Towards a better understanding of (post-)quantum security of symmetric key schemes

NTT Secure Platform Laboratories (and Nagoya University)
Akinori Hosoyamada
Introduction
Quantum Attacks against Symmetric Cryptosystems?

It has been said that symmetric key schemes would not to be much affected by quantum computers.
### Known Quantum Attacks

<table>
<thead>
<tr>
<th></th>
<th>Classical</th>
<th>Quantum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exhaustive Key Search</td>
<td>$O(2^n)$</td>
<td>$O(2^{n/2})$</td>
</tr>
<tr>
<td>Collision Finding</td>
<td>$O(2^{n/2})$</td>
<td>$O(2^{n/3})$</td>
</tr>
</tbody>
</table>

"2n-bit key suffices"
## Known Quantum Attacks: Today

<table>
<thead>
<tr>
<th>Attack</th>
<th>Classical</th>
<th>Quantum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exhaustive Key Search</td>
<td>$O(2^n)$</td>
<td>$O(2^{n/2})$</td>
</tr>
<tr>
<td>Collision Finding</td>
<td>$O(2^{n/2})$</td>
<td>$O(2^{n/3})$</td>
</tr>
<tr>
<td>Key Recovery on Even-Mansour</td>
<td>$O(2^{n/2})$</td>
<td>Polynomial time</td>
</tr>
<tr>
<td>Forgery against CBC-MAC</td>
<td>$O(2^{n/2})$</td>
<td>Polynomial time</td>
</tr>
</tbody>
</table>

Remark: The last two attacks assumes that quantum keyed oracles are available.
Quantum Attacks against Symmetric Cryptosystems?

It has been said that symmetric key schemes would not to be much affected by quantum computers

Symmetric key schemes may be significantly affected!!
  - Attacks by Kuwakado and Morii at ISIT2010, ISITA2012
  - Attacks by Kaplan et al. at CRYPTO2016
Quantum Attacks against Symmetric Cryptosystems?

It has been said that symmetric key schemes would not be much affected by quantum computers

Symmetric key schemes may be significantly affected!!
- Attacks by Kuwakado and Morii at ISIT2010, ISITA2012
- Attacks by Kaplan et al. at CRYPTO2016

Post-quantum security of symmetric schemes should be analyzed more carefully
Attack Models

Chosen Plaintext Attack

Enc. Oracle

Message

Ciphertext

Adversary

Computer
Attack Models

**Chosen Plaintext Attack**

- **Message** → **Enc. Oracle** → **Ciphertext**
  - **Adversary**
  - **Computer**

**Chosen Plaintext Attack**

- **Message** → **Enc. Oracle** → **Ciphertext**
  - **Adversary**
  - **Quantum Computer**
Attack Models

Chosen Plaintext Attack

Enc. Oracle

Message

Ciphertext

Adversary

Computer

Chosen Plaintext Attack

Q2 model, quantum query

Quantum Enc. Oracle

Quantum Superposed Message

Quantum Superposed Ciphertext

Adversary

Computer

Quantum Computer
A1. Classical algorithms can be converted into quantum algorithms.


A3. For hash functions, quantum query attacks are natural.

A4. If a scheme is secure against quantum query attacks, it can be used in cryptographic applications that run on quantum computers.

Question: Why should we consider quantum query attacks?
Question: Why should we consider quantum query attacks?

A1. Classical algorithms can be converted into quantum algorithms

quantum query attacks on obfuscated implementations?
Question: Why should we consider quantum query attacks?

A1. Classical algorithms can be converted into quantum algorithms
   quantum query attacks on obfuscated implementations?

A2. Quantum query attacks lead to more realistic [classical query + quantum computation] attacks
   Ex.) Offline Simon’s algorithm at Asiacrypt 2019.
Question: Why should we consider quantum query attacks?

A1. Classical algorithms can be converted into quantum algorithms. Quantum query attacks on obfuscated implementations?


A3. For hash functions, quantum query attacks are natural.
Question: Why should we consider quantum query attacks?

A1. Classical algorithms can be converted into quantum algorithms. Quantum query attacks on obfuscated implementations?

