On the Semidirect Discrete Logarithm Problem in Finite Groups

Christopher Battarbee[1], Giacomo Borin[2,14], Julian Brough[13], Ryann Cartor[3], Tobias Hemmert[13], Nadia Heninger[4], David Jao[5], Delaram Kahrobaei[6,7], Laura Maddison[8], Edoardo Persichetti[9], Angela Robinson[10], Daniel Smith-Tone[10,11], and Rainer Steinwandt[12]

[1] Sorbonne University, CNRS, LIP6, PolSys, Paris, France
[2] IBM Research Europe
[3] Clemson University, U.S.
[4] University of California, San Diego, U.S.
[5] University of Waterloo, Ontario, Canada.
[6] Departments of Computer Science and Mathematics, Queens College, City University of New York, U.S.
[7] Department of Computer Science and Engineering, Tandon School of Engineering, New York University, U.S.
[8] University of Ottawa, Ontario, Canada
[9] Florida Atlantic University, U.S.
[10] National Institute of Standards and Technology, U.S.
[11] University of Louisville, U.S.
[12] University of Alabama in Huntsville, U.S.
[13] Bundesamt für Sicherheit in der Informationstechnik, Bonn, Germany
[14] University of Zurich

Abstract. We present an efficient quantum algorithm for solving the semidirect discrete logarithm problem (SDLP) in *any* finite group. The believed hardness of the semidirect discrete logarithm problem underlies more than a decade of works constructing candidate post-quantum cryptographic algorithms from non-abelian groups. We use a series of reduction results to show that it suffices to consider SDLP in finite simple groups. We then apply the celebrated Classification of Finite Simple Groups to consider each family. The infinite families of finite simple groups admit, in a fairly general setting, linear algebraic attacks providing a reduction to the classical discrete logarithm problem. For the sporadic simple groups, we show that their inherent properties render them unsuitable for cryptographically hard SDLP instances, which we illustrate via a Baby-Step Giant-Step style attack against SDLP in the Monster Group.

Our quantum SDLP algorithm is fully constructive, up to the computation of maximal normal subgroups, for all but three remaining cases that appear to be gaps in the literature on constructive recognition of groups; for these cases SDLP is no harder than finding a linear representation. We conclude that SDLP is not a suitable post-quantum hardness assumption for any choice of finite group.

Keywords: Group-Based Cryptography, Semidirect Discrete Logarithm Problem, Post-Quantum Cryptography

1 Introduction

There has been a significant amount of research on *semidirect product* cryptography within the post-quantum community [24, 28, 41, 42, 23] since its introduction in 2013 by Habeeb et al. [24]. This approach aims to use the group-theoretic notion of the semidirect product to generalize the discrete logarithm problem (DLP) in a manner that resists quantum attacks. The resulting problem is called the *Semidirect Discrete Logarithm Problem* (SDLP), and is the subject of this paper.

The NIST Post-Quantum Standardization process [39] has motivated work on a wide variety of computational problems and candidate constructions for post-quantum cryptographic algorithms. While lattice-based cryptography may currently be the most well-represented among post-quantum schemes, there is a desire to have a diverse collection of candidates, computational hardness assumptions and algorithms. This would provide a hedge against cryptanalytic surprises (such as the late-breaking attacks against Rainbow and SIKE) and allow for different performance tradeoffs, as well as advanced functionalities.

In this light, SDLP is an appealing generalization of DLP over cyclic groups that can be used to define analogues of discrete logarithm-based cryptography over non-commutative (semi-)groups. SDLP offers an unusual degree of flexibility; almost all of the cryptosystems are defined for *any* finite group, and several are defined for finite semigroups. Battarbee et al. [7, 6] showed that the machinery of SDLP gives rise to a group action and suggests that this might allow efficiency improvements over other candidates for *group-action* based cryptography, especially in the realm of digital signature schemes.

Historically, cryptanalysis of SDLP-based schemes has been specific to a particular choice of group. For example, there have been several proposals of groups to be used with Semidirect Product Key Exchange (SDPKE), which is the analogue of Diffie-Hellman Key Exchange (DHKE) for SDLP [24, 28, 41, 42, 23]. Each of these proposals was later shown to be insecure due to some feature of the selected platform group [38, 43, 16, 37, 36]. However, analogously to the relationship between DHKE and the Diffie-Hellman problems, a break of SDPKE for some group does not demonstrate that SDLP is easy in that group. More recently, Imran and Ivanyos [25] showed that SDLP in a solvable group admits a reduction to standard quantum-vulnerable problems. While this work has eliminated some candidate constructions, it leaves unresolved the question motivating our work: is there any choice of finite group G such that SDLP in G is post-quantum secure?

