IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2022, No. 1, pp. 183–211. DOI:10.46586/tosc.v2022.i1.183-211

Quantum Period Finding is Compression Robust

Alexander May and Lars Schlieper

Faculty of Computer Science
Ruhr University Bochum, Bochum, Germany
{alex.may,lars.schlieper}@rub.de

Abstract. We study quantum period finding algorithms such as Simon and Shor (and its variant Ekerå-Håstad). For a periodic function f these algorithms produce – via some quantum embedding of f – a quantum superposition $\sum_x |x\rangle |f(x)\rangle$, which requires a certain amount of output qubits that represent $|f(x)\rangle$. We show that one can lower this amount to a single output qubit by hashing f down to a single bit in an oracle setting.
Namely, we replace the embedding of f in quantum period finding circuits by oracle access to several embeddings of hashed versions of f. We show that on expectation this modification only doubles the required amount of quantum measurements, while significantly reducing the total number of qubits. For example, for Simon's algorithm that finds periods in $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ our hashing technique reduces the required output qubits from n down to 1, and therefore the total amount of qubits from $2n$ to $n + 1$. We also show that Simon's algorithm admits real world applications with only $n + 1$ qubits by giving a concrete realization of a hashed version of the cryptographic Even-Mansour construction. Moreover, for a variant of Simon's algorithm on Even-Mansour that requires only classical queries to Even-Mansour we save a factor of (roughly) 4 in the qubits.
Our oracle-based hashed version of the Ekerå-Håstad algorithm for factoring n-bit RSA reduces the required qubits from $(\frac{3}{2} + o(1))n$ down to $(\frac{1}{2} + o(1))n$.

Keywords: Quantum period finding · Fourier transform · Simon · Shor · cryptographic applications

1 Introduction

Throughout this paper, we consider only logical qubits that are error-free. Although there is steady progress in constructing larger quantum computers, within the next years the number of qubits seems to be too limited for tackling problems of interesting size, e.g. for period finding applications in cryptography [KM12, KLLN16, LM17, SS17, RNSL17, HRS17, RS18].

Shor's algorithm [Sho94] for polynomial time factorization of n-bit numbers computes a superposition $\sum_x |x\rangle |f(x)\rangle$ with $2n$ *input qubits* representing the input $|x\rangle$ to f and n *output qubits* representing the output $|f(x)\rangle$ of the function.

However, it may not be necessary to implement a full-fledged $3n$-qubit Shor algorithm in order to factor numbers or compute discrete logarithms. Quantum computers with a very limited number of qubits might still serve as a powerful oracle that assists us in speeding up classical computations. For instance, Bernstein, Biasse and Mosca [BBM17] developed an algorithm that factors n-bit numbers with the help of only a sublinear amount of $n^{\frac{2}{3}}$ qubits in subexponential time that is (slightly) faster than the currently best known purely classical factorization algorithm.

Several other algorithms saved on the number of qubits in Shor's algorithm by shifting some more work into a classical post-processing, while – in contrast to [BBM17] – still

Received: 2021-09-01 Revised: 2021-11-23 Accepted: 2022-01-23 Published: 2022-03-11

preserving polynomial run time. Interestingly, all these algorithms concentrate on reducing the input qubits, while keeping n output qubits. Seifert [Sei01] showed that – using for the classical post-process simultaneous Diophantine approximations instead of continued fractions – the number of input qubits can be reduced from $2n$ to $(1+o(1))n$. For n-bit RSA numbers, which are a product of two $n/2$-bit primes, Ekerå and Håstad [EH17] reduced the number of input qubits down to $(\frac{1}{2}+o(1))n$, using some variant of the Hidden Number Problem [BV97] in the post-process. Thus, the Ekerå-Håstad version of Shor's algorithm factors n-bit RSA with a total of $(\frac{3}{2}+o(1))n$ qubits.

Mosca and Ekert [ME98] showed that one can reduce the number of input qubits even down to a single one, at the cost of an increased depth of Shor's quantum circuit.

Our contribution. We hash $f(x)$ in the output qubits down to t qubits, where t can be as small as 1. This can be realized using quantum embeddings of $h \circ f$ for different hash functions h, for which we assume oracle access. Our basic observation is that hashing preserves the periodicity of f. Namely, if $f(x) = f(x+s)$ for some period s and all inputs x then also

$$h(f(x)) = h(f(x+s)) \text{ for the period } s \text{ and all inputs } x.$$

The drawback of hashing is that h certainly introduces many more undesirable collisions $h(f(x)) = h(f(x'))$ where x, x' are not a multiple of s apart. Surprisingly, even for 1-bit range hash functions this plethora of undesirable collisions does not at all affect the correctness of our hashed quantum period finding algorithms, and only insignificantly increases the required number of measurements. Depending on the implementation of $h \circ f$ we suffer from an increased circuit depth. In some of our applications in this paper we double the circuit depth, while in other applications the circuit depth grows linearly in n.

More precisely, concerning correctness we show that a replacement of f by some hashed version of f has the following effects.

Simon's algorithm: In the input qubits, we still measure only vectors y that are orthogonal to the period s. The amplitudes of all other inputs cancel out.

Shor's algorithm: Let the period be $d = 2^r$, and let us use $q > r$ input qubits. Then we still measure in the input qubits only numbers y that are multiples 2^{q-r}. The amplitudes of all other inputs cancel out. In the case of general (not only power of two) periods we measure all inputs $y \neq 0$ with exactly half the probability as without hashing.

Our correctness property immediately implies that the original post-processing in Simon's algorithm (Gaussian elimination) and in Shor's algorithm (e.g. continued fractions) can still be used in the hashed version of the algorithms for period recovery.

However, this does not automatically imply that we achieve similar runtimes. Namely, in the original algorithms of Simon and Shor we measure all y having a non-zero amplitude with a uniform probability distribution. In Simon's algorithm for some period $s \in \mathbb{F}_2^n$ we obtain each of the 2^{n-1} many $y \in \mathbb{F}_2^n$ orthogonal to s with probability $\frac{1}{2^{n-1}}$. In Shor's algorithm with period $d = 2^r$, we measure each of the d many possible multiples y of 2^{q-r} with probability $\frac{1}{d}$.

These uniform probability distributions are destroyed by moving to the hashed version of the algorithms. Since $h(f(x)) = h(f(x'))$ for $x \neq x'$ happens for universal 1-bit range hash functions with probability $\frac{1}{2}$, the undesirable collisions put a probability weight of (roughly) $\frac{1}{2}$ on measuring $y = 0$ in the input qubits.

This seems to be bad news, since neither in Simon's algorithm does the zero vector y provide information about s, nor does in Shor's algorithm the zero-multiple y of 2^{q-r} provide information about d. However as good news, we show that besides putting probability weight $\frac{1}{2}$ on $y = 0$, hashing does not destroy the probability distribution stemming from the amplitudes of quantum period finding algorithms. Namely, we show that for the whole class of quantum period finding circuits that we consider – including Simon, Shor (and its variant Ekerå-Håstad) – the following result holds: If the probability to measure y is $p(y)$ when using f, then we obtain probability $p(y)/2$ to measure y when using $h \circ f$, where the latter probability is taken over the random choice of h from a family of 1-bit range universal hash functions.

Put differently, if we condition on the event that we do *not* measure $y = 0$ in the input bits (which happens in roughly every second measurement) in both cases – using f itself or its hashed version $h \circ f$ – we obtain exactly the same probability distribution for the measurements of any $y \neq 0$. This implies that our hashing approach preserves not only the correctness but also the runtime analysis of any processing of the measured data in a classical post-process. Thus, at the cost of only twice as many quantum measurements we save all but one of the output qubits. More generally, we show that at the cost of $\frac{1}{1-2^{-t}}$-times more measurements we may compression to t output qubits.

In particular, we show that the original Simon algorithm [Sim94] — that recovers for a periodic function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ its period in time polynomial in n with expected $n+1$ measurements using $2n$ qubits — admits an *oracle-based* hashed version with expected $2(n+1)$ measurements using only $n+1$ qubits. Moreover, we show that this leads to an *explicit* (non-oracle, efficiently constructable) realization of the quantum Even-Mansour attack [KM12, KLLN16] with only $n+1$ qubits , as well as an *explicit* distinguishing attack on 3-round Feistel ciphers with a single output qubit. However, these attacks work only in a quite strong attack model in which we have quantum access to Even-Mansour/Feistel and their inverse functions. For the quantum attack [BHN+19] on Even-Mansour with only classical access to the cipher, called Offline-Simon, we provide an *explicit* hashed realization that saves even (roughly) a factor of 4 in the number of qubits.

The original Ekerå-Håstad version of Shor's algorithm that computes discrete logarithms d in some abelian group G in polynomial time using $(1 + o(1)) \log d + \log(|G|)$ qubits requires in its *oracle-based* hashed version only $(1 + o(1)) \log d$ qubits. Moreover, the Ekerå-Håstad algorithm computes the factorization of an RSA modulus $N = pq$ of bit-size n in time polynomial in n using $(\frac{3}{2} + o(1))n$ qubits, whereas our *oracle-based* hashed version reduces this to only $(\frac{1}{2} + o(1))n$ qubits. We leave it as an open problem whether there exist an *explicit* hashed Ekerå-Håstad realization. For Ekerå-Håstad, one has to compute hashed versions of the exponentiation function $f_{a,N} : x \mapsto a^x \bmod N$. Notice that it is of course *not sufficient* to compute $f_{a,N}$ first, and afterwards hash the result, since this would require qubits for representing the full range of $f_{a,N}$.

For proving our theorems, we make use of universal hash function families, but we believe that this requirement is merely a proof artefact that guards against pathological functions. In practice, we may usually relax the hash function requirement. We conjecture that often a single h should still work. Even choosing h simply as the projection of f to a single bit should work for most functions of interest. We believe that it is of theoretical and practical interest to study in more generality, which classes of f admit a memory-efficient computation of their hashed versions. Recently, a first *explicit* application of our hashed Shor algorithm was given by Bonnetain, Leurent, Naya-Plasencia and Schrottenloher [BLNS21] for the MAC Poly1305.

Our paper is organized as follows. In Section 3 we present our first main result that Simon's algorithm is compression robust. In Sections 4 and 5 we provide explicit hashed Simon realizations of Even-Mansour and Feistel. In Section 6, we show how to explicitly realize hashed Offline-Simon.

Subsequently, we generalize the hash concept to Shor's algorithm and a more general class of period-finding circuits. For didactic reasons, we study in Section 7 first the simple case of Shor's algorithm for periods that are a power of two. In Section 8, as our second main result we generalize to any quantum circuits that fall in our period finding class. As a consequence, in Section 8 we obtain a hashed version of Shor's algorithm with general periods, and in Section 9 a hashed version of Ekerå-Håstad.

2 Preliminaries

Let us first recall some quantum notation. The reversible *quantum embedding* of a classical function f is defined as

$$U_f : |x\rangle\,|y\rangle \mapsto |x\rangle\,|y + f(x)\rangle\ .$$

For any $x, y \in \mathbb{F}_2^n$ we write their inner product as $\langle x, y\rangle = \sum_{i=1}^n x_i y_i \bmod 2$. The 1-qubit *Hadamard gate* realizes the mapping $H_1 : |x\rangle \mapsto \frac{1}{\sqrt{2}} \sum_{y\in\mathbb{F}_2} (-1)^{\langle x,y\rangle}\,|y\rangle$. Its n-qubit version is defined as the n-fold tensor product $H_n = \bigotimes_{i=1}^n H_1$. The n-qubit *Quantum Fourier Transform* (QFT) is the mapping

$$\mathrm{QFT}_n : |x\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{y\in\mathbb{F}_{2^n}} e^{2\pi i \frac{x}{2^n} y}\,|y\rangle\ .$$

Notice that $\mathrm{QFT}_1 = H_1$.

Definition 1. A hash function family $\mathcal{H}_t := \{h : \mathcal{D} \to \{0,1\}^t\}$ is *universal* if for all $x, y \in \mathcal{D}$, $x \neq y$ we have

$$\mathbb{P}_{h\in\mathcal{H}_t}\left[h(x) = h(y)\right] = \frac{1}{2^t}\ .$$

Efficient instantiations of (homomorphic) universal hash function families exist, e.g.

$$\mathcal{H}_t = \{h_r : \mathbb{F}_2^n \to \mathbb{F}_2^t \mid r \in (\mathbb{F}_2^n)^t\ ,\ h_r(x) = (\langle x, r_1\rangle, \ldots, \langle x, r_t\rangle)\}\ . \tag{1}$$

It is easy to see that *strongly 2-universal* hash function families as defined in [MU05] are universal in the sense of Definition 1.

3 Hashed-Simon

Let us briefly recall Simon's original algorithm. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be periodic with period $s \in \mathbb{F}_2^n \setminus \{0^n\}$, that is $f(x) = f(x + s)$ for all $x \in \mathbb{F}_2^n$. We call f a *Simon function* if it defines a $(2:1)$-mapping, i.e.

$$f(x) = f(y) \Leftrightarrow (y = x) \text{ or } (y = x + s).$$

The use of Simon functions allows for a clean theoretical analysis, although Simon's algorithm works also for more general periodic functions as shown in [AMR07, CvD08, LM17]. For ease of notation, we restrict ourselves to Simon functions.

The *Simon circuit* $Q_f^{\textsc{Simon}}$ from Figure 1 uses n input and n output qubits for realizing the embedding of f. It can easily be shown that in the n input qubits we measure only $y \in \mathbb{F}_2^n$ such that $y \perp s$, i.e. $ys = 0$.

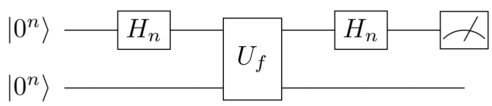


Figure 1: Quantum circuit $Q_f^{\textsc{Simon}}$

The Simon algorithm uses $Q_f^{\textsc{Simon}}$ until we have collected $n-1$ linearly independent vectors, from which we compute the unique vector s that is orthogonal to all of them using Gaussian elimination in time $\mathcal{O}(n^3)$.