A2. Quantum query attacks lead to more realistic \([\text{classical query} + \text{quantum computation}]\) attacks. Ex.) Offline Simon’s algorithm at Asiacrypt 2019.

A3. For hash functions, quantum query attacks are natural.

A4. If a scheme is secure against quantum query attacks, it can be used in cryptographic applications that run on quantum computers.
Quantum Query Attacks
## Known Quantum Attacks: Today

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Classical</th>
<th>Quantum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exhaustive Key Search</td>
<td>$O(2^n)$</td>
<td>$O(2^{n/2})$</td>
</tr>
<tr>
<td>Collision Finding</td>
<td>$O(2^{n/2})$</td>
<td>$O(2^{n/3})$</td>
</tr>
<tr>
<td>Key Recovery on Even-Mansour</td>
<td>$O(2^{n/2})$</td>
<td>Polynomial time</td>
</tr>
<tr>
<td>Forgery against CBC-MAC</td>
<td>$O(2^{n/2})$</td>
<td>Polynomial time</td>
</tr>
</tbody>
</table>

**Remark:** The last two attacks assumes that quantum keyed oracles are available.
## Known Quantum Attacks: Today

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Classical</th>
<th>Quantum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exhaustive Key Search</td>
<td>(O(2^n))</td>
<td>(O(2^{n/2}))</td>
</tr>
<tr>
<td>Collision Finding</td>
<td>(O(2^{n/2}))</td>
<td>Polynomial time</td>
</tr>
<tr>
<td>Key Recovery on Even-Mansour</td>
<td>(O(2^{n/2}))</td>
<td>Polynomial time</td>
</tr>
<tr>
<td>Forgery against CBC-MAC</td>
<td>(O(2^{n/2}))</td>
<td>Polynomial time</td>
</tr>
</tbody>
</table>

**Remark:** The last two attacks assumes that quantum keyed oracles are available.

Simon's algorithm

[The table includes the complexities of classical and quantum versions of different known quantum attacks, with a remark noting the added complexity due to quantum capabilities.]
Simon’s period finding algorithm

Problem

Suppose \( f : \{0,1\}^n \rightarrow S \) and \( s \in \{0,1\}^n \) satisfy
\[
\forall x \in \{0,1\}^n \ f(x \oplus s) = f(x)
\]
Given an oracle access to \( f \), find \( s \).

Classical algorithms: Exponential time

Simon’s quantum algorithm: Polynomial time [Sim97]

Suppose \( f: \{0, 1\}^n \to \{0, 1\} \) and \( s \in \{0, 1\}^n \) satisfy:

\[
\forall x \in \{0, 1\}^n, \quad f(x) \oplus s = f(x)
\]

Given an oracle access to \( f \), find \( s \).

**Problem**

To mount poly-time attacks, it is important to reduce the target problem to Simon’s problem.

Key-Recovery Attack on Even-Mansour

Even-Mansour cipher $E_{k_1,k_2}$
(P: public permutation)

Quantum CPA against Even-Mansour ciphers

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

- We can recover $k_1$ in polynomial time with Simon’s algorithm
- $k_2$ can easily be recovered since we have
  $$E_{k_1,k_2}(x) \oplus P(x \oplus k_1) = k_2$$

Various MACs/AEs are broken in poly-time...

If quantum queries are allowed, Simon’s algorithm breaks
- CBC-MAC
- PMAC
- GMAC
- GCM
- OCB
...

In polynomial time!

M. Kaplan, G. Leurent, A. Leverrier, and M. Naya-Plasencia: Breaking symmetric cryptosystems using quantum period finding (CRYPTO 2016)
### Luby-Rackoff (Feistel) Construction

#### Security in the classical setting

<table>
<thead>
<tr>
<th></th>
<th>PRP? (secure against CPA?)</th>
<th>SPRP? (secure against CCA?)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-round</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>3-round</td>
<td>○</td>
<td>×</td>
</tr>
<tr>
<td>4-round</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>5-round</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

M. Luby, C. Rackoff: How to construct pseudo-random permutations from pseudorandom functions (CRYPTO '85)
## Luby-Rackoff (Feistel) Construction

