# A REDUCTION OF SEMIGROUP DLP TO CLASSIC DLP

MATAN BANIN AND BOAZ TSABAN

Abstract. We present a polynomial-time reduction of the discrete logarithm problem in any periodic (or torsion) semigroup (Semigroup DLP) to the classic DLP in a sub*group* of the same semigroup. It follows that Semigroup DLP can be solved in polynomial time by quantum computers, and that Semigroup DLP has subexponential complexity whenever the classic DLP in the corresponding groups has subexponential complexity. We also consider several natural constructions of nonperiodic semigroups, and provide polynomial time solutions for the DLP in these semigroups.

## 1. introduction

For Discrete Logarithm Problem (DLP) based cryptography, it is desirable to find efficiently implementable groups for which sub-exponential algorithms for the DLP are not available. Thus far, the only candidates for such groups seem to be (carefully chosen) groups of points on elliptic curves [4, 7]. Groups of invertible matrices over a finite field, proposed in [9], where proved by Menezes and Wu [6] inadequate for this purpose. In their paper [3], Kahrobaei, Koupparis and Shpilrain propose to use *semigroups*—sets equipped with an associative multiplication, but where element are not necessarily invertible—as a platform for the Diffie–Hellman protocol. Specifically, they propose to use the semigroup of all matrices over a certain finite group-ring as a platform for the Diffie–Hellman protocol.

Let $S$ be a semigroup. The *order* of an element $g \in S$ is the cardinality of the set $\{\, g^k : k \in \mathbb{N} \,\}$, where $\mathbb{N} = \{1, 2, 3, \dots\}$ is the set of natural numbers. A semigroup $S$ is *periodic* (or *torsion*) if each element of $S$ has finite order. The DLP in a periodic semigroup $S$ is the problem of finding, given an element $g \in S$ and a power $h$ of $g$, a natural number $k$ such that $g^k = h$. *Semigroup DLP* is the general problem of solving the DLP in periodic semigroups.[1]

We will demonstrate that the Semigroup DLP is not harder than the classic DLP in groups. Moreover, the DLP in a semigroup $S$ reduces, in polynomial time, to the DLP in a subgroup $G$ of $S$. Thus, if there is a subexponential time algorithm for the DLP in subgroups of $S$ (which is the case, for example, when $S$ is a semigroup of matrices over a finite field, by the Menezes–Wu result [6]), then there is one for the DLP in $S$. In particular, as the DLP in groups is efficiently solvable by quantum computers, it follows that the Semigroup DLP is efficiently solvable by quantum computers.

**Related work.** The Semigroup DLP is known at least since K. McCurley's 1989 survey [5]. In [8], A. Myasnikov and A. Ushakov reduce the DLP in the semigroup proposed in

2010 *Mathematics Subject Classification.* 94A60.

*Key words and phrases.* discrete logarithm problem, quantum algorithms, semigroups.

[1] The nonperiodic case is discussed briefly in Section 3 below.

[3] to the DLP in a group of invertible matrices over a finite field, and deduce that the DLP in that semigroup can be solved by quantum computers. They achieve this goal by embedding the semigroup in the semigroup of all matrices over a finite field, and then applying Jordan Canonical Form theory to reduce the problem to the DLP in the group of invertible matrices over the same field. Our solution shows, in particular, that specialized methods are not necessary to solve this problem. We have reported our solution, without details, to Myasnikov and Ushakov, and suggested that they ask experts whether this was known. Following that, they consulted R. Steinwandt, who mentioned the problem to some, including A. Childs and G. Ivanyos. Unaware of our solution, Childs and Ivanyos came up with an independent proof that the Semigroup DLP can be solved by quantum computers [2, Sections 1–3]. The Childs–Ivanyos solution uses quantum computational assumptions, whereas our reduction uses only classic computational assumptions. Our approach is useful when quantum computers are not available, but subexponential algorithms are available for the DLP in the relevant groups, e.g., in matrix semigroups over finite fields.

## 2. A solution of the Semigroup DLP using a classic DLP oracle

2.1. **Assumptions.** Let $G$ be a finite cyclic group. We assume that the group $G$ is described by a specification of a generator $g$ of $G$ and an oracle for multiplication in $G$. We *do not* assume the explicit availability of an oracle for inversion in $G$.

