Research Article

Andrew Mendelsohn*, Edmund Dable-Heath, and Cong Ling

A small serving of mash: (Quantum) algorithms for SPDH-Sign with small parameters

https://doi.org/10.1515/jmc-2024-0025
received June 10, 2024; accepted January 13, 2025

Abstract: We find an efficient method to solve the semidirect discrete logarithm problem (SDLP) over finite nonabelian groups of order p^3 and exponent p^2 for certain exponentially large parameters. This implies an attack on SPDH-Sign,[1] a signature scheme based on the SDLP, for such parameters. In particular, SDLP instances over such groups are parameterised by an $n < (p-1)p^6$: we develop a method to solve instances when $n \leq \mathrm{poly}(\log p) \cdot p$. Letting λ be the security parameter of SPDH-Sign, which is taken $p = \exp \lambda$, we find we may solve instances of SDLP corresponding to SPDH-Sign instances with exponentially large p. However, for $n \approx p^2$ and larger, our method no longer completely solves the SDLP instances. We also study the linear hidden shift problem for a group action corresponding to SDLP and take a step towards proving the quantum polynomial time equivalence of SDLP and the semidirect computational Diffie–Hellman problem.

Keywords: semidirect product, discrete logarithm, signatures, group actions, cryptanalysis

MSC 2020: 11T71, 94A60, 68Q12

1 Introduction

In [1], the authors introduced a key exchange protocol. The security of their scheme was based on a discrete logarithm problem (DLP): given a group element g that generates a finite group G, and the element g^x for some $x \in \mathbb{N}$, can one recover x? Efficient classical solutions to the general DLP remain elusive, but Shor [2] gave an efficient quantum algorithm to solve the aforementioned problem. Thus, cryptography relying on the aforementioned discrete logarithm assumption is not post-quantum secure. However, the fruitfulness of the discrete logarithm assumption for classical cryptography has led to widespread use of diverse protocols relying on DLPs.

The field of post-quantum cryptography comprises several distinct topics: lattices, isogenies of elliptic curves, multivariate polynomials, and codes have all been used to develop cryptosystems believed no more vulnerable to attack by quantum adversaries than by classical adversaries. Another line of work refers back to the DLP above, asking: can the DLP be tweaked to yield a quantum-hard cryptographic problem? If this were possible, such a "tweaked" DLP may perhaps allow a large number of existing discrete logarithm-based protocols to be ported into a post-quantum setting.

[1] Pronounced "SPUD-Sign".

*** Corresponding author: Andrew Mendelsohn,** Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, United Kingdom, e-mail: am3518@ic.ac.uk
Edmund Dable-Heath: The Alan Turing Institute, London, NW1 2DB, United Kingdom, e-mail: edable-heath@turing.ac.uk
Cong Ling: Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, United Kingdom, e-mail: c.ling@imperial.ac.uk

Open Access. © 2025 the author(s), published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.

One contribution into this direction is the semidirect discrete logarithm problem (SDLP). This problem replaces the underlying finite cyclic group of [1] with a noncommutative group, constructed as the semidirect product of two groups, in an effort to boost the hardness of the problem. Informally, for some $x \in \mathbb{N}$, some finite group element $g \in G$, and an element in the automorphism group of G, $\phi \in \mathrm{Aut}(G)$, given the element

$$s_{g,\phi}(x) \coloneqq \phi^{x-1}(g) \cdot \phi^{x-2}(g) \cdot \ldots \cdot \phi(g) \cdot g,$$

the problem asks an adversary to recover x. Moving to this setting prevents an adversary from straightforwardly running the algorithm of Shor, which appears not to apply to such groups. Without loss of generality, x may be considered to be sampled from some finite group $\mathbb{Z}/n\mathbb{Z}$ for some integer n dividing the order of the group, $|G|$. Note when ϕ is the identity map, we recover the standard DLP.

This problem was recently analysed by Battarbee et al. [3,4]. In the former article, the authors gave a subexponential- (but not polynomial-) time algorithm for SDLP. In the latter article, the authors develop a signature scheme, SPDH-Sign, based on the hardness of the SDLP problem. In particular, the authors use the group

$$G = G_p \coloneqq \left\{ \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} : a, b \in \mathbb{Z}_{p^2}, a \equiv 1 \bmod p \right\}$$

to instantiate the SDLP problem, where $\mathbb{Z}_{p^2} \coloneqq \mathbb{Z}/p^2\mathbb{Z}$. To ensure a suitable level of security, one takes p to be a “cryptographic”-sized prime.

In this study, we contribute to the cryptanalysis of that scheme by performing further analysis on the SDLP problem using G_p.

1.1 Contributions

In this work, we provide four contributions to the study of SDLP. The first of these is to show that the structure of G_p enables an adversary to recover $x \bmod (p-1)$ from $s_{g,\phi}(x)$ in SDLP instances defined on elements of G_p. This allows one to recover x when x is defined modulo a small multiple of p. This is because of the semidirect product isomorphism

$$G_p \cong \mathbb{Z}/p^2\mathbb{Z} \rtimes \mathbb{Z}/p\mathbb{Z},$$

which is efficiently computable. We obtain

Theorem 4. *Let $(g, \phi) \in G_p \rtimes \mathrm{Aut}(G_p)$, where $g = (a, \varphi) \in G_p$, and $x \in \mathbb{Z}/n\mathbb{Z}$, where $s_{g,\phi}(n) = 1$. Then, given $s_{g,\phi}(x)$, there is a quantum polynomial time algorithm to find $x \bmod (p-1)$.*

The intuition behind this result is that the second coordinate of multiplication in the semidirect product $G_p \cong \mathbb{Z}/p^2\mathbb{Z} \rtimes \mathbb{Z}/p\mathbb{Z}$ behaves like multiplication in $\mathbb{Z}/p\mathbb{Z}$, and this enables one to extract information about x from the second coordinate of the element $s_{g,\phi}(x)$.

We have a simple implementation of our attack, which can be made available upon request of the authors.

In a different approach, we then show that one can recover $\phi^x(g)$ from the publicly available data $g, \phi, s_{g,\phi}(x)$ and that this also leaks information on x due to the structure of the automorphisms of G_p. In both of the aforementioned cases, we can recover x only when it is defined modulo a small multiple of p. When the security parameter of a scheme is denoted by λ, one has $p = \exp(\lambda)$; so our attacks hold against exponentially large parameter sizes. However, the element x may be defined modulo a larger integer than p, *prima facie* modulo an integer up to the size of the group used to instantiate the SDLP problem. Since $|G_p \rtimes \mathrm{Aut}(G)| = (p-1)p^6$, in the case of SPDH-Sign, one may take x to be defined in general as large as $((p-1)/p_i) \cdot p^6$, where p_i is the smallest prime factor of $p-1$ (see Theorem 3 for more details), and in these larger parameter instances, say $x \in \mathbb{Z}/(p-1)^i p^j\mathbb{Z}$ for $i + j \geq 2$, we do not currently see how to recover all of x.

