Quantum Linear Key-recovery Attacks Using the QFT

André Schrottenloher

Inria

Motivation

A block cipher $E_K : \mathbb{F}_2^n \to \mathbb{F}_2^n$

$$x \longrightarrow \boxed{E_K} \longrightarrow E_K(x)$$

Given access to the black-box E_K

- classical brute-force search of the key in $2^{|K|}$ evaluations of E
- quantum brute-force (Grover's search) in $\simeq 2^{|K|/2}$ evaluations of E

Valid **key-recovery attacks** must be below these bounds: **faster than brute force** (classically) or **faster than Grover** (quantumly).

Motivation (ctd.)

- **Linear cryptanalysis** is a powerful cryptanalysis technique
- **Advanced linear (key-recovery) attacks** use the FFT
- Many quantum algorithms use the **QFT** (quantum Fourier transform)

Is there a way to use the QFT in **quantum linear attacks**?

Previous & concurrent work

- **[KLLN16]**: quantum linear cryptanalysis using Grover's algorithm
- **[H22]**: using the QFT in some distinguishing attacks

Is there a way to use the QFT in quantum linear **key-recovery** attacks?

Kaplan, Leurent, Leverrier, Naya-Plasencia, "Quantum differential and linear cryptanalysis", ToSC 2016

Hosoyamada, "Quantum speed-up for multidimensional (zero correlation) linear and integral distinguishers", ePrint 2022

Outline

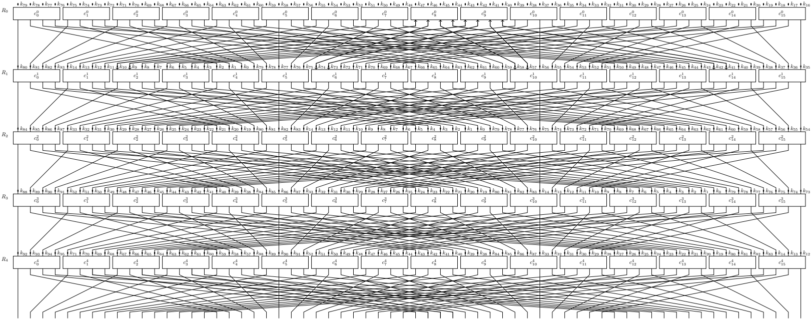
Linear Cryptanalysis

Linear cryptanalysis

- Exploits a **linear approximation** of E: choice of $(\alpha, \beta) \in \mathbb{F}_2^n$ such that $\alpha \cdot x \oplus \beta \cdot E(x)$ is biased
- The quality of an approximation (α, β) is related to its **ELP**
- If ELP is large enough, we have a **linear distinguisher**

Example

The Present block cipher:



Admits many one-bit approximations of the form:

$$\text{bit } i \text{ of input} = \text{bit } j \text{ of output } (\oplus 1) \text{ with probability } \tfrac{1}{2} + \varepsilon$$

Example (ctd.)

The **correlation** of approximation α, β:

$$\mathrm{cor} := \frac{1}{2^{n}} \sum_{x} (-1)^{\alpha \cdot x \oplus \beta \cdot E(x)}$$

- gets closer to ± 1 if the approximation is good
- $\simeq 2^{-n/2}$ for a random permutation,

... but around $\sqrt{\mathrm{ELP}} \simeq 2^{-30}$ for 22 rounds of Present.

We can **distinguish** 22-round Present from a random permutation.

The distinguisher: call the black-box and compute the correlation.

Last-rounds attack

- Take a block cipher with approximation α, β (e.g. 22-round Present)
- Append a couple **last rounds** with unknown subkey k

$\Longrightarrow$ search exhaustively for k using the distinguisher

$$x \longrightarrow \boxed{\underbrace{E_M}_{\text{Approximation } \alpha, \beta}} \longrightarrow \boxed{\underbrace{F_k}_{\text{Last rounds}}} \longrightarrow E_K(x) = F_k \circ E_M(x)$$

$\Uparrow$ We don't need to know the key

$\Uparrow$ We're guessing the key

Matsui, "Linear cryptanalysis method for DES cipher", EUROCRYPT 1993

Last-rounds attack

Using the whole codebook, time about $\mathcal{O}\left(2^{n} \times 2^{|k|}\right)$:

1. For each guess z of the subkey k, compute the **experimental correlation**:

$$\widehat{\mathrm{cor}}(z) := \frac{1}{2^{n}} \sum_{x} (-1)^{\alpha \cdot x} (-1)^{\beta \cdot F_z^{-1}(E_K(x))} \quad .$$

