Quantum computation of discrete logarithms in semigroups

Andrew M. Childs
Department of Combinatorics & Optimization
and Institute for Quantum Computing
University of Waterloo, Canada email: amchilds@uwaterloo.ca

Gábor Ivanyos
Institute for Computer Science and Control
Hungarian Academy of Sciences, Hungary
email: Gabor.Ivanyos@sztaki.mta.hu

July 8, 2014

Abstract

We describe an efficient quantum algorithm for computing discrete logarithms in semigroups using Shor's algorithms for period finding and the discrete logarithm problem as subroutines. Thus proposed cryptosystems based on the presumed hardness of discrete logarithms in semigroups are insecure against quantum attacks. In contrast, we show that some generalizations of the discrete logarithm problem are hard in semigroups despite being easy in groups. We relate a shifted version of the discrete logarithm problem in semigroups to the dihedral hidden subgroup problem, and we show that the constructive membership problem with respect to $k \geq 2$ generators in a black-box abelian semigroup of order N requires $\tilde{\Theta}(N^{\frac{1}{2}-\frac{1}{2k}})$ quantum queries.

1 Introduction

The presumed difficulty of computing discrete logarithms in groups is a common cryptographic assumption. For example, such an assumption underlies Diffie-Hellman key exchange, ElGamal encryption, and most elliptic curve cryptography. While such cryptosystems may be secure against classical computers, Shor showed that quantum computers can efficiently compute discrete logarithms [19]. Shor originally described an algorithm for computing discrete logarithms in the multiplicative group of a prime field, but it is well known that his approach efficiently computes discrete logarithms in any finite group, provided only that group elements have a unique encoding and that group operations can be performed efficiently.

Here we consider the closely-related problem of computing discrete logarithms in finite semigroups. A semigroup is simply a set equipped with an associative binary operation. In particular, a semigroup need not have inverses (and also need not have an identity element).

We work in a model of black-box semigroups (analogous to the model of black-box groups [2]). In this model, the elements of a semigroup S are uniquely represented by bit strings and we are given a black box that performs multiplication using this representation. In the quantum setting, this black box performs the multiplication reversibly (i.e., it performs the map $|x, y, z\rangle \mapsto |x, y, z \oplus xy\rangle$, where x, y, z are encodings of semigroup elements, xy is the encoding of the corresponding product, and $\oplus$ denotes bitwise addition modulo 2) and can be queried in superposition. It is conventional to charge unit cost for each query to the black box.

In the discrete logarithm problem for a semigroup S, we are given two elements $x, g \in S$ and are asked to find the smallest $a \in \mathbb{N} := \{1, 2, \ldots\}$ such that $g^a = x$ (or to determine that no such a exists). We write $a = \log_g x$.

At first glance, it may be unclear how a quantum computer could compute discrete logarithms in semigroups. Shor's discrete logarithm algorithm relies crucially on the function $(a, b) \mapsto g^a x^{-b}$, but x^{-b} is not defined in a semigroup. In fact, hardness of the semigroup discrete logarithm problem has been proposed as a cryptographic assumption that might be secure against quantum computers [11]. The particular scheme described in [11], based on matrix semigroups, has been broken by a quantum attack [16]. However, the algorithm of [16] uses a reduction from discrete logarithms in matrix groups to discrete logarithms in finite fields [14], so it does not apply to general semigroups.

Here we point out that in fact quantum computers can efficiently compute discrete logarithms in any finite semigroup. Our approach is a straightforward application of known quantum tools. The structure of the semigroup generated by g can be efficiently determined using the ability of a quantum computer to detect periodicity, as shown in Section 2. Once this structure is known, an algorithm to compute discrete logarithms follows easily, as explained in Section 3.

On the other hand, some problems for semigroups are considerably harder than for groups. In Section 4, we consider a shifted version of the discrete logarithm problem in semigroups, namely solving the equation $x = yg^a$ for a. This problem appears comparably difficult to the dihedral hidden subgroup problem, even though the corresponding problem in a group can be solved efficiently by computing a discrete logarithm. In Section 5, we consider the problem of writing a given semigroup element as a product of $k \geq 2$ given generators of a black-box abelian semigroup. This problem can also be solved efficiently in groups, whereas the semigroup version is provably hard, requiring $\Omega(N^{\frac{1}{2} - \frac{1}{2k}})$ quantum queries for an N-element semigroup. In fact, this bound is optimal up to logarithmic factors, as we show using the algorithm for the shifted discrete logarithm problem.