After this, we turn to abstract properties of the SDLP problem, which we consider as a group action problem. The action is of the abelian group $\mathbb{Z}/n\mathbb{Z}$ acting on the set $\mathcal{X}_{g,\phi} \coloneqq \{s_{g,\phi}(i) : i \in \mathbb{Z}/n\mathbb{Z}\}$; more details can be found below. We consider the “linear hidden shift” (LHS) problem and find that, as a corollary to our cryptanalytic attack, we can solve a special case of the LHS in quantum polynomial time. The particular LHS problem we consider in G_p, informally, is given $(g, \phi) \in G_p \rtimes \mathrm{Aut}(G_p)$, $\mathbf{x}_i \in (\mathbb{Z}/n\mathbb{Z})^\ell$, and $s_{g,\phi}(\sum_j s_j x_{i_j} + y)$ for $i = 1, \ldots, m$ and some unknown y, to recover $\mathbf{s} = (s_1 \ s_2 \ \ldots \ s_\ell)^T \leftarrow \{0, 1\}^\ell$. The assumption of the hardness of the LHS problem has been used to build advanced functionalities from group actions such as key-dependent message public key encryption [5], trapdoor claw-free functions [6], and pseudorandom generators [7]. We have

Theorem 5. *Let $g \in G_p$ and $\phi \in \mathrm{Aut}(G_p)$. Let $m \geq \ell + 1$. Then, there is a quantum polynomial time algorithm to solve* $\mathrm{LHS}_{\mathbb{Z}/n\mathbb{Z}, \mathcal{X}_{g,\phi}, \mathbf{s}, y}$.

It may thus be the case that it is not possible, or significantly more difficult, to build the advanced functionalities mentioned earlier from the SDLP, and it would be of interest to show that such functionalities can be built, since it would imply that the LHS property is stronger than necessary to realise group action-based protocols with those properties.

We then turn to an open problem from the study of Battarbee et al. [4]. In addition to SDLP, another problem, semidirect computational Diffie-Hellman (SCDH), was considered. This is the problem, given g, ϕ, $s_{g,\phi}(x)$ and $s_{g,\phi}(y)$, of computing $s_{g,\phi}(x + y)$. Of course, if one can solve SDLP, one may simply compute x from $s_{g,\phi}(x)$ and y from $s_{g,\phi}(y)$ and then compute $s_{g,\phi}(x + y)$ directly; but it is unknown if a solution to SCDH implies a solution to SDLP. We partially resolve this problem by demonstrating a quantum algorithm, which, given an oracle for a particular form of SCDH, which returns $s_{g,\phi}(2a)$ when given $s_{g,\phi}(a)$ for any a (denoted $\mathrm{SCDH}^2_{g,\phi,x}$), reduces SDLP to a hidden subgroup problem (HSP) instance, which can be efficiently solved with Shor’s period finding algorithm:

Theorem 6. *There is a quantum polynomial-time reduction from* $\mathrm{SDLP}_{g,\phi,x}$ *to* $\mathrm{SCDH}^2_{g,\phi,x}$.

We close by discussing the obstacles to a direct solution to SDLP via Shor’s algorithm.

1.2 Prior work

There is a burgeoning literature on noncommutative variants of the DLP, or schemes based on similar problems [3,4,8–11]. Attacks on variants of this problem can be found in previous studies [12–14]. The literature on cryptographic group actions includes refs [5,15–18].

We note the result [3, Theorem 6], which gives a method to solve SDLP, given access to a *group action* discrete logarithm oracle. This contrasts our work insofar as we merely require a discrete logarithm oracle for finite abelian groups.

In a concurrent and independent work (uploaded to the IACR Eprint server shortly prior to this article), Imran and Ivanyos [19] also provided cryptanalysis of the SDLP problem, in the idealised setting of black-box groups with unique labellings. We note the similarity to our work and note the greater generality of their approach, which applies to a variety of finite groups. However, our article includes results (on outer automorphisms, and relating SDLP and SCDH, for example) not covered by Imran and Ivanyos [19], and we consider our methods tailored to the choice of group suggested for SPDH-Sign a valuable contribution to the study of SDLP.

2 Preliminaries

2.1 Notations

We may write $[n]$ to denote the set $\{1, \ldots, n\}$. The arrow “←” may denote sampling from a set or sampling according to a distribution over a set; context will make which clear. If we write “$\overset{\$}{\leftarrow}$,” we mean sampling uniformly at random. The identity element of a group G will be denoted by e.

2.2 Group endomorphisms

To any finite group G are attached endomorphisms:

Definition 1. An endomorphism $\phi : G \to G$ is a homomorphism of groups from G to G.

If a group endomorphism ϕ is an isomorphism, we call ϕ an automorphism. The collection of all automorphisms of a finite group G forms a group, denoted $\mathrm{Aut}(G)$. The set of endomorphisms of G is denoted $\mathrm{End}(G)$.

2.3 Group actions

We define and give properties of group actions.

Definition 2. (Group action) A group action of a finite group G on a set X (sometimes called a G-set) is a map $\star : G \times X \to X$ satisfying
(1) for any $x \in X$, $e \star x = x$, and
(2) for any $g, h \in G$ and any $x \in X$, $(gh) \star x = g \star (h \star x)$.

A group action is effective if $|G| < \infty$ and standard group-theoretic operations can be performed in polynomial time. The following are standard properties of group actions:

Definition 3. A group action of G on X is
(1) transitive, if for any $x_1, x_2 \in X$, there exists a $g \in G$ satisfying $x_2 = g \star x_1$;
(2) faithful, if one has $g \star x = x$ for all $x \in X$ if and only if $g = e$;
(3) free, if one has $g = e$ if and only if there exists an $x \in X$ such that $x = g \star x$.

A free and transitive group action is called regular.

2.4 Semidirect product

We define the semidirect prodcuct of two groups.

Definition 4. (Semidirect product) Let G and H be finite groups. If there is an injective homomorphism

$$\rho : H \hookrightarrow \mathrm{Aut}(G),$$

then we can form a product of G and H, $G \rtimes_\rho H$, defined by the following multiplication rule: for (g, ϕ), $(h, \psi) \in G \times H$,

$$(g, \phi) \cdot (h, \psi) = (\rho(\psi)(g) \cdot h, \phi\psi),$$

where $\rho(\psi)(\cdot)$ is the action of the automorphism; this could be exponentiation (g^ψ) or conjugation ($\psi g \psi^{-1}$) or something more complicated. Note that this new group is noncommutative, i.e., swapping the order of multiplication can change the resulting group element on the right-hand side. If $H \subseteq \mathrm{Aut}(G)$, we can take ρ as the identity map and write $G \rtimes H$. In the literature, the product $G \rtimes \mathrm{Aut}(G)$ is sometimes called the holomorph of G, and denoted $\mathrm{Hol}(G)$. This construction is called the *external* semidirect product of G and H. It is a standard fact that $|G \rtimes H| = |G||H|$.

2.5 SDLP and SCDH

Recall the DLP in a finite abelian group G. Fix $g \in G$, which we will consider to be public. A challenger selects an integer x, computes $h = g^x$, and gives h to an adversary. The adversary has to recover x, which is defined modulo the order of $g \in G$. This can be solved in quantum polynomial time via Shor's algorithm [2], but is classically only solvable in exponential time.

