A Subexponential Quantum Algorithm for the Semidirect Discrete Logarithm Problem

Christopher Battarbee[1], Delaram Kahrobaei[1,2,3,5], Ludovic Perret[4], and Siamak F. Shahandashti[1]

[1]Department of Computer Science, University of York, UK
[2]Departments of Computer Science and Mathematics, Queens College, City University of New York, USA
[3]Initiative for the Theoretical Sciences, Graduate Center, City University of New York, USA
[4]Sorbonne University, CNRS, LIP6, PolSys, Paris, France
[5]Department of Computer Science and Engineering, Tandon School of Engineering, New York University, USA

September 6, 2022

Abstract

Group-based cryptography is a relatively young family in post-quantum cryptography. In this paper we give the first dedicated security analysis of a central problem in group-based cryptography: the so-called Semidirect Product Key Exchange (SPDKE). We present a subexponential quantum algorithm for solving SPDKE. To do this we reduce SPDKE to the Abelian Hidden Shift Problem (for which there are known quantum subexponential algorithms). We stress that this does not per se constitute a break of SPDKE; rather, the purpose of the paper is to provide a connection to known problems.

1 Introduction

The goal of Post-Quantum Cryptography (PQC) is to design cryptographic cryptosystems which are secure against classical and quantum adversaries. A topic of fundamental research for decades, the status of PQC drastically changed with the NIST standardisation process [1].

In July 2022, after five years and three rounds of selection, NIST selected a first set of PQC standards for Key-Encapsulation Mechanism (KEM) and Digital Signature Scheme DSS based on lattices and hash functions. The standardization process is still ongoing with a fourth round for KEM and a new NIST call for post-quantum DSS in 2023. Recent attacks [2, 3, 4] against round-3 multivariate signature schemes, Rainbow [2] and G*e*MSS [5], as well as the cryptanalysis of round-4 isogeny based KEM SIKE [6, 7], emphasise the need to continue the cryptanalysis effort in PQC as well as the diversity in the potential post-quantum hard problems.

A relatively young family of such problems come from *group-based* cryptography, see [8]. The protocol of this type we are interested in is called Semidirect Product Key Exchange (SPDKE), and was proposed by Habeeb et. al in 2013 [9]. It is a Diffie-Hellman-like, non-interactive key exchange protocol, which uses the group-theoretic notion of the semidirect product. Roughly speaking[1], we are interested in products of the form $\phi^{x-1}(g) \cdot \ldots \cdot \phi(g) \cdot g$, where g is an element of a (semi)group, ϕ is an endomorphism and $x \in \mathbb{N}$ is a positive integer.

Examples of concrete proposals for SPDKE can be found in [9, 10, 11, 12, 13]; respective cryptanalysis can be found in [14, 15, 16, 17, 18, 19]. The authors have a survey paper giving a summary of the back-and-forth on this topic [20]. Analogously to the Discrete Logarithm Problem, the security of SPDKE is heavily related to the following task: given a (semi)group element g, an endomorphism

[1] For more detail see Section 2.2

ϕ, and for some $x \in \mathbb{N}$ a product of the form $\phi^{x-1}(g) \cdot \ldots \cdot \phi(g) \cdot g$, recover the integer x. We refer to this task as the *Semidirect Discrete Logarithm Problem* (SDLP); in particular, successfully carrying it out allows recovery of the private exponents of participants in the key exchange.

The extant body of cryptanalytic work in this area does not address SDLP, instead using linear algebraic techniques to show that public information leads to key leakage. In fact, the complexity of SDLP and its relationship to more well-known hardness problems has not been well understood and was a significant open problem for researchers in this area. Our work in this paper takes the first steps towards addressing this problem. In particular, we show that there is a reduction of SDLP to the so-called *Abelian Hidden Shift Problem*, for which quantum subexponential algorithms are available.

1.1 Organisation of the Paper and Main Results

In line with [21], the bulk of the work will be the construction of a free, transitive group action, from which the main results quickly follow. To aid with this construction it will be useful to change our perspective slightly - towards this goal we introduce some notation, and recall the appropriate background mathematics.

In Section 2 we start by giving the definitions that allow the paper to be self-contained. We give a more detailed review of the mechanics of SPDKE, and introduce some new notation, and finish with a brief discussion of the Abelian Hidden Shift Problem.