Our Hashed-Simon (Algorithm 1) is identical to the Simon algorithm with the only difference that Q_f^{Simon} is replaced by $Q_{h\circ f}^{\text{Simon}}$, where in each iteration we instantiate $Q_{h\circ f}^{\text{Simon}}$ with some hash function h freshly drawn from a universal t-bit range hash function family $\mathcal{H}_t$. Notice that Simon can be considered as special case of Hashed-Simon, where we choose $t = n$ and the identity function $h = \text{id}$. This slightly abuses notation, since $\mathcal{H}_n = \{\text{id}\}$ is not universal. However, the following Lemma 1 holds without universality of $\mathcal{H}_n$. In Lemma 1 we show the correctness property of Hashed-Simon that by replacing Q_f^{Simon} with $Q_{h\circ f}^{\text{Simon}}$, we still measure only y orthogonal to s.

Algorithm 1: Hashed-Simon

Input : Simon function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$, universal $\mathcal{H}_t := \{h : \mathbb{F}_2^n \to \mathbb{F}_2^t\}$
Output : Period s of f

```
begin
    Set Y = ∅.
    repeat
        Run Q^SIMON_{h∘f} on |0^n⟩|0^t⟩ for some freshly chosen h ∈_R H_t.
        Let y be the measurement of the n input qubits.
        If y ∉ span(Y), then include y in Y.
    until Y contains n − 1 linear independent vectors
    Compute {s} = Y^⊥ \ {0^n} via Gaussian elimination.
    return s.
end
```

Lemma 1 (Orthogonality)**.** *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a Simon function with period s. Let $h : \mathbb{F}_2^n \to \mathbb{F}_2^t$ and $f_h = h \circ f : \mathbb{F}_2^n \to \mathbb{F}_2^t$. Let us apply $Q_{h\circ f}^{\text{Simon}}$ on $|0^n\rangle |0^t\rangle$. Then we obtain superposition*

$$\sum_{f_h(x)\in Im(f_h)} \sum_{\substack{y\in\mathbb{F}_2^n \\ y\perp s}} w_{y,f_h(x)} |y\rangle |f_h(x)\rangle \quad \textit{where } w_{y,f_h(x)} = \frac{1}{2^n} \sum_{x\in f_h^{-1}(f_h(x))} (-1)^{\langle x,y\rangle} .$$

Proof. Since f is a Simon function we have $f(x) = f(x+s)$ and therefore $f_h(x) = f_h(x+s)$. This implies $x \in f_h^{-1}(z)$ iff $x + s \in f_h^{-1}(z)$.

An application of $Q_{h\circ f}$ on input $|0^n\rangle |0^t\rangle$ yields for the operations $H_n \otimes I_t$ and U_{f_h}

$$|0^n\rangle |0^t\rangle \overset{H_n\otimes I_t}{\to} \frac{1}{2^{n/2}} \sum_{x\in\mathbb{F}_2^n} |x\rangle |0^t\rangle \overset{U_{f_h}}{\to} \frac{1}{2^{n/2}} \sum_{x\in\mathbb{F}_2^n} |x\rangle |f_h(x)\rangle .$$

Using $x \in f_h^{-1}(z)$ iff $x + s \in f_h^{-1}(z)$, we obtain

$$\frac{1}{2^{n/2}} \sum_{x\in\mathbb{F}_2^n} |x\rangle |f_h(x)\rangle = \frac{1}{2^{n/2}} \sum_{x\in\mathbb{F}_2^n} \frac{1}{2} (|x\rangle + |x+s\rangle) |f_h(x)\rangle$$

An application of H_n now yields

$$\begin{aligned} &\frac{1}{2^n} \sum_{x\in\mathbb{F}_2^n} \sum_{y\in\mathbb{F}_2^n} \frac{1}{2} \left((-1)^{\langle x,y\rangle} + (-1)^{\langle x+s,y\rangle}\right) |y\rangle |f_h(x)\rangle \\ =&\frac{1}{2^n} \sum_{x\in\mathbb{F}_2^n} \sum_{y\in\mathbb{F}_2^n} \frac{1}{2}(-1)^{\langle x,y\rangle} \left(1 + (-1)^{\langle s,y\rangle}\right) |y\rangle |f_h(x)\rangle \\ =&\frac{1}{2^n} \sum_{x\in\mathbb{F}_2^n} \sum_{\substack{y\in\mathbb{F}_2^n \\ y\perp s}} (-1)^{\langle x,y\rangle} |y\rangle |f_h(x)\rangle \ . \end{aligned}$$

The statement of the lemma follows. $\square$

From Lemma 1's superposition

$$\sum_{f_h(x)\in \mathrm{Im}(f_h)} \sum_{\substack{y\in\mathbb{F}_2^n\\ y\perp s}} \underbrace{\frac{1}{2^n} \sum_{x\in f_h^{-1}(f_h(x))} (-1)^{\langle x,y\rangle}}_{w_{y,f_h(x)}} |y\rangle\, |f_h(x)\rangle \tag{2}$$

we see that only $y\in\mathbb{F}_2^n$ with $y\perp s$ have a non-vanishing amplitude $w_{y,f_h(x)}$.

Assume that we measure some fixed $z = f_h(x)\in\{0,1\}^t$ in the output qubits. Then an easy calculation shows that Equation (2) collapses to

$$\sum_{\substack{y\in\mathbb{F}_2^n\\ y\perp s}} \underbrace{\frac{1}{(2^n\cdot|f_h^{-1}(z)|)^{1/2}} \sum_{x\in f_h^{-1}(z)} (-1)^{\langle x,y\rangle}}_{w_{y,z}} |y\rangle\,|z\rangle\ . \tag{3}$$

Recall that Lemma 1 contains the analysis of Simon's original algorithm as the special case $h = \mathrm{id}$. In this case, we know that by the definition of a Simon function $|f_h^{-1}(z)| = 2$ for all z and $\sum_{x\in f_h^{-1}(z)}(-1)^{\langle x,y\rangle}\in\{\pm 2\}$. Thus, all $y\perp s$ have amplitude $\pm\frac{1}{2^{(n-1)/2}}$. This means that a measurement yields the uniform distribution over all $y\perp s$.

The following lemma will be useful, when we analyze superpositions over all z.

Lemma 2. *Let $f:\mathbb{F}_2^n\mapsto\mathbb{F}_2^\ell$ be a Simon function. Then $\sum_{z\in\mathbb{F}_2^\ell} w_{y,z} = 0$ for all $y\neq 0$.*

Proof. Fix $y\neq 0^n$. If $y\not\perp s$ then all $w_{y,z}=0$ and thus the claim follows. Hence, in the following let $y\perp s$. If $z\notin f(\mathbb{F}_2^n)$ then $w_{y,z}=0$. Therefore

$$\sum_{z\in\mathbb{F}_2^\ell} w_{y,z} = \sum_{z\in f(\mathbb{F}_2^n)} w_{y,z}.$$

Since f is a Simon function, f is a (2:1)-mapping. Thus

$$\sum_{z\in f(\mathbb{F}_2^n)} w_{y,z} = \frac{1}{2}\sum_{x\in\mathbb{F}_2^n} w_{y,f(x)}.$$

Using the definition of $w_{y,f(x)}$ in Eq. (2) with $h=\mathrm{id}$ yields

$$\frac{1}{2}\sum_{x\in\mathbb{F}_2^n} w_{y,f(x)} = \frac{1}{2^{n+1}}\sum_{x\in\mathbb{F}_2^n}(-1)^{\langle x,y\rangle}\ .$$

Since for $y\neq 0$ we have $\sum_{x\in\mathbb{F}_2^n}(-1)^{\langle x,y\rangle}=0$, the claim follows. $\square$

Let us now develop some intuition for the amplitudes $w_{y,z}$ in Eq. (3) for 1-bit range hash functions $h:\mathbb{F}_2^n\to\mathbb{F}_2$. We expect that $|f_h^{-1}(z)|\approx 2^{n-1}$. We first look at the amplitude $w_{0^n,z}$ of $|y\rangle = |0^n\rangle$. Since for all $x\in\mathbb{F}_2^n$ we have $(-1)^{\langle x,y\rangle}=1$, the amplitude of $|0^n\rangle$ adds up to $w_{0^n,z} = \left(\frac{|f_h^{-1}(z)|}{2^n}\right)^{\frac{1}{2}} \approx \frac{1}{\sqrt{2}}$. Hence, we expect to measure the zero-vector 0^n with probability approximately $\frac{1}{2}$. This seems to be bad news, since the zero-vector is the only one orthogonal to s that does not provide any information about s.

However, we show that all $y\perp s$ with $y\neq 0^n$ still appear with significant amplitude. Intuitively, $\sum_{x\in f_h^{-1}(z)}(-1)^{\langle x,y\rangle}$ describes for $y\neq 0^n$ a random walk with $|f_h^{-1}(z)|$ steps.

Thus, this term should contribute on expectation roughly $|f_h^{-1}(z)|^{\frac{1}{2}}$ to the amplitude of $|y\rangle$. So we expect for all $y \perp s$ with $y \neq 0^n$ an amplitude of

$$w_{y,z} = \frac{\sum_{x \in f_h^{-1}(z)} (-1)^{\langle x,y \rangle}}{(2^n \cdot |f_h^{-1}(z)|)^{1/2}} \approx \frac{1}{2^{n/2}}.$$

This in turn implies that conditioned on the event that we do not measure 0^n (which happens with probability roughly $\frac{1}{2}$), we still obtain the uniform distribution over all remaining $y \perp s$.

We make our intuition formal in the following theorem.

Theorem 1. *Let $\mathcal{H}_t = \{h : \mathbb{F}_2^n \to \mathbb{F}_2^t\}$ be universal, and let f be a Simon function with period s. Then we measure in Algorithm* HASHED-SIMON *in the first n qubits any $y \perp s, y \neq 0$ with probability $\frac{1-2^{-t}}{2^{n-1}}$, where the probability is taken over the random choice of $h \in \mathcal{H}_t$.*

Proof. From Lemma 1 in the case $h = \mathrm{id}$, we conclude that SIMON gives us a superposition

$$\sum_{f(x) \in \mathrm{Im}(f)} \sum_{\substack{y \in \mathbb{F}_2^n \\ y \perp s}} w_{y,f(x)} |y\rangle |f(x)\rangle \text{ where } w_{y,f(x)} = \frac{1}{2^n} \sum_{x \in f^{-1}(f(x))} (-1)^{\langle x,y \rangle} .$$

For ease of notation let us denote $z = f(x)$. In particular for $z \notin \mathrm{Im}(f)$ we have $w_{y,z} = 0$. We measure any y with probability $\sum_{z \in \mathbb{F}_2^n} |w_{y,z}|^2$.

Let $p(y)$ denote the probability to measure y in the first n qubits. Since $w_{y,z} \in \mathbb{R}$, we obtain $|w_{y,z}|^2 = w_{y,z}^2$ and hence the identity

$$p(y) = \sum_{z \in \mathbb{F}_2^n} w_{y,z}^2 = \frac{1}{2^{n-1}} . \tag{4}$$

Let us now look at HASHED-SIMON with a t-bit range hash function $h \in \mathcal{H}_t$. From Lemma 1 we get

$$|\Phi_h\rangle = \sum_{z' \in \mathbb{F}_2^t} \sum_{\substack{y \in \mathbb{F}_2^n \\ y \perp s}} w_{y,z'} |y\rangle |z'\rangle .$$

With respect to the amplitudes $w_{y,z}$ of SIMON the superposition of HASHED-SIMON can be written as

$$|\Phi_h\rangle = \sum_{z' \in \mathbb{F}_2^t} \sum_{\substack{y \in \mathbb{F}_2^n \\ y \perp s}} \left(\sum_{z \in h^{-1}(z')} w_{y,z} \right) |y\rangle |z'\rangle ,$$

where

$$\sum_{z \in h^{-1}(z')} w_{y,z} = \sum_{z \in h^{-1}(z')} \frac{1}{2^n} \sum_{x \in f^{-1}(z)} (-1)^{\langle x,y \rangle} = \frac{1}{2^n} \sum_{x \in f_h^{-1}(z')} (-1)^{\langle x,y \rangle} = w_{y,z'} .$$

Let us denote by $p_h(y) = \mathbb{P}_{h \in \mathcal{H}_t}[y]$ the probability that we measure y in the first n qubits when applying $Q_{h \circ f}^{\textsc{Simon}}$. Our goal is to show that $p_h(y) = (1 - 2^{-t}) \cdot p(y) = \frac{1-2^{-t}}{2^{n-1}}$.

For some $h \in \mathcal{H}_t$ we denote $I_{h,z'} = \{z \in f(\mathbb{F}_2^n) \mid h(z) = z'\}$. Since $\dot{\bigcup}_{z' \in \mathbb{F}_2^t} I_{h,z'} = f(\mathbb{F}_2^n)$ and $w_{y,z} \in \mathbb{R}$, HASHED-SIMON yields

$$p_h(y) = \frac{1}{|\mathcal{H}_t|} \sum_{h \in \mathcal{H}_t} \sum_{z' \in \mathbb{F}_2^t} \left| \sum_{z \in I_{h,z'}} w_{y,z} \right|^2$$

$$= \frac{1}{|\mathcal{H}_t|} \sum_{h \in \mathcal{H}_t} \sum_{z' \in \mathbb{F}_2^t} \left(\sum_{z \in I_{h,z'}} w_{y,z} \right)^2 . \tag{5}$$

In Eq. (5) we obtain a cross-product $w_{y,z_1} w_{y,z_2}$ for $z_1 \neq z_2$ iff z_1, z_2 are in the same set $I_{h,z'}$, i.e. iff $h(z_1) = h(z_2)$. Using Definition 1 of a universal hash function family, we obtain $\mathbb{P}_{h \in \mathcal{H}_t}[h(z_1) = h(z_2)] = 2^{-t}$ for any $z_1 \neq z_2$. This implies that for exactly 2^{-t} of all $h \in \mathcal{H}_t$ we obtain $h(z_1) = h(z_2)$.