Battarbee et al. [4] replace G with $G \rtimes H$. Let $(g, \phi) \in G \rtimes H$. Select, for instance, $x = 2$, and compute

$$(g, \phi)^2 = (g, \phi) \cdot (g, \phi) = (\phi(g)g, \phi^2).$$

If a challenger gave an adversary the resulting group element, they could take the second component ϕ^2, solve the (abelian) DLP in H, and find that $x = 2$. Alternatively, they could solve a DLP in the cyclic group generated by (g, ϕ), denoted $\langle (g, \phi) \rangle$. More generally, for an arbitrary choice of x, we have

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

Clearly, if $x < |H|$, an adversary could always solve an abelian DLP to find x. If $x \geq |H|$, they could solve an abelian discrete logarithm to find $x \bmod |H|$. So one cannot release the second coordinate of $(g, \phi)^x$ and maintain secrecy of x. This leads to

Definition 5. (SDLP) The semidirect product DLP, $\mathrm{SDLP}_{g,\phi,x}$, is given

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

for some $x \in \mathbb{Z}^+$ and $(g, \phi) \in G \rtimes H$, to find x.

One can see that x is only defined modulo $|G \rtimes H| = |G| \times |H|$. Moreover, it is in fact only defined modulo the order of the group element chosen, $o(g, \phi)$, since if $x > o(g, \phi)$, then $(g, \phi)^x = (g, \phi)^{x \bmod o(g,\phi)}$. As a consequence, we may take $x \in \mathbb{Z}/n\mathbb{Z}$ for some $n | o(g, \phi)$. When one fixes a choice of (g, ϕ) and sets n to be the smallest integer such that $s_{g,\phi}(n) = 1$, the corresponding group action has particularly useful properties. We denote such a problem instance by $\mathrm{SDLP}_{g,\phi,x}$.

A related problem to SDLP is the SCDH problem:

Definition 6. (SCDH). Let G be a finite group, and let $(g, \phi) \in G \rtimes \mathrm{Aut}(G)$. Let $x, y \in \mathbb{N}$ and suppose we are given (g, ϕ), $s_{g,\phi}(x)$, and $s_{g,\phi}(y)$. The SCDH problem, $\mathrm{SCDH}_{g,\phi,x,y}$, is to compute $s_{g,\phi}(x + y)$.

In the study of Battarbee et al. [3], a subexponential quantum algorithm was given for SDLP over semigroups. In the following, a family of (semi)groups indexed by κ is "easy" if for a fixed κ, pairs $(g, \phi), (g', \phi') \in G_\kappa \rtimes \mathrm{End}(G_\kappa)$, and values $f(\kappa), f'(\kappa)$ (resp. $g(\kappa), g'(\kappa)$) denoting the number of operations required to solve SDLP (resp. SCDH) for (g, ϕ) and (g', ϕ'), respectively, then we have $f(\kappa) = O(f'(\kappa))$ (resp. $g(\kappa) = O(g'(\kappa))$). Then:

Theorem 1. [3, Theorem 10] *Let $\{G_\kappa\}_\kappa$ be an easy family of semigroups, and fix κ. For any pair $(g, \phi) \in G_\kappa \rtimes \mathrm{End}(G_\kappa)$, there is a quantum algorithm solving SDLP with respect to (g, ϕ) with time and query complexity $2^{O(\sqrt{\log \kappa})}$.*

In this work, we consider groups, rather than semigroups. We also note a group action interpretation of SDLP. Define

$$\mathcal{X}_{g,\phi} \coloneqq \{s_{g,\phi}(i) : i \in \mathbb{Z}/n\mathbb{Z}\}.$$

Then,

Definition 7. Let $(g, \phi) \in G \rtimes H$ and n be the smallest integer such that $s_{g,\phi}(n) = 1$. Define a group action of $\mathbb{Z}/n\mathbb{Z}$ on $\mathcal{X}_{g,\phi}$ by

$$\mathbb{Z}/n\mathbb{Z} \circlearrowright \mathcal{X}_{g,\phi} : x * s_{g,\phi}(y) = s_{g,\phi}(x + y).$$

This group action is free and transitive. We call this group action the semidirect product group action (SDPGA).

2.6 SPDH-sign

In the study of Battarbee et al. [4], a signature scheme was designed based on SDLP. The key generation and signing algorithms require multiple instances of SDLP to be published; we denote the number of samples by N, and refer to SPDH-Sign$_{g,\phi}(N)$ below. The key generation and signing algorithms are given by

Algorithm 1 Key generation algorithm

```
Gen($N$):
for $i \leftarrow 1, \ldots, N$ do
    $X_i \overset{\$}{\leftarrow} \mathcal{X}_{g,\phi}$
    $s_i \overset{\$}{\leftarrow} \mathbb{Z}_n$
    $Y_i \leftarrow s_i \star X_i$
end for
$sk \leftarrow (s_1, \ldots, s_N)$
$pk \leftarrow ((X_1, \ldots, X_N), (Y_1, \ldots, Y_N))$
return $(sk, pk)$
```

Algorithm 2 Signing algorithm

```
Sg($m, (sk, pk)$):
for $i \leftarrow 1, \ldots, N$ do
    $t_i \overset{\$}{\leftarrow} \mathbb{Z}_n$
    $I_i \leftarrow t_i \star X_i$
end for
$I \leftarrow (I_1, \ldots, I_N)$
$c \leftarrow H(I, m)$
for $i \leftarrow 1, \ldots, N$
    if $c_i = 0$ then
        $p_i \leftarrow t_i$
    else
        $p_i \leftarrow t_i - s_i$
    end if
end for
$p \leftarrow (p_1, \ldots, p_N)$
$(\sigma_1, \sigma_2) \leftarrow (I, p)$
return $(\sigma_1, \sigma_2)$
```

Note that it suffices to solve SDLP to break the scheme: if one can solve the SDLP problem, one can take the public key $pk = ((X_1, \ldots, X_N), (Y_1, \ldots, Y_N))$ of SPDH-Sign and extract the s_i, which comprise the secret key. For more on the security of SPDH-Sign, we refer the reader to the study of Battarbee et al. [4].

For the use of SPDH-Sign, one has to pick a particular group with which the scheme will be instantiated; the authors propose the use of the group

$$G = G_p \coloneqq \left\{ \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} : a, b \in \mathbb{Z}_{p^2}, a \equiv 1 \bmod p \right\}.$$

We note that we have $G_p \cong \mathbb{Z}/p^2\mathbb{Z} \rtimes \mathbb{Z}/p\mathbb{Z}$, where $\mathbb{Z}/p\mathbb{Z}$ acts on $\mathbb{Z}/p^2\mathbb{Z}$ via $a \star b = b^{1+pa}$. This isomorphism and its inverse are plainly efficiently computable.

When using such a group, p would be chosen to be a cryptographic prime, i.e., $p = \exp(\lambda)$, where λ is the security parameter of a SLDP-based scheme, such as SPDH-Sign.