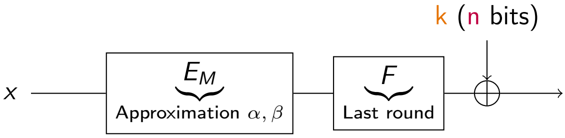
2. The good subkey k has (one of) the highest $|\widehat{\mathrm{cor}}(z)|$

Statistics

- Right subkey: $|\widehat{\mathrm{cor}}(k)|$ is around $\sqrt{\mathrm{ELP}}$
- Wrong subkey: $|\widehat{\mathrm{cor}}(z)|$ is around $2^{-n/2}$

FFT-based Linear Attacks

Improvement with the FFT



$$E_K(x) = k \oplus F \circ E_M(x)$$

$$\widehat{\mathrm{cor}}(z) = \frac{1}{2^n} \sum_x (-1)^{\alpha \cdot x} (-1)^{\beta \cdot F^{-1}(z \oplus E_K(x))} = \frac{1}{2^n} \sum_x (-1)^{\alpha \cdot E_K^{-1}(x)} (-1)^{\beta \cdot F^{-1}(z \oplus x)}$$

Principle: accelerate the computation of all $\widehat{\mathrm{cor}}(z)$.

Collard, Standaert, Quisquater, "Improving the time complexity of Matsui's linear cryptanalysis." ICISC 2007

Correlations via a discrete convolution

Introduce two functions f, g:

$$\begin{cases} f, g \ : \mathbb{F}_2^{n} \to \{-1, 1\} \\ f(x) := (-1)^{\alpha \cdot E_{K}^{-1}(x)} \\ g(x) := (-1)^{\beta \cdot F^{-1}(x)} \end{cases}$$

$$\widehat{\mathrm{cor}}(z) = \frac{1}{2^{n}} \sum_{x} f(x) g(z \oplus x) := \frac{1}{2^{n}} (f \star g)(z)$$

Compute all $\widehat{\mathrm{cor}}(z) \iff$ compute the discrete convolution of f and g

The convolution theorem

The Walsh-Hadamard transform of f:

$$\widehat{f}(y) = \sum_{x} (-1)^{x \cdot y} f(x)$$

"Under a Walsh-Hadamard transform, the convolution corresponds to a pointwise product"

$$(f \star g) = \frac{1}{2^{n}} \widehat{\widehat{f} \cdot \widehat{g}}$$

One computes $\widehat{f}$ via a Fast Walsh-Hadamard transform (FWHT) in time $\mathcal{O}(n2^{n})$.

Correlations via FWHT

1. Evaluate $f(x) = (-1)^{\alpha \cdot E_K^{-1}(x)} \to \mathcal{O}(2^n)$
2. Evaluate $g(x) = (-1)^{\beta \cdot F^{-1}(x)} \to \mathcal{O}(2^n)$
3. Compute $\widehat{f}, \widehat{g}$ via FWHT $\to \mathcal{O}(n2^n)$
4. Do a pointwise product $\to \mathcal{O}(2^n)$
5. Compute FWHT again $\to \mathcal{O}(n2^n)$
6. Find the highest outputs $\implies$ candidate keys

Improved time: $\mathcal{O}(n2^n)$ instead of $\mathcal{O}\left(2^n \times 2^{|k|}\right) = \mathcal{O}(2^n \times 2^n)$.

Correlation State and Applications

Quantum cryptanalysis basics

- The state of a quantum system is a **superposition**

$$\sum_{x \in \mathbb{F}_2^{\mathbf{n}}} \alpha_x \left|x\right\rangle \text{ with } \sum_x \left|\alpha_x\right|^2 = 1$$

- The amplitudes α_x are **not** immediately exploitable

Think of this as a probability distribution of outputs, where $|\alpha_x|^2$ is the probability to measure α_x.

Quantum search

Quantum search

Given a **setup** algorithm that produces: $\sum_x \alpha_x |x\rangle |\mathrm{flag}(x)\rangle$, we find x_g such that $\mathrm{flag}(x_g) = 1$ in $\mathcal{O}\left(\frac{1}{|\alpha_{x_g}|}\right)$ calls.

