Breaking Symmetric Cryptosystems Using Quantum Algorithms

Gaëtan Leurent

Joined work with:

Marc Kaplan Anthony Leverrier María Naya-Plasencia

Inria, France

FOQUS Workshop

Motivation

What would be the impact of quantum computers on symmetric cryptography?

- Some physicists think they can build quantum computers
- NSA thinks we need quantum-resistant crypto (or do they?)

Motivation

What would be the impact of quantum computers on symmetric cryptography?

- Some physicists think they can build quantum computers
- NSA thinks we need quantum-resistant crypto (or do they?)

Expected impact of quantum computers

- Some problems can be solved much faster with quantum computers
  - Up to exponential gains
  - But we don't expect to solve all NP problems

Impact on public-key cryptography

- RSA, DH, ECC broken by Shor's algorithm
  - Breaks factoring and discrete log in polynomial time
  - Large effort to develop quantum-resistant algorithms (*e.g.* NIST)

Impact on symmetric cryptography

- Exhaustive search of a k-bit key in time $2^{k/2}$ with Grover's algorithm
  - Common recommendation: double the key length (AES-256)
  - Is there more?

Expected impact of quantum computers

- Some problems can be solved much faster with quantum computers
  - Up to exponential gains
  - But we don't expect to solve all NP problems

Impact on public-key cryptography

- RSA, DH, ECC broken by Shor's algorithm
  - Breaks factoring and discrete log in polynomial time
  - Large effort to develop quantum-resistant algorithms (*e.g.* NIST)

Impact on symmetric cryptography

- Exhaustive search of a k-bit key in time $2^{k/2}$ with Grover's algorithm
  - Common recommendation: double the key length (AES-256)
  - Is there more?

Expected impact of quantum computers

- Some problems can be solved much faster with quantum computers
  - Up to exponential gains
  - But we don't expect to solve all NP problems

Impact on public-key cryptography

- RSA, DH, ECC broken by Shor's algorithm
  - Breaks factoring and discrete log in polynomial time
  - Large effort to develop quantum-resistant algorithms (*e.g.* NIST)

Impact on symmetric cryptography

- Exhaustive search of a k-bit key in time $2^{k/2}$ with Grover's algorithm
  - Common recommendation: double the key length (AES-256)
  - Is there more?

Security of symmetric cryptography

Classical approach

- Security of the protocol
  - Security proof assuming security of cryptographic operations
- Security of the modes (HMAC, CBC, ...)
  - Security proofs (assuming security of the primitive)
- Security of the primitives (AES, SHA-1, RSA, ...)
  - Studied with cryptanalysis

In the quantum setting

1. Study quantum cryptanalysis
2. Study modes of operations
   - Proofs in the quantum setting
   - Attacks in the quantum setting

Overview of the talk

Is AES secure in a quantum setting?

- Study classical cryptanalysis techniques in the quantum setting
  - Do we get a quadratic speedup?
  - Do we need a quantum encryption oracle?
  - How are different cryptanalysis techniques affected?

Quantum Differential and Linear Cryptanalysis
Kaplan, G. L., Leverrier, Naya-Plasencia [FSE '17 + ToSC]

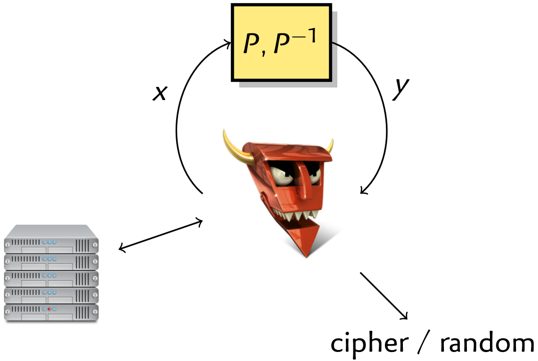
Are classical modes secure in the quantum setting?

- Encryption modes are secure [Unruh & al, PQC'16]
- Authentication modes broken by superposition queries

Breaking Symmetric Cryptosystems using Quantum Period Finding
Kaplan, G. L., Leverrier, Naya-Plasencia [CRYPTO '16]

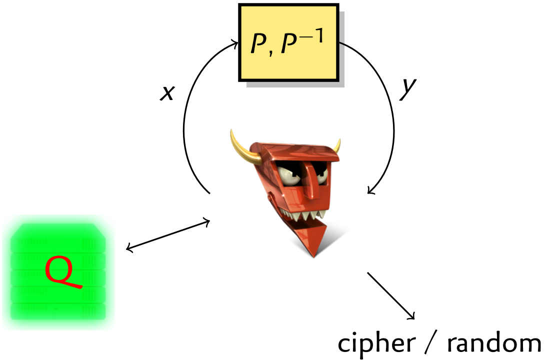
Security notions: Classical

- PRF security: given access to P/P^{-1}, distinguishing E from random
- Classical setting: classical computations
- Classical security: classical queries
- Cipher broken by adversary with
  - data $\ll 2^n$
  - time $\ll 2^k$
  - success $> 3/4$



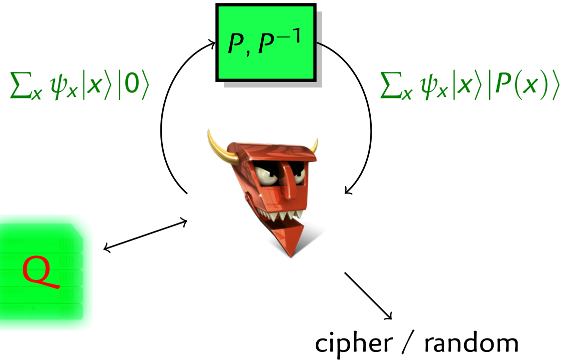
Security notions: Quantum Q1

- PRF security: given access to P/P^{-1}, distinguishing E from random
- Quantum setting: quantum computations
- Classical security: classical queries
- Cipher broken by adversary with
  - data $\ll 2^n$
  - time $\ll 2^{k/2}$
  - success $> 3/4$