Finally, in this section, we note an incorrect statement in the study of Battarbee et al. [4]. The authors write:

Theorem 2. [4, Theorem 9] *Let $(g, \phi) \in G_p \rtimes \mathrm{Aut}(G_p)$, where p is an odd prime. Suppose n is the smallest integer for which $s_{g,\phi}(n) = 1$. Then,*

$$n \in \{p, p^2, p^3, p^4, p^5, p^6, (p-1), p(p-1), p^2(p-1), p^3(p-1), p^4(p-1), p^5(p-1)\}.$$

The reasoning runs as follows. Since $n | \mathrm{ord}((g, \phi))$, and $\mathrm{ord}((g, \phi)) | G_p \ltimes \mathrm{Aut}(G)$, we must have $n|(p-1)p^6$ for some odd prime p, and $n \neq (p-1)p^6$ since this would imply $G_p \rtimes \mathrm{Aut}(G_p)$ were cyclic.

The reasoning is sound; the conclusion of the theorem statement, however, is false when $p \neq 3$: since p is prime, $p-1$ is not prime, and thus, the set of possibilities for n includes all elements of the set of divisors of $p-1$ multiplied by powers of p, up to p^6 – not just the 12 values stated earlier. For instance, $n = 2p$ is a possibility for all p. The statement should read:

Theorem 3. *Let $(g, \phi) \in G_p \rtimes \mathrm{Aut}(G_p)$, where p is an odd prime. Suppose n is the smallest integer for which $s_{g,\phi}(n) = 1$. Let $\{p_1, \ldots, p_t\}$ be the set of prime divisors of $p-1$. Then,*

$$n \in \left\{ p^j \prod_{i \in S} p_i \right\}_{j,S},$$

where $S \subset [t]$ runs over multisets S such that $\prod_{i \in S} p_i$ denotes the products of the p_i indexed by a subset of possible indices such that $\prod_{i \in S} p_i | p-1$, and j satisfies $j \in [5]$ if S satisfies $\prod_{i \in S} p_i = p-1$ and $j \in [6]$ otherwise.

We point this out for its relevance to our results in Section 4. If the number of prime factors of $p-1$ is bounded, one can compute n efficiently (quantumly) given p, g, and ϕ, using the methods of [4, Section 5] or [3, Algorithm 1].

3 On G_p and its automorphisms

In this section, we discuss properties of G_p, which we will exploit in the following, and in particular, give an explicit form for its automorphisms. Any finite group G has a set of automorphisms, denoted $\mathrm{Aut}(G)$, which form a group under composition. The structure of $\mathrm{Aut}(G)$ comprises two factors: the inner and outer automorphisms. These each form a subgroup of $\mathrm{Aut}(G)$.

Inner automorphisms are defined by conjugation: if $g \in G$ is an arbitrary group element, the map $c_h : g \mapsto hgh^{-1}$ can be checked to be an automorphism. The group formed by such maps is denoted $\mathrm{Inn}(G)$. Clearly, if h commutes with all other group elements, c_h is the trivial map; thus, when counting the number of inner automorphisms, we find that there are $|\mathrm{Inn}(G)| = \frac{|G|}{|\mathcal{Z}(G)|}$ of them, where $\mathcal{Z}(G) = \{g \in G : gh = hg$ for all $h \in G\}$ denotes the centre of the group.

The group of outer automorphisms, $\mathrm{Out}(G)$, is defined as

$$\mathrm{Out}(G) \coloneqq \mathrm{Aut}(G)/\mathrm{Inn}(G).$$

Hence, there are $|\mathrm{Out}(G)| = |\mathrm{Aut}(G)|/|\mathrm{Inn}(G)|$ outer automorphisms. We are interested in determining explicit forms of elements of these groups when $G = G_p$, for our subsequent cryptanalysis of SPDH-Sign. In the following, we let $g \in G_p$ and write $g = \begin{pmatrix} 1 + pm & b \\ 0 & 1 \end{pmatrix}$ for some $m \in \mathbb{Z}/p\mathbb{Z}$ and $b \in \mathbb{Z}/p^2\mathbb{Z}$. As in the study of Conrad [20], G_p is generated by elements r and s, where

$$r = \begin{pmatrix} 1+p & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad s = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

So a generic group element $g = \begin{pmatrix} 1 + pm & b \\ 0 & 1 \end{pmatrix}$ may be written $g = s^b r^m$, and group multiplication can be expressed

$$s^b r^m \cdot s^{b'} r^{m'} = s^{b+b'+pmb'} r^{m+m'}.$$

3.1 Inner automorphisms of G_p

We first consider inner automorphisms. Note that $(s^c r^n)^{-1} = (r^n)^{-1}(s^c)^{-1} = r^{-n}s^{-c} = s^{pcn}s^{-c}r^{-n}$, since $s^{pcn}s^{-c}r^{-n} \cdot s^c r^n = s^{pcn}s^{-pcn}r^0 = 1$. The inner automorphisms act on $s^b r^m$ by conjugation; i.e., if $\phi \in \mathrm{Inn}(G_p)$, then for some c and n,

$$\begin{aligned} \phi(s^b r^m) &= (s^c r^n)^{-1} s^b r^m (s^c r^n) = (s^c r^n)^{-1} s^{b+c+pmc} r^{m+n} \\ &= s^{pcn} s^{-c} r^{-n} s^{b+c+pmc} r^{m+n} = s^{pcn} s^{b+pmc-pn(b+c)} r^m \\ &= s^{pnc+b+pmc-pnb-pnc} r^m \\ &= s^{b+p(mc-nb)} r^m. \end{aligned}$$

We summarise this as

Lemma 1. *Let ϕ be an inner automorphism of G_p corresponding to conjugation by $s^c r^n$. Then, the action of ϕ on a generic group element $g = s^b r^m$ is given by*

$$\phi(g) = s^{b+p(mc-nb)} r^m.$$

We note that there are $|\mathrm{Inn}(G_p)| = \frac{|G_p|}{|\mathcal{Z}(G_p)|} = \frac{p^3}{p} = p^2$ inner automorphisms, since the centre of G_p is

$$\mathcal{Z}(G_p) = \left\{ \begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix} : b \in \mathbb{Z}/p^2\mathbb{Z}, b \equiv 0 \bmod p \right\} = \left\langle \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} \bmod p^2 \right\rangle.$$

3.2 Outer automorphisms of G_p

The form of the outer automorphisms is less obvious than that of the inner automorphsims; we have

Proposition 1. *The outer automorphisms of G_p are given by the maps*

$$\phi(s^b r^m) = s^{bw+pmu} r^m,$$

where ϕ corresponds to a pair $(u, w) \in \mathbb{Z}/p\mathbb{Z} \rtimes (\mathbb{Z}/p\mathbb{Z})^\times$.