After posting a preprint of this work, we learned of independent related work by Banin and Tsaban, who showed that the semigroup discrete logarithm problem can be solved efficiently using an oracle for the discrete logarithm problem

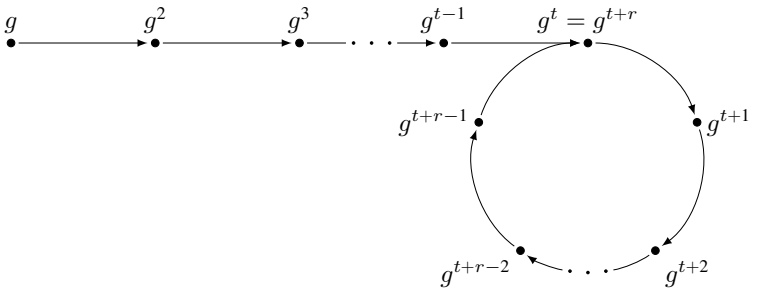

Figure 1: The semigroup $\langle g \rangle$.

in a cyclic group [3]. In particular, this implies a fast quantum algorithm for the semigroup discrete logarithm problem.

2 Finding the period and index of a semigroup element

Given a finite semigroup S, fix some element $g \in S$. The element g generates a subsemigroup $\langle g \rangle := \{g^j : j \in \mathbb{N}\}$ of S. The value

$$t := \min\{j \in \mathbb{N} : g^j = g^k \text{ for some } k \in j + \mathbb{N}\}$$

is called the *index* of g. The index exists since S is finite. The value

$$r := \min\{j \in \mathbb{N} : g^t = g^{t+j}\}$$

is called the *period* of g. These definitions are illustrated in Figure 1. If $j \geq t$, we say that g^j is in the *cycle* of g; if $j < t$, we say that g^j is in the *tail* of g.

We suppose that the elements of S are represented using $\log N$ bits, and we consider an algorithm to be efficient if it runs in time $\text{poly}(\log N)$. Since $|\langle g \rangle| = t + r$, clearly $t + r \leq N$. Typically, $\log N = \text{poly}(\log(t + r))$, in which case an efficient algorithm runs in time $\text{poly}(\log(t + r))$.

We claim that there is an efficient quantum algorithm to compute t and r. (Throughout this article, we consider bounded-error quantum algorithms.)

Lemma 1. *There is an efficient quantum algorithm to determine the index and the period of an element g of a black-box semigroup.*

Proof. First we find the period, as follows. Create the state $\frac{1}{\sqrt{M}} \sum_{j=1}^{M} |j\rangle|g^j\rangle$ for some sufficiently large M (it suffices to take $M > N^2 + N$). Note that we can compute g^j efficiently even for exponentially large j using repeated squaring, so this state can be made in polynomial time. Next, we discard the second register. To understand what happens when we do this, suppose we measure the second

register. If we obtain an element in the tail of g, then the first register is left in a computational basis state, which is useless. However, with probability at least $(M-t+1)/M \geq 1 - N/M$, we obtain an element in the cycle of g, and we are left with an r-periodic state

$$\frac{1}{\sqrt{L}} \sum_{j=0}^{L-1} |x_0 + jr\rangle$$

for some unknown $x_0 \in \{t, t+1, \ldots, t+r-1\}$, where L is either $\lfloor (M-t)/r \rfloor$ or $\lceil (M-t)/r \rceil$ (depending on the value of x_0). This is precisely the type of state that appears in Shor's period-finding algorithm (see for example [5, Algorithm 5]). After Fourier transforming this state over $\mathbb{Z}_M$ and measuring, we obtain the outcome $k \in \mathbb{Z}_M$ with probability

$$\Pr(k) = \frac{1}{LM} \left| \sum_{j=0}^{L-1} e^{2\pi i k(x_0+jr)/M} \right|^2 = \frac{\sin^2(\frac{\pi k r L}{M})}{LM \sin^2(\frac{\pi k r}{M})}.$$

A simple calculation (see for example [5, Eqs. (57)–(60)]) shows that the probability of obtaining a closest integer to one of the r integer multiples of M/r is at least $4/\pi^2$. By efficient classical postprocessing using continued fractions, we can recover r with constant probability by sampling from such a distribution [19]. Since we are in the cycle of g with overwhelming probability, the overall procedure succeeds with constant probability (which could be boosted by standard techniques).