The construction of the group action is contained entirely in Section 3. Several technical lemmas are proved and assembled as Theorem 3.6, in which the required group action is constructed.

In Section 4, we provide a preliminary reduction to the Abelian Hidden Shift Problem in Corollary 4.1. Before providing the full algorithm, we address some context-specific technicality as Procedure 4.2. The algorithm claimed in the title is finally given as Theorem 4.3.

We conclude the paper with a Conclusions section, which will reflect on the manner in which the main results were proved - in particular noting a surprising connection to isogeny-based cryptography.

Remark. *Shortly before posting this manuscript the authors were made aware of the work in [22]. The landmark result therein suggests that our* SDLP *is equivalent to the natural contextual analogue of the Computational Diffie-Hellman problem, which significantly strengthens the claim of security of* SPDKE. *We leave the full analysis to subsequent work.*

2 Preliminaries

2.1 Background Mathematics

We recall a number of group-theoretic notions used throughout this paper.
A *(semi)group* is a set G together with a binary operation $G \times G \to G$. Unless otherwise specified, we will write this operation multiplicatively, and require the following:

1. $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ for all $x, y, z \in G$
2. There is an identity element, written as 1, such that $g \cdot 1 = 1 \cdot g = g$ for all $g \in G$

Stopping here, we have a semigroup[2]. Suppose for some $g \in G$ there exists an element $h \in G$ such that $g \cdot h = h \cdot g = 1$; such an element is called the *inverse* of G, and is necessarily unique. If every $g \in G$ has an inverse, the semigroup is instead a group. The constructions that follow are defined for both groups and semigroups - to reflect this scenario we will use the catch-all notation (semi)groups.
For our purposes the (semi)groups will be non-abelian; that is, one cannot expect that $g \cdot h = h \cdot g$. One exception is the group we will write additively; in this case, the operation $g + h$ commutes,

[2] More precisely, a monoid.

and we require an inverse for every element. Note also that the identity is written as 0 in this case.

It will be useful for us to build a new (semi)group from an existing (semi)group. One way of doing this is via a structure called the *holomorph*[3]; let G a (semi)group and $End(G)$ its endomorphism group. The holomorph is the set $G \times End(G)$ equipped with multiplication

$$(g, \phi) \cdot (g', \phi') = (\phi'(g) \cdot g', \phi \circ \phi')$$

where $\circ$ refers to function composition.

Finally, we recall the notion of a *group action*. Let G a group and X a set: a group action is a function $\psi : G \times X \to X$. By convention we write $\psi(g, x)$ as $g * x$, and require the following: $1 \times x = x$ for all $x \in X$, and $g * (h * x) = (g \cdot h) * x$. The action is *free* if $g * x = x$ forces $g = 1$, and *transitive* if for any pair $x, y \in X$ there is a group element $g \in G$ such that $g * x = y$.

2.2 Semidirect Product Key Exchange

We here define in full SPDKE. One verifies by induction that holomorph exponentiation takes the form

$$(g, \phi)^x = (\phi^{x-1}(g) \cdot \ldots \phi(g) \cdot g, \phi^x)$$

The central idea of SPDKE is to use products of these form as a generalisation of Diffie-Helman Key-Exchange. It works as follows:

1. Suppose Alice and Bob agree on a public (semi)group G, as well as a group element g and endomorphism of G, say ϕ.

2. Alice picks a random secret integer x, and calculates the holomorph exponent $(g, \phi)^x = (A, \phi^x)$. She sends **only** A to Bob.

3. Bob similarly calculates (B, ϕ^y) corresponding to random, private integer y, and sends only B to Alice.

4. With her private automorphism ϕ^x Alice can now calculate her key as the group element $K_A = \phi^x(B) \cdot A$; Bob similarly calculates his key $K_B = \phi^y(A) \cdot B$.