Proof. Clearly, we have $\phi : G_p \to G_p$ such that $\phi(e) = e$. Let $g = s^b r^m$ and $g' = s^{b'} r^{m'}$. Observe

$$\phi(gg') = \phi(s^b r^m \cdot s^{b'} r^{m'}) = \phi(s^{b+b'+pmb'} r^{m+m'}) = s^{w(b+b'+pmb')+p(m+m')u} r^{m+m'}$$

and

$$\begin{aligned} \phi(g)\phi(g') &= s^{bw+pmu} r^m s^{b'w+pm'u} r^{m'} \\ &= s^{bw+pmu+b'w+pm'u+pm(b'w+pm'u)} r^{m+m'} \\ &= s^{bw+pmu+b'w+pm'u+pmb'w} r^{m+m'}. \end{aligned}$$

So ϕ is indeed multiplicative. Moreover, these are not inner automorphisms, which can be seen by inspecting the “twist” of b in the exponent by w. Note that there are $|\mathrm{Out}(G_p)| = |\mathrm{Aut}(G_p)|/|\mathrm{Inn}(G_p)| = (p-1)p^3/p^2 = (p-1)p$ outer automorphisms, and since the automorphisms mentioned earlier are obtained by pairs from $\mathbb{Z}/p\mathbb{Z} \rtimes (\mathbb{Z}/p\mathbb{Z})^\times$, and $|\mathbb{Z}/p\mathbb{Z} \rtimes (\mathbb{Z}/p\mathbb{Z})^\times| = p(p-1)$, we conclude we have found all the outer automorphisms. □

We conclude this section with the important observation.

Corollary 1. *Let $\phi \in \mathrm{Aut}(G_p)$. Then, for any $g = s^b r^m$, we have $\phi(g) = s^{b'} r^m$ for some b'; i.e., ϕ leaves r^m unchanged.*

Proof. Observation of the results of Lemma 1 and Proposition 1. □

4 “Making Mash” when $n \le \mathrm{poly}(\log p)p$

Here, we outline an attack on SPDH when n is “small” (although still exponential in the security parameter). The attack uses the structure of G_p to extract information on x from g, ϕ, and $s_{g,\phi}(x)$. We begin with a proposition:

Proposition 2. *Let $G = M \rtimes N$ be a semidirect product of finite groups with N acting on M via automorphisms. Consider the holomorph of G, $(M \rtimes N) \rtimes \mathrm{Aut}(G)$. Then, if N is simple, the maps induced on N by elements of $\mathrm{Aut}(G)$ are either the constant map $N \to \{e\}$ or automorphisms.*

Proof. Let $\phi \in \mathrm{Aut}(G)$. Writing $\phi(m, n) = (m', n')$, consider the induced map $\psi : N \to N, n \mapsto n'$. Since

$$\phi((m, n))\phi((m', n')) = \phi((m, n)(m', n')) = \phi((n'(m)m', nn')),$$

we have $\psi(n)\phi'(n') = \psi(nn')$. Moreover,

$$\phi((m, e))\phi((m', e)) = \phi((m, e)(m', e)) = \phi(mm', e),$$

so $\psi(e)\psi(e) = \psi(e)^2 = \psi(e)$ and $\psi(e)$ is an idempotent in a finite group; hence, $\psi(e) = e$. Thus, ψ is an endomorphism of N.

Since the image of a group under an endomorphism is a subgroup, we find that either $\psi(N) = N$ or $\psi(N) = \{e\}$. In the latter case, every element is mapped to e, and in the former, we have a homomorphism between finite groups of trivial kernel and thus an automorphism. □

We note that when $N = \mathbb{Z}/p\mathbb{Z}$, $\mathrm{End}(\mathbb{Z}/p\mathbb{Z}) = \mathbb{Z}/p\mathbb{Z}$.

We now give a general method to recover x when n is at most a small multiple of $|\mathrm{Aut}(N)|$, subject to a constraint on the group element $(g, \phi) \in G \rtimes \mathrm{Aut}(G)$, where $G = M \rtimes N$ is a semidirect product with M and N finite abelian, and N simple as in the previous proposition, and $g = (a, \varphi) \in G$. We then specialise to the particular case of G_p.

Theorem 4. *Let $G = M \rtimes N$ be a semidirect product with M and N finite abelian, and N simple. Suppose $|\mathrm{Aut}(N)| = \prod_i p_i$ for distinct primes p_i. Let $(g, \phi) \in G \rtimes \mathrm{Aut}(G)$, where $g = (a, \varphi) \in G$. Suppose that ϕ acts on φ as an automorphism ψ, sending $\varphi \mapsto \varphi^\alpha$ for some $\alpha \ne 0$. Then, given $s_{g,\phi}(x)$ for any $x \in \mathbb{Z}/n\mathbb{Z}$, there is a quantum polynomial time algorithm to find $x \bmod |\mathrm{Aut}(N)|$.*

Proof. The SDLP instance is to recover x from $s_{g,\phi}(x)$, which we may write

$$s_{g,\phi}(x) = \phi^{x-1}((a, \varphi))\phi^{x-2}((a, \varphi))\ldots\phi(a, \varphi)(a, \varphi),$$

where $g = (a, \varphi) \in M \rtimes N$. If ϕ acts as an induced automorphism ψ on φ sending φ to φ^α for some α, then since (g, ϕ) is public, evaluating $\phi(g)$ for φ^α and appealing to an abelian discrete logarithm oracle yields α. We can write $s_{g,\phi}(x)$ as

$$\phi^{x-1}((a, \varphi))\phi^{x-2}((a, \varphi))\ldots\phi((a, \varphi))(a, \varphi) = (\cdot,\ \psi^{x-1}(\varphi)\psi^{x-2}(\varphi)\ldots\psi(\varphi)\varphi),$$

for some unspecified first entry. The second entry above can be rewritten

$$(\varphi^{\alpha^{x-1}})(\varphi^{\alpha^{x-2}})\ldots(\varphi^{\alpha})\varphi = \varphi^{\alpha^{x-1}+\alpha^{x-2}+\ldots+\alpha+1}.$$

Another appeal to an abelian discrete logarithm oracle obtains the exponent

$$\alpha^{x-1} + \alpha^{x-2} + \ldots + \alpha + 1 \bmod |\mathrm{Aut}(N)|.$$

We now split into two cases: if $\alpha = 1$, then $\alpha^{x-1} + \alpha^{x-2} + \ldots + \alpha + 1 = x \bmod |\mathrm{Aut}(N)|$ and we are done. So suppose we are in the case of $\alpha \neq 1$.

By the chinese remainder theorem, it suffices to recover $x \bmod p_i$ from

$$b \coloneqq \alpha^{x-1} + \alpha^{x-2} + \ldots + \alpha + 1 \bmod p_i,$$

for all prime factors p_i of $|\mathrm{Aut}(N)|$ (which can be found efficiently with a quantum algorithm). To do this, rewrite

$$b = \alpha^{x-1} + \alpha^{x-2} + \ldots + \alpha + 1 = \frac{\alpha^x - 1}{\alpha - 1} \bmod p_i$$

and rearrange for

$$\alpha^x = b(\alpha - 1) + 1 \bmod p_i,$$

which can be done since we assumed $\alpha \neq 1$. A third appeal to an abelian discrete logarithm oracle gives $x \bmod p_i$, and hence, $x \bmod |\mathrm{Aut}(N)|$. □

Corollary 2. *Let* $n = \mathrm{poly}(\log p)p$ *and* $(g, \phi) \in G_p \rtimes \mathrm{Aut}(G_p)$. *Then, there is a quantum polynomial time algorithm to solve* $\mathrm{SDLP}_{g,\phi,x}$.

