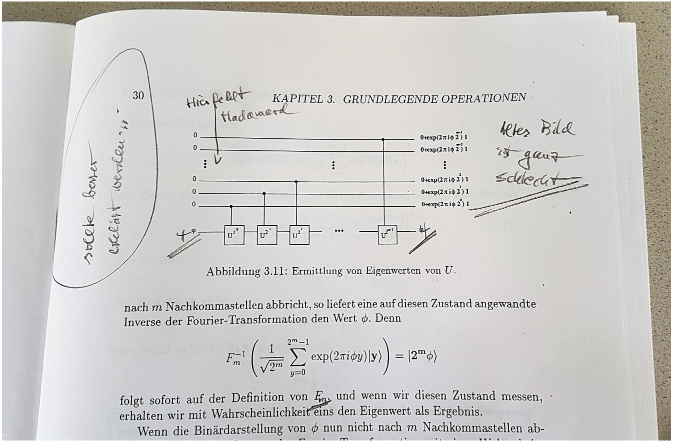
Main Message

- Quantum attacks on symmetric schemes understudied.
- Basic conclusion is: double the key-length.
- Two most popular generic ways of doing so:
  - Multiple-encryption
  - FX-construction
- Both not as good as you might think.
  - Multiple encryption: Kaplan 2014
  - FX construction: This talk

My Master Thesis (I/II)



My Master Thesis(II/II)



30 KAPITEL 3. GRUNDLEGENDE OPERATIONEN

Hier fehlt Hadamard

Altes Bild ist ganz schlecht

solche Boxen erklärt werden

Abbildung 3.11: Ermittlung von Eigenwerten von U.

nach m Nachkommastellen abbricht, so liefert eine auf diesen Zustand angewandte Inverse der Fourier-Transformation den Wert ϕ. Denn

$$F_m^{-1}\left(\frac{1}{\sqrt{2^m}}\sum_{y=0}^{2^m-1}\exp(2\pi i\phi y)|\mathbf{y}\rangle\right) = |\mathbf{2^m}\phi\rangle$$

folgt sofort auf der Definition von F_m, und wenn wir diesen Zustand messen, erhalten wir mit Wahrscheinlichkeit eins den Eigenwert als Ergebnis.

Wenn die Binärdarstellung von ϕ nun nicht nach m Nachkommastellen ab-

Horst Görtz Institute for IT-Security

Outline

From Bits to Qubits

One Qubit

The state x of one Qubit is a unit vector in $\mathbb{C}^2$.

Just notation:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Examples for states:

$$\begin{aligned} x_0 &= |0\rangle \approx 0 \\ x_1 &= |1\rangle \approx 1 \\ x_2 &= \alpha_0 |0\rangle + \alpha_1 |1\rangle \approx ? \end{aligned}$$

where

$$||\alpha_0||^2 + ||\alpha_1||^2 = 1$$

Two Qubits

Two Qubits

The state x of two Qubits is a unit vector in $\mathbb{C}^2 \otimes \mathbb{C}^2 \cong \mathbb{C}^4$.

(Not) just notation:

$$|0\rangle\,|0\rangle = |00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad |0\rangle\,|1\rangle = |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|1\rangle\,|0\rangle = |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle\,|1\rangle = |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Two Qubits

Two Qubits

The state x of two Qubits is a unit vector in $\mathbb{C}^2 \otimes \mathbb{C}^2 \cong \mathbb{C}^4$.

Examples for states:

$$
\begin{aligned}
x_0 &= |00\rangle \approx 00 \\
x_1 &= |10\rangle \approx 10 \\
x_2 &= \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \approx ?
\end{aligned}
$$

where

$$||\alpha_{00}||^2 + ||\alpha_{01}||^2 + ||\alpha_{10}||^2 + ||\alpha_{11}||^2 = 1$$

n Qubits

n Qubits

The state x of n Qubits is a unit vector in $\left(\mathbb{C}^2\right)^{\otimes n} \cong \mathbb{C}^{2^n}$.

Notation

For $x \in \mathbb{F}_2^n$ we denote

$$|x\rangle = |x_1, \ldots, x_n\rangle = |x_1\rangle \ldots |x_n\rangle = e_x$$

Examples:

$$\phi_1 = |x\rangle \approx x \quad \text{or} \quad \phi_2 = \sum_{x \in \mathbb{F}_2^n} \alpha_x |x\rangle \approx ?$$

where

$$\sum_{x \in \mathbb{F}_2^n} ||\alpha_x||^2 = 1$$

Computation: The principle

Given a quantum computer with n Qubits.

$$\phi = \sum_{x \in \mathbb{F}_2^n} \alpha_x \left| x \right\rangle$$

How do we conpute on that? How does the state change?

Computation: The principle

Given a quantum computer with n Qubits.

$$\phi = \sum_{x \in \mathbb{F}_2^n} \alpha_x \left| x \right\rangle$$

How do we conpute on that? How does the state change?

Computation $=$ Unitary Matrices

Any computation on a Quantum Computer corresponds to applying an unitary matrix.

Evolution of the state:

$$\phi \Rightarrow U\phi$$

As U is unitary:

$$||\phi||^2 = ||\mathcal{U}\phi||^2 = 1$$

Example: XOR

Two Qubit XOR:

XOR

Find U such that

$$|ab\rangle = |a\rangle\,|b\rangle \mapsto |a\rangle\,|a \oplus b\rangle$$

Example: XOR

Two Qubit XOR:

XOR

Find U such that

$$|ab\rangle = |a\rangle|b\rangle \mapsto |a\rangle|a \oplus b\rangle$$

On the basis we get:

$$U|00\rangle = |00\rangle \qquad U|01\rangle = |01\rangle$$
$$U|10\rangle = |11\rangle \qquad U|11\rangle = |10\rangle$$

Example: XOR

Two Qubit XOR:

XOR

Find U such that

$$|ab\rangle = |a\rangle\,|b\rangle \mapsto |a\rangle\,|a \oplus b\rangle$$

Example: XOR

Two Qubit XOR:

XOR

Find U such that

$$|ab\rangle = |a\rangle\,|b\rangle \mapsto |a\rangle\,|a \oplus b\rangle$$

A permutation matrix:

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

More general: Boolean Function

n Qubit Boolean Function:

$$f : \mathbb{F}_2^n \to \mathbb{F}_2$$

U_f on $(n+1)$ Qubits

Find U_f such that for all $a \in \mathbb{F}_2^n$ and $b \in \mathbb{F}_2$:

$$|ab\rangle = |a\rangle\,|b\rangle \mapsto |a\rangle\,|f(a) \oplus b\rangle$$

- U_f is quantum version of f
- Again a permutation matrix
- Efficient if f is efficient on classical computers.