Given the period of g, we can find its index by an efficient classical procedure. Observe that we can efficiently decide whether a given element g^j is in the tail or the cycle of g: simply compute g^r by repeated squaring and multiply by g^j to compute g^{j+r}. If $g^{j+r} = g^j$, then g^j is in the cycle of g; otherwise it is in the tail of g. Let

$$\gamma(g^j) := \begin{cases} 1 & \text{if } g^{j+r} = g^j \text{ (i.e., } g^j \text{ is in the cycle of } g) \\ 0 & \text{otherwise (i.e., } g^j \text{ is in the tail of } g). \end{cases}$$

The list $(\gamma(g), \gamma(g^2), \ldots, \gamma(g^N))$ consists of $t-1$ zeros followed by $N-t+1$ ones, so we can find t in $O(\log N)$ iterations by binary search. □

3 Computing discrete logarithms

We now show how to efficiently compute discrete logarithms in semigroups on a quantum computer.

Theorem 1. *There is an efficient quantum algorithm to compute* $\log_g x$ *on input* $x, g \in S$ *(or to determine if no such value exists).*

Proof. First, we use Lemma 1 to compute the index t and the period r of g. Then we determine whether x is in the tail or the cycle of g. As described in the proof of Lemma 1, this can be done efficiently by determining whether $xg^r = x$.

If x is in the tail of g, then we compute p, the smallest positive integer such that $\gamma(xg^p) = 1$. This can be done efficiently by using binary search to find the first 1 in the list $(\gamma(xg), \gamma(xg^2), \ldots, \gamma(xg^t))$. Then we can compute $\log_g x = t - p$.

On the other hand, suppose x is in the cycle of g. Then we use the well-known fact (see for example [9]) that $C := \{g^{t+j} \colon j \in \mathbb{Z}_r\}$ is a group with identity element g^{t+s} where $s = -t \bmod r$. In fact C is a cyclic group generated by g^{t+s+1}; in particular, for $j \geq t$ we have $g^{t+s+1}g^j = g^{j+1}$. Now we use Shor's discrete logarithm algorithm to compute $\log_{g^{t+s+1}} x$. While we cannot immediately compute the inverse of x in C, we know that the inverse of g^{t+s+1} is $g^{t+s+r-1}$, so we can compute the hiding function $f \colon \mathbb{Z}_r \times \mathbb{Z}_r \to C$ with $f(a,b) = x^a g^{(t+s+r-1)b} = x^a (g^{t+s+1})^{-b}$, which suffices to efficiently compute discrete logarithms in C. Thus we can compute $\log_g x = t + [(s + \log_{g^{t+s+1}} x) \bmod r]$.

Finally, given a candidate value a for $\log_g x$, we check whether $g^a = x$. If this check fails then we conclude that $\log_g x$ does not exist. This conclusion is correct (with bounded error) because the algorithm succeeds in finding $\log_g x$ (with bounded error) when it does exist. □

4 A shifted version of the discrete logarithm problem

While the discrete logarithm problem is no harder in semigroups than in groups, some problems that have efficient quantum algorithms in groups are more difficult in semigroups. In this section, we discuss a shifted version of the discrete logarithm problem that appears to be closely related to the dihedral hidden subgroup problem.

The shifted discrete logarithm problem is as follows: given $x, y, g \in S$, find some $a \in \mathbb{N}$ such that $x = yg^a$ (or determine that no such value exists). If S is a group, then this problem reduces to the ordinary discrete logarithm problem, since it suffices to find $a \in \mathbb{N}$ such that $g^a = y^{-1}x$. However, if S is a semigroup, then the best quantum algorithm we are aware of is the following.