Proof. We apply the theorem with $M = \mathbb{Z}/p^2\mathbb{Z}$ and $N = \mathbb{Z}/p\mathbb{Z}$, since as noted above, we have $G_p \cong \mathbb{Z}/p^2\mathbb{Z} \rtimes \mathbb{Z}/p\mathbb{Z}$, and note that by Corollary 1, any automorphism leaves the r component of a group element fixed, and so in the notation of the theorem, we always have $\alpha = 1$. We then obtain $x \bmod |\mathrm{Aut}(N)| = x \bmod (p-1)$ as in the proof of the theorem. If $n = \mathrm{poly}(\log p)p$, we can then find the true value of x by exhaustion in polynomial time, since there are $n/(p-1) = \mathrm{poly}(\log p)p/(p-1) = \mathrm{poly}(\log p)$ options for the true value of x. □

We note that such values for n are possible by Theorem 3.

The consequence of all this is that when instantiating SPDH-Sign with $G = G_p$, one should choose n to be at least $n \approx p^2$.

5 Attack in the style of Brown et al. [14]

In the study of Brown et al. [14], the scheme “MAKE” [8] was cryptanalysed, and Monico [21] extended the attack to the scheme “MOBS” [22]. The scheme uses square matrices whose entries are bitstrings of k bits equipped with the logical operations of OR and AND. Brown et al. [14] found (in the notation of Battarbee et al. [23]) that, given such a matrix M and an automorphism h of the space of such matrices, and writing $A \coloneqq h^{x-1}(M)\ldots h(M)M$, one could obtain $h(A)M = h^x(M)A$. From this, it was argued that MAKE and MOBS were insecure, since by linear algebra $h^x(M)$, and then h^x and finally, x, could be computed (although the efficacy of the attack was disputed in the study of Battarbee et al. [23]).

We note that one can obtain $\phi^x(g)$ given g, ϕ and $s_{g,\phi}(x)$, by computing

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

somewhat in the style of the attacks on MAKE and MOBS. It was known prior to this work that this element could be computed. Here, however, we observe that since we know g, one can then obtain further information on x.

In more detail and for $G = G_p$, suppose we have $g \in G_p$. Write $g = \begin{pmatrix} 1 + pa & b \\ 0 & 1 \end{pmatrix}$ for some $a \in \mathbb{Z}/p\mathbb{Z}$ and $b \in \mathbb{Z}/p^2\mathbb{Z}$. We then compute $\phi^x(g) = \begin{pmatrix} 1 + pa' & b' \\ 0 & 1 \end{pmatrix}$ for some $a' \in \mathbb{Z}/p\mathbb{Z}$ and $b' \in \mathbb{Z}/p^2\mathbb{Z}$. Here, we consider the case of inner automorphisms $\mathrm{Inn}(G_p)$ and of elements in $\mathrm{Out}(G_p) \coloneqq \mathrm{Aut}(G_p)/\mathrm{Inn}(G_p)$.

First, consider inner automorphisms. Recall that the inner automorphisms act on $s^b r^m$ by conjugation, and that by Lemma 1 if $\phi \in \mathrm{Inn}(G_p)$, then

$$\phi(s^b r^m) = s^{b+p(mc-nb)} r^m.$$

We then compute

$$\phi^x(s^b r^m) = s^{b+xp(mc-nb)} r^m.$$

We can multiply by r^{-m} to obtain $s^{b+xp(mc-nb)}$, use a discrete logarithm oracle to find $b + xp(mc - nb) \bmod p^2$, and then if $mc - nb \not\equiv 0 \bmod p$ rearrange to find $x \bmod p$.

In the case of outer automorphisms, we found in Proposition 1 that these are given by the maps

$$\phi(s^b r^m) = s^{bw+pmu} r^m,$$

where ϕ corresponds to a pair $(u, w) \in \mathbb{Z}/p\mathbb{Z} \rtimes (\mathbb{Z}/p\mathbb{Z})^\times$. We then compute

$$\phi^x(g) = s^{bw^x + pmu(w^{x-1} + \ldots + w+1)} r^m.$$

We can cancel the r^m, since it is public, for

$$s^{bw^x + pmu(w^{x-1} + \ldots + w+1)},$$

and we can hence recover $bw^x + pmu(w^{x-1} + \ldots + w + 1) \bmod p^2$ by solving the DLP instance. If we apply this to $\phi^{x-1}(g)$ as well, we may compute

$$\begin{aligned} & s^{bw^x + pmu(w^{x-1} + \ldots + w+1)} \cdot s^{-bw^{x-1} - pmu(w^{x-2} + \ldots + w+1)} \\ = \; & s^{bw^x - bw^{-1} + pmu(w^{x-1} + \ldots + w+1) - pmu(w^{x-2} + \ldots + w+1)} \\ = \; & s^{bw^{x-1}(w-1) + pmuw^{x-1}}, \end{aligned}$$

and we can then obtain $bw^{x-1}(w - 1) + pmuw^{x-1} \bmod p^2 = w^{x-1}(b(w - 1) + pmu) \bmod p^2$. If $b(w - 1) \not\equiv 0 \bmod p$, we can cancel the righthand factor for w^{x-1} and recover $x - 1 \bmod p$ from a discrete logarithm oracle.

We summarise the above as

Proposition 3. *Suppose $x \in \mathbb{Z}/n\mathbb{Z}$, $(g, \phi) \in G_p \rtimes \mathrm{Aut}(G_p)$, $g = s^b r^m$, and $s_{g,\phi}(x)$ is a SDLP instance. Then, if ϕ is an inner automorphism, there is a quantum polynomial time algorithm to compute $x \bmod p$, and if ϕ is an outer automorphism corresponding to a pair $(u, w) \in \mathbb{Z}/p\mathbb{Z} \rtimes (\mathbb{Z}/p\mathbb{Z})^\times$ such that $b(w - 1) \not\equiv 0 \bmod p$, then there is also a quantum polynomial time algorithm to recover $x \bmod p$.*

Finally, we note all automorphisms are obtained from composing inner and outer automorphisms.

6 SPDH and the LHS problem

In this section, we show that Theorem 4 implies a solution to a special case of the LHS problem defined in the study of Alamati et al. [5]. We begin by defining this problem formally. Let $\langle \mathbf{g}_i, \mathbf{s} \rangle \coloneqq \prod_j g_{i_j}^{s_j}$, where $g_{i_j} \in G$ and G is written multiplicatively.

Definition 8. The search LHS problem $\mathrm{LHS}_{G,X,\mathbf{s}}$ is hard over a regular group action $(G, X, \star)$ if for any $m = \mathrm{poly}(\lambda)$, $\mathbf{s} \leftarrow \{0, 1\}^\ell$, and for any PPT attacker $\mathcal{A}$, we have

$$\Pr[\mathcal{A}(\{(x_i, \mathbf{g}_i, (\langle \mathbf{g}_i, \mathbf{s} \rangle) \star x_i)\}_{i \in [m]}) \text{ outputs } \mathbf{s}] \leq \mathrm{negl}(\lambda),$$

where $\mathbf{g}_i \leftarrow G^\ell$ and $x_i \leftarrow X$ are sampled independently, over all random coins in the experiment.