Grover's exhaustive search:

1. take a key at random $\rightarrow \frac{1}{\sqrt{2^{|K|}}} \sum_z |z\rangle$
2. check if it's good $\rightarrow \frac{1}{\sqrt{2^{|K|}}} \sum_z |z\rangle |\mathrm{flag}\rangle$
3. use the quantum search black-box $\rightarrow$ time $\simeq \sqrt{2^{|K|}}$

Quantum Fourier Transform

Computing a Walsh-Hadamard transform on the amplitudes is easy

If $f : \{0,1\}^n \to \{-1,1\}$ is a function:

$$\frac{1}{2^{n/2}} \sum_x f(x) |x\rangle \xmapsto{H} \frac{1}{2^n} \sum_y \underbrace{\left(\sum_x (-1)^{x \cdot y} f(x) \right)}_{:= \widehat{f}(y)} |y\rangle$$

A typical thing to do: "sample at random"

$$|0\rangle \xmapsto{H} \frac{1}{2^{n/2}} \sum_x |x\rangle$$

The hero we need: a "correlation state"

$$|\mathsf{Cor}\rangle := \sum_{z} \widehat{\mathrm{cor}}(z)\,|z\rangle$$

Computing $|\mathrm{Cor}\rangle$

Recall the two functions f, g:

$$\begin{cases} f(x) := (-1)^{\alpha \cdot E_{K}^{-1}(x)} \\ g(x) := (-1)^{\beta \cdot F^{-1}(x)} \end{cases}$$

and

$$\widehat{\mathrm{cor}}(z) = \frac{1}{2^{n}} (f \star g)(z) = \frac{1}{2^{2n}} \widehat{\widehat{f} \cdot \widehat{g}}$$

We need:

$$\frac{1}{2^{2n}} \sum_{z} \widehat{\widehat{f} \cdot \widehat{g}}(z) \, |z\rangle = H \left(\frac{1}{2^{3n/2}} \underbrace{\sum_{y} \widehat{f}(y) \widehat{g}(y) \, |y\rangle}_{\text{So let's compute this}} \right)$$

Computing $|\mathrm{Cor}\rangle$ (ctd.)

1. Compute f in the amplitude (doable)

$$\sum_x f(x)\,|x\rangle$$

2. Apply H (easy)

$$\sum_y \widehat{f}(y)\,|y\rangle$$

3. Compute $\widehat{g}$ (**not easy**)

$$\sum_y \widehat{f}(y)\,|y\rangle\,|\widehat{g}(y)\rangle$$

4. Transfer $\widehat{g}(y)$ into the amplitude (doable)

$$\sum_y \widehat{f}(y)\widehat{g}(y)\,|y\rangle$$

Computing $|\mathrm{Cor}\rangle$ (ctd.)

There is a quantum algorithm that (on empty input $|0\rangle$) returns $|\mathrm{Cor}\rangle$.

The time complexity is dominated by:

- (a few) queries to E_K (to compute f)
- (a few) computations of $\widehat{g}$

Using the correlation state

Classical case

- We compute all $\widehat{\mathrm{cor}}(z)$
- We find the biggest one(s)

Quantum case

- We can compute $|\mathrm{Cor}\rangle = \sum_z \widehat{\mathrm{cor}}(z)\,|z\rangle$
- We **do not** have access to the values

$|\mathrm{Cor}\rangle$ is a superposition of subkey guesses where **the good guess has a higher amplitude**

Idea: use $|\mathrm{Cor}\rangle$ as a **shortcut** in an exhaustive key search.

Attack algorithm

Recall Grover's search:

1. take a key at random $\rightarrow \frac{1}{\sqrt{2^{|\mathsf{K}|}}} \sum_z |z\rangle$
2. check if it's good $\rightarrow \frac{1}{\sqrt{2^{|\mathsf{K}|}}} \sum_z |z\rangle |\text{flag}\rangle$
3. use the quantum search black-box $\rightarrow$ time $\simeq \sqrt{2^{|\mathsf{K}|}}$

Instead:

1. start from $|\text{Cor}\rangle \rightarrow \sum_z \widehat{\text{cor}}(z) |z\rangle$ **bigger on** k
2. check if the key guess is good
3. use the quantum search black-box $\rightarrow$ time **smaller than** $\sqrt{2^{|\mathsf{K}|}}$

Quantum - classical comparison

Classical cryptanalysis only needs to distinguish.

$\implies$ **extremely small** correlations are used

The speedup here depends directly on the correlation

$\implies$ we would like bigger correlations!

Conclusion

- Using the QFT to accelerate a **statistical** attack
- Still few (working) applications so far

Open question:

- Most issues would be solved if we had an efficient algorithm to find the largest correlation in $|\mathrm{Cor}\rangle$
- However, if $|\mathrm{Cor}\rangle$ is produced as a black-box, this seems very difficult

Report: ePrint 2023/184

Thank you!