Security notions: Quantum Q2

- PRF security: given access to P/P^{-1}, distinguishing E from random
- Quantum setting: quantum computations
- Quantum security: quantum (superposition) queries
- Cipher broken by adversary with
  - data $\ll 2^n$
  - time $\ll 2^{k/2}$
  - success $> 3/4$



About the models

Q1 model: classical queries

- Build a quantum circuit from classical values
- Example: breaking RSA with Shor's algorithm

Q2 model: superposition queries

- Access quantum circuit implementing the primitive with a secret key
- Example: breaking CBC-MAC with Simon's algorithm

- The Q2 model is very strong for the adversary
  - Simple and clean generalisation of classical oracle
  - Aim for security in the strongest (non-trivial) model
  - A Q2-secure block cipher is useful for security proofs of modes

Outline

Introduction

Grover's Algorithm

Quantum Differential Cryptanalysis
- Differential
- Truncated differential

Simon's Algorithm

Breaking Modes of Operation
- Forgery attack against CBC-MAC
- Other modes of operations

Slide attacks

Outline

Grover's algorithm

- Search for a marked element in a set X
- Set of marked elements M, with $|M| \geq \varepsilon \cdot |X|$

Classical algorithm

1: **loop**
2: $x \leftarrow \textsc{Setup}()$ ▷ Pick a random element in X, cost S
3: **if** $\textsc{Check}(x)$ **then** ▷ Check if it is marked, cost C
4: **return** x

- $1/\varepsilon$ repetitions expected
- Complexity $(S + C)/\varepsilon$

Grover's algorithm

- Search for a marked element in a set X
- Set of marked elements M, with $|M| \geq \varepsilon \cdot |X|$

Grover Algorithm (as a quantum walk)

Quantum algorithm to find a marked element using:

- SETUP: builds a uniform superposition of inputs in X
- CHECK: applies a control-phase gate to the marked elements

- Only $1/\sqrt{\varepsilon}$ repetitions needed
- Complexity $(S+C)/\sqrt{\varepsilon}$
- Can produce a uniform superposition of M
- Can provide an oracle without measuring (nesting)
- Variant to measure ε (quantum counting)

Grover's algorithm

- Search for a marked element in a set X
- Set of marked elements M, with $|M| \geq \varepsilon \cdot |X|$

Grover Algorithm (as a quantum walk)

Quantum algorithm to find a marked element using:

- SETUP: builds a uniform superposition of inputs in X
- CHECK: applies a control-phase gate to the marked elements

- Only $1/\sqrt{\varepsilon}$ repetitions needed
- Complexity $(S + C)/\sqrt{\varepsilon}$

- Can produce a uniform superposition of M
- Can provide an oracle without measuring (nesting)
- Variant to measure ε (quantum counting)

Brute-force attack

- We can use Grover's algorithm for a quantum brute-force key search

1. Capture a few known plaintext/ciphertext: $C_i = E_{\kappa^*}(P_i)$
2. SETUP: builds a uniform superposition of $\{0,1\}^k$ — $S = 1$
3. CHECK(κ): test whether $C_i = E_\kappa(P_i)$ — $\varepsilon = 2^{-k}, C = 1$

- Complexity $\mathcal{O}(2^{k/2})$
  - Quadratic gain
- Uses the Q1 model
  - Classical data (C_i, P_i)
  - Quantum circuit independant of the secret key κ^*

Outline

Introduction

Grover's Algorithm

Quantum Differential Cryptanalysis
- Differential
- Truncated differential

Simon's Algorithm

Breaking Modes of Operation
- Forgery attack against CBC-MAC
- Other modes of operations

Slide attacks

Differential distinguisher: classical

- Assume a *differential* $\delta_{in}, \delta_{out}$ given, with

$$h := -\log \Pr_x[E(x \oplus \delta_{in}) = E(x) \oplus \delta_{out}] \ll n,$$

Classical algorithm: search for right pairs

1: **for** $0 \leq i < 2^h$ **do**
2: $\quad x \leftarrow \textsc{Rand}()$
3: $\quad$ **if** $E(x \oplus \delta_{in}) = E(x) \oplus \delta_{out}$ **then**
4: $\quad\quad$ **return** `cipher`
5: **return** `random`

- Complexity $\mathcal{O}(2^h)$

Differential distinguisher: quantum

- Assume a *differential* $\delta_{\text{in}}, \delta_{\text{out}}$ given, with

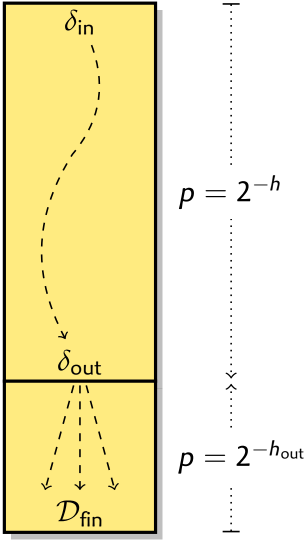
$$h := -\log \Pr_x[E(x \oplus \delta_{\text{in}}) = E(x) \oplus \delta_{\text{out}}] \ll n,$$

Quantum algorithm: Grover search for right pair

1. SETUP: builds a uniform superposition of $\{0,1\}^n$ — $S = 1$
2. CHECK(x): test whether $E(x \oplus \delta_{\text{in}}) = E(x) \oplus \delta_{\text{out}}$ — $\varepsilon = 2^{-h}, C = 1$