Further using $w_{y,z} = 0$ for $z \notin f(\mathbb{F}_2^n)$, we conclude that

$$p_h(y) = \sum_{z \in \mathbb{F}_2^n} w_{y,z}^2 + 2^{-t} \sum_{z_1 \neq z_2} w_{y,z_1} w_{y,z_2} .$$

From Lemma 2 we know that

$$0 = 2^{-t} \left(\sum_{z \in \mathbb{F}_2^n} w_{y,z} \right)^2 = 2^{-t} \sum_{z \in \mathbb{F}_2^n} w_{y,z}^2 + 2^{-t} \sum_{z_1 \neq z_2} w_{y,z_1} w_{y,z_2} .$$

An application of this identity together with Eq. (4) gives us

$$p_h(y) = (1 - 2^{-t}) \sum_{z \in \mathbb{F}_2^n} w_{y,z}^2 = (1 - 2^{-t}) \cdot p(y) = \frac{1 - 2^{-t}}{2^{n-1}} .$$

□

Corollary 1. *We measure in Algorithm* Hashed-Simon *in the first n qubits $y = 0$ with probability $2^{-t} + (1 - 2^{-t}) \cdot 2^{1-n}$.*

Theorem 2. *Let $\mathcal{H}_t = \{h : \mathbb{F}_2^n \to \mathbb{F}_2^t\}$ be universal, and let $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a Simon function with period $s \in \mathbb{F}_2^n$.* Hashed-Simon *recovers s with expected $\frac{n+1}{1-2^{-t}}$ applications of quantum circuits $Q_{h \circ f}^{\text{Simon}}$, $h \in_R \mathcal{H}_t$, that use only $n + t$ qubits.*

Proof. Let us define a random variable X_i, $1 \leq i < n$ for the number of applications of $Q_{h \circ f}^{\text{Simon}}$ until Hashed-Simon finds i linearly independent $y_1 \dots y_i$. Let E_i be the event that we already have $i - 1$ linearly independent $Y = \{y_1, \dots, y_{i-1}\}$ and we measure some $y_i \notin \text{span}(Y)$. Define $p_i = \mathbb{P}[E_i]$. Using Theorem 1, we obtain

$$p_1 = (1 - 2^{-t}) \cdot \frac{2^{n-1} - 1}{2^{n-1}}.$$

Since $|\text{span}\{y_1 \dots, y_{i-1}\}| = 2^{i-1}$, we obtain from Theorem 1 more generally

$$p_i = (1 - 2^{-t}) \cdot \frac{2^{n-1} - 2^{i-1}}{2^{n-1}}.$$

Clearly, X_i is geometrically distributed with parameter p_i. Let $X = X_1 + \ldots + X_{n-1}$ denote the number of required applications of $Q_{h \circ f}^{\text{Simon}}$ in Hashed-Simon. Then

$$\mathbb{E}[X] = \sum_{i=1}^{n-1} \mathbb{E}[X_i] = \sum_{i=1}^{n-1} \frac{1}{(1 - 2^{-t})} \cdot \frac{2^{n-1}}{2^{n-1} - 2^{i-1}} = \sum_{i=1}^{n-1} \frac{1}{(1 - 2^{-t})} \cdot \frac{2^{n-1} - 2^{i-1} + 2^{i-1}}{2^{n-1} - 2^{i-1}}$$
$$= \frac{1}{(1 - 2^{-t})} \cdot (n - 1) + \frac{1}{(1 - 2^{-t})} \cdot \sum_{i=1}^{n-1} \frac{2^{i-1}}{2^{n-1} - 2^{i-1}}.$$

Since $\lim_{n \to \infty} \sum_{i=1}^{n-1} \frac{2^{i-1}}{2^{n-1} - 2^{i-1}} \leq 1.6067$, the claim follows. □

Remark 1. With a similar analysis as in the proof of Theorem 2, we obtain an upper bound of $n + 1$ for the expected number of applications of Q_f^{Simon} in Simon's original algorithm.

4 Simon Attack on the Even-Mansour Construction

The famous Even-Mansour construction [EM97, DKS12] is an appealingly simple way of constructing a keyed pseudo-random permutation from an unkeyed public permutation $P : \mathbb{F}_2^n \to \mathbb{F}_2^n$ via

$$\mathrm{EM}_k : \mathbb{F}_2^n \to \mathbb{F}_2^n, \quad x \mapsto P(x+k)+k\ .$$

As shown by Even, Mansour [EM97] and Dunkelman et al [DKS12], the function EM_k offers quite strong security guarantees against classical adversaries.

However, Kuwakado and Morii [KM12] and Kaplan et al [KLLN16] showed that Even-Mansour is completely insecure against quantum superposition attacks. The key observation is that the function

$$f : \mathbb{F}_2^n \to \mathbb{F}_2^n, \quad x \mapsto P(x) + \mathrm{EM}_k(x) = P(x) + P(x+k) + k$$

satisfies $f(x) = f(x+k)$. Thus, an application of Simon's algorithm reveals as period the secret key k. This requires $2n$ qubits using SIMON. Let U_P and U_{EM_k} be quantum embeddings of P and EM_k. The quantum circuit for the attack is depicted in Figure 2.

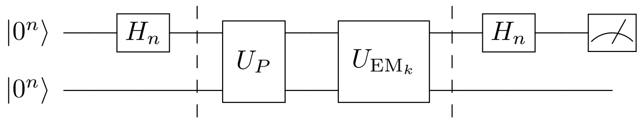


Figure 2: Quantum circuit for SIMON-attack on Even-Mansour.

4.1 Directly Realizing Hashed Even-Mansour: Simon Attack with $n+1$ qubits

From Theorem 2 we immediately conclude that we obtain a SIMON-attack with $n+1$ qubits using oracle access to hashed versions of f. In the following we show that we can directly (without oracles) construct hashed versions from P and EM_k.

First observe that P and EM_k are reversible functions $\mathbb{F}_2^n \to \mathbb{F}_2^n$, and thus may allow for direct quantum circuits that compute the function values of P, EM_k on the n input qubits, without using the generic universal quantum embedding strategy. Let us assume for the moment that we are able to construct for P an *in-place* quantum circuit Q_P, i.e. a circuit that acts on the n input qubits only. We show in the following that some natural choices for P in the Even-Mansour construction allow for such in-place realizations.

If P can be realized via Q_P in-place, then we can also realize EM_k in-place via some circuit Q_{EM_k}, where Q_{EM_k} just uses Q_P and adds in the key k (hardwired). Running the implementations of Q_P, Q_{EM_k} backwards realizes the inverse functions $P^{-1}, \mathrm{EM}_k^{-1}$. We denote the corresponding circuits by Q_P^{-1} and $Q_{\mathrm{EM}_k}^{-1}$. Notice that our hash method comes at the cost of doubling the circuit depth.

We take the following universal hash family from Equation (1) with $t=1$

$$\mathcal{H}_1 = \{h_r : \mathbb{F}_2^n \to \mathbb{F}_2 \mid r \in \mathbb{F}_2^n,\ h_r(x) = \langle x, r\rangle\}.$$

Notice that $\mathcal{H}_1$ is homomorphic, i.e. for all $h_r \in \mathcal{H}_1$ we have $h_r(x) + h_r(y) = h_r(x+y)$ by linearity of the inner product. As usual, we denote by U_{h_r} the universal embedding of h_r.

The $n+1$ qubit quantum circuit Q_{HS} in Figure 3 describes a HASHED-SIMON attack on Even-Mansour *without* the need for oracle access to hashed versions for f.
In Figure 3 we compute on the single output qubit of Q_{HS}

$$h_r(P(x)) + h_r(\mathrm{EM}_k(x)) = \langle P(x), r\rangle + \langle \mathrm{EM}_k(x), r\rangle = \langle P(x) + \mathrm{EM}_k(x), r\rangle$$

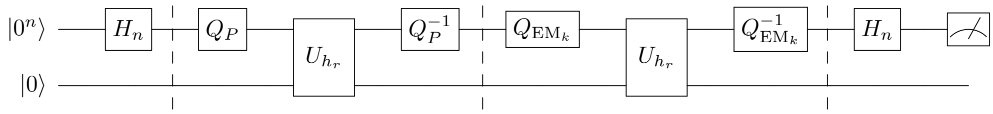


Figure 3: Quantum circuit Q_{HS} for a Hashed-Simon-Attack on Even-Mansour

$$= h_r(P(x) + \mathrm{EM}_k(x)) = h_r(f(x)).$$

Thus, the correctness of our construction follows.

It remains to show that we can compute P in-place.

In-place realization of P. There exist many lightweight permutations such as Gimli [BKL+17] that allow for (quantum) hardware-efficient implementations, for a list of candidates see the current second round NIST competition [NIS] or the work of Bonnetain and Jaques [BJ22]. For didactical reasons – since it is especially easy to describe and implement in-place – we choose the SiMeck cipher [YZS+15], for which we fix the key $k' \in \mathbb{F}_2^{n/2}$ to obtain a public permutation P, as also done in ACE [ATG+19].

SiMeck is a round-iterated Feistel cipher, see Figure 4 for one Feistel round $\mathcal{F}_{k'} : \mathbb{F}_2^n \to \mathbb{F}_2^n$.

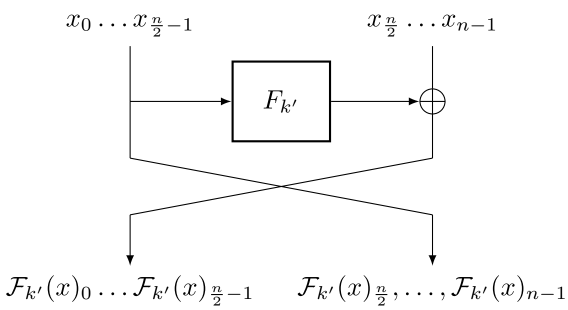


Figure 4: One round of $\mathcal{F}_{k'}$ of SiMeck, where $F_{k'} : \mathbb{F}_2^{\frac{n}{2}} \to \mathbb{F}_2^{\frac{n}{2}}$ is the round function.

Let $x = x_0 \dots x_{n-1}$. Then the i^{th} bit of $\mathcal{F}_{k'}(x)$ is

$$\mathcal{F}_{k'}(x)_i = \begin{cases} x_{i-n/2} & \text{if } i \geq n/2 \\ F_{k'}(x_0, \dots, x_{n/2-1})_i + x_{i+n/2} & \text{if } i < n/2 \end{cases}.$$

The round function $F_{k'} : \mathbb{F}_2^{n/2} \to \mathbb{F}_2^{n/2}$ in SiMeck is defined as

$$F_{k'}(x)_i = x_i \cdot x_{i+5 \bmod n/2} + x_{i+1 \bmod n/2} + k'_i .$$

We implement $\mathcal{F}_{k'}$ in-place as depicted in Figure 5. In our quantum circuit the AND-operation is realized by a Toffoli gate. Conditioned on $k'_i = 1$ we place a NOT-gate X (i.e. we hardwire k'_i).

Going from the public permutation P realized via SiMeck to Even-Mansour EM_k, we also hardwire the bits of k via NOT-gates, see Figure 6.

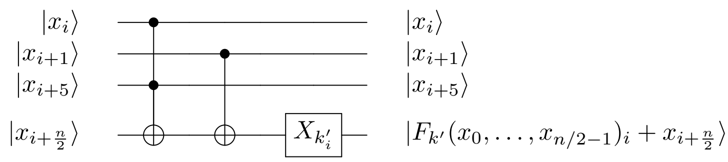


Figure 5: In-place realization of $\mathcal{F}_{k'}(x)_i$, $i < \frac{n}{2}$. If $k'_i = 1$ we place a NOT-gate X.

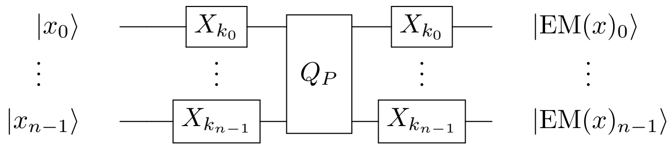


Figure 6: Quantum circuit for Even-Mansour. We place a NOT-gate X if $k_i = 1$.

5 Hashing 3-Round Feistel Distinguishers

In Section 4 via realized a hashed version of the Even-Mansour attack via splitting the periodic function in reversible blocks that in turn either have in-place implementations or oracles. More generally speaking, Hashed-Simon can be applied whenever the periodic function can be written as a sum of reversible functions, and in-place oracles/implementations are available for these functions and their inverses.

We will in the following showcase how this strategy can applied to mount a distinguishing attack on 3-round Feistel. Other potential applications of the method are the LRW construction, introduced by Liskov, Rivest and Wagner [LRW02], and 4-round Feistel distinguishers [IHM+19].

Kuwakado-Morii 3-round Feistel Distinguisher. In our distinguishing attack we get oracle access either to a realization of a 3-round Feistel (see Figure 7), or to a random permutation. Let us first assume that we are in the Feistel case, for which we describe the idea of the Kuwakado-Morii attack [KM10].

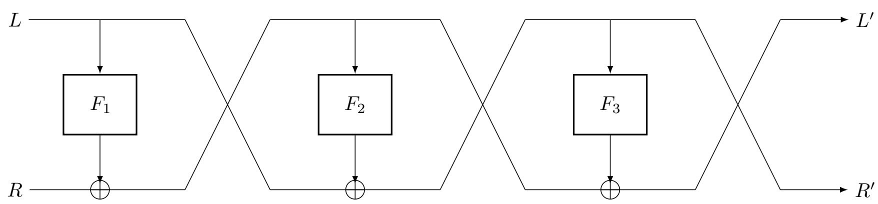


Figure 7: 3-round Feistel E, with $E(L, R) = (L', R')$.