Non classical: Conditional Flip

One Qubit, no classical equivalent:

Phase flipping

Consider U such that

$$|a\rangle \mapsto (-1)^a |a\rangle$$

$$U|0\rangle = |0\rangle \quad U|1\rangle = -|1\rangle$$

As a matrix:

$$U = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Last but not least: Hadamard

One one Qubit, again no classical equivalent:

Hadamard (ignoring scaling)

Consider U such that

$$|a\rangle \mapsto |0\rangle + (-1)^a |1\rangle$$

$$U|0\rangle = |0\rangle + |1\rangle \quad U|1\rangle = |0\rangle - |1\rangle$$

As a matrix:

$$U = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Last but not least: Hadamard

Generalization to n Qubits:

Hadamard on n Qubits

Consider $H^{\otimes n}$ such that

$$|a\rangle \mapsto \sum_x (-1)^{\langle a,x\rangle} |x\rangle$$

- $H^{\otimes n}$ is H applied to each Qubit.
- Thus, it is efficient if H is.
- Special case:

$$H^{\otimes n} |0\rangle = \sum_{x \in \mathbb{F}_2^n} |x\rangle$$

All Executions at Once

A small example

Putting things together: First H, then U_f.

$$\begin{aligned} |0\rangle\,|0\rangle &\mapsto \sum_{x\in\mathbb{F}_2^n} |x\rangle\,|0\rangle \\ &\mapsto \sum_{x\in\mathbb{F}_2^n} |x\rangle\,|f(x)\rangle \end{aligned}$$

We evaluated a function on all inputs at once!

Invisible

We cannot classicaly use the result w/o measuring.

Measurement

Make it classical

In order to use the output of a QC classically, we have to measure the state.

Consider an n-Qubit state:

$$\phi = \sum_{x \in \mathbb{F}_2^n} \alpha_x |x\rangle$$

Measurement

The measurement $M(\phi)$ of ϕ results in x with probability $||\alpha_x||^2$.

Measurement

Example on two Qubits

$$x = \frac{1}{\sqrt{2}}\left|00\right\rangle - \frac{1}{\sqrt{2}}\left|11\right\rangle$$

$$\begin{array}{lll} M(\phi) = 00 & \text{with probability} & 1/2 \\ M(\phi) = 11 & \text{with probability} & 1/2 \\ M(\phi) = 10 & \text{with probability} & 0 \\ M(\phi) = 00 & \text{with probability} & 0 \end{array}$$

Task of Quantum Computing

Make the correct/interessting result appear with overwhelming probability.

Outline

The Setting

Generic Search Problem

Given $f : \mathbb{F}_2^n \to \mathbb{F}_2$ such that

$$f(x) = \begin{cases} 1 & \text{if} \quad x = x_0 \\ 0 & \text{if} \quad x \neq x_0 \end{cases}$$

find x_0.

Classically: We need $\mathcal{O}(2^n)$ evaluations of f.

Grover's Solution

On a quantum computer, we get away with running time $\mathcal{O}(2^{n/2})$!

The Components

Hadamard $H^{\otimes n}$

$$|a\rangle \mapsto \sum_x (-1)^{\langle a,x\rangle} |x\rangle$$

U_f as phase flipping

$$|x\rangle \mapsto (-1)^{f(x)} |x\rangle$$

Missing piece: Reflection across the mean of α_x.

Reflection Across the Mean

Unitary Reflection Map

We consider the mapping

$$R = 2P - I$$

where

$$P = \left(\frac{1}{2^n}\right)_{i,j \in \{1..2^n\}}$$

Applied to $\phi = \sum_x \alpha_x |x\rangle$ we get

$$(R\phi)_j = (P - (I - P)\phi)_j = \overline{\alpha} - (\alpha_j - \overline{\alpha})$$

where

$$\overline{\alpha} = \frac{1}{2^n} \sum_x \alpha_x$$

Not discussed here: R is efficient if H is.

Grover's Algorithm

Grover's Algorithm

1. Start with $|0\rangle$
2. Apply $H^{\otimes n}$
3. Repeat t times
   1. Apply U_f as phase flipping
   2. Apply reflection R
4. Measure the state.

If $t \approx 2^{n/2}$ then result is x_0 with high probability.

Proof

No. But pictures.

