Safely Doubling your Block Ciphers for a Post-Quantum World

Ritam Bhaumik, André Chailloux, Paul Frixons and María Naya-Plasencia

Orange Labs, Caen, France
Inria, Paris, France

November 10th, 2022
Outline

- Quantum setting
- Existing quantum attacks
- State of the art of quantum security proofs
- First try and attack
- New construction: QuEME
- Instantiation: Double-AES
- Best attacks on Double-AES
Quantum setting
Quantum algorithms

**Grover’s search**

Grover’s search retrieve an element in time $O(\sqrt{n})$ among $n$ other elements.

**BHT collision search**

BHT algorithm retrieve a collision in time $O(n^{1/3})$ among $n$ elements.
Quantum algorithms

Shor’s algorithm

Shor’s algorithm solves the factorization problem and discrete logarithm exponentially faster than classical algorithms.

Simon’s period finding

Given $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ a function which admits a period i.e, there exists $s \in \{0, 1\}^n$ such that for all $x, y$ in $\{0, 1\}^n$, $f(x) = f(y) \iff x = y$ or $x = y \oplus s$.

Simon’s period finding find $s$ in $O(n^3)$ computations.
# Quantum models

<table>
<thead>
<tr>
<th>Model</th>
<th>Classical queries and classical computations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical model</td>
<td><img src="image" alt="Classical model" /></td>
</tr>
<tr>
<td>Q1 model</td>
<td><img src="image" alt="Q1 model" /></td>
</tr>
<tr>
<td>Q2 model</td>
<td><img src="image" alt="Q2 model" /></td>
</tr>
</tbody>
</table>
Quantum memory models

Small quantum memory
Polynomial size or at worst sub-exponential.

QRACM
Classical memory can be queried in superposition.

QRAQM
Quantum memory can be queried in superposition.
Naive impact of quantum computer

Public-key cryptography
The factorization problem and discrete logarithm happen to be the hard problems behind the most used cryptosystems.
⇒ Shor’s algorithm breaks them.
Replacement that uses other problems (NIST PQC competition).

Secret-key cryptography
Grover’s search speeds up the search for the secret keys.
⇒ Doubling the size of the keys should be enough.
Doubling the state size too [CNS17] not always enough
⇒ Need for a block cipher with 256-bit key and state (like Saturnin)
$LR_5$ is the natural proposition but still no proof after many years
Naive impact of quantum computer

Public-key cryptography
The factorization problem and discrete logarithm happen to be the hard problems behind the most used cryptosystems.  
⇒ Shor’s algorithm breaks them.

Replacement that uses other problems (NIST PQC competition).

Secret-key cryptography
Grover’s search speeds up the search for the secret keys.  
⇒ Doubling the size of the keys should be enough.

Doubling the state size too [CNS17] not always enough  
⇒ Need for a block cipher with 256-bit key and state (like Saturnin)  
$LR_5$ is the natural proposition but still no proof after many years.
Naive impact of quantum computer

Public-key cryptography
The factorization problem and discrete logarithm happen to be the hard problems behind the most used cryptosystems.
⇒ Shor’s algorithm breaks them.
Replacement that uses other problems (NIST PQC competition).

Secret-key cryptography
Grover’s search speeds up the search for the secret keys.
⇒ Doubling the size of the keys should be enough.

Doubling the state size too [CNS17] not always enough
⇒ Need for a block cipher with 256-bit key and state (like Saturnin)
$LR_5$ is the natural proposition but still no proof after many years
Existing quantum attacks
**Even-Mansour cipher**

Given $\Pi$ a public $n$-bit permutation and $k_1, k_2$ two $n$-bit secret keys,

$$EM : x \mapsto \Pi(x \oplus k_1) \oplus k_2.$$  

The Even-Mansour cipher is secure against classical attacks up to $2^{n/2}$ computations.

*Figure: Even-Mansour Cipher*
Q2 attack on the Even-Mansour cipher

Observe that $EM(x) \oplus \Pi(x) = \Pi(x \oplus k_1) \oplus k_2 \oplus \Pi(x) = EM(x \oplus k_1) \oplus \Pi(x \oplus k_1)$.

Then Simon’s algorithm applied to $x \mapsto EM(x) \oplus \Pi(x)$ retrieves $k_1$, we can find $k_2 = EM(x) \oplus \Pi(x \oplus k_1)$. 
$LR_3$ (3-round Feistel network)

**Figure:** $LR_3$ (3-round Feistel network)
We fix two values for the right part $R_0$ and $R_1$.
Observe that $L'(L, R_\beta) \oplus R_\beta$ only depends on $L \oplus f_1(R_\beta)$.
Then Simon’s algorithm applied to $(x, \beta) \mapsto L'(x, R_\beta) \oplus R_\beta$ retrieves $f_1(R_0) \oplus f_1(R_1)$.
Attack on $LR_4$
Given \( E_k \) a \( n \)-bit block cipher and \( k_1, k_2 \) two \( n \)-bit secret keys, \( EM : x \mapsto E_k(x \oplus k_1) \oplus k_2 \).
Attack on the FX construction

Like the Even-Mansour cipher, $FX(x) \oplus E_k(x)$ admits $k_1$ as a period. Then Simon’s algorithm applied to $x \mapsto FX(x) \oplus E_k(x)$ retrieve $k_1$, we can find $k_2 = FX(x) \oplus E_k(x \oplus k_1)$. 