- Complexity $\mathcal{O}(2^{h/2})$
  - Quadratic gain
- Uses the Q2 model
  - Superposition queries to E_{κ^*} with secret key κ^*

Last-Round attack: classical

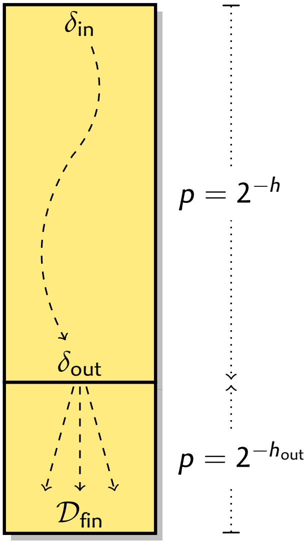


Classical algorithm

for $0 \le i < 2^h$ **do**
    $x \leftarrow \textsc{Rand}()$
    ▷ Filter possible output differences
    if $E(x) \oplus E(x \oplus \delta_{\text{in}}) \in \mathcal{D}_{\text{fin}}$ **then**
        Find last key candidates for $(x, x \oplus \delta_{\text{in}})$
        Try all possibilities for remaining key bits

- Finding partial key candidates costs $C_{k_{\text{out}}}$
  - Between 1 and $2^{k_{\text{out}}}$
- $T = 2^h + 2^{h-n+\Delta_{\text{fin}}} \cdot \left(C_{k_{\text{out}}} + 2^{k-h_{\text{out}}}\right)$

Last-Round attack: quantum Q2

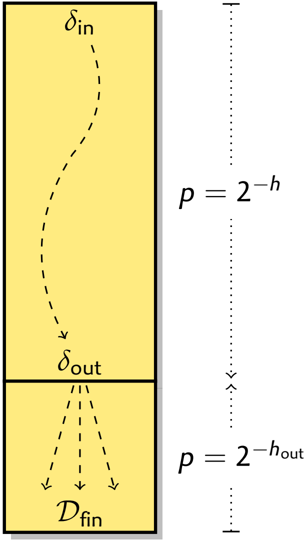


Quantum algorithm: Grover search for right pair

1. SETUP: builds a uniform superposition of $X = \{x : E(x) \oplus E(x \oplus \delta_{\text{in}}) \in \mathcal{D}_{\text{fin}}\}$ using nested Grover algorithm $S = 2^{(n-\Delta_{\text{fin}})/2}$
2. CHECK(x): Find last key cand. for $(x, x \oplus \delta_{\text{in}})$ Run nested Grover over remaining key bits $\varepsilon = 2^{n-h-\Delta_{\text{fin}}}, C = C^*_{k_{\text{out}}} + 2^{(k-h_{\text{out}})/2}$

- Repeat key recovery with right pair
- Finding partial key candidates costs $C^*_{k_{\text{out}}}$
  - Between 1 and $2^{k_{\text{out}}/2}$
- $T = 2^{h/2} + 2^{(h-n+\Delta_{\text{fin}})/2} \cdot \left(C^*_{k_{\text{out}}} + 2^{(k-h_{\text{out}})/2}\right)$

Last-Round attack: quantum Q1



- Previous attack uses superposition queries
- Alternatively, make 2^h classical queries
  - Interesting if $2^h < 2^{k/2}$
  - E.g. AES-256

Quantum algorithm: Grover search for right pair

1. SETUP: builds superposition of classical data using quantum memory $S = 1$
2. CHECK(x): same as Q2

$\varepsilon = 2^{n-h-\Delta_{\text{fin}}}, C = C^*_{k_{\text{out}}} + 2^{(k-h_{\text{out}})/2}$

- $T = 2^h + 2^{(h-n+\Delta_{\text{fin}})/2} \cdot \left(C^*_{k_{\text{out}}} + 2^{(k-h_{\text{out}})/2}\right)$

Truncated differential cryptanalysis

- Use a vector space of input / output differences: $\mathcal{D}_{\text{in}}, \mathcal{D}_{\text{out}}$ given (dim. $\Delta_{\text{in}}, \Delta_{\text{out}}$), with

$$h := -\log \Pr_{x,\delta \in \mathcal{D}_{\text{in}}} [E(x \oplus \delta) \oplus E(x) \in \mathcal{D}_{\text{out}}] \ll n - \Delta_{\text{out}}$$

Classical distinguisher: use structures

- Encrypt $2^{\Delta_{\text{in}}}$ plaintexts $x \oplus \mathcal{D}_{\text{in}}$, build $2^{2\Delta_{\text{in}-1}}$ pairs x_i, x_j
- Detect when there is y_1, y_2 s.t. $y_1 \oplus y_2 \in \mathcal{D}_{\text{out}}$: truncate to $\mathcal{D}_{\text{out}}^{\perp}$, find collisions
- Complexity $\mathcal{O}(2^{h-\Delta_{\text{in}}})$

Quantum algorithm: Grover search for structure with right pair

1. Setup: builds a uniform superposition of $\{0,1\}^n$ $S = 1$
2. Check(x): test whether $\exists y_1, y_2 \in x \oplus \mathcal{D}_{\text{in}}$ s.t. $y_1 \oplus y_2 \in \mathcal{D}_{\text{out}}$ $\varepsilon = 2^{-h+2\Delta_{\text{in}}}$, $C = ?$

- Complexity $\mathcal{O}(2^{h/2-\Delta_{\text{in}}/3})$ — less than quadratic speedup

Collision search

- Search for collisions in a list L of N elements

Classical algorithm

1: $\textsc{Sort}(L)$
2: **for** $0 < i < |L|$ **do**
3: **if** $L[i] = L[i+1]$ **then return** $L[i]$
4: **return** $\perp$