Example of Grover's Algo

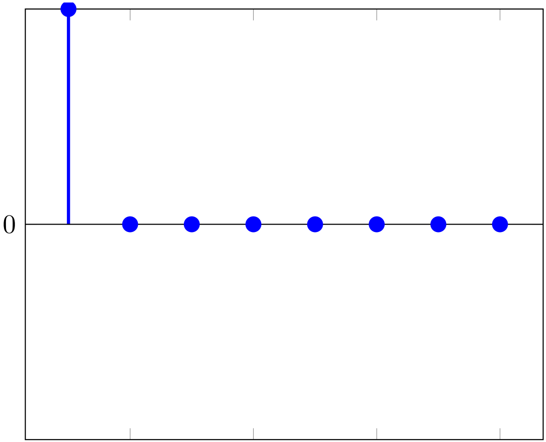
With 3 Qubits

$$f : \mathbb{F}_2^3 \rightarrow \mathbb{F}_2$$

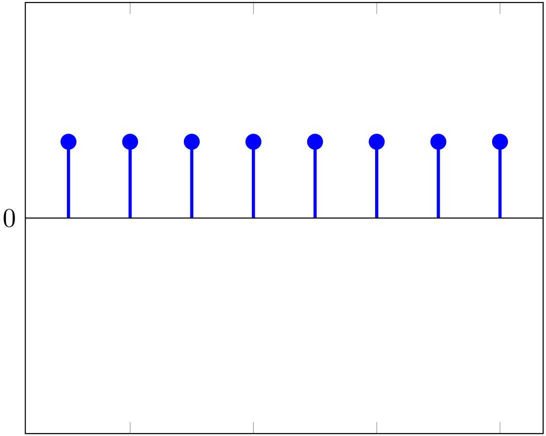
where

$$f(x) = 1 \Leftrightarrow x = 3$$

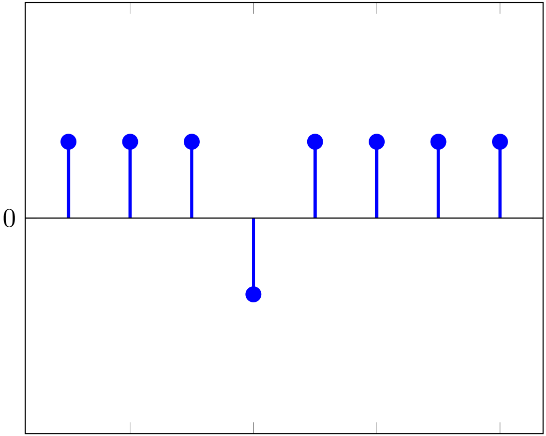
Example of Grover's Algo



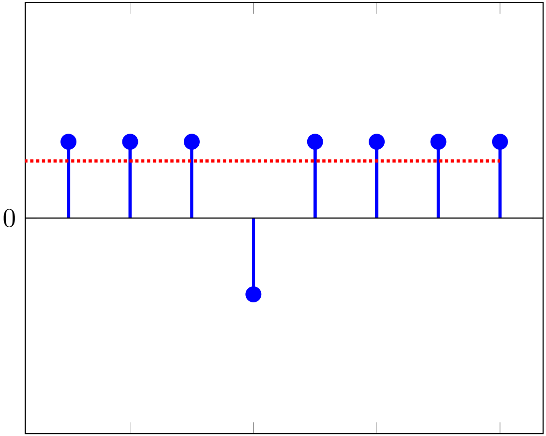
Example of Grover's Algo



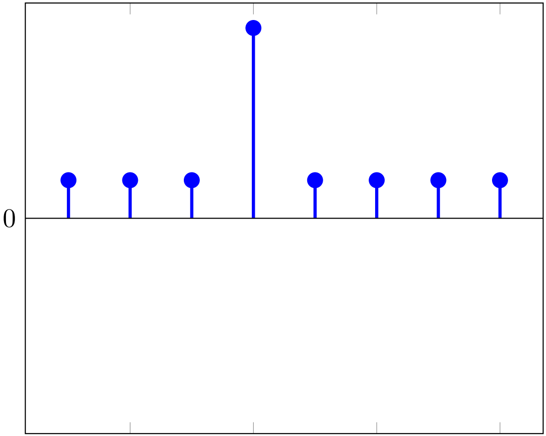
Example of Grover's Algo



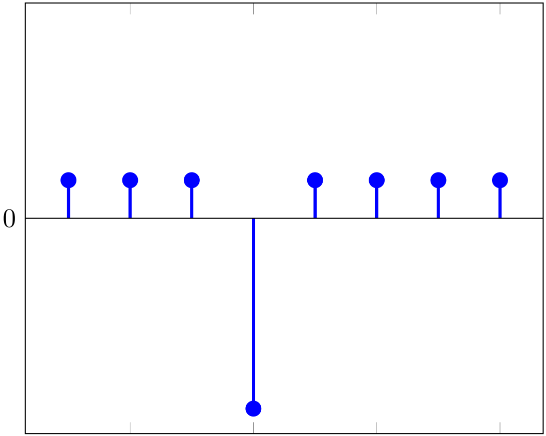
Example of Grover's Algo



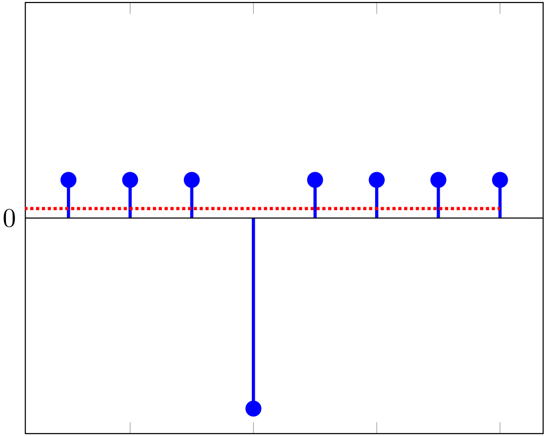
Example of Grover's Algo



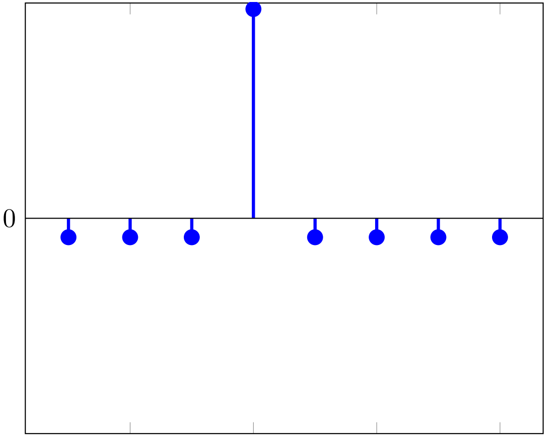
Example of Grover's Algo



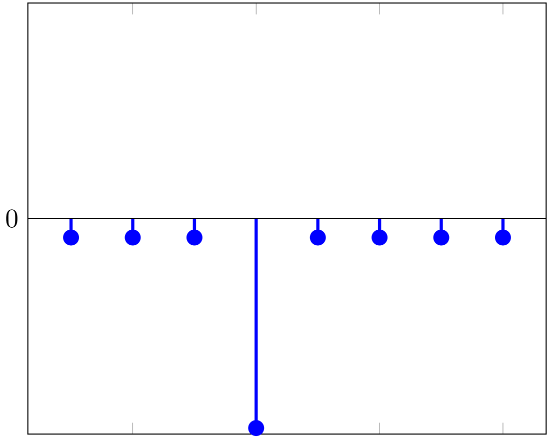
Example of Grover's Algo



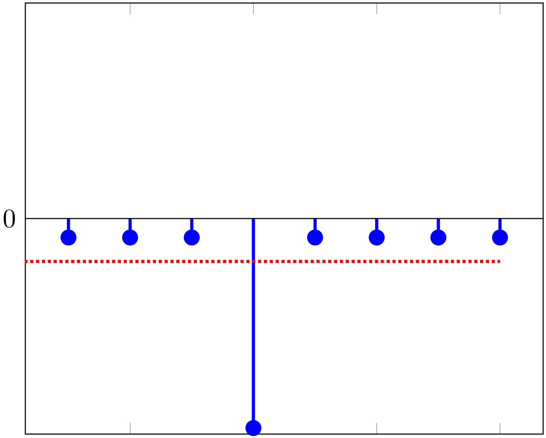
Example of Grover's Algo



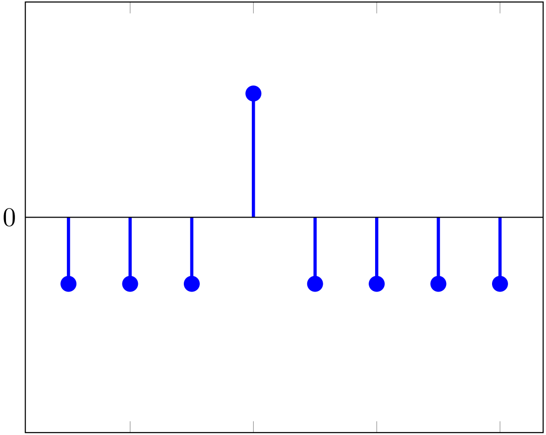
Example of Grover's Algo



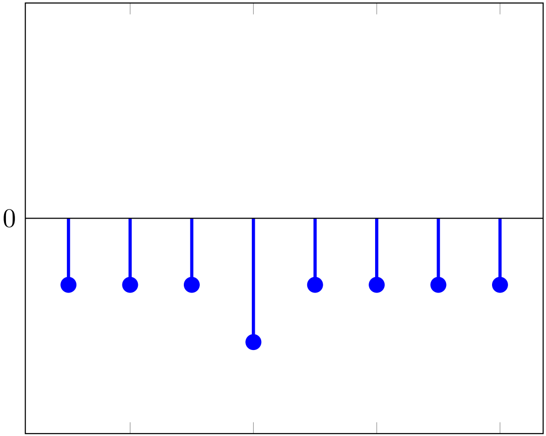
Example of Grover's Algo



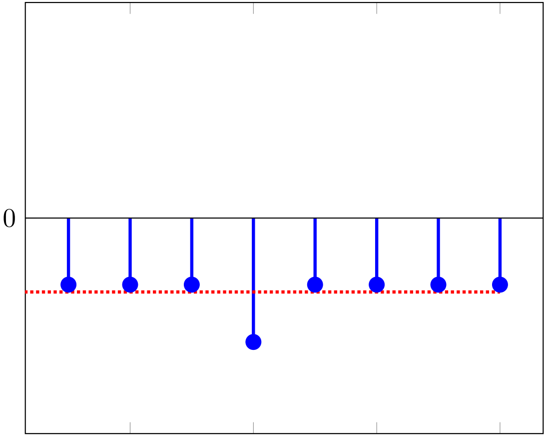