We have

$$\begin{aligned}
\phi^x(B) \cdot A &= \phi^x(\phi^{y-1}(g) \cdot \ldots \cdot g) \cdot (\phi^{x-1}(g) \cdot \ldots \cdot g) \\
&= (\phi^{x+y-1} \cdot \ldots \cdot \phi^x(g)) \cdot (\phi^{x-1}(g) \cdot \ldots \cdot g) \\
&= (\phi^{x+y-1} \cdot \ldots \cdot \phi^y(g)) \cdot (\phi^{y-1}(g) \cdot \ldots \cdot g) \\
&= \phi^y(A) \cdot B
\end{aligned}$$

so $K := K_A = K_B$. Note that $A \cdot B \neq K$ as a consequence of our insistence that the group operation is non-commutative. It is immediate that the security of this scheme is related to SDLP, since $A = \phi^{x-1}(g) \cdot \ldots \cdot g$, and A, g, ϕ are all available to an eavesdropper - recovering Alice's private exponent is therefore exactly SDLP.

Writing these products in full will quickly become rather cumbersome. We therefore introduce some non-standard notation, which is useful both for convenience of exposition and the required shift in perspective we will introduce in this paper.

Definition 2.1. Let G be a finite, non-commutative (semi)group, $g \in G$, and $\phi \in End(G)$. We define the following function:

$$\begin{aligned}
s : G \times End(G) \times \mathbb{N} &\to G \\
(g, \phi, x) &\mapsto \phi^{x-1}(g) \cdot \ldots \cdot \phi(g) \cdot g
\end{aligned}$$

[3] The holomorph is itself a special case of the notion of a semidirect product group, hence the terminology.

Notice also that when g, ϕ are fixed - as in the case of the key exchange - the function s is really only taking integer arguments, continuing our analogy with the standard notion of group exponentiation. Indeed, in this language, `SDLP` has the following form:

Definition 2.2 (Semidirect Discrete Logarithm Problem)**.** Let G be a finite, non-commutative (semi)group, $g \in G$, and $\phi \in End(G)$. The *Semidirect Discrete Logarithm Problem* is the following task: given g, ϕ and $s(g, \phi, x)$ for some $x \in \mathbb{N}$, recover the integer x.

2.3 Abelian Hidden Shift Problem

Once we have constructed the requisite group action there is a canonical reduction to a known hardness problem, which will give us our quantum subexponential algorithms. The known hardness problem is as follows:

Definition 2.3 (Abelian Hidden Shift Problem)**.** Let A be a finite abelian group and X a set. Suppose there are two injective functions $f, g : A \to S$ that differ by a shift: that is, there is a group element $s \in A$ such that $g(a) = f(a + s)$ for each $a \in A$. In this scenario we say that f, g *hide* s. Provided with f, g, the *Abelian Hidden Shift Problem* is to recover the shift value s.

We will make use of the following fact, due to Kuperberg [23, Proposition 6.1]:

Theorem 2.4. *Let $|A| = N$ in the above setup. There exists a quantum algorithm of time complexity $2^{\mathcal{O}(\sqrt{\log N})}$ for solving the Abelian Hidden Shift Problem.*

3 Structure of the Exponents

In order to define our action we must first examine the structure of the set $X = \{s(g, \phi, i) : i \in \mathbb{N}\}$. Certainly this is neither a group nor a semigroup - numerous counterexamples can be found whereby multiplication of elements in this set are not contained in the set - but we can make some progress by borrowing from the standard theory of monogenic semigroups; presented, for example, in [24]. Since $X \subset G$, X is finite - the set $\{x \in \mathbb{N} : \exists y \quad s(g, \phi, x) = s(g, \phi, y)\}$ must therefore be non-empty, else it is in bijection with the natural numbers. We may therefore choose its smallest element, say n. By definition of n the set $\{x \in \mathbb{N} : s(g, \phi, n) = s(g, \phi, n + x)\}$ must also be non-empty, so we may again pick its smallest element and call it r.
The structure of X is further restricted by the following result:

Lemma 3.1. *Let $x, y \in \mathbb{N}$, then*

$$\phi^x \left(s(g, \phi, y)\right) \cdot s(g, \phi, x) = s(g, \phi, x + y)$$

Proof. Note that $s(g, \phi, x + y) = \phi^{x+y-1}(g) \cdot \ldots \cdot g$. Since ϕ preserves multiplication, applying ϕ^x to $s(g, \phi, y)$ adds x to the exponent of each term. Multiplication on the right by $s(g, \phi, x)$ then completes the remaining terms of $s(g, \phi, x + y)$. □