- Complexity $\widetilde{\mathcal{O}}(N)$

Quantum algorithmic: Ambainis' element distinctness

- Quantum walk algorithm to find collisions
- Complexity $\mathcal{O}(N^{2/3})$ — less than quadratic speedup!
- Uses memory $\mathcal{O}(N^{2/3})$

Collision search

- Search for collisions in a list L of N elements

Classical algorithm

1: $\textsc{Sort}(L)$
2: **for** $0 < i < |L|$ **do**
3: **if** $L[i] = L[i+1]$ **then return** $L[i]$
4: **return** $\perp$

- Complexity $\widetilde{\mathcal{O}}(N)$

Quantum algorithmic: Ambainis' element distinctness

- Quantum walk algorithm to find collisions
- Complexity $\mathcal{O}(N^{2/3})$ — less than quadratic speedup!
- Uses memory $\mathcal{O}(N^{2/3})$

Truncated differential cryptanalysis

- Use a vector space of input / output differences: $\mathcal{D}_{\text{in}}, \mathcal{D}_{\text{out}}$ given (dim. $\Delta_{\text{in}}, \Delta_{\text{out}}$), with

$$h := -\log \Pr_{x,\delta\in\mathcal{D}_{\text{in}}}[E(x\oplus\delta)\oplus E(x)\in\mathcal{D}_{\text{out}}] \ll n-\Delta_{\text{out}}$$

Classical distinguisher: use structures

- Encrypt $2^{\Delta_{\text{in}}}$ plaintexts $x\oplus\mathcal{D}_{\text{in}}$, build $2^{2\Delta_{\text{in}-1}}$ pairs x_i, x_j
- Detect when there is y_1, y_2 s.t. $y_1\oplus y_2\in\mathcal{D}_{\text{out}}$: truncate to $\mathcal{D}_{\text{out}}^{\perp}$, find collisions
- Complexity $\mathcal{O}(2^{h-\Delta_{\text{in}}})$

Quantum algorithm: Grover search for structure with right pair

1. SETUP: builds a uniform superposition of $\{0,1\}^n$ $S = 1$
2. CHECK(x): test whether $\exists y_1, y_2 \in x\oplus\mathcal{D}_{\text{in}}$ s.t. $y_1\oplus y_2\in\mathcal{D}_{\text{out}}$ $\varepsilon = 2^{-h+2\Delta_{\text{in}}}$, $C = 2^{2\Delta_{\text{in}}/3}$

- Complexity $\mathcal{O}(2^{h/2-\Delta_{\text{in}}/3})$ — less than quadratic speedup

Summary: simplified complexities

- Simple differential distinguisher

$$D_C = 2^h \qquad D_{Q1} = 2^h = D_C \qquad D_{Q2} = 2^{h/2} = \sqrt{D_C}$$
$$T_C = 2^h \qquad T_{Q1} = 2^h = T_C \qquad T_{Q2} = 2^{h/2} = \sqrt{T_C}$$

- Simple differential LR attack

$$D_C = 2^h \qquad D_{Q1} = 2^h = D_C \qquad D_{Q2} = 2^{h/2} = \sqrt{D_C}$$
$$T_C = 2^h + C_k \qquad T_{Q1} = 2^h + C_k^* \qquad T_{Q2} = 2^{h/2} + C_k^* \approx \sqrt{T_C}$$

- Truncated differential distinguisher

$$D_C = 2^{h-\Delta_{\text{in}}} \qquad D_{Q1} = 2^{h-\Delta_{\text{in}}} = D_C \qquad D_{Q2} = 2^{h/2-\Delta_{\text{in}}/3} > \sqrt{D_C}$$
$$T_C = 2^{h-\Delta_{\text{in}}} \qquad T_{Q1} = 2^{h-\Delta_{\text{in}}} = T_C \qquad T_{Q2} = 2^{h/2-\Delta_{\text{in}}/3} > \sqrt{T_C}$$

- Truncated differential LR attack *Assuming > 1 filtered pairs / structure*

$$D_C = 2^{h-\Delta_{\text{in}}} \qquad D_{Q1} = 2^{h-\Delta_{\text{in}}} = D_C \qquad D_{Q2} = 2^{h/2-(n-\Delta_{\text{fin}})/6} > \sqrt{D_C}$$
$$T_C = 2^{h-\Delta_{\text{in}}} + C_k \qquad T_{Q1} = 2^{h-\Delta_{\text{in}}} + C_k^* \qquad T_{Q2} = 2^{h/2-(n-\Delta_{\text{fin}})/6} + C_k^* > \sqrt{T_C}$$

Conclusion

- Quantification of classical attacks using Grover and Ambainis
  - Differential, truncated differential and linear cryptanalysis
  - Independent work on quantum differential cryptanalysis [Zhou, Lu, Zhang & Sun, QIP]

- "It's complicated"
- Up to quadratic speedup
  - A cipher secure against classical cryptanalysis, is secure against this kind of quantum cryptanalysis.
- Truncated differential attacks have less than quadratic speedup
  - Can become worse than Grover key search (not an attack)
  - The best quantum attack is not always a quantum version of the best classical attack
  - Concrete examples: LAC, KLEIN
- Data complexity can only be reduced using quantum queries
- Cipher with $k > n$ are most likely to see quadratic speedup
  - Attacks with classical queries (Q1 model) possible

Outline

Previous work: breaking Even-Mansour encryption

Kuwakado & Morii [ISITA '12]

The Even-Mansour cipher can be broken with quantum queries

Even-Mansour cipher [Even & Mansour, Crypto '97]

- Simple block cipher construction, from a public permutation P
  - $E_\kappa(x) = P(x \oplus \kappa_1) \oplus \kappa_2$

$$x \rightarrow \oplus \kappa_1 \rightarrow \boxed{P} \rightarrow \oplus \kappa_2 \rightarrow E_\kappa(x)$$