By *DLP oracle for $G$* we mean an oracle that, whenever provided with the group $G$ (i.e., its generator $g$ and a multiplication oracle) and a power $h$ of $g$, returns a natural number $k$, of size polynomial in the relevant parameters (including the length of $g$ and the order of $G$), such that $h = g^k$.

The assumption that the oracle does not request an explicit inversion oracle is important for our reduction. However, we are not aware of any practical constraint imposed by it. For example:

(1) The known DLP algorithms do not request inversion, or can be easily transformed into ones without inversion.
(2) If $m$ is any nonzero multiple of the order of a group element $h$, then $h^{m-1} = h^{-1}$. Thus, it suffices to have a method to find a nonzero multiple of the order of a group element.

*Example* 1. Shor's proof that quantum computers can solve the DLP efficiently shows that quantum computers can find, efficiently, the order of any group element.

*Example* 2. Assume that $g$ is an element of a group $H$ of known cardinality $m := |H|$, and the number $\log m$ is polynomial in the relevant parameters. Then $g^m = 1$.

Let $A$ be an $n \times n$ matrix over a finite field $\mathbb{F}$. The vector space $\mathbb{F}^n$ decomposes to the direct sum $\ker(A^n) \oplus \operatorname{im}(A^n)$. Let $r$ be the dimension of $\operatorname{im}(A^n)$. Then the matrix $A^n$ is conjugate to a direct sum of an invertible $r \times r$ matrix and the zero matrix. It follows that the group generated by $A^n$ is isomorphic to $\mathrm{GL}_r(\mathbb{F})$ the group of all invertible $r \times r$ matrices of $\mathbb{F}$, which is a subgroup of $\mathrm{GL}_n(\mathbb{F})$. Thus, for $m := |\,\mathrm{GL}_n(\mathbb{F})|$, we have that $(A^n)^m$ is the identity element of the group generated by $A^n$.

Once we show that the DLP in the semigroup generated by the (not necessarily invertible) matrix $A$ can be reduced to the DLP in the mentioned group with a multiplication oracle, there is no need for an explicit inversion oracle in that group. We thus obtain a generalization of the quantum-based cryptanalysis of the Kahrobaei–Koupparis–Shpilrain scheme [3] in [8]. Moreover, this provides a subexponential cryptanalysis using standard computational power.

In the Semigroup DLP, we assume that each element in the considered semigroups has a unique representation; equivalently for our purposes, a canonical representative that can be computed in polynomial time. We also assume that multiplication in the given semigroup can be carried out in polynomial time.

### 2.2. **The reduction.**

The following lemma should be well known. For completeness, we include a proof.

**Lemma 3.** *Let $S$ be a periodic semigroup, and $g$ be a member of $S$. Let $l, n$ be minimal*[2] *with $g^{l+n} = g^l$, and let $t$ be minimal with $tn > l$. Then the set*

$$G := g^l \cdot \{\, g^k : 0 \leq k < n \,\} = \{\, g^{l+k} : 0 \leq k < n \,\}$$

*is a cyclic group of order $n$, with identity element $g^{tn}$ and generator $g^{tn+1}$. Moreover, $g^{tn} = g^{sn}$ for all $s \geq t$.*

*Proof.* As $g^{l+n} = g^l$, the set $G$ is closed under products. The element $g^{tn}$ is neutral: As

$$g^{tn} g^l = g^{tn+l} = g^{l+tn} = g^l,$$

we have that $g^{tn} g^l g^k = g^l g^k$ for each element $g^l g^k \in G$. Let $s \geq t$. Then $g^{sn} = g^{tn+(s-t)n} = g^{tn}$.

Inversion: Given $g^{l+k} \in G$, let $d$ be such that $l + k + d = tn - l \pmod{n}$. Then

$$g^{l+k} g^{l+d} = g^l g^{l+k+d} = g^l g^{tn-l} = g^{tn}.$$

Generator: As $g^{tn}$ is neutral, for each element $g^a$ in $G$ we have that

$$g^a = (g^{tn})^a g^a = g^{tna+a} = (g^{tn+1})^a. \quad \square$$

Let $S$ be a semigroup and let $g$ be an element of $S$. Let $l$, $n$, $t$ and $G$ be as in Lemma 3.