Remark. *One can entirely symmetrically swap the roles of x and y in the above argument, which gives two ways of calculating $s(g, \phi, x + y)$. In essence, therefore, this result gives us a slightly more elegant proof of the correctness of* `SPDKE`*.*

As a consequence of Lemma 3.1 and the definitions of n, r we have

$$\begin{aligned} s(g, \phi, n + 2r) &= \phi^r(s(g, \phi, n + r)) \cdot s(g, \phi, r) \\ &= \phi^r(s(g, \phi, n)) \cdot s(g, \phi, r) \\ &= s(g, \phi, n + r) = s(g, \phi, n) \end{aligned}$$

We conclude, by extending this argument in the obvious way, that $s(g, \phi, n + qr) = s(g, \phi, n)$ for each $q \in \mathbb{N}$. In fact, we have the following:

Lemma 3.2. *Fix g, ϕ and define n, r as above. One has that*

$$s(g, \phi, n + x + qr) = s(g, \phi, n + x)$$

for all $x, q \in \mathbb{N}$.

We will frequently invoke Lemma 3.2. Indeed, we immediately get that the set X cannot contain values other than $\{g, ..., s(g, \phi, n), ..., s(g, \phi, n+r-1)\}$. If any of the values in $\{g, ..., s(g, \phi, n-1)$ are equal we contradict the minimaltiy of n, and if any of the values in $\{s(g, \phi, n), ..., s(g, \phi, n+r-1)\}$ are equal we contradict the minimality of r. We have shown the following:

Theorem 3.3. *Fix $g \in G$ and $\phi \in End(G)$. The set $X = \{s(g, \phi, i) : i \in \mathbb{N}\}$ has size $n + r - 1$ for integers n, r dependent on g, ϕ. In particular*

$$X = \{g, ..., s(g, \phi, n), ..., s(g, \phi, n + r - 1)\}.$$

We refer to the set $\{g, ..., s(g, \phi, n - 1)\}$ as the *tail* of X, and the set $\mathcal{C} = \{s(g, \phi, n), ..., s(g, \phi, n + r - 1)\}$ as the *cycle* of X. The values n and r are called the *index* and *period* of X.
One can see that unique natural numbers correspond to each element in the tail, but infinitely many correspond to each element in the cycle. In fact, each element of the cycle corresponds to a unique residue class modulo r, shifted by the index n. This is a rather intuitive fact, but owing to its usefulness we will record it formally. In the following we assume the function $\mod$ returns the canonical positive residue.

Theorem 3.4. *Let $x, y \in \mathbb{N}$. We have*

$$s(g, \phi, n + x) = s(g, \phi, n + y)$$

if and only if $x \mod r = y \mod r$.

Proof. In the reverse direction, setting $x' = x \mod r$ and $y' = y \mod r$, we have by Lemma 3.2 that $s(g, \phi, n + x) = s(g, \phi, n + x')$ and $s(g, \phi, n + y) = s(g, \phi, n + y')$. By assumption $x' = y'$, and $0 \leq x', y' < r$. The claim follows since we know values in the range $\{s(g, \phi, n), ..., s(g, \phi, n+r-1)\}$ are distinct by Theorem 3.3.
On the other hand, suppose $s(g, \phi, n + y) = s(g, \phi, n + x)$ but $x \not\equiv y \mod r$. Without loss of generality we can write $y = x' + u + qr$ for some $q \in \mathbb{N}, 0 < u < r$ and $x' = x \mod r$. By remarks made in the discussion of theorem 2.2, we must have

$$s(g, \phi, n + x') = s(g, \phi, n + x' + u).$$

There are now three cases to consider; we claim each of them gives a contradiction.
First, suppose $x' + u = r$, then $s(g, \phi, n + x') = s(g, \phi, n)$. Since $x' < r$ we contradict minimality of r. The case $x' + u < r$ gives a similar contradiction.
Finally, if $x' + u > r$, without loss of generality we can write $x' + u = r + v$ for some positive integer v, so we have $s(M, \phi, n + x') = s(M, \phi, n + v)$. Since $x' \neq v$ (else we contradict $u < r$), and both values are strictly less than r, we have a contradiction, since distinct integers of this form give distinct evaluations of s. □