### Security in the quantum setting

<table>
<thead>
<tr>
<th>Rounds</th>
<th>PRP? (secure against CPA?)</th>
<th>SPRP? (secure against CCA?)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-round</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>3-round</td>
<td>× [KM10]</td>
<td>×</td>
</tr>
<tr>
<td>4-round</td>
<td>☐ [HI19]</td>
<td>× [IHMSI19]</td>
</tr>
<tr>
<td>5-round</td>
<td>☐ [HI19]</td>
<td>?</td>
</tr>
</tbody>
</table>


[HI19] A. Hosoyamada, T. Iwata: 4-Round Luby-Rackoff construction is a qPRP. (Asiacrypt 2019)
Other Quantum Query Attacks

- Speed-up for differential/linear cryptanalysis [KLLN16b]
- Key recovery attacks on Feistel by using the quantum distinguishers [HS18b, IHMSI19]
- The attack with Kuperberg’s algorithm [BN18]
- The attack on the FX construction by Leander and May [LM17]
- Speed-up for Demiric-Secluk meet-in-the-middle attack [HS18b, BNS19]


Attacks with Classical Query + Quantum Computation
Attack Models

Chosen Plaintext Attack

Enc. Oracle

Message → Ciphertext

Adversary

Computer

Q1 model, classical query

Enc. Oracle

Message → Ciphertext

Adversary

Quantum Computer
Quantum query attack with Simon’s algorithm is applicable

Simple On-Off MITM attack is applicable in the classical setting

Even if quantum queries are not allowed and just a small quantum computer is available, by using Simon’s algorithm we can mount a memory-efficient attack.

# Offline Simon’s algorithm (AC 2019)

(Q1 / Classical query) attacks on Even-Mansour

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Query</th>
<th>Q. Mem</th>
<th>C. Mem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kuwakado &amp; Morii [KM12]</td>
<td>$2^n/3$</td>
<td>$2^n/3$</td>
<td>$2^n/3$</td>
<td>$2^n/3$</td>
</tr>
<tr>
<td>Hosoyamada &amp; Sasaki [HS18a]</td>
<td>$2^{3n/7}$</td>
<td>$2^{3n/7}$</td>
<td>Poly(n)</td>
<td>$2^n/7$</td>
</tr>
<tr>
<td><strong>Offline Simon</strong></td>
<td>$2^n/3 (&lt; 2^{3n/7})$</td>
<td>$2^n/3$</td>
<td>poly(n)</td>
<td>poly(n)</td>
</tr>
</tbody>
</table>

Note: Polynomial factors are ignored. Only classical queries are allowed to keyed oracles. No parallelized computations.

Other classical query attacks

• Differential / Linear Cryptanalysis [KLLN16b]
• Online-Offline meet-in-the-middle attacks [HS18a]
• Demiric-Selçuk meet-in-the-middle attacks [BNH19,HS18b]

and more...


Generic Attacks on Hash
Collision Attack on Hash

Collision of $f$
Collision Attack on Hash

The number of queries required to find a collision

Classical: $\Theta(N^{1/2})$

Quantum: $\Theta(N^{1/3})$

[BHT97,Zha15]


MultiCollision Attack on Hash

3-Collision of $f$
MultiCollision Attack on Hash

The number of queries required to find an $\ell$-collision

Classical: $\Theta(N^{(\ell-1)/\ell})$ [STKT08]

Quantum: $\Theta\left(\frac{2^{\ell-1} - 1}{N \cdot 2^{\ell-1}}\right)$

[HSX17, HSTX19, LZ19]


MultiCollision Attack on Hash

<table>
<thead>
<tr>
<th>$\ell$ (multiplicity)</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical (◯)</td>
<td>$\frac{1}{N^2}$</td>
<td>$\frac{2}{N^3}$</td>
<td>$(N^4)$</td>
<td>$(N^5)$</td>
</tr>
<tr>
<td>Quantum (●)</td>
<td>$\frac{1}{N^3}$</td>
<td>$\frac{3}{N^7}$</td>
<td>$\frac{15}{N^{31}}$</td>
<td>$\frac{31}{N^{63}}$</td>
</tr>
</tbody>
</table>

$\text{log}_N(\text{Query})$

![Graph showing the relationship between $\ell$ and $\text{log}_N(\text{Query})$]
Other generic attacks on hash