**Reduction 4.** *Finding $n$, using a DLP oracle for $G$.*

*Procedure.* Fix a number $N$ with $N \gg l + n$. This can be done by beginning with a fixed number $N$, and doubling it until the following procedure works. Choose random number $k \in \{\lceil N/2 \rceil, \dots, N\}$, and compute $h := g^k$. The distribution of $h$ is statistically indistinguishable from the uniform distribution on $G$. As $\varphi(n)/n$ is greater than $1/(e^\gamma \log\log n + 3/\log\log n)$ for $n > 2$ [1, Theorem 8.8.7] (and is at least $1/2$ for $n = 1$ or $2$), we assume, for a while, that $h$ is a generator of $G$.

It is known that the order of a group $G$ can be computed given a generator $h$ and a DLP oracle for that group. Briefly, this can be done, using our notation, as follows. Choose a random number $k \in \{1, \dots, N\}$, and compute $k' := \log_h(h^k)$. It may be that calling the oracle twice with the same input, we obtain different values of $k'$. However, the distribution of $k'$ depends only on $k \bmod n$ and not on $k$ itself. Thus, taking the greatest common divisor of $O(1)$ differences $k - k'$ of this kind, we will obtain $n$.

[2] It does not matter whether we first minimize $l$ and then $n$, or vice versa.

Now, in any case, $h$ generates a subgroup of $G$, whose order is found by the above-mentioned algorithm. This order divides the order of $G$. Repeating this procedure for $O(\log\log n)$ elements $h$, the maximum (or least common multiple) of the obtained orders will be the order of $G$. □

We now find $t$, using our knowledge of $n$. The identity element of a group is its unique element $e$ satisfying $e = e^2$, an idempotent element. By Lemma 3, we need to find the minimal $t$ such that $g^{tn}$ is an idempotent. Given that $g^{sn} = g^{tn}$ for all $s \geq t$ and $g^{sn} \neq g^{tn}$ for $s < t$, this can be done by binary search, as in the following algorithm.

**Algorithm** *Finding the minimal $t$ such that $g^{tn}$ is the identity element of $G$*

```
$b \leftarrow 1$
while $g^{nb} \neq (g^{nb})^2$
        $b \leftarrow 2b$
$e \leftarrow g^{nb}$ (this is the identity element of $G$)
$a \leftarrow \frac{b}{2}$
repeat
    $c \leftarrow \frac{a+b}{2}$
    if $g^{nc} \neq e$
        then
            $a \leftarrow c$
        else
            $b \leftarrow c$
until $b - a = 1$
return $b$
```

*Remark* 5. One can use a variation of the above algorithm, that precomputes a logarithmic number of powers $g^{2^i}$, and replaces each power computation by one multiplication. This applies to all algorithms in this paper.

Let $g^x$ be given. There are two cases to consider. First, assume that $g^x \in G$. Find $t$ and $n$ as above. Compute $tn$. Let $r = g^{tn+1}$, a generator of $G$. Using the oracle, we obtain a number $x'$ such that $r^{x'} = g^x$. Then $g^x = r^{x'} = g^{x'(tn+1)}$. Take $k = x'(tn+1)$. Then $g^k = g^x$, and we are done. We next treat the remaining case.

The following immediate fact shows that membership in $G$ can be tested efficiently.

**Lemma 6.** *For each $x \in \mathbb{N}$, we have that $g^x \in G$ if and only if $g^n g^x = g^x$.* □

If $g^x \notin G$, we use binary search to find the minimal $b$ such that $g^{bn} g^x \in G$, as follows.