Lemma 2. *There is a quantum algorithm that, on inputs $x, y, g \in S$, finds $a \in \mathbb{N}$ such that $x = yg^a$ (or determines if no such value exists) in time $2^{O(\sqrt{\log |S|})}$. Furthermore, there is an algorithm using only $\mathrm{poly}(\log |S|)$ quantum queries.*

Proof. Similarly to $j \mapsto g^j$, the function $j \mapsto yg^j$ has index

$$\tilde{t} := \min\{j \in \mathbb{N} \colon yg^j = yg^k \text{ for some } k \in j + \mathbb{N}\}$$

and period

$$\tilde{r} := \min\{j \in \mathbb{N} \colon yg^{\tilde{t}} = yg^{\tilde{t}+j}\};$$

we say that yg^j is in the cycle if $j \geq \tilde{t}$ and in the tail if $j < \tilde{t}$. The period $\tilde{r}$ and the index $\tilde{t}$ can be computed efficiently along the same lines as described in Section 2.

The case where x is in the tail can be treated as in Section 3. If x is in the cycle, so that $x = yg^{\tilde{t}+\ell}$ for some nonnegative integer ℓ, then we must solve a constructive orbit membership problem for a permutation action of the group $\mathbb{Z}_{\tilde{r}}$ on the set of elements of the form $yg^{\tilde{t}+j}$. Specifically, the action of $j' \in \mathbb{Z}_{\tilde{r}}$ is multiplication by $g^{j'}$ and we must find the element $\ell \in \mathbb{Z}_{\tilde{r}}$ transporting $yg^{\tilde{t}}$ to x. To this end, we consider the efficiently computable function $f\colon \mathbb{Z}_2 \ltimes \mathbb{Z}_{\tilde{r}} \to S$ with $f(0,j) = yg^{\tilde{t}+j}$ and $f(1,j) = xg^j$. The function $f(0,j)$ is injective since it has period $\tilde{r}$. Furthermore, $f(1,j) = xg^j = yg^{\tilde{t}+\ell+j} = f(0,j+\ell)$, i.e., $f(1,j)$ is a shift of $f(0,j)$ by ℓ. Therefore, f hides the subgroup $\langle(1,-\ell)\rangle$ of the dihedral group $\mathbb{Z}_2 \ltimes \mathbb{Z}_{\tilde{r}}$ (i.e., it is constant on the cosets of this subgroup and distinct on different cosets). It follows that the Kuperberg sieve [12] finds ℓ (and hence $a = \tilde{t} + \ell$) in time $2^{O(\sqrt{\log \tilde{r}})}$. Finally, since the dihedral hidden subgroup problem can be solved with only polynomially many quantum queries to the hiding function [6], we can solve the shifted discrete logarithm problem in a black-box semigroup S with only $\mathrm{poly}(\log |S|)$ queries.

As in the proof of Theorem 1, given a candidate value a, we can check whether $x = yg^a$. If this check fails, we can conclude (with bounded error) that no solution exists. □

The dihedral hidden subgroup problem (DHSP) is apparently hard. Despite considerable effort (motivated by a close connection to lattice problems [17]), Kuperberg's algorithm remains the best known approach, and it is plausible that there might be no efficient quantum algorithm. Note that the DHSP can be reduced to a quantum generalization of the constructive orbit membership problem, namely, orbit membership for a permutation action on pairwise orthogonal quantum states [7, Proposition 2.2]. Thus, intuitively, a solution of the shift problem for a (classical) permutation action (such as in the shifted discrete logarithm problem) should exploit that the action is on classical states, unless it also solves the DHSP.

In Section 5, we describe another variant of the discrete logarithm problem that is even harder than the shifted discrete logarithm problem, requiring exponentially many queries. We also show that our lower bound for that problem is nearly optimal using the algorithm of Lemma 2 as a subroutine.

5 Constructive semigroup membership

Given an abelian semigroup generated by $g_1, \ldots, g_k$ and a semigroup element $x \in \langle g_1, \ldots, g_k\rangle$, the *constructive membership problem* asks us to find $a_1, \ldots, a_k \in \mathbb{N}_0 := \{0, 1, 2, \ldots\}$ with $a_1 + \cdots + a_k \geq 1$ such that $x = g_1^{a_1} \cdots g_k^{a_k}$. The notation g_i^0 simply indicates that no factor of g_i is present, so solutions with $a_i = 0$ for some values of i are well defined even though the semigroup need not have an identity element.