• Collision finding with polynomial number of qubits [CNS17]
  – The BHT algorithm finds a collision in time $N^{1/3}$ but requires $N^{1/3}$ qubits...
  – Even if only poly-qubits are available, collision can be found in time $N^{2/5} (< N^{1/2})$

• Acceleration for the k-xor problem [Amb07, GNS18]

• Multi-target preimage search [BB17, CNS17]
  – Applicable to key recovery in multi-key/user setting

Challenges for the future in cryptanalysis
Attacks on keyed primitives

- More attacks on concrete primitives
- Applications of quantum algorithms other than Simon (period finding), Grover, Quantum-walk-search
- New quantum algorithms (attacks) that are specific to concrete symmetric key schemes
- Other applications of quantum algorithms in the classical query model

and more...
Generic attacks on hash

• New Time-Memory tradeoff for inverting functions that is better than the classical tradeoff?
Time-Memory tradeoff for inverting function

- \( f \): random function/permutation (n-bit to n-bit) / \( \mathcal{A} \): adversary

1. \( \mathcal{A} \) runs precomputation with \( h \) and store (classical/quantum) data of size \( S \)
2. \( \mathcal{A} \) receives a randomly chosen \( y \)
3. \( \mathcal{A} \) tries to find \( x \) s.t. \( f(x) = y \) in time \( T \) by using the stored data

Classical tradeoff between \( T \) and \( S \):

\[
T = \frac{2^n}{S}
\]

Quantum tradeoff between \( T \) and \( S \):

So far, there does not exist any tradeoff that is better than \( T = \frac{2^n}{S} \) when \( S = 1 \) but it is not clear what kind of tradeoff is possible when \( S > 1 \)…
Time-Memory tradeoff for inverting function

- \( f \): random function/permutation (n-bit to n-bit) / \( \mathcal{A} \): adversary

1. \( \mathcal{A} \) runs precomputation with \( h \) and store (classical/quantum) data of size \( S \)
2. \( \mathcal{A} \) receives a randomly chosen \( y \)
3. \( \mathcal{A} \) tries to find \( x \) s.t. \( f(x) = y \) in time \( T \) by using the stored data

**Classical tradeoff between \( T \) and \( S \):**
\[
T = 2^n / S \quad \text{(if \( f \) is a random permutation)}
\]
Time-Memory tradeoff for inverting function

- \( f \): random function/permutation (n-bit to n-bit) / \( \mathcal{A} \): adversary
  1. \( \mathcal{A} \) runs precomputation with \( h \) and store (classical/quantum) data of size \( S \)
  2. \( \mathcal{A} \) receives a randomly chosen \( y \)
  3. \( \mathcal{A} \) tries to find \( x \) s.t. \( f(x) = y \) in time \( T \) by using the stored data

Classical tradeoff between \( T \) and \( S \):
\[
T = 2^n / S \quad \text{(if } f \text{ is a random permutation)}
\]

Quantum tradeoff between \( T \) and \( S \):

So far, there does not exist any tradeoff that is better than \( T = 2^n / S \)
Grover search achieves \( T = 2^{n/2} \) when \( S = 1 \) but it is not clear what kind of trade-off is possible when \( S > 1 \)...
Security Proofs / Lower bounds
What has already been done?

Generic bounds on random functions (query complexity)
- Preimages of random functions: $\Theta(N) \to \Theta(N^{1/2})$
- RP-RF switch: $\Theta(N^{1/2}) \to \Theta(N^{1/3})$
- Multicollision-Finding problem: $\Theta(N^{(\ell-1)/\ell}) \to \Theta(N^{(2^\ell-1-1)/2^\ell-1})$
- k-xor: $\Theta(N^{1/k}) \to \Theta(N^{1/(k+1)})$

Red: Classical Bound
Blue: Quantum Bound
What has already been done?

Security proofs for specific schemes
(against quantum query attacks, w/o algebraic assumptions)

- CPA security of encryption modes (CTR, CBC, OFB,...) (@PQCrypto2016)
- Generic composition for AE (@PQCrypto2016)
- PRF security of NMAC/HMAC (@CRYPTO2017)
- Sponge-like construction
  - PRF security of sponge with keyed (secret) permutation (@CRYPTO2017)
  - Collision-resistance (collapsing) of sponge with public function (@PQCrypto2018)
- Indifferentiability of (fixed-length) Merkle-Damgaard (@CRYPTO2019)
- PRP security of 4-Round Luby-Rackoff (Feistel) (@Asiacrypt 2019)
What is difficult in the quantum setting?