**Algorithm** *Finding the minimal $b$ such that $g^{bn} g^x \in G$*

```
$b \leftarrow 1$
while $g^{bn} g^x \notin G$
        $b \leftarrow 2b$
$a \leftarrow \frac{b}{2}$
repeat
    $c \leftarrow \frac{a+b}{2}$
    if $g^{cn} g^x \notin G$
        then
```

```
            $a \leftarrow c$
        **else**
            $b \leftarrow c$
**until** $b - a = 1$
**return** $b$
```

Similarly, if $g^k \in G$ and $k$ is known, we can use binary search to find the maximal $c$ such that $k - cn > 0$ and $g^{k-cn} \in G$.

**Reduction 7.** *Computing a discrete logarithm of $g^x$, using a DLP oracle for $G$.*

*Procedure.* It remains to consider the case where $g^x \notin G$. Let $r = g^{tn+1}$ be the generator of $G$. Use the above algorithm to find the minimal $b$ such that $g^{bn}g^x \in G$. As $n$ is the order of $G$, for each $a$ with $g^{bn+x} = g^a$, we have that $bn + x \leq a$.

Using the oracle, compute $x' := \log_r g^{bn+x}$. Then $g^{bn+x} = g^{x'(tn+1)}$, and thus $bn + x \leq x'(tn+1)$. Note that the number $x'(tn+1)$ is known. Using binary search, find the maximal $c$ such that $g^{x'(tn+1)-cn} \in G$. Then $bn+x = x'(tn+1) - cn$, and thus $x = x'(tn+1) - cn - bn$ is found. □

## 3. Comments on nonperiodic semigroups

Our consideration of periodic semigroups is natural in the context of the DLP, but it is still interesting to consider the case where the element $g$ of $S$ has *infinite* order. In this case, the semigroup $\langle g \rangle$ generated by $g$ is isomorphic to the additive semigroup of natural numbers $\mathbb{N}$. However, the isomorphism may be infeasible to compute.

The consideration of the DLP in infinite semigroups was proposed by Vladimir Shpilrain, at the conference *Algebraic Methods in Cryptography*, Ruhr Universität Bochum, Germany, 2005. As the second named author commented in that conference, there is a correlation between the bitlength of $g^k$ and the power $k$ (for each fixed coding of the semigroup elements). In such a case, for distinct powers $g^k$ and $g^l$, one may take many random powers $g^{kr}, g^{lr}$, compare their lengths, and decide by majority whether $k < l$. This implies a binary search algorithm for solving the DLP in the ambient semigroup. We have verified, by experiments, that this approach succeeds 100% of the time in Artin's braid group, with the Garside normal form as the length function.

Let $p$ be a prime number, and $\pi$ be a random permutation on $\{0, \dots, p-1\}$. The permutation $\pi$ defines a group structure $G_\pi = (\{0, \dots, p-1\}, +_\pi)$ by viewing $\pi$ as an isomorphism from the group $(\mathbb{Z}_p, +)$ to $G_\pi$. In other words, addition in $G_\pi$ is defined by

$$x +_\pi y := \pi(\pi^{-1}(x) + \pi^{-1}(y)) \bmod p,$$

and the generator of $G_\pi$ is $\pi(1)$. Shoup's classic theorem asserts that the DLP in $G_\pi$ cannot be solved with less than $O(\sqrt{p})$ queries to the group operation [10]. Consequently, the following problem is very appealing.

**Problem 8.** Let $p$ be a prime number. Can an *infinite* cyclic semigroup $C$ be constructed, using a random permutation $\pi$ on $\{0, \dots, p-1\}$ as an oracle, such that the DLP in $C$ cannot be solved with less than $O(\sqrt{p})$ group operations?

For this problem, one must define a reasonable distribution on $\mathbb{N}$ for choosing the DLP exponent. To this end, one may either use a Gaussian distribution with large enough deviation, as is often done in Lattice-based cryptography, or fix a large enough interval $\{1, 2, \dots, N\}$ and choose the exponent uniformly from that interval.

A simple approach to Problem 8 is to define a permutation $P$ on $\mathbb{N}$ as follows: Given a natural number $k$, represent $k$ in base $p$ and apply the permutation $\pi$ to the least significant digit of $k$.