From Figure 7 we see that

$$R' = L + F_2(R + F_1(L)).$$

Let us define via

$$E : \mathbb{F}_2^{n/2} \times \mathbb{F}_2^{n/2} \to \mathbb{F}_2^{n/2} \times \mathbb{F}_2^{n/2}, (L, R) \mapsto (L', R')$$

the application of 3-round Feistel, and we denote by $E(L, R)_{R'} = R'$ the projection on the last $n/2$ output bits. Moreover, we restrict L to only two input values α_0, α_1. We define

$$f : \mathbb{F}_2 \times \mathbb{F}_2^{n/2} \to \mathbb{F}_2^{n/2}, \quad (b, x) \mapsto F_2(x + F_1(\alpha_b)) = E(\alpha_b, x)_{R'} + \alpha_b, \tag{6}$$

which satisfies

$$f(0,x) = F_2(x + F_1(\alpha_0)) = f(1, x + F_1(\alpha_0) + F_1(\alpha_1)).$$

Therefore, f is periodic with the (unknown) period $(1, F_1(\alpha_0) + F_1(\alpha_1))$.

In case our oracle E realizes a random permutation, the function $f(b,x)$ is periodic with negligible probability. Thus, via an application of Simon's algorithm we can test periodicity and thus distinguish both cases.

Let U_E be the quantum embedding of E, and let $\alpha_0 = 0^{n/2}$, $\alpha_1 = 10^{n/2-1}$. Then the quantum distinguishing circuit is depicted in Figure 8.

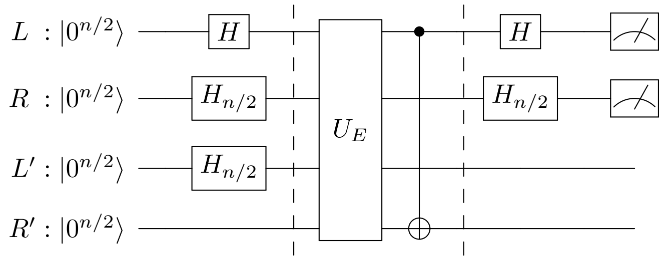


Figure 8: Quantum circuit Q_{S} for a Simon-distinguisher on 3-round Feistel. The H and CNOT gate on the first register are applied only on the first qubit.

Notice that the H-gate on the first qubit generates a uniform superposition over $b \in \{0,1\}$, and thus over $\alpha_0 = 0^{n/2}$ and $\alpha_1 = 10^{n/2-1}$. The $H_{n/2}$-gate on the output-register is used to hide the value of L', which is necessary for the proper projection to R', see [HS18] for details.

5.1 Directly Realizing Hashed 3-round Feistel

Recall from Equation (6) that $f(b,x) = E(\alpha_b, x)_{R'} + \alpha_b$. Assume that we obtain oracle access to in-place realizations U_E of the oracle function E, as well as $U_{E^{-1}}$ for the inverse E^{-1}.

We use the universal homomorphic hash family from Equation (1) with $t = 1$ for $n/2$ bits

$$\mathcal{H}_1 = \{h_r : \mathbb{F}_2^{n/2} \to \mathbb{F}_2 \mid r \in \mathbb{F}_2^{n/2},\ h_r(x) = \langle x, r\rangle\}.$$

Analogous to Section 4, we first compute a hash of $E(\alpha_b, x)$, uncompute E, and then add a hash of α_b.

The resulting $n+1$ qubit quantum circuit Q_{HS} is depicted in Figure 9. It provides an *explicit realization* of a Hashed-Simon distinguishing attack on 3-round Feistel *without* the need for oracle access to hashed versions of f.

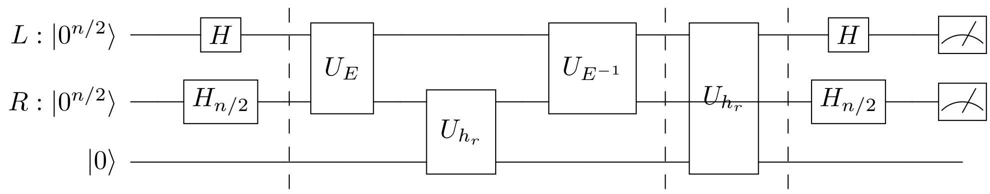


Figure 9: Quantum circuit Q_{HS} for a Hashed-Simon-distinguisher on 3-round Feistel.

Notice that the second hash function in Figure 9 hashes the first $n/2$ qubits to the output qubit. In total we compute on the output qubit of Q_{HS}

$$h_r(E(\alpha_b, x)_{R'}) + h_r(\alpha_b) = \langle E(\alpha_b, x)_{R'}, r\rangle + \langle \alpha_b, r\rangle = \langle E(\alpha_b, x)_{R'} + \alpha_b, r\rangle$$

$$= h_r(E(\alpha_b, x)_{R'} + \alpha_b) = h_r(f(x)).$$

Thus, the correctness of our construction follows.

6 Hashing Offline Even-Mansour to a Quarter of its Bits

While Simon's attack on the Even-Mansour cipher

$$\text{EM}_k : \mathbb{F}_2^n \to \mathbb{F}_2^n,\ x \mapsto P(x+k)+k$$

from Section 4 with only $\mathcal{O}(n)$ queries nicely illustrates the power of quantum computations on symmetric cryptography, it also uses a strong model giving an attacker full *quantum access* to EM_k.

Recently, Bonnetain et al [BHN+19] proposed an algorithm called Offline-Simon that in a more realistic model, where an attacker gets only *classical access* to EM_k, achieves a *polynomial* speedup over classical attacks. More precisely, the Even-Mansour attack with Offline-Simon runs in time $\tilde{\mathcal{O}}\left(2^{\frac{n}{3}}\right)$ using $\frac{4}{3}cn^2 + o(n^2)$ qubits, for some constant c (chosen as $c = \frac{5}{3}$ in [BHN+19]). This qubit analysis uses the very mild assumption that Even-Mansour's public permutation $P : \mathbb{F}_2^n \to \mathbb{F}_2^n$ can be implemented with $o(n^2)$ qubits. Efficiently computable P's are usually computable with $\mathcal{O}(n)$ qubits.

We show in the following that our hashing technique reduces the number of required qubits to only $\frac{1}{3}cn^2 + o(n^2)$, thereby saving roughly a factor of 4. Moreover, as opposed to Section 4 we do neither require the realization of any inverse functions nor require in-place realizations.

For simplicity of exposition we first explain the Offline-Simon technique when applied to Even-Mansour. Moreover, we ignore the fact that Even-Mansour is not a perfect 2:1-function, which only insignificantly affects the analysis as shown in [KLLN16, SS17, LM17].

In Theorem 4, we show how to apply Offline-Simon in combination with hashing in a more general setting. As a direct corollary we obtain that the FX-construction attack [LM17] admits a hashed version with only half the qubits.

Offline-Simon. Recall from Section 4 that the main observation of the quantum attack on Even-Mansour is that the function $\text{EM}_k(x) + P(x)$ is periodic with period k. In this function only EM_k is key-dependent. The idea of Offline-Simon is to define a key-dependent function

$$g : \mathbb{F}_2^{\frac{n}{3}} \to \mathbb{F}_2^n,\ x \mapsto \text{EM}_k(x||0^{\frac{2n}{3}}),$$

for which we only have classical access, and a key-independent function with quantum access

$$f_{k'} : \mathbb{F}_2^{\frac{n}{3}} \to \mathbb{F}_2^n,\ x \mapsto P(x||k') \text{ for some } k' \in \mathbb{F}_2^{\frac{2n}{3}}.$$

We write the secret key as $k = (k_1||k_2) \in \mathbb{F}_2^{\frac{n}{3}} \times \mathbb{F}_2^{\frac{2n}{3}}$. Notice that

$$\begin{aligned} g(x) + f_{k_2}(x) &= \text{EM}_{(k_1||k_2)}(x||0^{\frac{2n}{3}}) + P(x||k_2) \\ &= P(x + k_1||k_2) + k + P(x||k_2) \end{aligned}$$

is periodic in $k_1 \in \mathbb{F}_2^{\frac{n}{3}}$. As opposed to Section 4 instead of period k we obtain period k_1, thereby reducing the period length from n to $\frac{n}{3}$ bits. However, $g + f_{k'}$ is periodic only for the choice $k' = k_2$, otherwise it behaves like a random function (by the property of P). Therefore, Offline-Simon searches for $k' = k_2$ with a Grover search in time $\tilde{\mathcal{O}}(2^{\frac{n}{3}})$, using a Grover function that returns 1 iff $g + f_{k'}$ is periodic. Such a function has been designed by Leander and May [LM17] using Simon's algorithm. In a nutshell, Offline-Simon proceeds as follows. Let $c \geq 1$ be a small constant. Using $2^{n/3}$ classical queries to g we determine $g(x)$ for all $x \in \mathbb{F}_2^{\frac{n}{3}}$. We then build the uniform superposition

$$|\Phi_g\rangle := \bigotimes_{j=1}^{cn} \left(\sum_{x_j \in \mathbb{F}_2^{\frac{n}{3}}} |x_j\rangle \, |g(x_j)\rangle \right),$$

where for simplicity of exposition we omit amplitudes.

In addition, we take a uniform superposition over all key candidates k', i.e., we obtain

$$\sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \, |\Phi_g\rangle := \sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \bigotimes_{j=1}^{cn} \left(\sum_{x_j \in \mathbb{F}_2^{\frac{n}{3}}} |x_j\rangle \, |g(x_j)\rangle \right).$$

Using $\mathcal{O}(n)$ quantum queries to $f_{k'}$ we construct the superposition

$$\sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \, |\Phi_{g+f_{k'}}\rangle := \sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \bigotimes_{j=1}^{cn} \left(\sum_{x_j \in \mathbb{F}_2^{\frac{n}{3}}} |x_j\rangle \, |(g + f_{k'})(x_j)\rangle \right).$$

Eventually, we use Hadamard on the $|x_j\rangle$'s to obtain cn copies of a typical SIMON superposition

$$\begin{aligned}\sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \, |\Phi\rangle := \sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \otimes & \left(\sum_{x_1, y_1 \in \mathbb{F}_2^{\frac{n}{3}}} (-1)^{\langle x_1, y_1\rangle} \, |y_1\rangle \, |(g + f_{k'})(x_1)\rangle \right) \otimes \dots \\ \otimes & \left(\sum_{x_{cn}, y_{cn} \in \mathbb{F}_2^{\frac{n}{3}}} (-1)^{\langle x_{cn}, y_{cn}\rangle} \, |y_{cn}\rangle \, |(g + f_{k'})(x_{cn})\rangle \right).\end{aligned} \tag{7}$$

The state $|\Phi\rangle$ is then used do check for periodicity of $g + f_{k'}$.

Let us for a moment assume that we measure $|\Phi\rangle$. This measurement yields cn independent SIMON-samples $y_1, \dots, y_{cn}$ with $c \geq 1$. If $g + f_{k'}$ is periodic, then the $y_1, \dots, y_{cn}$ span a subspace of dimension at most $n - 1$. In the case of a non-periodic $g + f_{k'}$, with high probability (depending on c) the $y_1, \dots, y_{cn}$ span a subspace of full dimension n. Thus, we may decide whether $g + f_{k'}$ is periodic by applying Gaussian elimination to the basis vectors $y_1, \dots, y_{cn}$.

Instead of measuring $|\Phi\rangle$, we use a quantum version of Gaussian elimination which can be realized with $o(n^2)$ ancilla-qubits[1] in polynomial depth. This dimension computation procedure is then used as a Grover function for k', which returns label 1 if $g + f_{k'}$ is periodic, and else with high probability label 0. A proper choice of c guarantees with sufficiently large probability a correct label.

Notice that $|\Phi\rangle$ in Equation (7) has cn copies of a state with $\frac{n}{3}$ input qubits for y and n output qubits for $(g + f_{k'})(x)$. Thus ignoring low order terms we need $\frac{4}{3}cn^2$ qubits.

Hashed-Offline-Simon. Let us use our t-output bit homomorphic hash function family from Equation (1)

$$\mathcal{H}_t = \{h_r : \mathbb{F}_2^n \to \mathbb{F}_2^t \mid r \in (\mathbb{F}_2^n)^t, \; h_r(x) = (\langle x, r_1\rangle, \dots, \langle x, r_t\rangle)\}.$$

[1]To make the Gaussian dimension computation reversible, it suffices to add ancillary qubits to save the position of the pivot element in each step. Thus only $\mathcal{O}(\log(n) \cdot n)$ ancilla bits are required.

Let c' be a small constant. In the following we see how c' relates to c from OFFLINE-SIMON.

Choose $h_j \in_R \mathcal{H}_t$ for $1 \le j \le c'n$. As in OFFLINE-SIMON, we first classically query on-the-fly for all x_j the value $g(x_j)$ and compute $h_j(g(x_j))$ for every j to create

$$|\Phi_{h\circ g}\rangle := \bigotimes_{j=1}^{c'n} \left(\sum_{x_j \in \mathbb{F}_2^{\frac{n}{3}}} |x_j\rangle \, |h_j(g(x_j))\rangle \right).$$

We then take the superposition over all key candidates k'

$$\sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \, |\Phi_{h\circ g}\rangle := \sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \bigotimes_{j=1}^{c'n} \left(\sum_{x_j \in \mathbb{F}_2^{\frac{n}{3}}} |x_j\rangle \, |h_j(g(x_j))\rangle \right).$$

Second, using $\mathcal{O}(n)$ quantum queries to $h_j \circ f_{k'}(x_j) = h_j(P(x_j||k'))$ (a combination of each h_j with the quantum circuit for $f_{k'}$) we construct[2] the superposition

$$\sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \, |\Phi_{h\circ(g+f_{k'})}\rangle := \sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \bigotimes_{j=1}^{c'n} \left(\sum_{x_j \in \mathbb{F}_2^{\frac{n}{3}}} |x_j\rangle \, |(h_j \circ g + h_j \circ f_{k'})(x_j)\rangle \right).$$

By the homomorphic property of $h_j \in \mathcal{H}_t$ we have $h_j \circ g + h_j \circ f_{k'} = h_j \circ (g + f_{k'})$.