- Security proof
  - Attacker is given oracle access to P and E
  - "If P is a random permutation, attacks against E_κ with time T and data D are possible only if $DT > 2^n$"

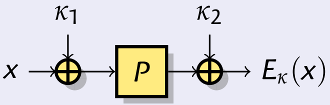
Previous work: breaking Even-Mansour encryption

Kuwakado & Morii *[ISITA '12]*

The Even-Mansour cipher can be broken with quantum queries

Even-Mansour cipher *[Even & Mansour, Crypto '97]*

- Simple block cipher construction, from a public permutation P
  - $E_\kappa(x) = P(x \oplus \kappa_1) \oplus \kappa_2$



- Security proof
  - Attacker is given oracle access to P and E
  - "If P is a random permutation, attacks against E_κ with time T and data D are possible only if $DT > 2^n$"

Classical attack against Even-Mansour

Slide with a twist attack [Biryukov & Wagner, Eurocrypt '00]

Using $2^{n/2}$ known plaintext $y_i = E_\kappa(x_i)$

1. Assume that a pair of plaintext satisfy $x' = x \oplus \kappa_1$
   - $E_\kappa(x) = P(\underbrace{x \oplus \kappa_1}_{x'}) \oplus \kappa_2, \qquad E_\kappa(x') = P(\underbrace{x' \oplus \kappa_1}_{x}) \oplus \kappa_2$
   - $E_\kappa(x) \oplus P(x') = E_\kappa(x') \oplus P(x) = \kappa_2$
   - $E_\kappa(x) \oplus P(x) = E_\kappa(x') \oplus P(x')$
2. Attacker computes $y_i \oplus P(x_i) = E_\kappa(x_i) \oplus P(x_i)$, looks for collisions
3. When $y_i \oplus P(x_i) = y_j \oplus P(x_j)$, try $\kappa_1 = x_i \oplus x_j$

Quantum attack against Even-Mansour

Kuwakado & Morii, [ISITA '12]

The Even-Mansour cipher can be broken with quantum queries

- Build the same function as in the classical attack:

$$\begin{aligned} f : \mathbb{B}^n &\to \mathbb{B}^n \\ x &\mapsto E_\kappa(x) \oplus P(x) = P(x \oplus \kappa_1) \oplus P(x) \oplus \kappa_2. \end{aligned}$$

$$f(x) = f(x \oplus \kappa_1)$$

- There is a quantum algorithm to recover κ_1 with $\mathcal{O}(n)$ queries
  - Simon's algorithm (period-finding)
  - Superposition queries to f: $\sum_x \psi_x |x\rangle|0\rangle \mapsto \sum_x \psi_x |x\rangle|f(x)\rangle$

Quantum attack against Even-Mansour

Kuwakado & Morii, [ISITA '12]

The Even-Mansour cipher can be broken with quantum queries

- Build the same function as in the classical attack:

$$\begin{aligned} f : \mathbb{B}^n &\to \mathbb{B}^n \\ x &\mapsto E_\kappa(x) \oplus P(x) = P(x \oplus \kappa_1) \oplus P(x) \oplus \kappa_2. \end{aligned}$$

$$f(x) = f(x \oplus \kappa_1)$$

- There is a quantum algorithm to recover κ_1 with $\mathcal{O}(n)$ queries
  - Simon's algorithm (period-finding)
  - Superposition queries to f: $\sum_x \psi_x |x\rangle |0\rangle \mapsto \sum_x \psi_x |x\rangle |f(x)\rangle$

Quantum attack against Even-Mansour

Kuwakado & Morii, [ISITA '12]

The Even-Mansour cipher can be broken with quantum queries

- Build the same function as in the classical attack:

$$\begin{aligned} f : \mathbb{B}^n &\to \mathbb{B}^n \\ x &\mapsto E_\kappa(x) \oplus P(x) = P(x \oplus \kappa_1) \oplus P(x) \oplus \kappa_2. \end{aligned}$$

$$f(x) = f(x \oplus \kappa_1)$$

1. Build a quantum circuit for f, from a circuit for E_κ
2. Apply Simon's algorithm to recover κ_1

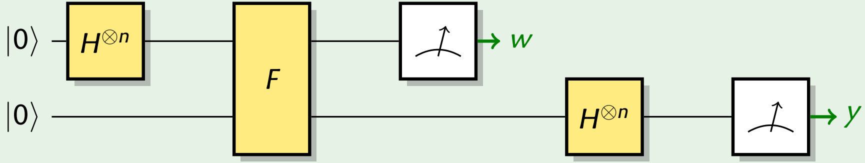
Simon's Algorithm

Definition (Simon's problem)

Given $f : \mathbb{B}^n \to \mathbb{B}^n$ such that there exists $\delta \in \mathbb{B}^n$ with $f(x) = f(x') \Leftrightarrow x \oplus x' \in \{0^n, \delta\}$, find δ.

- Classical algorithms require $\mathcal{O}(2^{n/2})$ queries (finding collisions)
- Simon's algorithm require $\mathcal{O}(n)$ quantum queries

One step of Simon's algorithm returns $y \perp \delta$



Simon's Algorithm

Definition (Simon's problem)

Given $f : \mathbb{B}^n \to \mathbb{B}^n$ such that there exists $\delta \in \mathbb{B}^n$ with $f(x) = f(x') \Leftrightarrow x \oplus x' \in \{0^n, \delta\}$, find δ.