This natural generalization of the discrete logarithm problem is easy for abelian groups (see for example [10, Theorem 5]). In that case, let $r_i := |\langle g_i \rangle|$ for all $i \in \{1, \ldots, k\}$, $r := |\langle x \rangle|$, and $L := \mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_k} \times \mathbb{Z}_r$. The values $(r_1, \ldots, r_k, r)$ can be computed efficiently by Shor's order-finding algorithm [19]. Now consider the function $f\colon L \to G$ defined by $f(a_1, \ldots, a_k, b) = g_1^{a_1} \cdots g_k^{a_k} x^{-b}$. This function hides the subgroup $H := \{(a_1, \ldots, a_k, b) \in L\colon g_1^{a_1} \cdots g_k^{a_k} = x^b\} \leq L$, so generators of H can be found in polynomial time [15]. To solve the constructive membership problem, it suffices to find the solutions with $b = 1 \bmod r$. This corresponds to a system of linear Diophantine equations, so it can be solved classically in polynomial time (see for example [18, Corollary 5.3b]).

Here we show that the constructive membership problem in semigroups is considerably harder. Specifically, given a black-box semigroup S, we need exponentially many quantum queries (in $\log |S|$) to solve the constructive membership problem with respect to $k \geq 2$ generators.

Theorem 2. *For any fixed $k \in \mathbb{N}$, there is a black-box semigroup S with k generators for which at least $\Omega(|S|^{\frac{1}{2} - \frac{1}{2k}})$ quantum queries are required to solve the constructive membership problem.*

Proof. For any $n \in \mathbb{N}$, consider the abelian semigroup

$$S = \{g_1^{a_1} \cdots g_k^{a_k} \colon a_1, \ldots, a_k \in \mathbb{N}_0,\, 1 \leq a_1 + \cdots + a_k \leq n\} \cup \{0\}$$

generated by $g_1, \ldots, g_k$, with the following multiplication rules:

$$\begin{aligned} 0(g_1^{a_1} \cdots g_k^{a_k}) &= 0 \\ (g_1^{a_1} \cdots g_k^{a_k})(g_1^{b_1} \cdots g_k^{b_k}) &= \begin{cases} g_1^{a_1+b_1} \cdots g_k^{a_k+b_k} & \text{if } \sum_{i=1}^k (a_i + b_i) \leq n \\ 0 & \text{if } \sum_{i=1}^k (a_i + b_i) > n. \end{cases} \end{aligned}$$

Let $\Sigma := \{(a_1, \ldots, a_{k-1}) \in \mathbb{N}_0^{k-1} \colon a_1 + \cdots + a_{k-1} \leq n\}$. We show that the problem of inverting a black-box permutation $\pi\colon \Sigma \to \Sigma$ (i.e., computing $\pi^{-1}(\sigma)$ for any fixed $\sigma \in \Sigma$ given a black box for π) reduces to constructive semigroup membership in a black-box version of S with respect to the generators $g_1, \ldots, g_k$. Since inverting a permutation of m points requires $\Omega(\sqrt{m})$ quantum queries [1], $|\Sigma| = \binom{n+k-1}{k-1} = \Theta(n^{k-1})$, and $|S| = \binom{n+k}{k} = \Theta(n^k)$, this shows that constructive semigroup membership requires $\Omega(\sqrt{n^{k-1}}) = \Omega(|S|^{\frac{1}{2} - \frac{1}{2k}})$ queries.

To construct the black-box semigroup, we specify an encoding

$$\mathrm{enc}\colon S \to \{(a_1, \ldots, a_k) \in \mathbb{N}_0^k \colon 1 \leq a_1 + \cdots + a_k < n\} \cup \Sigma \cup \{0\}$$

defined by

$$\begin{aligned} \mathrm{enc}(g_1^{a_1} \cdots g_k^{a_k}) &:= (a_1, \ldots, a_k) \qquad \text{if } a_1 + \cdots + a_k < n \\ \mathrm{enc}(g_1^{a_1} \cdots g_{k-1}^{a_{k-1}} g_k^{n - a_1 - \cdots - a_{k-1}}) &:= \pi(a_1, \ldots, a_{k-1}) \\ \mathrm{enc}(0) &:= 0. \end{aligned}$$