We are almost ready to define our group action; first, however, we must specify the group acting on the cycle. The previous result implies that we are in some sense interested in the action of residue classes, but in order to use the usual notion of integers $\mod r$, denoted here by $\mathbb{Z}_r$, we would need to be comfortable letting the function s take negative integer inputs. In fact, a well-behaved notion of the output of s on negative integers can be constructed provided one is willing to let g, ϕ be invertible. Fortunately, we need not restrict ourselves to this case, and instead consider the following object:

Definition 3.5. We write $\mathbb{N}_r = \{[i]_r : 0 \leq i < r\}$, where $[i]_r = \{k \in \mathbb{N} : k \equiv i \mod r\}$. We define the operation, written additively, by $[i]_r + [j]_r = [i + j]_r$.

It is easy to see that $\mathbb{N}_r$ is a finite abelian group[4]. We conclude the section by defining its action on the cycle $\{s(g,\phi,n),...,s(g,\phi,n+r-1)\}$.

Theorem 3.6. *Let $\mathcal{C}$ be the cycle as described above with size r. The additive group $\mathbb{N}_r$ acts freely and transitively on $\mathcal{C}$.*

Proof. Define the action $\psi : \mathbb{N}_r \times \mathcal{C} \to \mathcal{C}$ by

$$\psi([j]_r, s(g,\phi,n+i)) = \phi^{[j]_r}(s(g,\phi,n+i)) \cdot s(g,\phi,[j]_r)$$

This object is, of course, a set: indeed, it is equal to

$$\{\phi^{j+kr}(s(g,\phi,n+i)) \cdot s(g,\phi,j+kr) : k \in \mathbb{N}_0\}$$

which, by Proposition 3.1, is exactly the set $\{s(g,\phi,n+i+j+kr) : k \in \mathbb{N}\}$. By Theorem 3.4, every element of this set is equal to $s(g,\phi,n+i+j)$, so we will harmlessly abuse notation by writing

$$\psi([j]_r, s(g,\phi,n+i)) = s(g,\phi,n+i+j)$$

Moreover, the output[5] of ψ is indeed in $\mathcal{C}$, since $s(g,\phi,n+i+j) = s(g,\phi,n+(i+j) \mod r)$, and $0 \leq (i+j) \mod r < r$. In general, when $y > r$ we are free to dispense with the slightly more cumbersome modular notation and write $s(g,\phi,n+y)$ instead of $s(g,\phi,n+(y \mod r))$.
Let us check an action is indeed defined. Certainly $[0]_r$ fixes every element of $\mathcal{C}$. Let $[j]_r,[k]_r \in \mathbb{N}_r$; then, writing $\psi([j]_r, s(g,\phi,n+i))$ as $[j]_r * s(g,\phi,n+i)$ in the conventional fashion, we have

$$\begin{aligned}
[k]_r * ([j]_r * s(g,\phi,n+i)) &= [k]_r * s(g,\phi,n+i+j)\\
&= s(g,\phi,n+i+j+k)\\
&= [j+k]_r * s(g,\phi,n+i)\\
&= ([k]_r + [j]_r) * s(g,\phi,n+i)
\end{aligned}$$

where any protests that the sum exceeds r are countered by the well-definedness of modular addition.
Now let us see that the action is free and transitive. If $s(g,\phi,n+i)$ is fixed by $[j]_r$ then Theorem 3.4 gives that $i+j \equiv i \mod r$, so $[j]_r = [0]_r$. Thus the action is free. Fix $s(g,\phi,n+i)$ and $s(g,\phi,n+j)$; then $[r-i+j]_r \in \mathbb{N}_r$ is such that

$$[r-i+j]_r * s(g,\phi,n+i) = s(g,\phi,n+i+r-i+j) = s(g,\phi,n+j)$$

as required for transitivity. □

4 Hidden Shift Problem

The connection between the Abelian Hidden Shift Problem problem and group actions has been noticed before, and is given in the context of isogeny-based cryptography in [21]. In what follows, we use our Theorem 3.6 to update their reduction.