1. It is not trivial how to record queries
   - Copying the values of queries disturbs the adversary’s quantum states, which leads to changing its behavior significantly

2. “Lazy Sampling” is not available
   - In classical proofs, the value $F(x)$ of a random function $F$ is randomly chosen on the fly when the adversary queries $x$ to $F$
   - At most one value is fixed per each classical query
   - In the quantum setting, the adversary may query a superposition of all possible $x$ at the same time…
The Compressed Oracle Technique

**Compressed Oracle Technique [Zha19]**

- It enables us to do “Lazy sampling” to some extent for random functions in the quantum setting.
- The important observation: Sometimes recorded information should be “forgotten”.
- Many applications:
  - Quantum Indifferentiability of Merkle-Damgaard [Zha19]
  - Lower bound for multicollision finding problem [LZ19]
  - Quantum PRP security of 4-round Luby-Rackoff [IH19]
  - etc...

---

The Compressed Oracle Technique

One remark:
Zhandry’s compressed oracle technique cannot be applied to permutations
## Remarks on query lower bound

<table>
<thead>
<tr>
<th>Research Area</th>
<th>Problems</th>
<th>Backward query?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantum computation</td>
<td>Worst case</td>
<td>×</td>
</tr>
<tr>
<td>Public key crypto</td>
<td>Average case (randomized)</td>
<td>×</td>
</tr>
<tr>
<td>Symmetric key crypto</td>
<td>Average case (randomized)</td>
<td>○</td>
</tr>
</tbody>
</table>
Remarks on query lower bound

<table>
<thead>
<tr>
<th>Research Area</th>
<th>Problems</th>
<th>Backward query?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantum computation</td>
<td>Worst case</td>
<td>×</td>
</tr>
<tr>
<td>Public key crypto</td>
<td>Average case (randomized)</td>
<td>×</td>
</tr>
<tr>
<td>Symmetric key crypto</td>
<td>Average case (randomized)</td>
<td>○</td>
</tr>
</tbody>
</table>
It is hard to treat permutations...

- So far there is no published results on quantum proof techniques for public random permutation or ideal cipher

- Exception: One-wayness of Davies-Meyer Compression function
  - Giving security proofs by computing statistical distance
  - (so far & as far as I know) the only published results on quantum proofs for schemes in ideal permutation model / ideal cipher model w/o algebraic assumptions

Challenges for the future

- Generic and strong proof technique to treat random permutations / ideal ciphers
  - The compressed oracle technique: Since F is a random function, F(x) and F(y) are independent, which means that the quantum registers for F(x) and F(y) are not entangled.
  - If we try to apply the compressed oracle technique to a random permutation P, P(x) and P(y) are **not** independent, which means that the quantum registers for P(x) and P(y) will be **entangled**

Quantum entanglement always make things extremely difficult…
Challenges for the future

- Generic and strong proof technique to treat random permutations / ideal ciphers
  - The compressed oracle technique: Since F is a random function, F(x) and F(y) are independent, which means that the quantum registers for F(x) and F(y) are not entangled
  - If we try to apply the compressed oracle technique to a random permutation P, P(x) and P(y) are not independent, which means that the quantum registers for P(x) and P(y) will be entangled

Quantum entanglement always make things extremely difficult...

Solved??
Czajkowski, Majenz, Schaffner, Zur: Quantum lazy sampling and game-playing proofs for quantum indifferentiability. (ePrint 2019/428)
Summary
Summary

- Recent results show many unexpected attacks are possible in the quantum setting
  - Many schemes are broken in poly-time with quantum queries
  - Simon’s algorithm is applicable even if only classical queries are allowed
  - Various new tradeoffs

- There are lots of challenging but interesting topics to study
  - Time-memory tradeoffs for inverting functions?
  - Proof techniques for permutations?
  - AES can be broken with quantum algorithms?

Thank you!