We can compute $\mathrm{enc}(gh)$ using at most one call to π given the encodings $\mathrm{enc}(g)$, $\mathrm{enc}(h)$ of any $g, h \in S$. Now suppose we can solve the constructive membership problem for some semigroup element with encoding $\sigma \in \Sigma$, with respect to the generators $g_1, \ldots, g_k$ with encodings $(1, 0, \ldots, 0), \ldots, (0, \ldots, 0, 1)$. Then we can find the values $a_1, \ldots, a_{k-1}$ such that $\mathrm{enc}(g_1^{a_1} \cdots g_{k-1}^{a_{k-1}} g_k^{n-a_1-\cdots-a_{k-1}}) = \sigma$, so that $(a_1, \ldots, a_{k-1}) = \pi^{-1}(\sigma)$, thereby inverting π. □

Note that Theorem 2 gives a lower bound on the worst-case query complexity. In fact, the same lower bound holds if we are given a random element of Σ. However, we leave the problem of the average-case quantum query complexity where, say, x is chosen uniformly from the semigroup, as an open problem.

We also show that for any fixed k, the lower bound of Theorem 2 is nearly tight.

Theorem 3. *For any fixed $k \in \mathbb{N}$, there is a quantum algorithm to solve the constructive membership problem for $x \in S = \langle g_1, \ldots, g_k \rangle$ with respect to $g_1, \ldots, g_k$ in time $|S|^{\frac{1}{2}-\frac{1}{2k}+o(1)}$. Furthermore, the quantum query complexity of this problem is at most $|S|^{\frac{1}{2}-\frac{1}{2k}} \operatorname{poly}(\log |S|)$.*

To prove this, we use the following simple observations.

Lemma 3. *Let S be a finite abelian semigroup and let $x, g_1, \ldots, g_k \in S$. Let $(a_1, \ldots, a_k)$ be the lexicographically first k-tuple from $\mathbb{N}_0^k$ such that $x = g_1^{a_1} \cdots g_k^{a_k}$. Then $(a_1 + 1) \cdots (a_k + 1) \leq |S|$.*

Proof. Assume for a contradiction that $(a_1 + 1) \cdots (a_k + 1) > |S|$. Then, by the pigeonhole principle, there must exist $c_1, \ldots, c_k, d_1, \ldots, d_k \in \mathbb{N}_0$ with $c_i, d_i \leq a_i$ (for all $i = 1, \ldots, k$) such that $g_1^{c_1} \cdots g_k^{c_k} = g_1^{d_1} \cdots g_k^{d_k}$ and $(c_1, \ldots, c_k) \neq (d_1, \ldots, d_k)$. Suppose without loss of generality that $(c_1, \ldots, c_k)$ is lexicographically smaller than $(d_1, \ldots, d_k)$. Let $b_i := a_i + c_i - d_i$ for all i, and note that $a_i - d_i \geq 0$. Thus $g_1^{a_1} \cdots g_k^{a_k} = g_1^{d_1} \cdots g_k^{d_k} g_1^{a_1-d_1} \cdots g_k^{a_k-d_k}$ and $g_1^{b_1} \cdots g_k^{b_k} = g_1^{c_1} \cdots g_k^{c_k} g_1^{a_1-d_1} \cdots g_k^{a_k-d_k}$. This implies $g_1^{b_1} \cdots g_k^{b_k} = x$. Also, for the first index i with $c_i \neq d_i$, we have $c_i < d_i$. Therefore $(b_1, \ldots, b_k)$ is lexicographically smaller than $(a_1, \ldots, a_k)$, a contradiction. □

Lemma 4. *For any $r, L \in \mathbb{N}$, let*

$$D(r, L) := \{(a_1, \ldots, a_r) \in \mathbb{N}_0^r \colon (a_1 + 1) \cdots (a_r + 1) \leq L\}.$$

Then for fixed r, $|D(r, L)| = O(L \log^{r-1} L)$.