Corollary 4.1. *Given $s(g,\phi,n+i), s(g,\phi,n+j)$, one can recover the value $[k]_r$ such that $[k]_r * s(g,\phi,n+i) = s(g,\phi,n+j)$ provided one can solve the Abelian Hidden Shift Problem.*

Proof. Set $f_A, f_B : \mathbb{N}_r \to \mathcal{C}$ as $f_A([x]_r) = [x]_r * s(g,\phi,n+i)$ and $f_B([x]_r) = [x]_r * s(g,\phi,n+j)$. Then

$$\begin{aligned}
f_B([x]_r) &= [x]_r * s(g,\phi,n+j)\\
&= [x]_r * ([k]_r * s(g,\phi,n+i))\\
&= ([x]_r + [k]_r) * s(g,\phi,n+i)\\
&= f_A([x]_r + [k]_r)
\end{aligned}$$

[4] For the reader uncomfortable with this unconventional definition, one can think of it as the group $< x : x^r = 1 >$; in particular, the element x^{-1} is defined only in terms of positive powers of x.

[5] Or, more accurately, the content of the singleton set that ψ outputs is an element of $\mathcal{C}$.

In other words, f_A, f_B hide $[k]_r$. To complete the setup of a hidden shift problem we require the functions to be injective, which follows from the action being free and transitive. □

Consider again the problem of recovering x given $g, \phi, s(g, \phi, x)$. It follows from the above that if one can somehow find an N with $N > n$, all that remains is to solve an instance of the Abelian Hidden Shift Problem. The point is that this latter part of the task is of quantum subexponential complexity, as discussed in Section 2.3. Before we can prove our main theorem, however, we will have to deal with some context-specific technicality.

4.1 Non-empty Tails

Notice in the case that the cycle constitutes the whole of the set of exponents (equivalently, when $n = 1$) we have described a method of quantum subexponential complexity for recovering x from $g, \phi, s(g, \phi, x)$ for any $x \in \mathbb{N}$, since in this case $g = s(g, \phi, 1)$ is in the cycle. In particular, we have described a method of carrying out this task provided one has access to a commutative hidden shift problem oracle, and the best known such oracle is quantum subexponential.
In order to deal with the case that $n > 1$ and g is in the tail, not the cycle, it is therefore reasonable to assume access to a commutative hidden shift oracle. We detail below how one gets around this technicality below; the procedure is adapted from [25, Reduction 2.2, pp. 2–3].

Procedure 4.2. Pick some positive integer N. This integer N must be, in a sense detailed below, sufficiently large, but at the outset we have no way of verifying if N has such a property. We therefore need to perform a 'sanity check' at the end of the procedure; if this is failed, we know our initial choice of N was not sufficiently large, so we return to the outset replacing N with $2N$. Choose some random $k \in \{\lceil N/2 \rceil, ..., N\}$ and calculate $s(g, \phi, k)$. We assume N is sufficiently large to guarantee $s(g, \phi, k)$ is in the cycle; that is, we assume $\lceil N/2 \rceil \geq n$. Choose another random $k' \in \{1, ..., N\}$, set $h = s(g, \phi, k)$, and calculate $h' = \phi^{k'}(h) \cdot s(g, \phi, k')$. We now have two cycle elements; we know by our work above that with access to a hidden shift oracle we can recover the positive residue class $[l]_r$ that acts on h to give h'. In particular this gives us r, and therefore our sanity check: we should have that $\phi^r(h) \cdot s(g, \phi, r) = h$. If not, N was not sufficiently large, and we return to the outset as described.
With access to r it is a simple matter to recover n, since by definition of n, r, if $s(g, \phi, m + r) \neq s(g, \phi, m)$ then $m < n$. On the other hand, if $s(g, \phi, m + r) = s(g, \phi, m)$ we must have $m \geq n$. We can therefore carry out binary search to find the smallest m such that $s(g, \phi, m) = s(g, \phi, m + r)$; by definition, this smallest m is exactly n.

Since the requirement of N to be sufficiently large is that $N \geq n$, we expect to have to make $\mathcal{O}(\log n)$ oracle calls. By Theorem 2.4, each oracle call has quantum time complexity $2^{\mathcal{O}(\sqrt{logr})}$. We conclude after appropriate manipulation that the time complexity of finding r is $2^{\mathcal{O}(\log\log n + \sqrt{\log r})}$. The final binary search procedure is again $\mathcal{O}(\log n)$-time, so the overall complexity is $2^{\mathcal{O}(\log\log n + \sqrt{\log r})}$.