**Definition 9.** For a permutation $P$ on $\mathbb{N}$, let $S_P := (\mathbb{N}, +_P)$ be the semigroup obtained by declaring $P$ an isomorphism from $(\mathbb{N}, +)$ to $(\mathbb{N}, +_P)$, that is, where

$$x +_P y := P(P^{-1}(x) + P^{-1}(y)).$$

The semigroup $S_P$ is generated by its element $P(1)$. In the DLP for $S_P$ we are provided with the generator $g := P(1)$ and, since we use additive notation, the element $kg = P(k)$ for some random secret $k$ in a large interval, and we need to find $k$. The addition operator $+_P$ is given as an oracle.

Consider the simple case with $P$ defined using a permutation $\pi$ on $\{0, \dots, p-1\}$ as in the last paragraph preceding Definition 9. Assume that the exponent $k$ is in $\{1, \dots, N\}$, where $N$ is exponential in the security parameters. If $p$ is much larger than $N$ (say, exponential in $N$), then the element $\pi(1)$ is likely to be too large to store. Thus, for the problem to be reasonable, we request that $p$ is smaller, or not much larger than $N$. Still, the length-based algorithm described above solves the DLP in semigroups constructed this way. We thus consider the following, more general construction.

**Definition 10.** Let $m_0 = 0$. For each $n = 0, 1, 2, \dots$, let $m_{n+1} > m_n$ be a natural number and $P_n$ be a permutation on $\{m_n, \dots, m_{n+1} - 1\}$. The *block permutation* $P = \bigoplus_{n=1}^{\infty} P_n$ on $\mathbb{N}$ is defined as follow: Given a natural number $k$, let $n$ be the unique natural number such that $m_n \leq k < m_{n+1}$. Then

$$P(k) := P_n(k).$$

Let $P$ be a block permutation, with $m_n$ growing rapidly. E.g., $m_n := 2^{2^n}$. Then the length-based algorithm fails in $S_P$. Indeed, consider distinct $k$ and $l$. For large random $r$, the values $kr$ and $lr$ are, with high probability, in the same block $\{m_n, \dots, m_{n+1} - 1\}$, and consequently, nothing about the size of $k$ and $l$ can be learned from $P(kr)$ and $P(lr)$.

However, for each block permutation $P$ on $\mathbb{N}$, the DLP in the semigroup $S_P$ associated to $P$ can be solved in polynomial time. Indeed, given $P(k)$, let $n$ be such that $m_n \leq P(k) < m_{n+1}$; equivalently, $m_n \leq k < m_{n+1}$. For each $m$, we are able to compute $P(m) = mP(1)$ and $P(k + m) = P(k) +_P P(m)$. Since $m_{n+1} \leq P(r)$ if and only if $m_{n+1} \leq r$, we can find the minimal $m$ such that $m_{n+1} \leq P(k + m)$ by binary search on $m$, and this is also the minimal $m$ such that $m_{n+1} \leq k + m$, that is, $k + m = m_{n+1}$. We then have $k = m_{n+1} - m$.

This approach can be extended to more complicated situations. We conclude this discussion with an interesting example.

Let $P$ be a block permutation as above, with $m_n$ growing rapidly. For each $n$, let $k_n := (m_n + m_{m+1})/2$, and let $Q$ be a block permutation consisting of random permutations $Q_n$ on the blocks $\{k_n, \dots, k_{n+1} - 1\}$. Let $R = QP$, the composition of $Q$ and $P$, and consider the semigroup $S_R$ associated to the permutation $R$. For a pair of elements $k$ and $l$ that we wish to compare, either both are in the same $P$-block, or both are in the same $Q$-block, so there seems to be little hope to see which is larger. We provide, however, with an efficient solution for the DLP in the semigroup $S_R$.