For SDPGA: the search LHS problem is hard over $(\mathbb{Z}/n\mathbb{Z}, \mathcal{X}_{g,\phi}, \star)$ if for any $m = \mathrm{poly}(\lambda)$ and for any PPT attacker $\mathcal{A}$, we have

$$\Pr[\mathcal{A}(\{(s_{g,\phi}(y_i), \mathbf{x}_i, (\langle \mathbf{x}_i, \mathbf{s} \rangle) \star s_{g,\phi}(y_i))\}_{i \in [m]}) \quad \text{outputs } \mathbf{s}] \leq \mathrm{negl}(\lambda),$$

where $\mathbf{x}_i \leftarrow (\mathbb{Z}/n\mathbb{Z})^\ell$, $\mathbf{s} \leftarrow \{0,1\}^\ell$, $(g,\phi) \leftarrow G_p \rtimes \mathrm{Aut}(G_p)$, and $s_{g,\phi}(y_i) \leftarrow \mathcal{X}_{g,\phi}$ are sampled independently, over all random coins in the experiment. Note that additively,

$$(\langle \mathbf{x}_i, \mathbf{s} \rangle) \star s_{g,\phi}(y_i) = \left(\sum_j s_j x_{i_j} \right) \star s_{g,\phi}(y_i) = s_{g,\phi}\left(\sum_j s_j x_{i_j} + y_i \right).$$

We consider the special case in which the same $s_{g,\phi}(y)$ is used for all $s_{g,\phi}(y_i)$, $i = 1, \ldots, \ell$. We denote this case by $\mathrm{LHS}_{\mathbb{Z}/n\mathbb{Z}, \mathcal{X}_{g,\phi}, \mathbf{s}, y}$. We now prove our result:

Theorem 5. *Let* $(g,\phi) \in G_p \rtimes \mathrm{Aut}(G_p)$ *and let* $m \geq \ell + 1$. *Then, there is a quantum polynomial time algorithm to solve* $\mathrm{LHS}_{\mathbb{Z}/n\mathbb{Z}, \mathcal{X}_{g,\phi}, \mathbf{s}, y}$.

Proof. Write $x' = \sum_j s_j x_{i_j}$. We are given the (g,ϕ), $\mathbf{x}_i$, and $s_{g,\phi}(x' + y)$. We therefore use the method of Theorem 4 to find $b_i \coloneqq x' + y \bmod (p-1)$, for $i = 1, \ldots, m$. This gives us the m equations

$$\begin{aligned} b_1 &= \sum_j s_j x_{1_j} + y \bmod (p-1) \\ &\vdots \\ b_m &= \sum_j s_j x_{m_j} + y \bmod (p-1). \end{aligned}$$

This is m equations in the $\ell + 1$ unknown values of $s_1, \ldots, s_\ell, y$ with known coefficients $x_{i_j} \bmod (p-1)$. We may set $x_{i_{j+1}} = 1$ for all i as the "coefficient" of y. Since $s_i \in \{0,1\}$ for all i the modulo operation leaves s_i unchanged. Thus, when $m \geq \ell + 1$, we can solve this system of equations for the s_i(and $y \bmod (p-1)$), and so solve the search LHS instance. □

7 On the equivalence of SCDH and SDLP

Here, we reduce SDLP to the SCDH problem via an efficient quantum algorithm. Since SCDH reduces to SDLP trivially, this establishes the quantum polynomial equivalence of the two problems, stated as an open problem in [4]. We note the results of previous studies [24,25] on the corresponding problem in the commutative case; our result is analogous to [24, Theorem 1]. We do this by transforming SDLP instances into HSP instances, assuming the presence of a SCDH oracle. Recall:

Definition 9. (HSP) Let $f : G \to S$ be a function from finite group G to a set S that is constant on the cosets of some $H \leq G$; i.e. $f(g) = f(g')$ if and only if $gH = g'H$. Given f, G, S, find a generating set of H.

We refer below to $\mathrm{SCDH}^2_{g,\phi,x}$, which is the general SCDH problem restricted to the task of doubling in the argument of $s_{g,\phi}(x)$; i.e., one solves $\mathrm{SCDH}^2_{g,\phi,x}$ if given g, ϕ, and $s_{g,\phi}(x)$, one computes $s_{g,\phi}(x + x) = s_{g,\phi}(2x)$. Note that this is weaker than a general SCDH oracle, which returns $s_{g,\phi}(a + b)$ given $s_{g,\phi}(a)$ and $s_{g,\phi}(b)$ for any $a, b \in \mathbb{Z}/n\mathbb{Z}$.

Theorem 6. *There is a quantum polynomial-time reduction from* $\mathrm{SDLP}_{g,\phi,x}$ *to* $\mathrm{SCDH}^2_{g,\phi,x}$.

Proof. Let $x \in \mathbb{Z}/n\mathbb{Z}$, $(g,\phi) \in G \rtimes \mathrm{Aut}(G)$, and suppose we are given $s_{g,\phi}(x)$. We assume that given (g,ϕ), $s_{g,\phi}(x)$, and $s_{g,\phi}(y)$, we are able to compute $s_{g,\phi}(x + y)$ in the case $x = y$. In particular, we can then compute $s_{g,\phi}(ax)$ for any a in (classical) polynomial time by computing $s_{g,\phi}(2x) = s_{g,\phi}(x + x)$, writing a in base 2, and then repeatedly doubling and adding in the argument of $s_{g,\phi}(\cdot)$ appropriately.

We then define a map $f : \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z} \to \mathcal{X}_{g,\phi}$, $(a,b) \mapsto \phi^b(s_{g,\phi}(ax))s_{g,\phi}(b)$. This can be rewritten $f(a,b) = s_{g,\phi}(ax + b)$. Observe that if $f(a,b) = f(a',b')$, then we must have $ax + b = a'x + b' \bmod n$, since the group action of $\mathbb{Z}/n\mathbb{Z}$ on $G \rtimes \mathrm{Aut}(G)$ is regular. We then find that $f(a,b) = f(a',b')$ if and only if $(a,b) = (a',b') + \lambda(1,-x)$. This is an HSP instance, which can be solved in quantum polynomial time via Shor. □

We note that our result assumes a perfect $\mathrm{SCDH}^2_{g,\phi,x}$ oracle; we leave for future work the adaptation of the results of Montgomery and Zhandry [25], which hold for algorithms solving CDH with non-negligible advantage.

8 Relation of SDLP to HSP

In this final section we explain why, we could not solve the SDLP problem via a reduction to an HSP instance in an analogous manner to the abelian DLP.

DLP is reduced to HSP via the map $f(a,b) = s^a g^b$, where $g^x = s$, with $a, b \in \mathbb{Z}/n\mathbb{Z}$. Then, $f(a,b) = g^{ax+b}$, and $f(a,b) = f(a',b')$ if and only if $(a,b) = (a',b') + \lambda(1,-x)$.