Eventually, Hadamard on $|x\rangle$ creates $c'n$ copies of a SIMON superposition

$$\begin{aligned} \sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \, |\Phi_h\rangle := \sum_{k' \in \mathbb{F}_2^{\frac{2n}{3}}} |k'\rangle \otimes \left(\sum_{x_1, y_1 \in \mathbb{F}_2^{\frac{n}{3}}} (-1)^{\langle x_1, y_1 \rangle} |y_1\rangle \, |h_1 \circ (g + f_{k'})(x_1)\rangle \right) \otimes \dots \\ \otimes \left(\sum_{x_{c'n}, y_{c'n} \in \mathbb{F}_2^{\frac{n}{3}}} (-1)^{\langle x_{c'n}, y_{c'n} \rangle} |y_{c'n}\rangle \, |h_{c'n} \circ (g + f_{k'})(x_{c'n})\rangle \right). \end{aligned} \tag{8}$$

The remaining steps following Equation (8) are the same as for Equation (7) above.

Thus, in contrast to OFFLINE-SIMON we hash the n output qubits to t output qubits. However, we have to chose t with some care. On the one hand, taking $t = 1$ minimizes the number of bits per copy from $\frac{4}{3}n$ to only $\frac{1}{3}n + 1$. On the other hand, by Theorem 1 the choice $t = 1$ results in (roughly) half the y_i's being zero, which in turn forces us to set $c' \ge 2c$. Thus, in total we obtain at least $2cn(\frac{1}{3}n + 1)$ qubits instead of $cn(\frac{4}{3}n)$, saving at most a factor of (roughly) 2.

We will show in Theorem 3 that the choice $t = \log_2 n$ saves us a factor of (roughly) 4 by reducing the bits per copy to (the achievable minimum) $\frac{1}{3}n + o(n)$ while increasing cn insignificantly to only $c'n = cn + o(n)$.

Theorem 3. *A* HASHED-OFFLINE-SIMON *attack with only classical access to the Even Mansour cipher* EM_k *computes* k *in time* $\tilde{\mathcal{O}}\left(2^{\frac{n}{3}}\right)$ *using* $\frac{1}{3}cn^2 + o(n^2) = \frac{5}{9}n^2 + o(n^2)$ *qubits, instead of* $\frac{4}{3}cn^2 + o(n^2) = \frac{20}{9}n^2 + o(n^2)$ *qubits for the non-hashed version.*

Proof. We know from [BHN+19] and the above discussion that the non-hashed version OFFLINE-SIMON works for the choice $c = \frac{5}{3}$ in time $\tilde{\mathcal{O}}\left(2^{\frac{n}{3}}\right)$ using $\frac{4}{3}cn^2 + o(n^2) = \frac{20}{9}n^2 +$

[2] This can be done by computing iteratively $f_{k'}(x_j)$, hashing it with h_j, and uncomputing $f_{k'}(x_j)$ to reuse the qubits for $f_{k'}(x_{j+1})$. Using this iterative procedure instead of $c'n$ times we only need once the qubits for representing $f_{k'}$.

$o(n^2)$ qubits, where $o(n^2)$ also accounts for the required ancilla qubits. The value c is chosen so that the error of the Grover function is bounded by $2^{-2n/3}$ and the success probability is bounded by $\Theta(1)$.

We choose $t = \log_2(n)$ for the hash length of our family $\mathcal{H}_t$. We measure each y_j in $|\Phi\rangle$ from Equation (7) with probability $\frac{1}{2^{n/3-1}}$. By Theorem 1, we measure each $y_j \neq 0$ in $|\Phi_h\rangle$ from Equation (8) with probability $\frac{1-2^{-t}}{2^{n/3-1}}$. This implies that conditioned on measuring $y_j \neq 0$, the probability distribution of all y_j is preserved.

Let X_i be a Bernoulli random variable that takes value 1 iff we measure $y_i = 0$. We have $p := \mathbb{P}[X_i = 1] = \frac{1}{n} + \frac{n-1}{n \cdot 2^{n/3-1}} < \frac{2}{n}$ for sufficiently large n. Let

$$n' = \frac{n}{\ln(n)}.$$

We choose $c'n = cn + n'$ in Hashed-Offline-Simon.

Let BAD be the event that less than cn of our $c'n$ measurements are non-zero vectors $y_i \neq 0$. Notice that in the event BAD we do not have sufficiently many non-zero vectors for Hashed-Offline-Simon. Let $X = \sum_{i=1}^{c'n} X_i$ be a random variable for the number of 0-measurements. Then $\mu := \mathbb{E}[X] = pc'n < 2c + \frac{2n'}{n} = 2c + \frac{2}{\ln(n)} < 4$ for sufficiently large n.

Application of a Chernoff bound yields

$$\begin{aligned}\mathbb{P}[\texttt{BAD}] = \mathbb{P}\left[X \geq c'n - cn\right] = \mathbb{P}\left[X \geq n'\right] = \mathbb{P}\left[X \geq \left(1 + \left(\frac{n'}{\mu} - 1\right)\right) \cdot \mu\right] \\ \leq \left(\frac{e^{n'/\mu - 1}}{(n'/\mu)^{n'/\mu}}\right)^{\mu} = e^{n' - \mu - \ln(n'/\mu)\cdot n'} = e^{-n + o(n)} < o(1) \cdot 2^{-2n/3} ,\end{aligned}$$

for sufficiently large n.

We see that by our choice of c' in Hashed-Offline-Simon the error of the Grover function increases by at most a factor of $(1 + o(1))$. Following the analysis from [BHN+19] this implies that we still have a success-probability of $\Theta(1)$.

In $|\Phi_h\rangle$ we obtain $c'n$ copies of $n/3 + t$ and therefore a total qubit amount of

$$c'n \cdot \left(\frac{n}{3} + \log_2 n\right) + o(n^2) = (cn + o(n))\left(\frac{n}{3} + o(n)\right) + o(n^2) = \frac{1}{3}cn^2 + o(n^2) = \frac{5}{9}n^2 + o(n^2) ,$$

where the term $o(n^2)$ also accounts for the qubits required to compute $f_{k'}$ as well the ancilla qubits for Gaussian elimination and the space for k'. □

Remark 2. Our factor-4 save in qubits comes at a linear increase of the circuit depth. In the following theorem we show that our iterative calculation of $h \circ f_{k'}$ increases the anyway exponential (!) depth of Offline-Simon by at most a factor of $2cn(1 + o(1))$.

Theorem 3 generalizes as follows.

Theorem 4. *Given an* Offline-Simon *attack with* $f_{k'}, g : \mathbb{F}_2^n \to \mathbb{F}_2^{\ell(n)}$ *and* $k' \in \mathbb{F}_2^m$*, where* $f_{k'}$ *can be computed in space* $o(n^2)$ *and depth* $\omega(1)$*. A* Hashed-Offline-Simon *attack requires only* $cn^2 + m + o(n^2)$ *instead of* $cn^2 + cn\ell(n) + m + o(n^2, n\ell(n))$ *qubits, while increasing the (exponential) depth of the circuit by a factor of at most* $2cn \cdot (1 + o(1))$.

Proof. In the proof of Theorem 3 we simply replace $f_{k'}, g : \mathbb{F}_2^{\frac{n}{3}} \to \mathbb{F}_2^n$ by $f_{k'}, g : \mathbb{F}_2^n \to \mathbb{F}_2^{\ell(n)}$, and take the memory requirement of $k' \in \mathbb{F}_2^m$ into account. This yields a total qubit amount of

$$(cn + o(n)) \cdot (n + o(n)) + m + o(n^2) = cn^2 + m + o(n^2) .$$

The depth of Hashed-Offline-Simon differs from that of Offline-Simon in the computation of the following two functions.

First, while Offline-Simon checks for periodicity of cn vectors, Hashed-Offline-Simon checks for periodicity of $c'n = cn(1+o(1))$ vectors. This increases the depth only by a $(1+o(1))$-factor.

Second and more important, in Offline-Simon we compute the cn values of $f_{k'}$'s in parallel, i.e. depth(f), while in Hashed-Offline-Simon the $c'n$ values of $h \circ f_{k'}$ are computed and uncomputed iteratively to save space. Since h's depth is bounded by n, this iterative procedure requires depth at most

$$c'n(2 \cdot \mathrm{depth}(f) + n) = \mathrm{depth}(f)\left(2c'n + \frac{n}{\mathrm{depth}(f)}\right) = \mathrm{depth}(f)\left(2cn(1+o(1))\right),$$

proving our theorem. □

Remark 3. An application of Theorem 4 with the setting of a linear $\ell(n) = dn$ and $m = o(n^2)$ shows that our hashing technique saves roughly a $(d+1)$-factor in the number of qubits.

We obtain the following corollary for the FX-construction defined as

$$\mathrm{FX}_{k_0,k_1,k_2} : \mathbb{F}_2^n \to \mathbb{F}_2^n,\ x \mapsto P_{k_0}(x+k_1)+k_2,$$

where P is a key dependent permutation with $k_0 \in \mathbb{F}_2^m$, and $k_1, k_2 \in \mathbb{F}_2^n$.

Corollary 2. *For the FX-construction with $f_{k'}, g : \mathbb{F}_2^n \to \mathbb{F}_2^n$, $f_{k'} := P_{k'}$, $g := FX_{k_0,k_1,k_2}$, and $k_0 \in \mathbb{F}_2^{o(n^2)}$,* Hashed-Offline-Simon *saves at least a factor of* $2(1-o(1))$ *in qubits in comparison to* Offline-Simon.

7 Hashed Shor: Special Periods

Let us briefly recall Shor's algorithm. Let $f : \mathbb{Z} \to \mathbb{Z}$ be periodic with period $d \in \mathbb{N}$, i.e. $d > 0$ is minimal with the property $f(x) = f(x+d)$ for all $x \in \mathbb{Z}$. For ease of notation, let us first focus on applying Shor's algorithm for factorization. In Section 9 we will also see an application for discrete logarithms.

Let $N \in \mathbb{N}$ be a composite n-bit number of unknown factorization, and let a be chosen uniformly at random from $\mathbb{Z}_N^*$, the multiplicative group modulo N. Let us define the function $f : \mathbb{Z} \to \mathbb{Z}_N$, $x \mapsto a^x \bmod N$. Notice that f is periodic with $d = \mathrm{ord}_N(a)$, since $f(x+d) = a^{x+d} = a^x a^{\mathrm{ord}(a)} = a^x = f(x)$. It is well-known that we can compute a non-trivial factor of N in probabilistic polynomial time given $d = \mathrm{ord}_N(a)$ [Sho97]. We encode the inputs of f with q qubits.

In order to find d, Shor uses the quantum circuit $Q_f^{\textsc{Shor}}$ from Figure 10 with oracle-access to f. In $Q_f^{\textsc{Shor}}$ we measure in the q input qubits with high probability y's that are close to some multiple of $\frac{2^q}{d}$. The original Shor algorithm then measures sufficiently many y's (a constant number is sufficient) to extract d in a classical post-process.

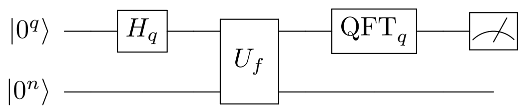


Figure 10: Quantum circuit $Q_f^{\textsc{Shor}}$

Our Hashed-Shor (Algorithm 2) simply replaces circuit $Q_f^{\textsc{Shor}}$ with its hashed version $Q_{h\circ f}^{\textsc{Shor}}$, where we use oracle-access to hashed versions of f. Notice that Shor is a special case of Hashed-Shor for the choice $t = \lceil \log_2(N) \rceil$ and $\mathcal{H}_t = \{\mathrm{id}\}$. For this choice $\mathcal{H}_t$ is not universal, but we do not need universality in the following Lemma 3 about the superposition produced by $Q_{h\circ f}$. From Lemma 3 we conclude correctness of Hashed-Shor for any t-bit range hash function h.