Proof. By induction on r, we show that $|D(r, L)| \leq L(\frac{3}{2} \log_2 L)^{r-1}$ for every integer $L > 1$. Clearly $|D(1, L)| = L$. We have $(a_1, \ldots, a_{r+1}) \in D(r + 1, L)$ if and only if $(a_1, \ldots, a_r) \in D(r, \lfloor L/(a_{r+1} + 1) \rfloor)$. Therefore

$$\begin{aligned} |D(r+1, L)| &= \sum_{a=1}^{L} |D(r, \lfloor L/a \rfloor)| \leq \sum_{a=1}^{L} \lfloor L/a \rfloor (\tfrac{3}{2} \log_2 \lfloor L/a \rfloor)^{r-1} \\ &\leq \sum_{a=1}^{L} (L/a)(\tfrac{3}{2} \log_2 L)^{r-1} \leq L(\tfrac{3}{2} \log_2 L)^r \end{aligned}$$

where we used the fact that for every integer $L > 1$, $\sum_{a=1}^{L} \frac{1}{a} < \frac{3}{2} \log_2 L$. □

We are now ready to prove the lower bound for constructive semigroup membership.

Proof of Theorem 3. By Lemma 3, there are some $a_1, \ldots, a_k \in \mathbb{N}_0$ with $x = g_1^{a_1} \cdots g_k^{a_k}$ and some $j \in \{1, \ldots, k\}$ such that $\prod_{i \neq j}(a_i + 1) \leq |S|^{(k-1)/k}$. To see this, note that $\prod_{j=1}^{k} \prod_{i \neq j}(a_i + 1) = \left(\prod_{j=1}^{k}(a_j + 1)\right)^{k-1} \leq |S|^{k-1}$. Thus, for each $j \in \{1, \ldots, k\}$, we perform a Grover search [8] over the set

$$\{(a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_k) \in \mathbb{N}_0^{k-1} \colon \prod_{i \neq j}(a_i + 1) \leq |S|^{(k-1)/k}\},$$

where for each $(k-1)$-tuple we use Lemma 2 (with $y = \prod_{i \neq j} g_i^{a_i}$ and $g = g_j$) to find a_j such that $x = g_1^{a_1} \cdots g_k^{a_k}$ (or to exclude its existence). By Lemma 4, the running time of this procedure is $k|S|^{\frac{k-1}{2k} + o(1)} = |S|^{\frac{1}{2} - \frac{1}{2k} + o(1)}$. Using the query-efficient (but not time-efficient) algorithm for the dihedral hidden subgroup problem in place of Kuperberg's algorithm, we require only $|S|^{\frac{1}{2} - \frac{1}{2k}} \operatorname{poly}(\log |S|)$ queries. □

While Theorem 2 shows that the constructive membership problem is provably hard in black-box semigroups, the problem is also known to be NP-hard in explicit semigroups. In particular, Beaudry proved NP-completeness of membership testing in abelian semigroups of transformations of (small) finite sets [4].

6 Discussion

We have considered quantum algorithms for the semigroup discrete logarithm problem and some natural generalizations thereof. While discrete logarithms can be computed efficiently by a quantum computer even in semigroups, the shifted semigroup discrete logarithm problem appears comparable in difficulty to the dihedral hidden subgroup problem, and the constructive membership problem in a black-box semigroup with respect to multiple generators is provably hard. Thus, while hardness of the discrete logarithm problem in semigroups is not a good assumption for quantum-resistant cryptography, one might build quantum-resistant cryptosystems based on the presumed hardness of other problems in semigroups.

Testing membership in abelian semigroups is related to a cryptographic problem known as the semigroup action problem (SAP) [13]. Given an (abelian) semigroup S acting on a set M and two elements $x, y \in M$, the SAP asks one to find an element $s \in S$ such that $x = sy$. Constructive membership testing in a monoid (i.e., a semigroup with an identity element, which can be adjoined artificially if necessary) is an instance of SAP: consider S acting on itself by multiplication and let y be the identity. (More precisely, to obtain a decomposition with respect to generators $g_1, \ldots, g_k$, consider the natural action of

$\langle g_1 \rangle \times \cdots \times \langle g_k \rangle$ on S.) On the other hand, the SAP over an abelian semigroup can be reduced to membership of x in a subsemigroup generated by y and S of the abelian semigroup $S' = S \cup M \cup \{0\}$ with a semigroup operation that naturally extends the multiplication of S and the action of S on M. In particular, the SAP for a cyclic semigroup action reduces to an instance of the shifted discrete logarithm problem discussed in Section 4.