Example of Grover's Algo



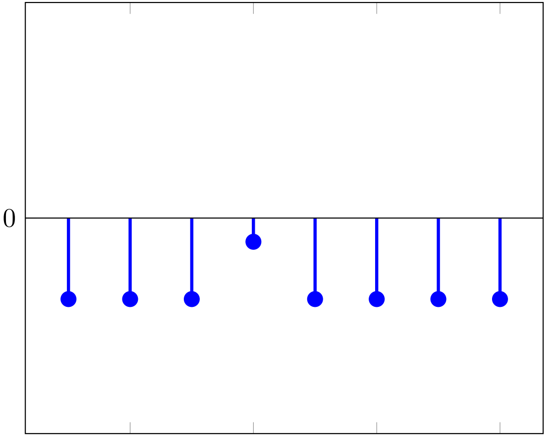
Example of Grover's Algo



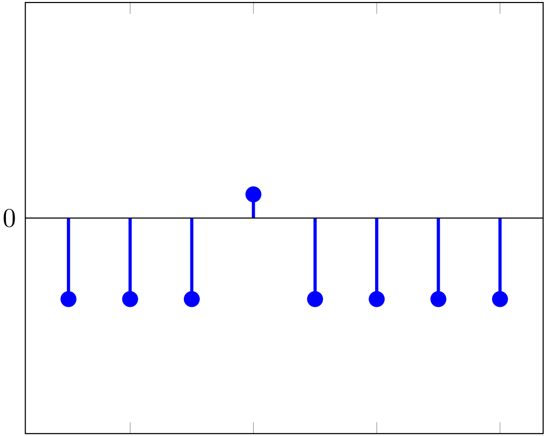
Example of Grover's Algo



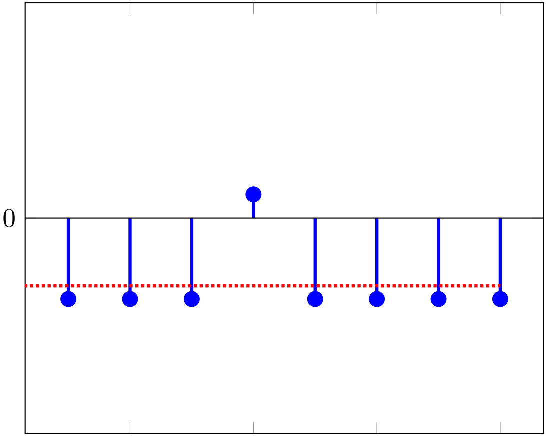
Example of Grover's Algo



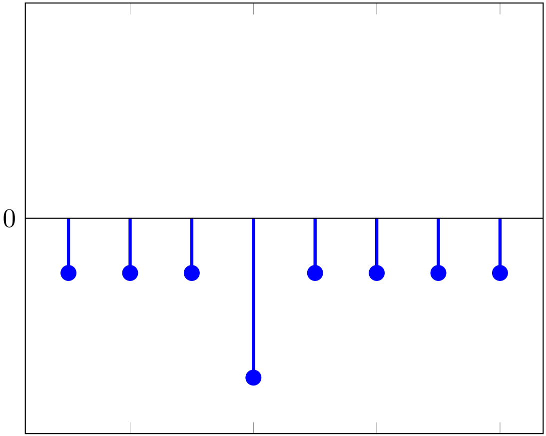
Example of Grover's Algo



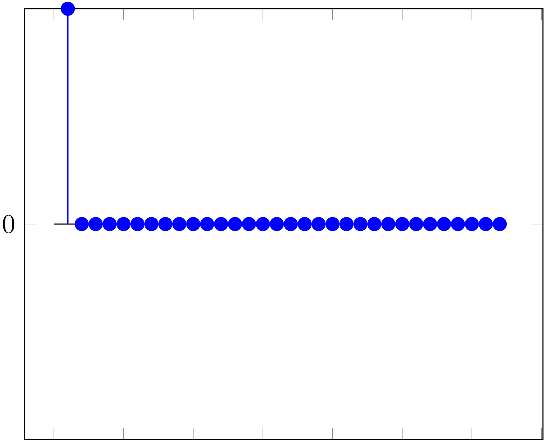
Example of Grover's Algo



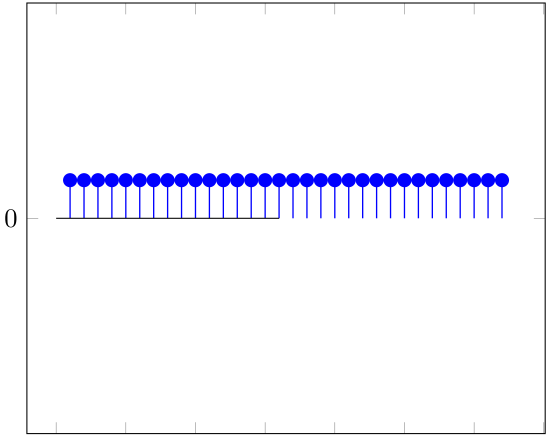
Example of Grover's Algo



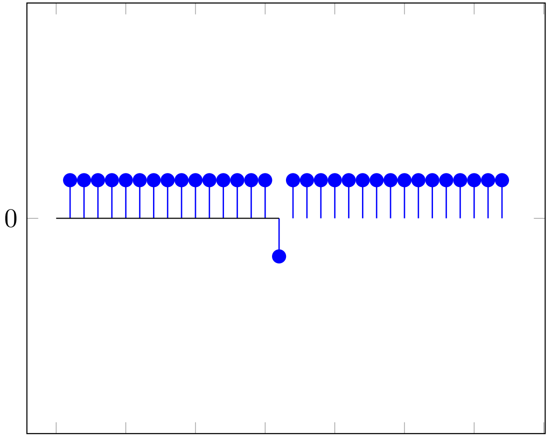
Example of Grover's Algo



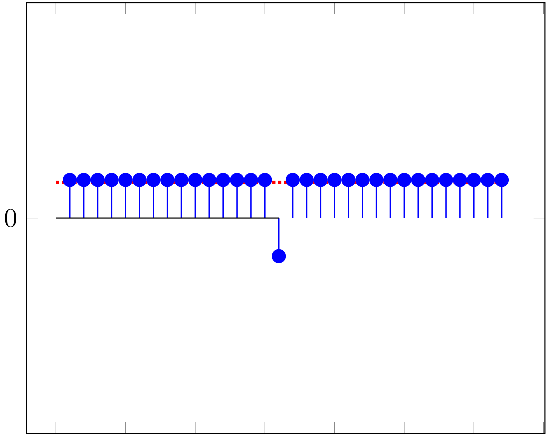
Example of Grover's Algo



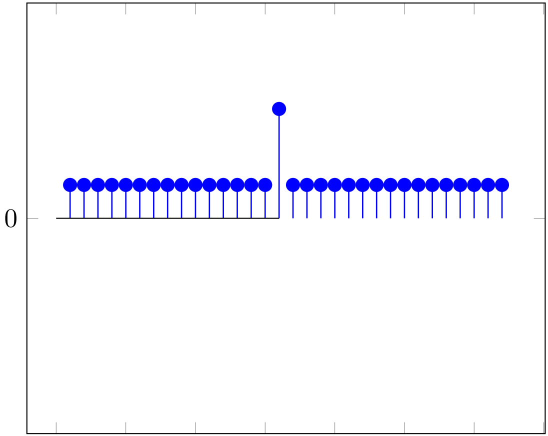
Example of Grover's Algo



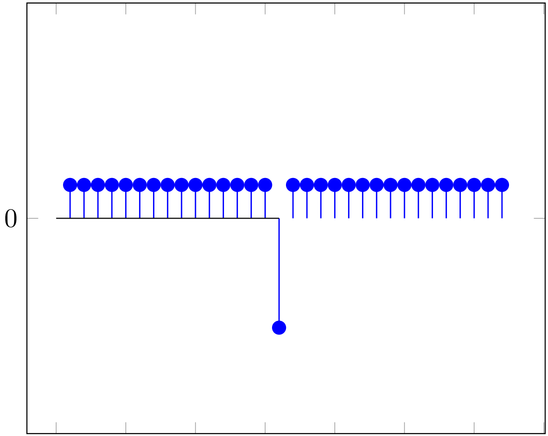
Example of Grover's Algo



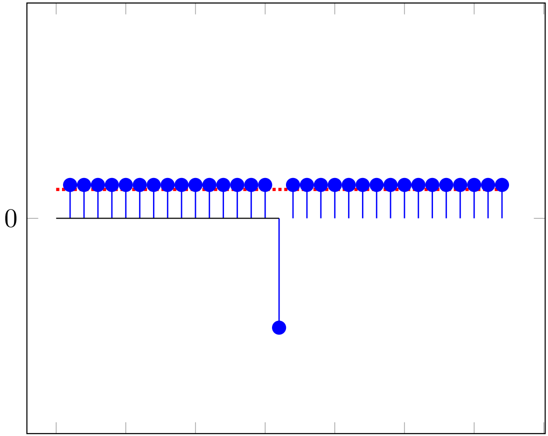
Example of Grover's Algo



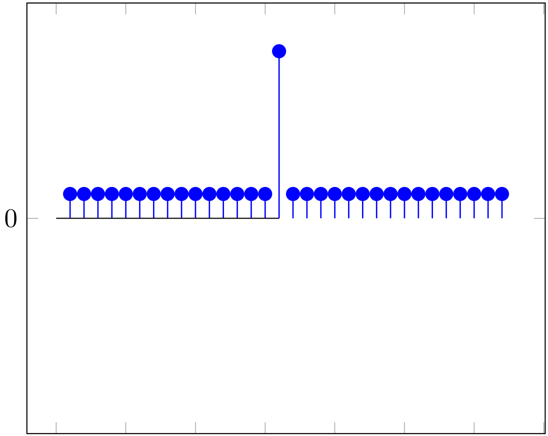
Example of Grover's Algo



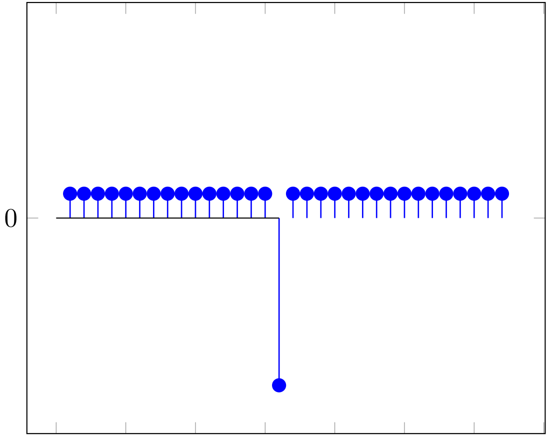
Example of Grover's Algo



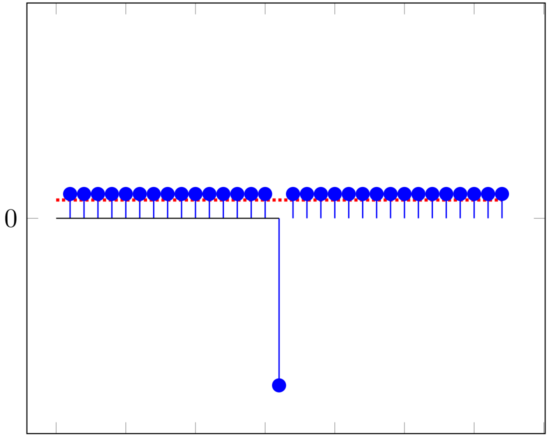