Algorithm 2: Hashed-Shor

Input : $f : \mathbb{Z} \to \mathbb{Z}_N$, universal $\mathcal{H}_t := \{h : \mathbb{Z}_N \to \{0,1\}^t\}$
Output : Period d of f

begin
  Set $Y = \emptyset$.
  repeat
    Run $Q^{\text{Shor}}_{h\circ f}$ on $|0^q\rangle\,|0^t\rangle$ for some freshly chosen $h \in_R \mathcal{H}_t$.
    Let y be the measurement of the q input qubits.
    If $y \neq 0$, then include y in Y.
  until $|Y|$ *is sufficiently large.*
  Compute d from Y in a classical post-process.
  return d
end

Lemma 3. *Let $N \in \mathbb{N}$, $a \in \mathbb{Z}_N^*$ with $d = ord_N(a)$ and $f(x) = a^x \bmod N$. Let $h : \mathbb{Z}_N \to \{0,1\}^t$. Define $M_z := \{k \in \mathbb{Z}_d \mid h(a^k \bmod N) = z\}$. An application of quantum circuit $Q^{\text{Shor}}_{h\circ f}$ on input $|0^q\rangle\,|0^t\rangle$ yields a superposition*

$$|\Phi_h\rangle = \sum_{y=0}^{2^q-1} \sum_{z\in\{0,1\}^t} \frac{1}{2^q} \sum_{k\in M_z} \sum_{\substack{c\geq 0:\\ cd+k<2^q}} e^{2\pi i \frac{cd+k}{2^q} y}\,|y\rangle\,|z\rangle\ . \tag{9}$$

Proof. In $Q^{\text{Shor}}_{h\circ f}$, we apply on input $|0^q\rangle\,|0^t\rangle$ first the operation $H_q \otimes I_t$ followed by $U_{h\circ f}$. This results in superposition

$$\frac{1}{\sqrt{2^q}} \sum_{x=0}^{2^q-1} |x\rangle\,|h(a^x \bmod N)\rangle\ .$$

Let $x = cd + k$ with $k \in \mathbb{Z}_d$. Since $a^x = a^{cd+k} \equiv a^k \mod N$, the value of $f(x)$ depends only on $k = (x \bmod d)$. Therefore, we rewrite the above superposition as

$$\frac{1}{\sqrt{2^q}} \sum_{z\in\{0,1\}^t} \sum_{k\in M_z} \sum_{\substack{c\geq 0:\\ cd+k<2^q}} |cd+k\rangle\,|z\rangle\ ,$$

Eventually, an application of QFT_q yields

$$|\Phi_h\rangle = \sum_{y=0}^{2^q-1} \sum_{z\in\{0,1\}^t} \underbrace{\frac{1}{2^q} \sum_{k\in M_z} \sum_{\substack{c\geq 0:\\ cd+k<2^q}} e^{2\pi i \frac{cd+k}{2^q} y}}_{w_{y,z}}\,|y\rangle\,|z\rangle\ .$$

□

Remark 4. For the choice $h = id$, Lemma 3 provides an analysis of Shor's original quantum circuit Q^{Shor}_f. This choice implies $M_z = \{k \in \mathbb{Z}_d \mid a^k = z \bmod N\}$. Therefore, we obtain the superposition

$$|\Phi\rangle = \sum_{y=0}^{2^q-1} \sum_{k=0}^{d-1} \frac{1}{2^q} \sum_{\substack{c\geq 0:\\ cd+k<2^q}} e^{2\pi i \frac{cd+k}{2^q} y}\,|y\rangle\,|a^k \bmod N\rangle\ . \tag{10}$$

and the amplitudes of $|y\rangle\,|z_k\rangle$ with $z_k = a^k \bmod N$ are

$$w_{y,z_k} = \frac{1}{2^q} \sum_{\substack{c\geq 0:\\ cd+k<2^q}} e^{2\pi i \frac{cd+k}{2^q} y} .$$

For didactical reasons and ease of notation, let us look in the subsequent section at the special case of periods d that are powers of two. In Section 8, we analyse the general d case.

7.1 Periods that are a power of two

Let $d = 2^r$ for some $r \in \mathbb{N}$ with $r \leq q$. Then $\max\{c \in \mathbb{N} \mid cd+k < 2^q\} = \frac{2^q}{d} - 1 = 2^{q-r} - 1$, independent of $k \in \mathbb{Z}_d$. Hence, let us define $m = \frac{2^q}{d}$ and $z_k = a^k \mod N$. Using $md = 2^q$, this allows us to rewrite Eq. (10) and Eq. (9) as

$$|\Phi\rangle = \sum_{y=0}^{2^q-1} \sum_{k=0}^{d-1} \left(\frac{1}{\sqrt{d}} \cdot e^{2\pi i \frac{k}{2^q} y} \right) \cdot \left(\frac{1}{\sqrt{m \cdot 2^q}} \sum_{c=0}^{m-1} e^{2\pi i \frac{cd}{2^q} y} \right) |y\rangle\,|z_k\rangle \tag{11}$$

respectively for $h : \mathbb{Z}_N \to \{0,1\}^t$ as

$$|\Phi_h\rangle = \sum_{y=0}^{2^q-1} \sum_{z\in\{0,1\}^t} \left(\frac{1}{\sqrt{d}} \sum_{k\in M_z} e^{2\pi i \frac{k}{2^q} y} \right) \cdot \left(\frac{1}{\sqrt{m \cdot 2^q}} \sum_{c=0}^{m-1} e^{2\pi i \frac{cd}{2^q} y} \right) |y\rangle\,|z\rangle\ . \tag{12}$$

Notice that the factor

$$\frac{1}{\sqrt{m \cdot 2^q}} \sum_{c=0}^{m-1} e^{2\pi i \frac{cd}{2^q} y}$$

is identical in $|\Phi\rangle$ and its hashed version $|\Phi_h\rangle$. Further notice that the factor is independent of z_k and z. In the following lemma we show that for a measurement of any $|y\rangle$, where y is a multiple of m, this factor contributes to the probability with $\frac{1}{d}$.

Lemma 4. *Let $d = 2^r \leq 2^q$ and $y = \ell m$ for some $0 \leq \ell < d$. Then we have*

$$\left| \frac{1}{\sqrt{m \cdot 2^q}} \sum_{c=0}^{m-1} e^{2\pi i \frac{cd}{2^q} y} \right|^2 = \frac{1}{d} .$$

Proof. Since $y = \ell m = \ell \frac{2^q}{d}$ we obtain

$$\left| \frac{1}{\sqrt{m \cdot 2^q}} \sum_{c=0}^{m-1} e^{2\pi i \frac{cd}{2^q} y} \right|^2 = \frac{1}{m \cdot 2^q} \left| \sum_{c=0}^{m-1} e^{2\pi i c\ell} \right|^2 = \frac{m^2}{m \cdot 2^q} = \frac{m}{2^q} = \frac{1}{d} .$$

□

We now show that the same common factor ensures that in both superpositions $|\Phi\rangle$ and its hashed version $|\Phi_h\rangle$ we never measure some $|y\rangle$ if y is not a multiple of m.

Lemma 5. *Let $d = 2^r \leq 2^q$ and $y \in \{0, \ldots, 2^q - 1\}$ with $m \nmid y$. Then we measure $|y\rangle$ in either $|\Phi\rangle$ or $|\Phi_h\rangle$ from Equation (11) or Equation (12) with probability 0.*

Proof. Let $y = m\ell + k$ with $0 < k < \ell$. It suffices to show that $\left| \sum_{c=0}^{m-1} e^{2\pi i \frac{cd}{2^q} y} \right|^2 = 0$. Using $\frac{d}{2^q} = \frac{1}{m}$, we obtain

$$\left| \sum_{c=0}^{m-1} e^{2\pi i \frac{cd}{2^q} y} \right|^2 = \left| \sum_{c=0}^{m-1} e^{2\pi i \frac{c}{m}(m\ell+k)} \right|^2 = \left| \sum_{c=0}^{m-1} \left(e^{2\pi i \frac{k}{m}} \right)^c \right|^2 = 0 .$$

□

Hence, we conclude from Lemmata 3, 4 and 5 that in both $|\Phi\rangle$ and its hashed version $|\Phi_h\rangle$ we *always* measure some $|y\rangle$ for which $y = \ell m = \frac{\ell 2^q}{d}$. Assume that $\gcd(\ell, d) = 1$, then we directly read off d from y. If ℓ is uniformly distributed in the interval $[0, d)$ this happens with sufficient probability to compute d in polynomial time.

Indeed, in SHOR's original algorithm ℓ is uniformly distributed since the first factor in Eq. (11) satisfies for any y

$$\sum_{k=0}^{d-1} \left| \frac{1}{\sqrt{d}} \cdot e^{2\pi i \frac{k}{2^q} y} \right|^2 = \frac{1}{d} \sum_{k=0}^{d-1} \left| e^{2\pi i \frac{k}{2^q} y} \right|^2 = \frac{1}{d} \sum_{k=0}^{d-1} 1 = 1 \ .$$

Similar to the reasoning in Section 3, we show that in the case of the hashed version $|\Phi_h\rangle$ we obtain any $|y\rangle$ with $y \neq 0$ with probability of at least $\frac{1}{2d}$, where the probability is taken over the random choice of the hash function. This implies that we measure for $|\Phi_h\rangle$ the useless $y = 0$ with at most probability $1 - \frac{d-1}{2d} \approx \frac{1}{2}$.

Theorem 5. *Let $N \in \mathbb{N}$, $a \in \mathbb{Z}_N^*$ with $d = ord_N(a)$ a power of two and $f(x) = a^x \bmod N$. Let $\mathcal{H}_t = \{h : \mathbb{Z}_N \to \{0,1\}^t\}$ be universal. Then we measure in* HASHED-SHOR *in the q input qubits any $y = \ell m$, $0 < \ell < d$ with probability $\frac{1-2^{-t}}{d}$, where the probability is taken over the random choice of $h \in \mathcal{H}_t$.*

Proof. Let us denote by $p_h = \mathbb{P}_{h \in \mathcal{H}_t}[y]$ the probability that we measure y in HASHED-SHOR in the q input qubits. By Lemma 3, Eq. (12) and Lemma 4 we know that for all $y = \ell m = \ell \frac{2^q}{d}$ we have

$$p_h = \frac{1}{|\mathcal{H}_t|} \sum_{h \in \mathcal{H}_t} \sum_{z \in \{0,1\}^t} \left| \frac{1}{\sqrt{d}} \sum_{k \in M_z} e^{2\pi i \frac{k}{2^q} y} \right|^2 \cdot \frac{1}{d} = \frac{1}{d^2} \cdot \frac{1}{|\mathcal{H}_t|} \sum_{h \in \mathcal{H}_t} \sum_{z \in \{0,1\}^t} \left| \sum_{k \in M_z} e^{2\pi i \frac{k\ell}{d}} \right|^2 .$$

Recall that $M_z := \{k \in \mathbb{Z}_d \mid h(a^k \bmod N) = z\}$. Observe that for $k_1 \neq k_2$ we obtain a cross-product $e^{2\pi i \frac{k_1 \ell}{d}} \cdot \overline{e^{2\pi i \frac{k_2 \ell}{d}}} = e^{2\pi i \frac{(k_1 - k_2)\ell}{d}}$ iff k_1, k_2 are in the same set $M_z, z \in \{0,1\}^t$, i.e. iff $h(a^{k_1} \bmod N) = h(a^{k_2} \bmod N)$. Using Definition 1 of a universal hash function family, we obtain $\mathbb{P}_{h \in \mathcal{H}_t}[h(a^{k_1} \bmod N) = h(a^{k_2} \bmod N)] = 2^{-t}$ for any $k_1 \neq k_2$. This implies that for exactly 2^{-t} of all $h \in \mathcal{H}_t$ we obtain $h(a^{k_1} \bmod N) = h(a^{k_2} \bmod N)$. Therefore,

$$p_h = \frac{1}{d^2} \cdot \left(\sum_{k=0}^{d-1} \left| e^{2\pi i \frac{k\ell}{d}} \right|^2 + 2^{-t} \sum_{k_1 \in \mathbb{Z}_d} \sum_{k_2 \neq k_1 \in \mathbb{Z}_d} e^{2\pi i \frac{(k_1 - k_2)\ell}{d}} \right) .$$

Since $k_1 - k_2 \neq 0$, we can rewrite as

$$p_h = \frac{1}{d^2} \cdot \left(d + \frac{d}{2^t} \sum_{k \in \mathbb{Z}_d \setminus \{0\}} e^{2\pi i \frac{k\ell}{d}} \right) = \frac{1}{d^2} \cdot \left(d - \frac{d}{2^t} \right) = \frac{1 - 2^{-t}}{d} \ .$$

□

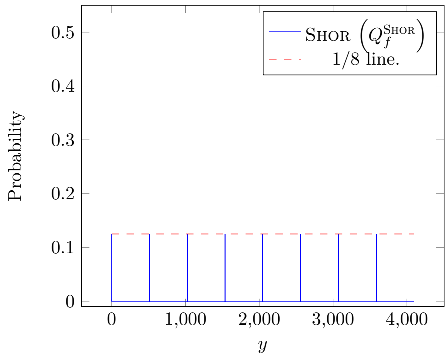
From Theorem 5 we see that in the hashed version $|\Phi_h\rangle$ we measure every $y = m\ell, y \neq 0$ with probability $\frac{1-2^{-t}}{d}$, whereas in comparison in $|\Phi\rangle$ we measure every $y = m\ell$ with probability $\frac{1}{d}$. It follows that in Eq. (12) the scaling factor

$$\mathcal{S} = \sum_{z \in \{0,1\}^t} \left| \frac{1}{\sqrt{d}} \sum_{k \in M_z} e^{2\pi i \frac{k}{2^q} y} \right|^2 \tag{13}$$

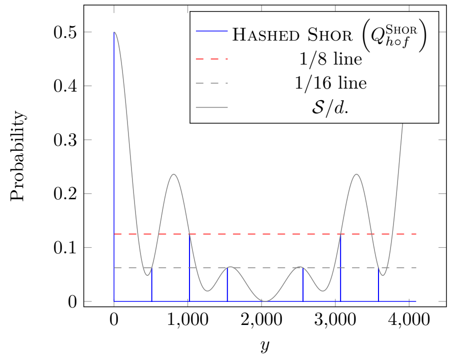
takes on expected value $1 - 2^{-t}$ for $y = \ell m$, $0 < \ell < d$ taken over all $h \in \mathcal{H}_t$. Notice that $\mathcal{S}$ is a symmetric function in y, i.e. $\mathcal{S}(y) = \mathcal{S}(2^q - y)$.

Let us look at an example to illustrate how the probabilities behave. We choose $N = 51 = 3 \cdot 17$, $a = 2$ and $q = 12$. This implies $d = \text{ord}_N(a) = 8$ and $m = \frac{2^q}{d} = 512$. In $|\Phi\rangle$ we measure some $y = m\ell = 512\ell$, $0 \le \ell < d = 8$ with probability $\frac{1}{8}$ each, as illustrated in Figure 11a.