A natural open question raised by our work is the quantum complexity of the shifted semigroup discrete logarithm problem: is this task indeed as hard as the DHSP, or is there a faster algorithm using additional structure? In general, it might also be interesting to develop new quantum-resistant cryptographic primitives based on hard semigroup problems.

Acknowledgments

We thank Rainer Steinwandt for suggesting the problem of computing discrete logarithms in semigroups and for helpful references. We thank Robin Kothari for pointing out that the lower bound of Theorem 2 generalizes from $k = 2$ to $k > 2$. We also thank the Dagstuhl research center and the organizers of its 2013 seminar on Quantum Cryptanalysis, where this work was started. AMC received support from NSERC, the Ontario Ministry of Research and Innovation, and the US ARO. GI received support from the Hungarian Research Fund (OTKA, Grant NK105645) and from the Centre for Quantum Technologies at the National University of Singapore.

References

[1] Andris Ambainis, Quantum lower bounds by quantum arguments, *Journal of Computer and System Sciences* **64** (2002), 750–767, preliminary version in STOC 2000.

[2] László Babai and Endre Szemerédi, On the complexity of matrix group problems I, in: *25th Symposium on Foundations of Computer Science*, pp. 229–240, 1984.

[3] Matan Banin and Boaz Tsaban, *A reduction of semigroup DLP to classic DLP*, arXiv:1310.7903.

[4] Martin Beaudry, Membership testing in commutative transformation semigroups, *Information and Computation* **79** (1988), 84–93, preliminary version in ICALP 1987.

[5] Andrew M. Childs and Wim van Dam, Quantum algorithms for algebraic problems, *Reviews of Modern Physics* **82** (2010), 1–52.

[6] Mark Ettinger and Peter Høyer, On quantum algorithms for noncommutative hidden subgroups, *Advances in Applied Mathematics* **25** (2000), 239–251.

[7] Katalin Friedl, Gábor Ivanyos, Frédéric Magniez, Miklos Santha and Pranab Sen, Hidden translation and translating coset in quantum computing, *SIAM Journal on Computing* **43** (2014), 1–24, preliminary version in STOC 2003.

[8] Lov K. Grover, Quantum mechanics helps in searching for a needle in a haystack, *Physical Review Letters* **79** (1997), 325–328, preliminary version in STOC 1996.

[9] John M. Howie, *Fundamentals of semigroup theory*, LMS Monographs 12, Oxford University Press, 1995.

[10] Gábor Ivanyos, Frédéric Magniez and Miklos Santha, Efficient quantum algorithms for some instances of the non-abelian hidden subgroup problem, *International Journal of Foundations of Computer Science* **14** (2003), 723–739, preliminary version in SPAA 2001.

[11] Delaram Kahrobaei, Charalambos Koupparis and Vladimir Shpilrain, Public key exchange using matrices over group rings, *Groups Complexity Cryptology* **5** (2013), 97–115.

[12] Greg Kuperberg, A subexponential-time quantum algorithm for the dihedral hidden subgroup problem, *SIAM Journal on Computing* **35** (2005), 170–188.

[13] Gérard Maze, Chris Monico and Joachim Rosenthal, Public Key Cryptography based on Semigroup Actions, *Advances in Mathematics of Communications* **1** (2007), 489–507, preliminary version in ISIT 2002.

[14] Alfred J. Menezes and Yi-Hong Wu, The discrete logarithm problem in $GL(n, q)$, *Ars Combinatoria* **47** (1997), 23–32.

[15] Michele Mosca and Artur Ekert, The hidden subgroup problem and eigenvalue estimation on a quantum computer, in: *Proceedings of the 1st NASA International Conference on Quantum Computing and Quantum Communication*, Lecture Notes in Computer Science 1509, Springer-Verlag, 1999.

[16] Alexei D. Myasnikov and Alexander Ushakov, *Quantum algorithm for the discrete logarithm problem for matrices over finite group rings*, Cryptology ePrint Archive, Report 2012/574.

[17] Oded Regev, Quantum computation and lattice problems, *SIAM Journal on Computing* **33** (2004), 738–760, preliminary version in FOCS 2002.

[18] Alexander Schrijver, *Theory of Linear and Integer Programming*, Wiley-Interscience, 1986.

[19] Peter W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Journal on Computing* **26** (1997), 1484–1509, preliminary version in FOCS 1994.