Q2 attack on the FX construction
Other quantum attacks

Other Simon-based attacks

- Other constructions have been broken (CBC-MAC, PMAC, GMAC, GCM, OCB,...)
- Q1 attacks (Offline-Simon algorithm)
- Linearization attacks
State of the art of quantum security proofs
Polynomial degree minimization

For a given quantum algorithm, build a family of oracles such that:

- The oracles are indexed by few integer variables (one or two in practice).
- The application of the quantum algorithm is a polynomial on the index.
- The family is “large”.

Proofs made with this technique

Grover’s search and BHT collision search are optimal.
Recording oracle

For a given attacking quantum algorithm,

- The queries made by the attacker can be recorded as a database (in superposition)
- The superposition can be examined to determine whether it differs from random or not
- We can deduce the advantage of attackers.

Proofs made with this technique

$LR_4$ is a quantum Pseudo-Random Function.
First Try
First try and attack

\[ S = S(L, R) \quad T = T(L, R) \]

**Figure**: EME construction
Modification of Simon’s algorithm

Simon's period finding

Given $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ a function which admits a period i.e, there exists $s \in \{0, 1\}^n$ such that for all $x, y$ in $\{0, 1\}^n$, $f(x) = f(y) \iff x = y$ or $x = y \oplus s$.

Simon's period finding find $s$ in $O(n^3)$ computations.

Quick description of Simon’s algorithm

Simon’s algorithm starts by making $|\phi_f\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$.

Then it measures the second register and we get a superposition $\frac{1}{\sqrt{2}} (|x_0\rangle + |x_0 + s\rangle)$.

By applying an Hadamard gate, we get a $y$ such that $y \cdot s = 0$. 
Modification of Simon’s algorithm

Relaxation of Simon’s algorithm
It is not necessary to have access to $f$ if we have the superposition $|\phi_f\rangle$. We can also take a superposition on a restricted space $A$, $|\phi_{f,A}\rangle = \frac{1}{\sqrt{|A|}} \sum_{x\in A} |x\rangle |f(x)\rangle$.

Practical use
By considering the null function restricted to $A = \{(x, y)|f(x) = f(y)\}$, we can search for $s$ such that $f(x) = f(y) \Rightarrow f(x \oplus s) = f(y \oplus s)$ in $O(n2^{n/3})$ operations.
Q2 Attack

$S(L_0, R_0) = S(L_1, R_1)$ is equivalent to

$$f(E_1(L_0) \oplus E_2(R_0)) \oplus f(E_1(L_1) \oplus E_2(R_1)) = E_1(L_0) \oplus E_1(L_1).$$

If we guess the key of $E_2$, we can reverse it. Then, by considering the function

$$F : (b, R) \mapsto S(L_b, E_2^{-1}(R)),$$

we can recover $s = (1, E_1(L_0) \oplus E_1(L_1))$ in time $\tilde{O}(2^{k/2} + 2^{n/3})$. 
Our attack impacts all mixing layers of the form:

\[ M(x, y) = \Pi_2(f(\Pi_1(x, y)), x, y) \]

with \( \Pi_1 \) and \( \Pi_2 \) two linear functions. The period is different but the procedure is the same.
New Construction
Figure: QuEME construction
**Security proofs**

<table>
<thead>
<tr>
<th>Classical security</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuEME is proven to be secure up to $2^n$ classical queries.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quantum security</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuEME is proven to be secure up to $2^{n/6}$ quantum queries.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Security claim</th>
</tr>
</thead>
<tbody>
<tr>
<td>We claim QuEME to be secure up to $2^n$ quantum queries.</td>
</tr>
</tbody>
</table>
Instantiation : Double-AES
Key extension

$K = (k_1 \Vert k_2)$

$k_3 = k_1 \oplus k_2$

$k_4 = k_1 \oplus (k_2 \ll 1)$

Block ciphers

Taking the same block cipher would induce weak keys by having the same permutation for multiple blocks.
Quick description of AES

The state of AES is composed of elements of $\mathbb{F}_{256}$ organized in a $4 \times 4$ matrix:

$$
\begin{bmatrix}
\alpha_0 & \alpha_4 & \alpha_8 & \alpha_{12} \\
\alpha_1 & \alpha_5 & \alpha_9 & \alpha_{13} \\
\alpha_2 & \alpha_6 & \alpha_{10} & \alpha_{14} \\
\alpha_3 & \alpha_7 & \alpha_{11} & \alpha_{15}
\end{bmatrix}
$$
Modification of AES-128

Composition of a round of AES

AES-128 is composed of 10 rounds which are composed of:

- AddKey xors the state with the round key;
- SubBytes which applies the AES Sbox on all individual elements $\alpha_i$;
- ShiftRows which shifts the $i$-th row by $i$ position;
- MixColumns which multiplies each column by a fixed matrix.