- Classical algorithms require $\mathcal{O}(2^{n/2})$ queries (finding collisions)
- Simon's algorithm require $\mathcal{O}(n)$ quantum queries

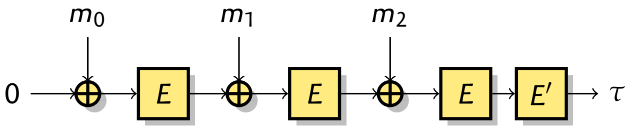
Weaker promise

$f(x) = f(x') \Leftarrow x \oplus x' \in \{0^n, \delta\}$ *i.e.* $\forall x,\ f(x) = f(x \oplus \delta)$

- There are extra collisions $f(x) = f(x')$ with arbitrary $x \oplus x'$
- If there is no structure in these collisions, we can still recover δ
- Complexity increase by a factor $\mathcal{O}(1/(1-\varepsilon))$, with $\varepsilon = \max_{t \neq \{0,\delta\}} \Pr_x[f(x) = f(x \oplus t)]$

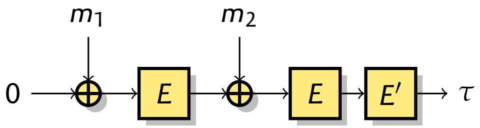
Outline

CBC-MAC



- One of the first MAC [NIST, ANSI, ISO, '85?]
- Based on CBC encryption mode
- Security proof [Bellare, Kilian & Rogaway '94]
  - "If E is a secure block cipher, there are no forgery attacks against CBC-MAC with less than $2^{n/2}$ blocs"

Classical Attack against CBC-MAC



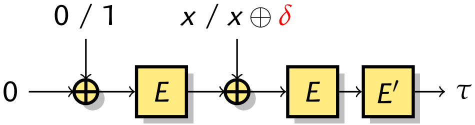
- Collision attack: two sets of $2^{n/2}$ messages

- $A_x = 0 \parallel x$
- $\mathrm{MAC}(A_x) = E'(E(x \oplus E(0)))$

- $B_y = 1 \parallel y$
- $\mathrm{MAC}(B_y) = E'(E(y \oplus E(1)))$

- Collision (A_x, B_y)?
  - The MAC collide iff $x \oplus E(0) = y \oplus E(1)$
  - Deduce $\delta = E(0) \oplus E(1) = x \oplus y$
  - Produce forgeries: $\mathrm{MAC}(0 \parallel m \parallel m') = \mathrm{MAC}(1 \parallel m \oplus \delta \parallel m')$

Quantum attack against CBC-MAC



- Consider the following function:

$$\begin{aligned} f : \mathbb{B} \times \mathbb{B}^n &\to \mathbb{B}^n \\ b, x &\mapsto \mathrm{MAC}(b \parallel x) = E'\big(E(x \oplus E(b))\big) \\ f(0,x) &= E'(E(x \oplus E(0))) \\ f(1,x) &= E'(E(x \oplus E(1))) \end{aligned}$$

- $f(b,x) = f(b \oplus 1, x \oplus \delta)$, with $\delta = E(0) \oplus E(1)$
  - Simon's algorithm recovers $1 \parallel \delta$
  - Produce forgeries: $\mathrm{MAC}(0 \parallel m) = \mathrm{MAC}(1 \parallel m \oplus \delta)$

Attack structure

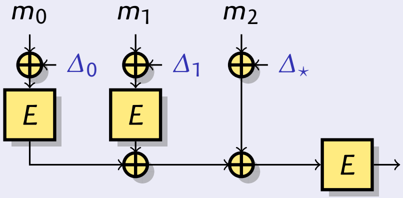
1. Define a function f with $f(x \oplus \delta) = f(x)$ for some interesting δ
2. Build quantum circuit for f, use Simon's algorithm to recover δ
   - $t = \mathcal{O}(n)$ quantum queries
3. Use δ to produce forgeries
   - One classical query gives two messages/MAC pairs
   - Repeat until more valid messages than queries $(t + 1 \text{ times})$

Applications of Simon's algorithm

- Breaks most common MAC and AEAD modes
- Corresponds to classical attacks with $2^{n/2}$ queries
  - Query f with $2^{n/2}$ values, look for collisions

PMAC: Parallelisable MAC with secret offsets

PMAC



- With 2-block msg, $\approx$ CBC-MAC
- Same attack

$$
\begin{aligned}
f : \mathbb{B} \times \mathbb{B}^n &\to \mathbb{B}^n \\
b, x &\mapsto \mathrm{MAC}(b \,\|\, x) \\
f(b,x) &= E(E(b \oplus \Delta_0) \oplus x \oplus \Delta_\star) \\
f(b,x) &= f(b \oplus 1, x \oplus \delta) \\
\delta &= E(\Delta_0) \oplus E(\Delta_0 \oplus 1)
\end{aligned}
$$

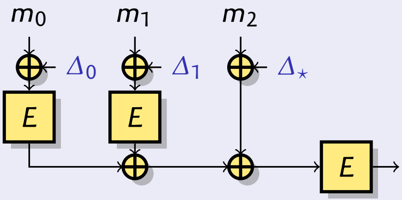
PMAC variant

- No message xored into state
- Alternative attack

$$
\begin{aligned}
f : \mathbb{B}^n &\to \mathbb{B}^n \\
x &\mapsto \mathrm{MAC}(x \,\|\, x) \\
f(x) &= E(E(x \oplus \Delta_0) \oplus E(x \oplus \Delta_1)) \\
f(x) &= f(x \oplus \delta) \\
\delta &= \Delta_0 \oplus \Delta_1
\end{aligned}
$$