Example of Grover's Algo



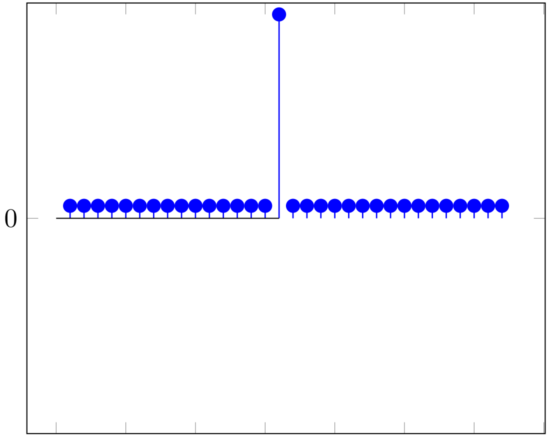
Example of Grover's Algo



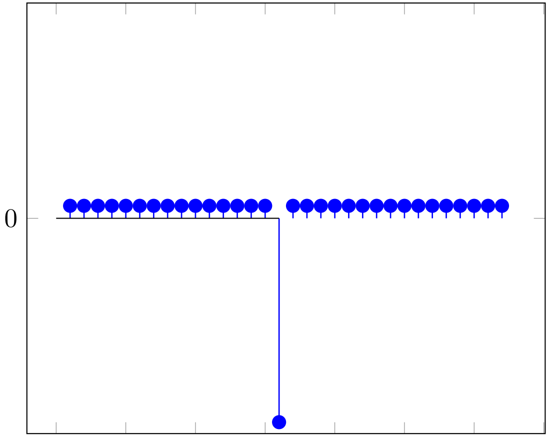
Example of Grover's Algo



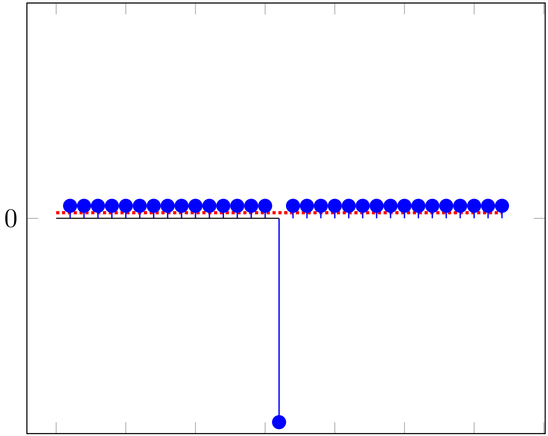
Example of Grover's Algo



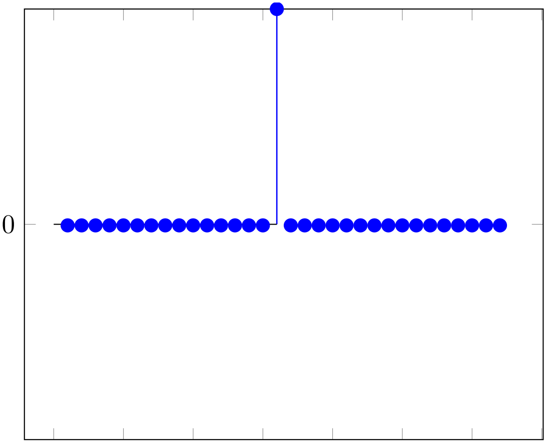
Example of Grover's Algo



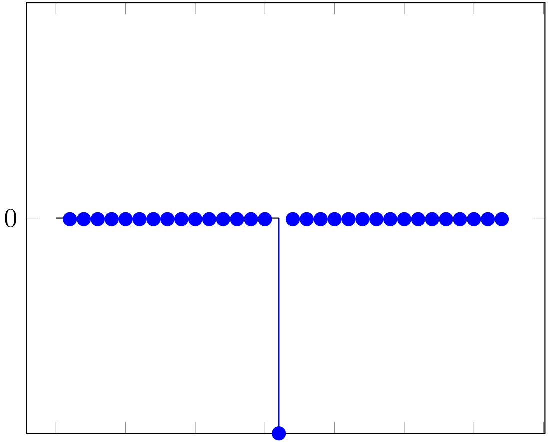
Example of Grover's Algo



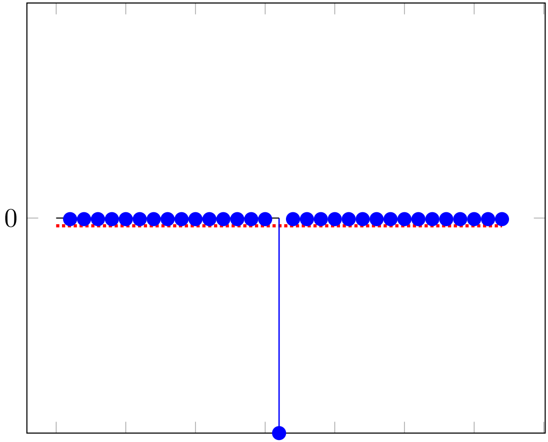
Example of Grover's Algo



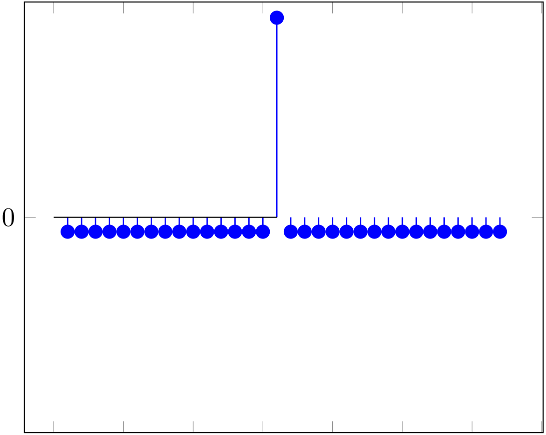
Example of Grover's Algo



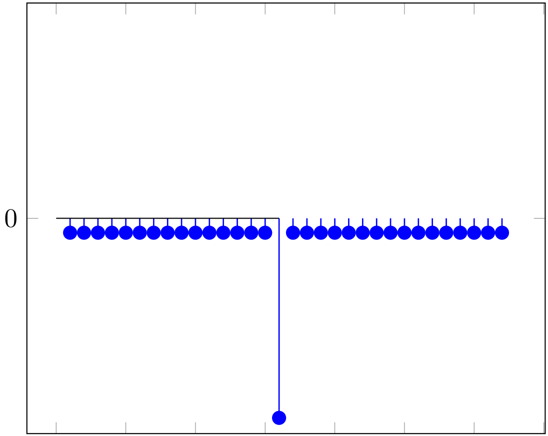
Example of Grover's Algo



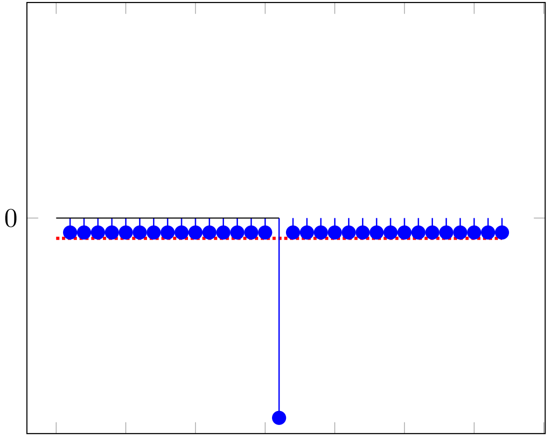
Example of Grover's Algo



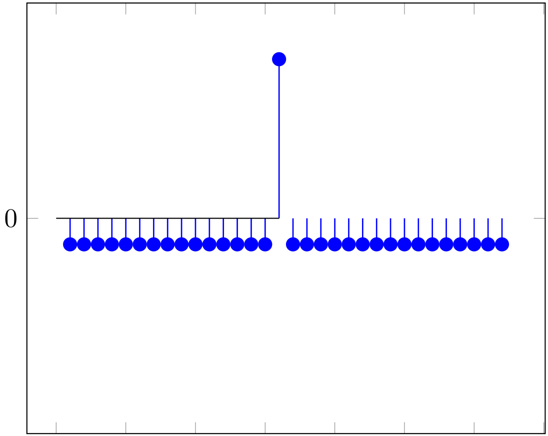
Example of Grover's Algo



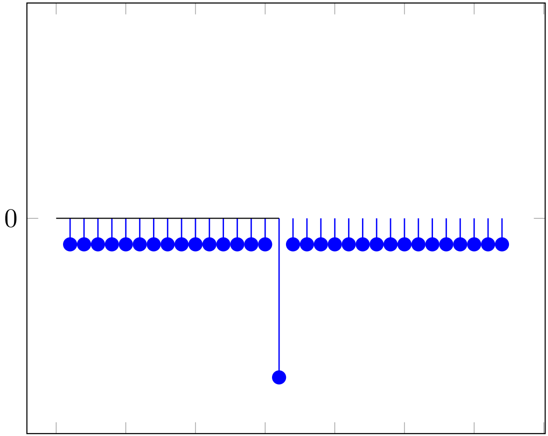
Example of Grover's Algo



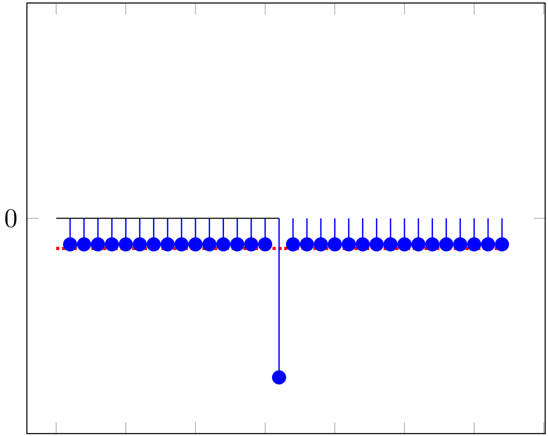
Example of Grover's Algo



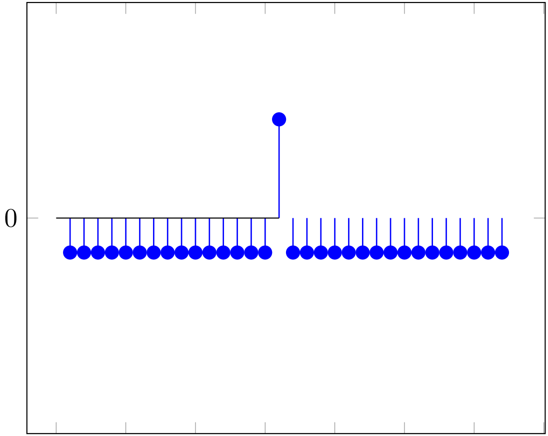
Example of Grover's Algo



Example of Grover's Algo