The last round omits the Mixcolumns operation and applies one extra AddKey.
## Modification of AES-128

### AES key schedule

\[ K = K_0 = k_0 \| k_1 \| k_2 \| k_3 \text{ and } K_{i+1} = (k_{4i+4} \| k_{4i+5} \| k_{4i+6} \| k_{4i+7}) \text{ for } i \text{ from } 0 \text{ to } 9 \]

\[
\begin{align*}
k_{4i+4} &= \text{SubWord} \left( \text{RotWord}(k_{4i+3}) \right) \oplus k_{4i} \oplus r_{ci} \\
k_{4i+5} &= k_{4i+4} \oplus k_{4i+1} \\
k_{4i+6} &= k_{4i+5} \oplus k_{4i+2} \\
k_{4i+7} &= k_{4i+6} \oplus k_{4i+3}
\end{align*}
\]

with \( r_{ci} = \begin{pmatrix} X^i \mod X^8 + X^4 + X^3 + X + 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \)
Modification of AES-128

We modify the round constant for each block $E_j$ for $j \in \{1, 2, 3, 4\}$:

$$rc_{i,j} = \begin{pmatrix} X^i \mod X^8 + X^4 + X^3 + X + 1 \\ j \\ 0 \\ 0 \end{pmatrix}$$
Instantiation

**Double-AES**
We propose Double-AES with 10 rounds of AES for each blocks and claim a unified security claim for both classical and quantum attackers with $T^2/p > 2^{224}$ where $T$ is the time complexity of the attack and $p$ is the probability of success.

**Double-AES-7**
We conjecture Double-AES-7 (with 7 rounds of AES for each blocks) to also provide the target security.

**Double-AES-6-MC**
We conjecture Double-AES-6-MC (with 6 rounds of AES for each blocks but the last round include a MixColumn operation) to also provide the target security.
Best attacks on Double-AES and preliminary
Cancelation of the first middle round

If we introduce a difference in $L$ but not in $R$, on the first round of the middle encryption, the difference in the plaintext and the first key cancel each other.
For a pair of plaintexts \((L_1 \| R), (L_2 \| R)\), we get a ciphertext pair \((S_1 \| T_1), (S_2 \| T_2)\).

From a guess of \(k_3\), we get \(E_3^{-1}(S_1) \oplus E_3^{-1}(S_2)\).

From a guess of \(k_1\), we get \(E_1(L_1) \oplus E_1(L_2)\) and with 3 bytes of \(E_2(R)\), we get 16 possibilities for a byte of \(E_3^{-1}(S_1) \oplus E_3^{-1}(S_2)\).
Using more pairs

One pair can filter one guess of \( (k_1, k_3, E_2(R)) \) out of 16. Then by using 70 pairs instead of one, we do not need to guess more elements and filter out every wrong guess.

Complexity

This attack takes \( 2^{107.5} \) time and memory.
From a guess of $k_1$, we can build a set of plaintext $(L_i, R)$ such that $E_1(L_i)$ is constant except the bytes 0 and 8 that take the value $i$. We then use the square property of 3-round AES. We recover the balanced byte from the ciphertext and the guess of 5 bytes of $k_3$. 

![Diagram of AES encryption process]

1. $E_1$ takes input $L_i$ and produces $E_1(L_i)$.
2. $E$ takes $E_1(L_i)$ and produces $E$.
3. $E_2$ takes $E$ and produces $E_2$.
4. $E_3$ takes $E_2$ and produces $E_3$.
5. $E_4$ takes $E_3$ and produces $T$.
Attack on X-3-3

Figure: Square-like property on the middle part.
Attack on X-3-3

Figure: Recovery on the bottom part.
Using more sets

One set can filter one guess of out of 256. Then by using 21 sets instead of one, we do not need to guess more elements and filter out every wrong guess.

Complexity

This attack takes $2^{96.5}$ time and data.
Conclusion
Conclusion

- We extend the properties exploitable for a Simon-based attack and apply it on EME.
- We develop QuEME, a quantum-safe construction for block ciphers.
- We propose Double-AES, a new block-cipher ready for you to experiment.
Future work

- Other ways to extend the properties retrievable by Simon’s algorithm.
- Application of our quantum attack on $LR_5$
- Further cryptanalysis of Double-AES.