4.2 Solving SDLP

It remains to assemble the procedure given in Section 4.1 and Corollary 4.1. Indeed, doing so will gives the claim of the title, which we restate for completeness:

Theorem 4.3. *There exists an algorithm which solves* SDLP *in quantum subexponential time.*

Proof. First, as described in Procedure 4.2, one uses the oracle to recover the index n, with complexity $2^{\mathcal{O}(\log\log n + \sqrt{\log r})}$. With knowledge of the index we can use the oracle again to find the positive residue class $[k]_r$ such that $[k]_r * s(g, \phi, n) = s(g, \phi, x)$. Without loss of generality let k the smallest integer representative of this residue class; it follows that $x = n + k$. Again by Theorem 2.4, the quantum time complexity of recovering this residue class is $2^{\mathcal{O}(\sqrt{logr})}$, from which the claim follows. □

5 Conclusion

We have provided the first dedicated analysis of SDLP, showing a reduction to a well-studied problem. Perhaps the most surprising aspect of the work is the progress made by a simple rephrasing; we made quite significant progress through rather elementary methods, and we suspect much more can be made within this framework.

The reader may notice that we have shown that SPDKE is an example of a commutative action-based key exchange, and that breaking all such protocols can be reduced to the Abelian Hidden Shift Problem. Indeed, this work shows the algebraic machinery of SPDKE is a candidate for what Couveignes calls a *hard homogenous space*[6] [27], which was not known until now. In line with the naming conventions in this area we propose a renaming of SPDKE to SPDH, which stands for 'Semidirect Product Diffie Hellman', and should be pronounced *spud*.

We would also like to reiterate the sentiment expressed in the abstract. The purpose of this paper is not to claim a general purpose break of SPDKE (or, indeed, SPDH) - the algorithm presented is subexponential in complexity, which has been treated as tolerable in classical contexts. Instead, the point is to show a connection between SDLP and a known hardness problem, thereby providing insight on a problem about which little was known.

5.1 Acknowledgements

We thank Chloe Martindale of the University of Bristol and Samuel Jaques of the University of Oxford for fruitful discussion on this topic at the recent ICMS conference *Foundations and Applications of Lattice-based Cryptography.* We furthermore thank the organisers of this conference for facilitating this conversation.

References

[1] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. *Report on Post-Quantum Cryptography.* Research report NISTIR 8105. NIST, 2016. URL: http://csrc.nist.gov/publications/drafts/nistir-8105/nistir_8105_draft.pdf.

[2] Ward Beullens. "Breaking Rainbow Takes a Weekend on a Laptop". In: *IACR Cryptol. ePrint Arch.* (2022), p. 214. URL: https://eprint.iacr.org/2022/214.

[3] Chengdong Tao, Albrecht Petzoldt, and Jintai Ding. "Efficient Key Recovery for All HFE Signature Variants". In: *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I.* Ed. by Tal Malkin and Chris Peikert. Vol. 12825. Lecture Notes in Computer Science. Springer, 2021, pp. 70–93. URL: https://doi.org/10.1007/978-3-030-84242-0%5C_4.

[4] John Baena, Pierre Briaud, Daniel Cabarcas, Ray A. Perlner, Daniel Smith-Tone, and Javier A. Verbel. "Improving Support-Minors rank attacks: applications to GeMSS and Rainbow". In: *IACR Cryptol. ePrint Arch.* (2021), p. 1677. URL: https://eprint.iacr.org/2021/1677.

[5] Antoine Casanova, Jean-Charles Faugere, Gilles Macario-Rat, Jacques Patarin, Ludovic Perret, and Jocelyn Ryckeghem. "GeMSS : a great multivariate short signature". PhD thesis. UPMC-Paris 6 Sorbonne Universités; INRIA Paris Research Centre, PolSys Team, 2017.

[6] Wouter Castryck and Thomas Decru. *An efficient key recovery attack on SIDH (preliminary version).* Cryptology ePrint Archive, Paper 2022/975. 2022. URL: https://eprint.iacr.org/2022/975.

[6] Another major example of which arises from the theory of isogenies between elliptic curves - see, for example, [26]

[7] Luciano Maino and Chloe Martindale. *An attack on SIDH with arbitrary starting curve.* Cryptology ePrint Archive, Paper 2022/1026. 2022. URL: `https://eprint.iacr.org/2022/1026`.