Example of Grover's Algo



Generalization of Grover: Amplitude Amplification

Brassard, Høyer ('97) generalized the idea:
Given

- A classically efficient function that decides if a state is good or bad
- A quantum algorithm that results in a good state with probability p.

$\mathcal{O}(p^{-1/2})$ iterations of generalized Grover will result in a good state with large probability.

Outline

Quantum Attacks on Symmetric Crypto

Basically two attacks known:

Simon's Algorithm

Used to e.g. break Even-Mansour

Grover's Algorithm

Used to speed-up brute force

Grover's Algorithm to break block ciphers

Generic block cipher

$$\text{Enc}(m) = E_k(m)$$

$$m \longrightarrow \boxed{E_k} \longrightarrow c$$

Conversion into Grover's problem (given a message/cipher-text pair):

$$f(x) = \begin{cases} 1 & \text{if } E_x(m) = c \\ 0 & \text{else} \end{cases}$$

The Attack

Apply Grover's Algorithm to f. Recover k in time $\mathcal{O}(2^{n/2})$.

Simon's Algorithm

Simon's Algorithm

Given $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ such that $\exists s$

$$F(x) = F(x + s) \quad \forall x$$

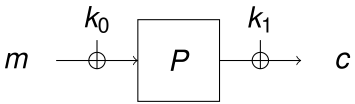
than one can recover s in linear time.

- Originally: $F(x) = F(y) \Leftrightarrow y = x + s$
- Used by Kuwakado and Morii to break Even-Mansour
- Extended to many modes in [KLLNP]

Simon's Algorithm to break EM

The Even-Mansour scheme:

$$\text{Enc}(m) = E(m + k_0) + k_1$$



Conversion into Simon's problem:

$$F(x) = \text{Enc}(x) + P(x)$$

Then

$$F(x) = F(x + k_0)$$

The Attack (with quantum queries)

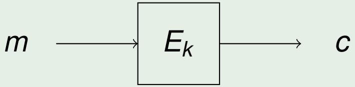
Apply Simon's algorithm to F. Recover k_0 in linear time.

Outline

Combine?

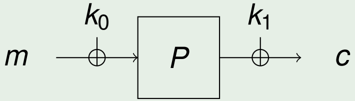
We can break:

Generic Ciphers



Time: $\mathcal{O}(2^{n/2})$

Even-Mansour

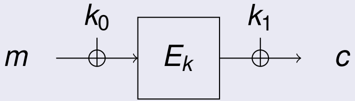


Time: $\mathcal{O}(n)$

What about combining this?