In that spirit, one might try setting $f(a,b,c) = (s_{g,\phi}(x), \phi^a)^c (g,\phi)^b$. Then, if $a = x$, we have $f(a,b,c) = (g,\phi)^{cx+b}$, and we would have defined a map from an abelian group into the cyclic group $\langle (g,\phi) \rangle$, as is done for DLP. The condition $a = x$ seems problematic, however. Note $f(a,b,c) = f(a',b',c')$ if $(a,b,c) = (x,b',c') + \lambda(0,-x,1)$, as (some) solutions have the form $(x,0,0) + \langle (0,-x,1) \rangle$, which is an affine line in $(\mathbb{Z}/n\mathbb{Z})^3$. This, however, is not a "period" in the sense of Shor that Shor's algorithm for the HSP requires. Thus, an obstacle for defining the required map is the "hiding" of ϕ^x, which prevents an adversary for defining a map into $\langle (g,\phi) \rangle$.

One might observe that we are not given a group element, but merely an element of the orbit $\mathcal{X}_{g,\phi}$ of (g,ϕ) under the action of $\mathbb{Z}/n\mathbb{Z}$. This might prompt one to attempt to define a map $f : \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z} \to \mathcal{X}_{g,\phi}$ in the spirit of the aforementioned map. This would seek to define a map $f(a,b) = s_{g,\phi}(ax+b)$. Then, since the group action is regular, $f(a,b) = f(a',b')$ if and only if $(a,b) = (a',b') + \lambda(1,-x)$, and we could use Shor's period finding algorithm. Since we can add b in the argument, to define such a map, one would first have to define a map $f'(a) = s_{g,\phi}(ax)$. Referring to the previous section, one can see that this is in fact how Theorem 6 was proved, since the possibility of defining such a map follows from assuming SCDH. However, it seems that without the SCDH assumption, one cannot compute $s_{g,\phi}(ax)$ given the available information. This thus can be seen as an obstacle to a complete quantum solution to SDLP.

Funding information: This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC), Grant Numbers EP/X037010/1 and EP/Y037243/1.

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

Conflict of interest: The authors state no conflict of interest.

References

[1] Diffie W, Hellman M. New directions in cryptography. IEEE Trans Inform Theory. 1976;22(6):644–54.

[2] Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science; 1994. p. 124–34.

[3] Battarbee C, Kahrobaei D, Perret L, Shahandashti SF. A subexponential quantum algorithm for the semidirect discrete logarithm problem; 2022. Presented at NIST's Fourth PQC Standardization Conference. Cryptology ePrint Archive, Paper 2022/1165. https://eprint.iacr.org/2022/1165.

[4] Battarbee C, Kahrobaei D, Perret L, Shahandashti SF. SPDH-sign: towards efficient, post-quantum group-based signatures. In: Johansson T, Smith-Tone D, editors. Post-quantum cryptography. Switzerland: Springer Nature; 2023. p. 113–38.

[5] Alamati N, De Feo L, Montgomery H, Patranabis S. Cryptographic group actions and applications. In: Moriai S, Wang H, editors. ASIACRYPT 2020. Cham: Springer International Publishing; 2020. p. 411–39.

[6] Alamati N, Malavolta G, Rahimi A. Candidate trapdoor claw-free functions from group actions with applications to quantum protocols. In: Kiltz E, Vaikuntanathan V, editors. TCC 2022. vol. 13747 of LNCS. Switzerland: Springer Nature; 2022. p. 266–93.

[7] Alamati N, Patranabis S. Cryptographic primitives with hinting property. In: Agrawal S, Lin D, editors. ASIACRYPT 2022. vol. 13791 of LNCS. Switzerland: Springer Nature; 2022. p. 33–62.

[8] Rahman N, Shpilrain V. MAKE: A matrix action key exchange. J Math Cryptol. 2022;16(1):64–72.

[9] Habeeb M, Kahrobaei D, Koupparis C, Shpilrain V. Public key exchange using semidirect product of (semi)groups. In: Jacobson M, Locasto M, Mohassel P, Safavi-Naini R, editorsApplied u. Berlin Heidelberg: Springer; 2013. p. 475–86.

[10] Kahrobaei D, Shpilrain V. Using semidirect product of (semi)groups in public key cryptography. In: Beckmann A, Bienvenu L, Jonoska N, editors. Pursuit of the Universal. Cham: Springer International Publishing; 2016. p. 132–41.

[11] Battarbee C, Kahrobaei D, Shahandashti SF. Semidirect product key exchange: the state of play; 2023. Cryptology ePrint Archive, Paper 2023/594. https://eprint.iacr.org/2023/594.

[12] Roman'kov V. Linear decomposition attack on public key exchange protocols using semidirect products of (semi)groups. CoRR. 2015; http://arxiv.org/abs/1501.01152.

[13] Battarbee C, Kahrobaei D, Shahandashti SF. Cryptanalysis of semidirect product key exchange using matrices over non-commutative rings. Math Cryptol. 2022 March;1(2):2–9. https://journals.flvc.org/mathcryptology/article/view/130528.

[14] Brown DRL, Koblitz N, LeGrow JT. Cryptanalysis of MAKE. J Math Cryptol. 2022;16(1):98–102.

[15] Couveignes JM. Hard Homogeneous Spaces; 2006. Cryptology ePrint Archive, Paper 2006/291. https://eprint.iacr.org/2006/291.

[16] Gnilke OW, Zumbrägel J. Cryptographic group and semigroup actions. J Algebra Appl. 2024;23(07):2530001.

[17] Castryck W, Vander Meeren N. Two remarks on the vectorization problem. In: Isobe T, Sarkar S, editors. INDOCRYPT 2022. vol. 13774 of LNCS. Cham: Springer International Publishing; 2022. p. 658–78.

[18] D'Alconzo G, Di Scala AJ. Representations of group actions and their applications in cryptography. Finite Fields Appl. 2024;99:102476.

[19] Imran M, Ivanyos G. Efficient quantum algorithms for some instances of the semidirect discrete logarithm problem. Designs Codes Cryptography. 2024 May;92:2825–43.

[20] Conrad K. Groups of order p^3; https://kconrad.math.uconn.edu/blurbs/grouptheory/groupsp3.pdf.

[21] Monico C. Remarks on MOBS and cryptosystems using semidirect products; 2021. Cryptology ePrint Archive, Paper 2021/1114. https://eprint.iacr.org/2021/1114.

[22] Rahman N, Shpilrain V. MOBS: matrices over bit strings public key exchange. La Matematica. 2024 June;3:1198–206.

[23] Battarbee C, Kahrobaei D, Tailor D, Shahandashti SF. On the efficiency of a general attack against the MOBS cryptosystem. J Math Cryptol. 2022;16(1):289–97.

[24] Galbraith S, Panny L, Smith B, Vercauteren F. Quantum equivalence of the DLP and CDHP for group actions. Math Cryptol. 2021 June;1(1):40–4. https://journals.flvc.org/mathcryptology/article/view/122741.

[25] Montgomery H, Zhandry M. Full quantum equivalence of group action DLog and CDH, and more. In: Agrawal S, Lin D, editors. ASIACRYPT 2022. vol. 13791 of LNCS. Switzerland: Springer Nature; 2022. p. 3–32.