Let us assume we have in Hashed-Shor $M_0 = \{2, 3, 4, 7\}$ (using $t = 1$). This fully specifies the scaling function $\mathcal{S}$ from Eq. (13). Thus, each amplitude from $|\Phi\rangle$ is multiplied by $\mathcal{S}$, as illustrated in Figure 11b.



(a) Shor with $d = 8$, $m = 512$. The probabilities are independent of the measurement of the output qubits z_k.



(b) Hashed-Shor with $d = 8$, $m = 512$, $t = 1$ and $M_0 = \{2, 3, 4, 7\}$.

Figure 11: Probability distributions for Shor and Hashed-Shor

Theorem 6. *Let $N \in \mathbb{N}$, $a \in \mathbb{Z}_N^*$ with $d = ord_N(a)$ a power of two and $f(x) = a^x \mod N$. Let $\mathcal{H}_t = \{h : \mathbb{Z}_N \to \{0,1\}^t\}$ be universal, and let x be represented by q input qubits in $Q_{h\circ f}^{\text{Shor}}$. Then Hashed-Shor finds f's period d with expected $\frac{2}{1-2^{-t}} \le 4$ applications of quantum circuits $Q_{h\circ f}^{\text{Shor}}$, $h \in_R \mathcal{H}_t$, that use only $q + t$ qubits.*

Proof. In Shor we compute the fraction $\frac{y}{2^q} = \frac{\ell}{d}$ in reduced form. Since d is a power of two, this fraction reveals d in its denominator iff ℓ is odd. Using Theorem 5, we measure $y = m\ell$ with an odd ℓ, $0 < \ell < d$ with probability $\frac{d}{2} \cdot \frac{1-2^{-t}}{d} = \frac{1-2^{-t}}{2}$. Thus, we need on expectation $\frac{2}{1-2^{-t}} \le 4$ applications of $Q_{h\circ f}^{\text{Shor}}$ to find f's period d.

Notice that we can check the validity of d via testing the identity $a^d \stackrel{?}{=} 1 \bmod N$. $\square$

For comparison, we need in Shor's original algorithm with the non-hashed version of f on expectation 2 measurements until we find d.

8 Hashed Period-Finding Including Shor

Notice that we proved in Theorem 1 and Theorem 5 that when we move to the hashed version of our quantum circuits all probabilities to measure some $y \neq 0$ decrease exactly by a factor of $(1 - 2^{-t})$ (over the random choice of the t-bit hash function).

The same is true for finding arbitrary (non power of two) periods with circuit $Q_{h\circ f}^{\text{Shor}}$. However, this does not immediately follow from the proof of Theorem 5, because the proof builds on the special form of superposition $|\Phi_h\rangle$ from Eq. (12) that only holds if d is a power of two. Here we show a more general result for period finding algorithms that applies for Shor's original circuit as well as for its Ekerå-Håstad variant in the subsequent section. To this end let us define a generic period finding quantum circuit Q_f^{Period} (see Figure 12). In Figure 12 we denote by Q_1, Q_2 any quantum circuitry that acts on the q

input qubits. For example, for Simon's circuit we have $Q_1 = Q_2 = H_q$ (see Figure 1). For Shor's circuit we have $Q_1 = H_q$ and $Q_2 = \mathrm{QFT}_q$. In the following Theorem 7 we define explicitly a *cancellation criterion* that this circuitry Q_1, Q_2 has to fulfill. An important feature of $Q_f^{\textsc{Period}}$ is however that we apply f only once.

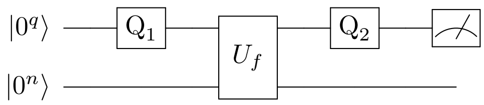


Figure 12: Quantum circuit $Q_f^{\textsc{Period}}$

Now let us use our generic period finding circuit $Q_f^{\textsc{Period}}$ inside a generic period finding algorithm Period that uses a certain number of measurements of $Q_f^{\textsc{Period}}$ and some classical post-processing. If we replace in Period the circuit $Q_f^{\textsc{Period}}$ by its hashed variant $Q_{h\circ f}^{\textsc{Period}}$ then we call the resulting algorithm Hashed-Period (Algorithm 3).

Algorithm 3: Hashed-Period

Input : $f : \{0,1\}^q \to \{0,1\}^n$, universal $\mathcal{H}_t := \{h : \{0,1\}^n \to \{0,1\}^t\}$
Output : Period d of f

```
begin
    Set Y = ∅.
    repeat
        Run Q^PERIOD_{h∘f} on |0^q⟩|0^t⟩ for some freshly chosen h ∈_R H_t.
        Let y be the measurement of the q input qubits.
        If y ≠ 0^q, then include y in Y.
    until |Y| is sufficiently large.
    Compute d from Y in a classical post-process.
    return d
end
```

Notice that Period can be considered as special case of Hashed-Period, where we choose $t = n$ and the identity function $h = id$. This slightly abuses notation, since $\mathcal{H}_n = \{id\}$ is not universal.

The proof of the following theorem closely follows the reasoning in the proof of Theorem 1. Here we show that the probabilities decreases by exactly a factor of $1 - 2^{-t}$ in the hashed version if a certain *cancellation criterion* (Equation (14)) is met.

Theorem 7. *Let $f : \{0,1\}^q \to \{0,1\}^n$ and $\mathcal{H}_t = \{h : \{0,1\}^n \to \{0,1\}^t\}$ be universal. Let $Q_f^{\textsc{Period}}$ be a quantum circuit that on input $|0^q\rangle |0^n\rangle$ yields a superposition*

$$|\Phi\rangle = \sum_{y\in\{0,1\}^q} \sum_{f(x)\in Im(f)} w_{y,f(x)} |y\rangle |f(x)\rangle \ \textit{satisfying} \sum_{f(x)\in Im(f)} w_{y,f(x)} = 0 \textit{ for any } y \neq 0. \tag{14}$$

Let us denote by $p(y)$, respectively $p_h(y)$, the probability to measure some $|y\rangle$, $y \neq 0$ in the q input qubits when applying $Q_f^{\textsc{Period}}$, respectively $Q_{h\circ f}^{\textsc{Period}}$ with $h \in_R \mathcal{H}_t$. Then $p_h(y) = (1 - 2^{-t}) \cdot p(y)$.

Proof. For ease of notation let us denote $z = f(x)$. By definition, we have $p(y) = \sum_{z\in \mathrm{Im}(f)} |w_{y,z}|^2$.

Now let us find an expression for $p_h(y)$ when using $Q_{h\circ f}^{\textsc{Period}}$. For $h \in \mathcal{H}_t$ we denote

$I_{h,z'} = \{z \in \mathrm{Im}(f) \mid h(z) = z'\}$. Since $\dot\bigcup_{z' \in \mathbb{F}_2^t} I_{h,z'} = \mathrm{Im}(f)$, we obtain

$$p_h(y) = \frac{1}{|\mathcal{H}_t|} \sum_{h \in \mathcal{H}_t} \sum_{z' \in \mathrm{Im}(h)} \left| \sum_{z \in I_{h,z'}} w_{y,z} \right|^2 . \tag{15}$$

In Eq. (15) we obtain a cross-product $w_{y,z_1}\overline{w_{y,z_2}}$ for $z_1 \neq z_2$ iff z_1, z_2 are in the same set $I_{h,z'}$, $z' \in \mathrm{Im}(h)$, i.e. iff $h(z_1) = h(z_2)$. Using Definition 1 of a universal hash function family, we obtain $\mathbb{P}_{h \in \mathcal{H}_t}[h(z_1) = h(z_2)] = 2^{-t}$ for any $z_1 \neq z_2$. This implies that for exactly 2^{-t} of all $h \in \mathcal{H}_t$ we obtain $h(z_1) = h(z_2)$. We conclude that

$$p_h(y) = \sum_{z \in \mathrm{Im}(f)} |w_{y,z}|^2 + 2^{-t} \cdot \sum_{z_1 \neq z_2} w_{y,z_1}\overline{w_{y,z_2}}.$$

Our prerequisite $\sum_{z \in \mathrm{Im}(f)} w_{y,z} = 0$ for any $y \neq 0$ implies

$$\begin{aligned} 0 = 2^{-t} \left| \sum_{z \in \mathrm{Im}(f)} w_{y,z} \right|^2 &= 2^{-t} \sum_{z \in \mathrm{Im}(f)} |w_{y,z}|^2 + 2^{-t} \sum_{z_1 \neq z_2} w_{y,z_1}\overline{w_{y,z_2}} \\ &= p_h(y) - (1 - 2^{-t}) \cdot \sum_{z \in \mathrm{Im}(f)} |w_{y,z}|^2. \end{aligned}$$

Together with the definition of $p(y)$ we conclude that

$$p_h(y) = (1 - 2^{-t}) \cdot \sum_{z \in \mathrm{Im}(f)} |w_{y,z}|^2 = (1 - 2^{-t}) \cdot p(y).$$

□

We already showed in Lemma 2 that Simon's circuit Q_f^{Simon} fulfills the cancellation criterion (Equation (14)) of Theorem 7. Thus, the statement of Theorem 1 directly follows from Theorem 7. However, for an improved intelligibility we preferred to prove Theorem 1 directly.

In the following Lemma 6 we show that Q_f^{Shor} also meets the cancellation criterion. Thus, going to the hashed version in Shor's algorithm immediately scales all probabilities by a factor of $1 - 2^{-t}$ for $y \neq 0$.

Lemma 6. *On input $|0^q\rangle |0^n\rangle$ the quantum circuit Q_f^{Shor} yields a superposition*

$$|\Phi\rangle = \sum_{y \in \{0,1\}^q} \sum_{f(x) \in Im(f)} w_{y,f(x)} |y\rangle |f(x)\rangle \ \textit{satisfying} \sum_{f(x) \in Im(f)} w_{y,f(x)} = 0 \textit{ for any } y \neq 0.$$

Proof. From Equation (10) we know that Q_f^{Shor} yields a superposition

$$|\Phi\rangle = \sum_{y=0}^{2^q-1} \sum_{f(x) \in \mathrm{Im}(f)} w_{y,f(x)} |y\rangle |f(x)\rangle \text{ with } \sum_{f(x) \in \mathrm{Im}(f)} w_{y,f(x)} = \sum_{k=0}^{d-1} \frac{1}{2^q} \sum_{\substack{c \geq 0: \\ cd+k < 2^q}} e^{2\pi i \frac{cd+k}{2^q} y} .$$

We conclude for $y \neq 0$ that

$$\sum_{f(x) \in \mathrm{Im}(f)} w_{y,f(x)} = \sum_{k=0}^{d-1} \frac{1}{2^q} \sum_{\substack{c \geq 0: \\ cd+k < 2^q}} e^{2\pi i \frac{cd+k}{2^q} y} = \sum_{r=0}^{2^q-1} \frac{1}{2^q} e^{2\pi i \frac{r}{2^q} y} = \frac{1}{2^q} \sum_{r=0}^{2^q-1} \left(e^{2\pi i \frac{y}{2^q}} \right)^r = 0 .$$

□

Since by Theorem 7 the use of hashed versions at most halves all probabilities for $y \neq 0$, we expect that Hashed-Period requires at most twice as many measurements as Period. This is more formally shown in the following Theorem 8.

Theorem 8. *Let $f : \{0,1\}^q \to \{0,1\}^n$ have period d, and let $\mathcal{H}_t = \{h : \{0,1\}^n \to \{0,1\}^t\}$ be universal. Assume that* Period *succeeds to find d with probability ρ with an expected number of m measurements, using some Q_f^{Period} with $q+n$ qubits that satisfies the cancellation criterion (Equation (14)) of Theorem 7. Then* Hashed-Period *succeeds to find d with probability ρ using $Q_{h\circ f}^{\text{Period}}$, $h \in_R \mathcal{H}_t$, with only $q+t$ qubits and an expected number of $\frac{m}{1-2^{-t}}$ measurements.*

Proof. We first show the factor of $\frac{1}{1-2^{-t}}$ difference in the expected number of measurements. In the case of Q_f^{Period} we measure some $y \neq 0$ with probability $\sum_{y=1}^{2^q-1} p(y)$, whereas for $Q_{h\circ f}^{\text{Period}}$ we measure $y \neq 0$ with $1-2^{-t}$ times the probability $\sum_{y=1}^{2^q-1}(1-2^{-t}) \cdot p(y) = (1-2^{-t}) \cdot \sum_{y=1}^{2^q-1} p(y)$ according to Theorem 7. This implies that on expectation we need $\frac{1}{1-2^{-t}}$ as many measurements.

It remains to show that Hashed-Period has the same success probability ρ as Period to compute the period d. To this end we show that conditioned on $y \neq 0$, both circuits Q_f^{Period} and $Q_{h\circ f}^{\text{Period}}$ yield an identical probability distribution for the measured $|y\rangle$ in the q input qubits.

Let $p(y)$, respectively $p_h(y)$, be the probability that we measure $|y\rangle$ in the q input qubits using Q_f^{Shor}, respectively $Q_{h\circ f}^{\text{Shor}}$. Since Period conditions on measuring $y \neq 0$ we obtain in the case of Q_f^{Shor} the probabilities

$$\frac{p(y)}{\sum_{y=1}^{2^q-1} p(y)} \text{ for any } y \neq 0.$$

In the case $Q_{h\circ f}^{\text{Shor}}$, we obtain using Theorem 7 the same probabilities

$$\frac{(1-2^{-t}) \cdot p(y)}{\sum_{y=1}^{2^q-1}(1-2^{-t}) \cdot p(y)} = \frac{p(y)}{\sum_{y=1}^{2^q-1} p(y)} \text{ for any } y \neq 0.$$

Since both probability distributions are identical, the success probability ρ is identical as well, independent of any specific post-process for computing d. ☐

Since by Lemma 6 Shor's circuit Q_f^{Shor} satisfies the cancellation criterion of Theorem 7, Theorem 8 implies that we can implement Shor's algorithm oracle-based with $q+t$ instead of $q+n$ qubits at the cost of only $\frac{1}{1-2^{-t}}$ times as many measurements. In other words, for $t=1$ we save all but one of the output qubits at the cost of twice as many measurements.