The FX-Construction

FX-Construction



Question

How to attack the FX construction in a quantum setting?

Attacking the FX construction

Question

How to attack the FX construction in a quantum setting?

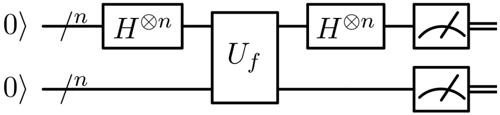
This is actually a question about:

Combining Simon and Grover

How to combing Simon's and Grover's algorithm?

Let's have a closer look.

Inside Simon's Algorithm



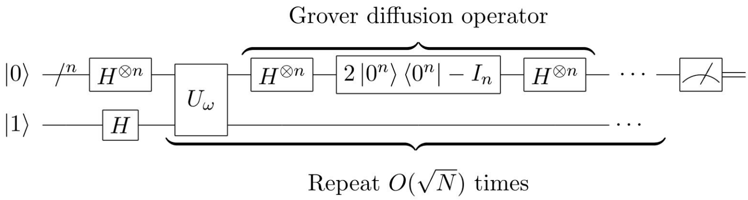
Key-features:

- Requires to implement Enc$(x) + P(x)$ as unitary embedding.
- Running once and measuring results in x s.t.

$$\langle k_0, x \rangle = 0$$

- Running $n + \epsilon$ times results in k_0 by solving linear equations

Inside Grover's Algorithm (Amplitude Amplification)



Key-features:

- Requires a quantum algorithm $\mathcal{A}$ with initial success probability p.
- Requires phase-flipping for good states
- Running $p^{-1/2}$ times results in a good state with high prob.

Combining: Avoid Measurements

Approach: Use Simon's algo for $\mathcal{A}$

Problem

Measuring not allowed in $\mathcal{A}$ for Grover. Simon's algo requires measuring.

Combining: Avoid Measurements

Approach: Use Simon's algo for $\mathcal{A}$

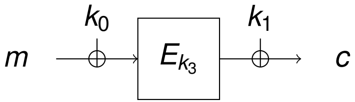
Problem

Measuring not allowed in $\mathcal{A}$ for Grover. Simon's algo requires measuring.

Sketch of the solution:

- Run $n+\epsilon$ Simons in parallel
- Linear algebra to compute candidate for k_0
- Check against message/cipher-text pairs
- If that fits: flip the phase

Parallel Simon: A bit more details



Running Simon's Algorithm in parallel results in states

$$\begin{aligned}\phi &= \sum_{k_3', x=(x_1,\ldots,x_s)} \alpha_{k_3',x} \left|k\right\rangle \left|x\right\rangle \\ &= \sum_{k_3', x=(x_1,\ldots,x_s)} \alpha_{k_3',x} \left|k\right\rangle \left|x_1,\ldots,x_s\right\rangle\end{aligned}$$

such that

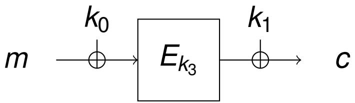
$$\alpha_{x,k_3} \neq 0 \Rightarrow \langle x_i, k_0 \rangle = 0$$

for all i.

Question

How do we continue without measuring?

Parallel Simon: A bit more details



$$\phi = \sum_{k_3', x=(x_1,\ldots,x_s)} \alpha_{k_3',x} |k\rangle |x\rangle$$

such that

$$\alpha_{k_3,x} \neq 0 \Rightarrow \langle x_i, k_0 \rangle = 0$$

for all i. We have to identify good states.

Good States

States where $k_3' = k_3$.

Parallel Simon: A bit more details

Good States

States where $k_3' = k_3$.

Given $|k\rangle\,|x_1,\ldots,x_s\rangle$ we compute

$$U = \langle x_1,\ldots,x_s\rangle^{\perp}$$

- If $\dim U = n$ state is bad.
- If $\dim U < n-1$ state is bad.

Otherwise:

We found our candidate key

$$U = \langle k_0' \rangle$$

Parallel Simon: A bit more details

We found our candidate key

$$U = \langle k_0' \rangle$$

Here:

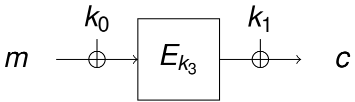
- Check if k_3', k_0' matches with known cipher-text/plain-text pairs
- YES: state is good.
- NO: state is bad.

Efficient

Classification of states is efficient.

Remains: Check that error probability is small.

Result



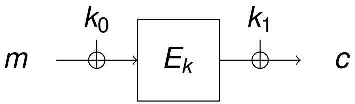
Result

The FX construction can be broken in time $\mathcal{O}(2^{n/2})$. Quantum computer gets n times bigger.

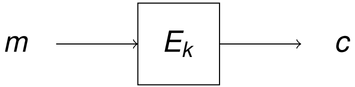
Outline

Conclusion

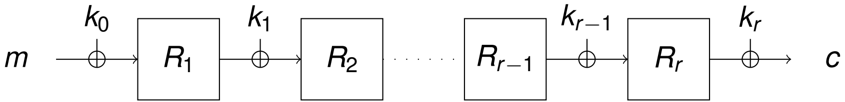
In a quantum world



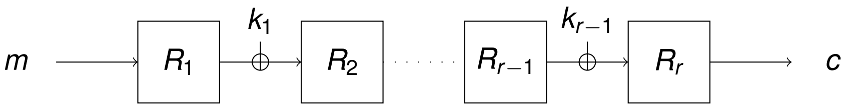
is as secure (linear overhead) as



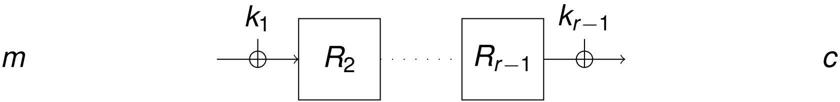
Key-Alternating Ciphers



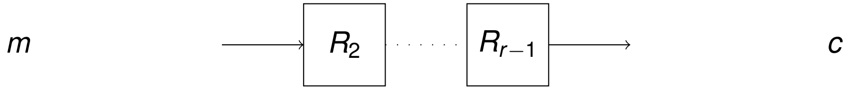
Key-Alternating Ciphers



Key-Alternating Ciphers



Key-Alternating Ciphers



Key-Alternating Ciphers

m c

Polynomial attack on key-alternating ciphers

Key-Alternating Ciphers

m c

Polynomial attack on key-alternating ciphers does not work like that

Future Work

Possible future topics:

- Correct attacks on key-alternating ciphers
- Other applications of Simon/Grover combination

Thank you.