PMAC: Parallelisable MAC with secret offsets

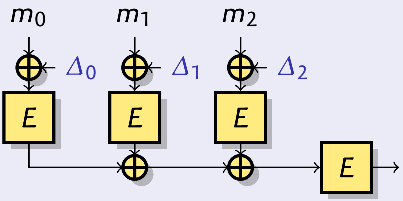
PMAC



- With 2-block msg, $\approx$ CBC-MAC
- Same attack

$$
\begin{aligned}
f : \mathbb{B} \times \mathbb{B}^n &\to \mathbb{B}^n \\
b, x &\mapsto \mathrm{MAC}(b \,\|\, x) \\
f(b,x) &= E(E(b \oplus \Delta_0) \oplus x \oplus \Delta_\star) \\
f(b,x) &= f(b \oplus 1, x \oplus \delta) \\
\delta &= E(\Delta_0) \oplus E(\Delta_0 \oplus 1)
\end{aligned}
$$

PMAC variant

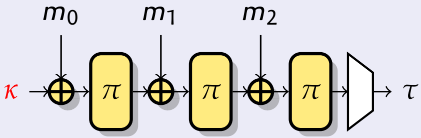


- No message xored into state
- Alternative attack

$$
\begin{aligned}
f : \mathbb{B}^n &\to \mathbb{B}^n \\
x &\mapsto \mathrm{MAC}(x \,\|\, x) \\
f(x) &= E(E(x \oplus \Delta_0) \oplus E(x \oplus \Delta_1)) \\
f(x) &= f(x \oplus \delta) \\
\delta &= \Delta_0 \oplus \Delta_1
\end{aligned}
$$

Sponge-based modes

Full-width sponge



- Same structure as CBC-MAC
- Same attack

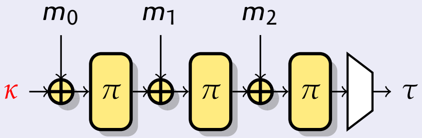
$$
\begin{aligned}
f : \mathbb{B} \times \mathbb{B}^n &\to \mathbb{B}^n \\
b, x \quad &\mapsto \mathrm{MAC}(b \,\|\, x) \\
f(b,x) &= \pi(\pi(\kappa \oplus b) \oplus x) \\
f(b,x) &= f(b \oplus 1, x \oplus \delta) \\
\delta &= \pi(\kappa) \oplus \pi(\kappa \oplus 1)
\end{aligned}
$$

Normal sponge

- Can't cancel the full state difference
- No attack found

Sponge-based modes

Full-width sponge



- Same structure as CBC-MAC
- Same attack

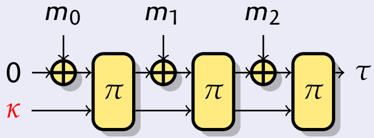
$$\begin{aligned} f : \mathbb{B} \times \mathbb{B}^n &\to \mathbb{B}^n \\ b, x &\mapsto \mathrm{MAC}(b \,\|\, x) \end{aligned}$$

$$f(b,x) = \pi(\pi(\kappa \oplus b) \oplus x)$$

$$f(b,x) = f(b \oplus 1, x \oplus \delta)$$

$$\delta = \pi(\kappa) \oplus \pi(\kappa \oplus 1)$$

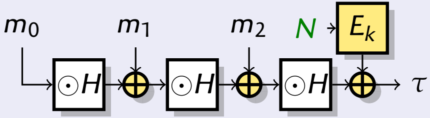
Normal sponge



- Can't cancel the full state difference
- No attack found

Nonce-based modes

Nonce at the end (GMAC)



- Same structure as CBC-MAC
- Same attack

$$
\begin{aligned}
f_N : \mathbb{B} \times \mathbb{B}^n &\to \mathbb{B}^n \\
b, x &\mapsto \mathrm{MAC}(b \,\|\, x) \\
f_N(b,x) &= b \cdot H^2 \oplus x \cdot H \oplus E_k(N) \\
f_N(b,x) &= f(b \oplus 1, x \oplus \delta) \\
\delta &= H
\end{aligned}
$$

Nonce at the beginning (CCM)

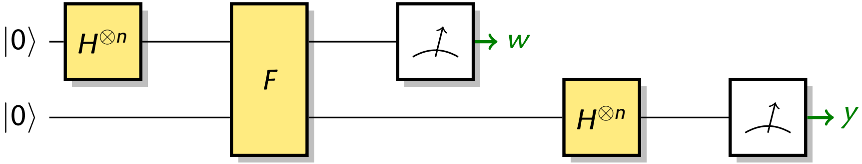
- State difference depend on N
- No fixed period δ
- No attack found

Dealing with the nonce

- We can't really apply Simon's algorithm to f_N
  - We don't choose N
  - Each oracle call will use a different N
- Luckily, one step of Simon's algorithm makes a single call to f_N
  - The family f_N satisfies Simon's promise with the same δ
  - One step gives y with $y \perp \delta$
  - Classical repetition, classical linear algebra

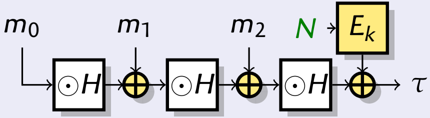
Dealing with the nonce

- We can't really apply Simon's algorithm to f_N
  - We don't choose N
  - Each oracle call will use a different N
- Luckily, one step of Simon's algorithm makes a single call to f_N
  - The family f_N satisfies Simon's promise with the same δ
  - One step gives y with $y \perp \delta$
  - Classical repetition, classical linear algebra



Nonce-based modes

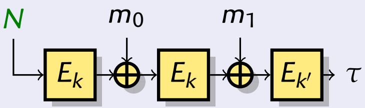
Nonce at the end (GMAC)



- Same structure as CBC-MAC
- Same attack

$$\begin{aligned} f_N : \mathbb{B} \times \mathbb{B}^n &\to \mathbb{B}^n \\ b, x &\mapsto \mathrm{MAC}(b \,\|\, x) \\ f_N(b,x) &= b \cdot H^2 \oplus x \cdot H \oplus E_k(N) \\ f_N(b,x) &= f(b \oplus 1, x \oplus \delta) \\ \delta &= H \end{aligned}$$