9 Oracle-Based Hashed Ekerå-Håstad

In 2017, Ekerå and Håstad [EH17] proposed a variant of Shor's algorithm for computing the discrete logarithms of $x = g^d$ in polynomial time with only $(1+o(1))\log d$ input qubits. The Ekerå-Håstad algorithm saves input qubits in comparison to Shor's original discrete logarithm algorithm whenever d is significantly smaller than the group order.

An interesting application of such a small discrete logarithm algorithm is the factorization of n-bit RSA moduli $N = pq$, where p, q are primes of the same bit-size. Let $g \in_R \mathbb{Z}_N^*$. Then $\text{ord}_N(g)$ divides $\phi(N)/2 = (p-1)(q-1)/2 = \frac{N+1}{2} - \frac{p+q}{2}$. Therefore

$$x := g^{\frac{N+1}{2}} = g^{\frac{p+q}{2}} \bmod N.$$

Hence, we obtain a discrete logarithm instance in $\mathbb{Z}_N^*$ where the desired logarithm $d = \frac{p+q}{2}$ is of size only roughly $\frac{n}{2}$ bits, whereas group elements have to be represented with n bits. Notice that the knowledge of $d = \frac{p+q}{2}$ together with $N = pq$ immediately yields the factorization of N in polynomial time.

The Ekerå-Håstad algorithm computes d with $(\frac{1}{2} + \frac{1}{s})n$ input and n output qubits, using a classical post-process that takes time polynomial in n and s^s. Choosing $s = \frac{\log n}{\log \log n}$, we obtain a polynomial time factoring algorithm with a total of $(\frac{3}{2} + o(1))n$ qubits.

In the following, we show that the Ekerå-Håstad algorithm is covered by our framework of quantum period finding algorithms which fulfill the cancellation criterion of Equation (14) from Theorem 7. Thus, by Theorem 8 we can save all but 1 of the n output qubits via (oracle-based) hashing, at the cost of only doubling the number of quantum measurements. This in turn leads to a polynomial time (oracle-based) factorization algorithm for n-bit RSA numbers using only $(\frac{1}{2} + o(1))n$ qubits. Concerning discrete logarithms, with our (oracle-based) hashing approach we can quantumly compute d from g and g^d in polynomial time using only $(1 + o(1)) \log d$ qubits.

Let $(g, x = g^d, S(G))$ be a discrete logarithm instance with $m = \log d$. Here $S(G)$ specifies how we compute in the group G generated by g, e.g. $S(G) = N$ specifies that we compute modulo N in the group $G = \mathbb{Z}_N^*$. Define

$$f_{g,x,S(G)}(a, b) = a^x \cdot x^{-b} = g^{a-bd}.$$

The Ekerå-Håstad quantum circuit $Q_f^{\text{Ekerå-Håstad}}$ from Figure 13 computes on input $|0^{m+\ell}\rangle\, |0^\ell\rangle\, |0^n\rangle$, where $\ell := \frac{m}{s}$, a superposition

$$|\Phi\rangle = \frac{1}{2^{m+2\ell}} \sum_{a,j=0}^{2^{m+\ell}-1} \sum_{b,k=0}^{2^\ell - 1} e^{2\pi i (aj + 2^m bk)/2^{m+\ell}} \, |j, k, f_{g,x,S(G)}(a, b)\rangle \ . \qquad (16)$$

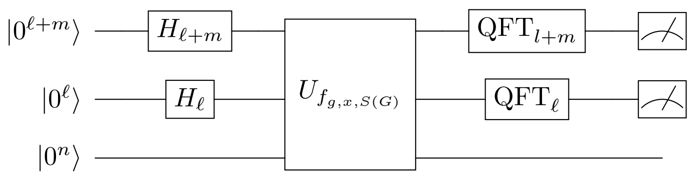


Figure 13: Quantum circuit $Q_f^{\text{Ekerå-Håstad}}$

The main step in the analysis of Ekerå-Håstad shows that we measure in the $m + 2\ell = (1 + \frac{2}{s})m = (1 + \frac{2}{s}) \log d$ input qubits with high probability so-called *good pairs* (j, k) that help us in computing d via some lattice reduction technique.

In the following Lemma 7, we show that $Q_f^{\text{Ekerå-Håstad}}$ satisfies our cancellation criterion of Theorem 7. Thus, we conclude from Theorem 7 that by moving to the 1-bit hashed version $Q_{h \circ f}^{\text{Ekerå-Håstad}}$ we lower the probabilities of measuring good (j, k) only by a factor of $\frac{1}{2}$ (averaged over all hash functions).

Lemma 7. *Let $(g, x, S(G))$ be a discrete logarithm instance and $f_{g,x,S(G)}(a, b) = g^a x^{-b}$. On input $|0^{2\ell+m}\rangle\, |0^n\rangle$ the quantum circuit $Q_f^{\text{Ekerå-Håstad}}$ yields a superposition*

$$|\Phi\rangle = \sum_{y \in \{0,1\}^{m+2\ell}} \sum_{f(x) \in Im(f)} w_{y,f(x)}\, |y\rangle\, |f(x)\rangle \ \textit{satisfying} \sum_{f(x) \in Im(f)} w_{y,f(x)} = 0 \textit{ for any } y \neq 0 \ .$$

Proof. From Eq. (16) with $y = (j,k)$ we know that $Q_f^{\text{Ekerå-Håstad}}$ yields a superposition

$$|\Phi\rangle = \sum_{j=0}^{2^{m+\ell}-1} \sum_{k=0}^{2^{m+\ell}-1} \sum_{f(x)\in \mathrm{Im}(f)} w_{(j,k),f(x)} \, |j,k\rangle \, |f(x)\rangle \text{ with}$$

$$\sum_{f(x)\in \mathrm{Im}(f)} w_{(j,k),f(x)} = \frac{1}{2^{m+2\ell}} \sum_{a=0}^{2^{m+\ell}-1} \sum_{b=0}^{2^{\ell}-1} e^{2\pi i(aj+2^m bk)/2^{m+\ell}}.$$

Hence for $y \neq 0$ we obtain

$$\begin{aligned}\sum_{f(x)\in \mathrm{Im}(f)} w_{(j,k),f(x)} &= \frac{1}{2^{m+2\ell}} \left(\sum_{a=0}^{2^{m+\ell}-1} e^{2\pi i a j/2^{m+\ell}} \right) \cdot \left(\sum_{b=0}^{2^{\ell}-1} e^{2\pi i b k/2^{\ell}} \right) \\ &= \frac{1}{2^{m+2\ell}} \left(\sum_{a=0}^{2^{m+\ell}-1} \left(e^{2\pi i j/2^{m+\ell}} \right)^a \right) \cdot \left(\sum_{b=0}^{2^{\ell}-1} \left(e^{2\pi i k/2^{l}} \right)^b \right) .\end{aligned}$$

Since by prerequisite $(j,k) \neq (0,0) \in \mathbb{Z}_{2^{m+\ell}} \times \mathbb{Z}_{2^\ell}$, we have $j \neq 0 \bmod 2^{m+\ell}$ or $k \neq 0 \bmod 2^\ell$. This implies that at least one of the factors is identical 0. □

By Theorem 8, replacing in the Ekerå-Håstad algorithm the quantum circuit $Q_f^{\text{Ekerå-Håstad}}$ by single output bit circuits $Q_{h\circ f}^{\text{Ekerå-Håstad}}$ comes at the cost of only twice the number of measurements. Since the Ekerå-Håstad algorithm finds discrete logarithms d in polynomial time using only $m + 2\ell = (1 + o(1)) \log d$ input qubits, we obtain from Theorem 8 the following corollary.

Corollary 3. *Ekerå-Håstad's Shor variant admits an* oracle-based *hashed version that*

1. *computes discrete logarithms* d *from* g, g^d *in polynomial time using* $(1 + o(1)) \log d$ *qubits,*
2. *factors* n*-bit RSA numbers in time polynomial in* n *using* $(\frac{1}{2} + o(1))n$ *qubits.*

Open Problem: Can we modify our oracle-based approach into a real-world application similar to the results in Section 4 for the Simon algorithm? That is, can we define (not necessary single bit) hashed versions of the exponentiation function without first computing the full function value?

Acknowledgments

This work was funded by DFG under Germany's Excellence Strategy - EXC 2092 CASA - 390781972.

References

[AMR07] Gorjan Alagic, Cristopher Moore, and Alexander Russell. Quantum algorithms for simon's problem over general groups. In *SODA*, pages 1217–1224. SIAM, 2007.

[ATG+19] Mark Aagaard, Riham Al Tawy, Guang Gong, Kalikinkar Mandal, and Raghvendra Rohit. ACE: An authenticated encryptionand hash algorithm. *Submission to the NIST LWC Competition*, 2019.

[BBM17] Daniel J. Bernstein, Jean-François Biasse, and Michele Mosca. A low-resource quantum factoring algorithm. In *PQCrypto*, volume 10346 of *Lecture Notes in Computer Science*, pages 330–346. Springer, 2017.

[BHN+19] Xavier Bonnetain, Akinori Hosoyamada, María Naya-Plasencia, Yu Sasaki, and André Schrottenloher. Quantum attacks without superposition queries: The offline simon's algorithm. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 552–583. Springer, 2019.

[BJ22] Xavier Bonnetain and Samuel Jaques. Quantum period finding against symmetric primitives in practice. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(1):1–27, 2022.

[BKL+17] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 299–320, 2017.

[BLNS21] Xavier Bonnetain, Gaëtan Leurent, María Naya-Plasencia, and André Schrottenloher. Quantum linearization attacks. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 422–452. Springer, 2021.

[BV97] Dan Boneh and Ramarathnam Venkatesan. Rounding in lattices and its cryptographic applications. In *SODA*, pages 675–681. ACM/SIAM, 1997.

[CvD08] Andrew Childs and Wim van Dam. Quantum algorithms for algebraic problems. *Review of Modern Physics*, 82, 12 2008.

[DKS12] Orr Dunkelman, Nathan Keller, and Adi Shamir. Minimalism in cryptography: The even-mansour scheme revisited. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 336–354. Springer, 2012.

[EH17] Martin Ekerå and Johan Håstad. Quantum algorithms for computing short discrete logarithms and factoring RSA integers. *CoRR*, abs/1702.00249, 2017.

[EM97] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *J. Cryptology*, 10(3):151–162, 1997.

[HRS17] Thomas Häner, Martin Roetteler, and Krysta M. Svore. Factoring using $2n+2$ qubits with toffoli based modular multiplication. *Quantum Information & Computation*, 17(7&8):673–684, 2017.

[HS18] Akinori Hosoyamada and Yu Sasaki. Quantum demiric-selçuk meet-in-the-middle attacks: Applications to 6-round generic feistel constructions. In Dario Catalano and Roberto De Prisco, editors, *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, volume 11035 of *Lecture Notes in Computer Science*, pages 386–403. Springer, 2018.

[IHM+19] Gembu Ito, Akinori Hosoyamada, Ryutaroh Matsumoto, Yu Sasaki, and Tetsu Iwata. Quantum chosen-ciphertext attacks against feistel ciphers. In Mitsuru Matsui, editor, *Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings*, volume 11405 of *Lecture Notes in Computer Science*, pages 391–411. Springer, 2019.

[KLLN16] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *CRYPTO (2)*, volume 9815 of *Lecture Notes in Computer Science*, pages 207–237. Springer, 2016.

[KM10] Hidenori Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round feistel cipher and the random permutation. In *IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings*, pages 2682–2685. IEEE, 2010.

[KM12] Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type even-mansour cipher. In *Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012, Honolulu, HI, USA, October 28-31, 2012*, pages 312–316, 2012.

[LM17] Gregor Leander and Alexander May. Grover meets simon - quantumly attacking the fx-construction. In *ASIACRYPT (2)*, volume 10625 of *Lecture Notes in Computer Science*, pages 161–178. Springer, 2017.

[LRW02] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, pages 31–46, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[ME98] Michele Mosca and Artur Ekert. The hidden subgroup problem and eigenvalue estimation on a quantum computer. In *Quantum Computing and Quantum Communications, First NASA International Conference, QCQC'98, Palm Springs, California, USA, February 17-20, 1998, Selected Papers*, pages 174–188, 1998.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and computing - randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[NIS] NIST. Lightweight cryptography (lwc) standardization: Round 2 candidates announced.

[RNSL17] Martin Roetteler, Michael Naehrig, Krysta M. Svore, and Kristin E. Lauter. Quantum resource estimates for computing elliptic curve discrete logarithms. In *ASIACRYPT (2)*, volume 10625 of *Lecture Notes in Computer Science*, pages 241–270. Springer, 2017.

[RS18] Martin Roetteler and Krysta M. Svore. Quantum computing: Codebreaking and beyond. *IEEE Security & Privacy*, 16(5):22–36, 2018.

[Sei01] Jean-Pierre Seifert. Using fewer qubits in shor's factorization algorithm via simultaneous diophantine approximation. In *CT-RSA*, volume 2020 of *Lecture Notes in Computer Science*, pages 319–327. Springer, 2001.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *FOCS*, pages 124–134. IEEE Computer Society, 1994.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[Sim94] Daniel R. Simon. On the power of quantum computation. In *FOCS*, pages 116–123. IEEE Computer Society, 1994.

[SS17] Thomas Santoli and Christian Schaffner. Using simon's algorithm to attack symmetric-key cryptographic primitives. *Quantum Information & Computation*, 17(1&2):65–78, 2017.

[YZS+15] Gangqiang Yang, Bo Zhu, Valentin Suder, Mark D. Aagaard, and Guang Gong. The simeck family of lightweight block ciphers. In *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, pages 307–329, 2015.