[8] Delaram Kahrobaei, Ramon Flores, and Marialaura Noce. “Group-based Cryptography in the Quantum Era”. In: *The Notices of American Mathematical Society, to appear* (2022). URL: `https://arxiv.org/abs/2202.05917`.

[9] Maggie Habeeb, Delaram Kahrobaei, Charalambos Koupparis, and Vladimir Shpilrain. “Public key exchange using semidirect product of (semi) groups”. In: *International Conference on Applied Cryptography and Network Security.* Springer. 2013, pp. 475–486.

[10] Dima Grigoriev and Vladimir Shpilrain. “Tropical cryptography II: extensions by homomorphisms”. In: *Communications in Algebra* 47.10 (2019), pp. 4224–4229.

[11] Delaram Kahrobaei and Vladimir Shpilrain. “Using semidirect product of (semi) groups in public key cryptography”. In: *Conference on Computability in Europe.* Springer. 2016, pp. 132–141.

[12] Nael Rahman and Vladimir Shpilrain. “MAKE: A matrix action key exchange”. In: *Journal of Mathematical Cryptology* 16.1 (2022), pp. 64–72.

[13] Nael Rahman and Vladimir Shpilrain. “MOBS: Matrices Over Bit Strings public key exchange”. In: *https://eprint.iacr.org/2021/560* (2021).

[14] Alexei Myasnikov and Vitaliĭ Roman’kov. “A linear decomposition attack”. In: *Groups Complexity Cryptology* 7.1 (2015), pp. 81–94.

[15] Steve Isaac and Delaram Kahrobaei. “A closer look at the tropical cryptography”. In: *International Journal of Computer Mathematics: Computer Systems Theory* (2021), pp. 1–6.

[16] Daniel Brown, Neal Koblitz, and Jason Legrow. “Cryptanalysis of ‘MAKE’”. In: *Journal of Mathematical Cryptology* 16.1 (2022), pp. 98–102.

[17] Chris Monico. “Remarks on MOBS and cryptosystems using semidirect products”. In: *arXiv preprint arXiv:2109.11426* (2021).

[18] Christopher Battarbee, Delaram Kahrobaei, and Siamak F. Shahandashti. “Cryptanalysis of Semidirect Product Key Exchange Using Matrices Over Non-Commutative Rings”. In: *Mathematical Cryptology* 1.2 (2022), pp. 2–9.

[19] Christopher Battarbee, Delaram Kahrobaei, Dylan Tailor, and Siamak F Shahandashti. “On the efficiency of a general attack against the MOBS cryptosystem”. In: *arXiv:2111.05806; to appear in Journal of Mathematical Cryptology* (2022).

[20] Christopher Battarbee, Delaram Kahrobaei, and Siamak F Shahandashti. “Semidirect Product Key Exchange: the State of Play”. In: *arXiv preprint arXiv:2202.05178* (2022).

[21] Andrew Childs, David Jao, and Vladimir Soukharev. “Constructing elliptic curve isogenies in quantum subexponential time”. In: *Journal of Mathematical Cryptology* 8.1 (2014), pp. 1–29.

[22] Hart Montgomery and Mark Zhandry. *Full Quantum Equivalence of Group Action DLog and CDH, and More.* ASIACRYPT 2022, Cryptology ePrint Archive, Paper 2022/1135. 2022. URL: `https://eprint.iacr.org/2022/1135`.

[23] Greg Kuperberg. “A subexponential-time quantum algorithm for the dihedral hidden subgroup problem”. In: *SIAM Journal on Computing* 35.1 (2005), pp. 170–188.

[24] John Mackintosh Howie. *Fundamentals of semigroup theory.* 12. Oxford University Press, 1995.

[25] Matan Banin and Boaz Tsaban. “A reduction of semigroup DLP to classic DLP”. In: *Designs, Codes and Cryptography* 81.1 (2016), pp. 75–82.

[26] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. "CSIDH: an efficient post-quantum commutative group action". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2018, pp. 395–427.

[27] Jean-Marc Couveignes. "Hard homogeneous spaces". In: *Cryptology ePrint Archive* (2006). URL: `https://eprint.iacr.org/2006/291.pdf`.