Nonce at the beginning (CCM)



- State difference depend on N
- No fixed period δ
- No attack found

Quantum security of modes of operations

Applications of Simon's algorithm

Common MAC and AEAD modes broken with superposition queries:

- CBC-MAC, PMAC, GMAC, GCM, OCB
- 8 CAESAR candidates: **AEZ**, **CLOC**, **COLM**, Minalpher, **OCB**, OMD, **OTR**, POET

Secure modes

- Common encryption modes are mostly quantum-secure
  [Unruh, Targhi, Tabia & Anand, PQC'16]
- Efficient MACs & AEAD secure against quantum attacks?
  - Boneh & Zhandry: quantum safe Carter-Wegman MAC, where the randomness depend on the message
  - Alagic and Russell: replace xor by other group operation
- Do we have the right security definition?

Outline

Introduction

Grover's Algorithm

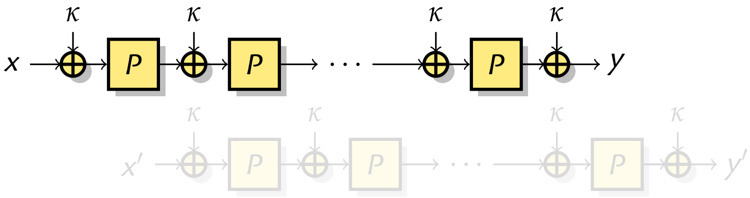
Quantum Differential Cryptanalysis
- Differential
- Truncated differential

Simon's Algorithm

Breaking Modes of Operation
- Forgery attack against CBC-MAC
- Other modes of operations

Slide attacks

Classical slide attacks

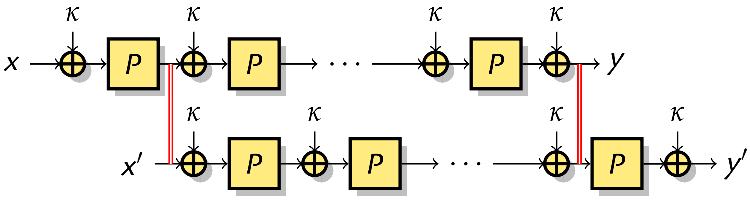


- Cryptanalysis of block ciphers [Biryukov & Wagner, FSE '99]
- Applicable if all rounds are identical

$E_\kappa(P(x \oplus \kappa)) = P(E_\kappa(x)) \oplus \kappa$

1. Assume a pair $x' = P(x \oplus \kappa)$, then $y' = P(y) \oplus \kappa$
   - $x \oplus P^{-1}(x') = P(y) \oplus y' = \kappa$
   - $x \oplus P(y) = P^{-1}(x') \oplus y'$
2. Attacker looks for collision between
   - $x_i \oplus P(y_i)$
   - $P^{-1}(x_j) \oplus y_j$
3. When $x_i \oplus P(y_i) = P^{-1}(x_j) \oplus y_j$, try $\kappa = x_i \oplus P^{-1}(x_j)$

Classical slide attacks



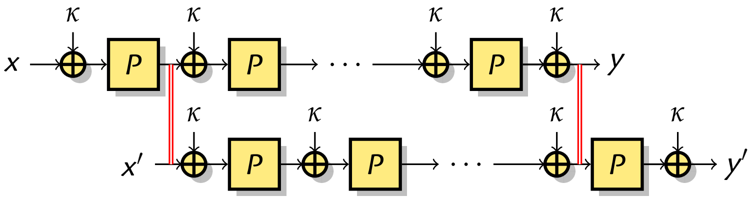
- Cryptanalysis of block ciphers
- Applicable if all rounds are identical

[Biryukov & Wagner, FSE '99]

$E_\kappa(P(x \oplus \kappa)) = P(E_\kappa(x)) \oplus \kappa$

1. Assume a pair $x' = P(x \oplus \kappa)$, then $y' = P(y) \oplus \kappa$
   - $x \oplus P^{-1}(x') = P(y) \oplus y' = \kappa$
   - $x \oplus P(y) = P^{-1}(x') \oplus y'$
2. Attacker looks for collision between
   - $x_i \oplus P(y_i)$
   - $P^{-1}(x_j) \oplus y_j$
3. When $x_i \oplus P(y_i) = P^{-1}(x_j) \oplus y_j$, try $\kappa = x_i \oplus P^{-1}(x_j)$

Quantum slide attacks



- $E_\kappa(P(x \oplus \kappa)) = P(E_\kappa(x)) \oplus \kappa$
- Build function inspired by the classical attack:

$$f : \mathbb{B} \times \mathbb{B}^n \to \mathbb{B}^n$$

$$b, x \quad \mapsto \begin{cases} x \oplus P(E_\kappa(x)) & \text{if } b = 0, \\ x \oplus E_\kappa(P(x)) & \text{if } b = 1. \end{cases}$$

- $f(0, x) = P(E_\kappa(x)) \oplus x = E_\kappa(P(x \oplus \kappa)) \oplus \kappa \oplus x = f(1, x \oplus \kappa)$
  - Simon's algorithm recovers $1 \parallel \kappa$

Conclusion

Applications of two quantum algorithms on symmetric crypto

1. Grover's Algorithm (and variant)
   - Quadratic speedup for some cryptanalysis techniques
2. Simon's Algorithm
   - $\mathcal{O}(n)$ attacks against common MAC and AEAD modes
   - $\mathcal{O}(n)$ slide attack

- There are more quantum attacks than Grover key search for symmetric crypto
  - Against primitives and modes
- Most of our attacks require superposition queries