Let $R(l)$ and $n$ be given. We will show how to test whether $m_n \leq l$, assuming that the numbers $m_n$ and $k_n$ are known. Let $M := 127$, for example. Compute $R(m_n)$, and the values $R(l), R(l+1), \dots, R(l+M)$. We consider all possible cases.
*Case 1.* Assume that one of the computed values $R(l+i)$ is $R(m_n)$. Then $l+i = m_n$, and then $l = m_n$ if $i = 0$, and $l < m_n$ if $i > 0$, and we are done.
*Case 2.* Assume that we are not in Case 1, that is, $m_n \notin \{l, l+1, \dots, l+M\}$. Then $l < m_n - M$ or $m_n < l$.
*Case 2.1.* Assume that $l < m_n - M$. Since $l, l+1, \dots, l+M < m_n$, we have that $P(l), P(l+1), \dots, P(l+M) < m_n \leq k_n$, and thus $QP(l), QP(l+1), \dots, QP(l+M) < k_n$. Thus, in this case, all values $R(l), \dots, R(l+M)$ are smaller than $k_n$.
*Case 2.2.* Assume that $m_n < l$. If $m_{n+1} \leq l+M$, then $k_n \leq m_{n+1} \leq P(l+M)$, and thus $k_n \leq QP(l+M) = R(l+M)$, and this proves that we are not in Case 2.1, and thus that $m_n < l$. It remains to consider the case where $m_n < l, l+1, \dots, l+M < m_{n+1}$. In this case, the numbers $P(l), P(l+1), \dots, P(l+M)$ are $M+1$ uniformly distributed numbers in the set $\{m_n, m_n+1, \dots, m_{n+1}-1\}$, subject only to the condition that they are distinct. Since $k_n$ is the middle point of this set, we have that, with overwhelming probability, at least one of these numbers $P(l+i)$ is greater than $k_n$. Then, as in Case 2.1, we have that $k_n \leq QP(l+i) = R(l+i)$, this proves that $m_n < l$.

In summary, the following decision procedure, for the question whether $l \leq m_n$, is correct with overwhelming probability: If $R(m_n) \in \{R(l), \dots, R(l+M)\}$, decide according to the $i$ with $R(m_n) = R(l+i)$. If $R(m_n) \notin \{R(l), \dots, R(l+M)\}$: If $\max\{R(l), \dots, R(l+M)\} < k_n$, decide that $l < m_n$. Otherwise, decide that $m_n < l$.

The problem of finding a meaningful infinite cyclic semigroup where the DLP has no polynomial-time solution remains, thus, challenging.

**Acknowledgments.** We thank the referees for their work on the evaluation of this paper and for their useful comments.

## References

[1] E. Bach, J. Shallit, **Algorithmic Number Theory**, Vol I: Efficient Algorithms, MIT Press Series in the Foundations of Computing, Cambridge, MA, 1996.

[2] A. Childs, G. Ivanyos, *Quantum computation of discrete logarithms in semigroups*, Journal of Mathematical Cryptology **8** (2014), 405–416.

[3] D. Kahrobaei, C. Koupparis, V. Shpilrain, *Public key exchange using matrices over group rings*, Groups Complexity Cryptology **5** (2013), 97–115.

[4] N. Koblitz, *Elliptic curve cryptosystems*, Mathematics of computation **48.177** (1987), 203–209.

[5] K. McCurley, *The Discrete Logarithm Problem*, Proceedings of Symposia in Appplied Mathematics **42** (1989), American Mathematical Society, 49–74.

[6] A. Menezes, Y. Wu, *The discrete logarithm problem in $GL(n,q)$*, Ars Combinatoria **47** (1997), 23–32.

[7] V. Miller, *Use of elliptic curves in cryptography*, Advances in Cryptology—CRYPTO'85 Proceedings, Springer–Berlin–Heidelberg, 1986.

[8] A. Myasnikov, A. Ushakov, *Quantum algorithm for discrete logarithm problem for matrices over finite group rings*, Groups Complexity Cryptology **6** (2014), 31–36.

[9] R. Odoni, V. Varadharajan, P. Sanders, *Public key distribution in matrix rings*, Electronics Letters **20** (1984), 386–387.

[10] V. Shoup, *Lower bounds for discrete logarithms and related problems*, Advances in Cryptology—EUROCRYPT 97, Lecture Notes in Computer Science **1233** (1997), 256–266.

*E-mail address*, Matan Banin: `baninmmm@gmail.com`

*E-mail address*, Boaz Tsaban: `tsaban@math.biu.ac.il`
*URL*, Boaz Tsaban: `http://www.cs.biu.ac.il/~tsaban`

Department of Mathematics, Bar-Ilan University, Ramat Gan 5290002, Israel