This question has remained unanswered for over a decade of active research in the area. In this work, we prove that the answer is negative. Our result makes use of the famous Classification of Finite Simple Groups and develops

The corresponding authors of this work, Christopher Batterbee and Giacomo Borin, can be reached at `christopher.battarbee@lip6.fr` and `sdlp@gbor.in`, respectively.

a generalization of the "decomposition" methods of [25]. In particular, we will repeatedly use the "recursion tool" of [25] to reduce an instance of SDLP in an arbitrary finite group to several instances of SDLP in finite simple groups. Since there is a relatively short and known list of all possible finite simple groups, we then devise quantum and classical algorithms for solving SDLP or reducing it to the problem of finding a linear representation of the group, that we can solve (up to some technical detail concerning constructive recognition of groups) in each family of finite simple groups.

Our contributions are highlighted below.

- We develop a more sophisticated method of decomposition into "smaller" instances of SDLP, based on the ideas of [25]. In particular we show that, for SDLP in an arbitrary finite group G, one can always generate logarithmically-many instances of SDLP in simple groups; moreover, solving these instances of SDLP suffices to solve SDLP in the group G.
- We solve SDLP in non-sporadic simple groups by studying their representations and, building on another idea of [25], give a reduction to the classical DLP after some linear algebra calculations of polylogarithmic complexity.
- We propose an adaptation of Shanks' Baby-Step-Giant-Step algorithm which efficiently (and classically) solves SDLP in sporadic groups, exploiting the relatively low orders of their elements. This completes our claim that one can solve SDLP in a practical manner in an arbitrary finite group G.

While our work eliminates hope for quantum-secure SDLP-based cryptography over finite groups, the corresponding problem for semigroups, which is featured in some previous proposals [24], remains an interesting open problem. Indeed, evidence suggests that some group-theoretic problems may be harder to solve on semigroups than on groups. For example, Childs and Ivanyos [17] prove an exponential lower bound on the number of quantum queries required to solve the constructive semigroup membership problem on a black-box semigroup, whereas the corresponding problem for black-box groups is known to be quantum polynomial-time since it simply reduces to DLP. We remark also that our techniques are unlikely to translate to the infinite case of SDLP.

1.1 Paper Organization and Contributions

We prove the following main results.

Theorem 1. *Let G be a finite black-box group. Given an oracle computing maximal normal subgroups, in order to solve* SDLP *in G, it suffices to solve* SDLP *in at most $\log|G|$ many simple groups. We can compute the information defining these instances of* SDLP *in simple groups in quantum polynomial time in $\log|G|$.*

Theorem 2. *Let G be a finite black-box group and suppose there is an efficient linear (or projective) representation of G of dimension n. One can solve* SDLP *in G in quantum polynomial time in n and $\log|G|$.*

Corollary 1. *Let S be a finite simple black-box group, that is not one of the groups ${}^2F_4(2^{2n+1})$ or ${}^3D_4(2^e)$. One can solve* SDLP *in S in quantum polynomial time in* $\log|S|$.

We will explicitly discuss SDLP in the two groups omitted by Corollary 1 in Section 6. The techniques for computing arbitrary maximal normal subgroups comes from the literature on various computational group theoretic problems, in particular the task of computing composition series of groups. The literature here does not appear to be completely resolved, and we discuss it in Appendix A. The rest of our paper is organized as follows (which also gives a guide to the structure of our results). Section 2 gives some background on group theory and some of the computational problems that arise in this work. This section also summarizes the main results of [25] that we generalize in this work. In Section 3, we go into more detail on the main decomposition tool, and generalize it in several steps to finite simple groups. In Section 4, we give a generic method to solve SDLP for any finite group using its linear representation. Combining the results in these two sections gives an efficient reduction of SDLP in any group to SDLP in finite simple groups, as well as an algorithm solving SDLP with running time dependent on the faithful dimension in simple groups. In Section 5, we use the classification of finite simple groups to iterate through each of the families of finite simple groups in turn. Given the previous computational reductions, the main question for each of these families is to construct an efficient linear representation from a black-box group; this is known to be in probabilistic quantum polynomial time for all but two minor special cases. Finally, the sporadic groups can be easily dispensed with, either via a brute-force search or via an adapted baby-step giant-step algorithm. We conclude in Section 6 that SDLP on finite groups is not a reliable candidate for quantum-resistant cryptography.

2 Preliminaries

The semidirect discrete logarithm problem arises from the study of the semidirect product of a group G by its own automorphism group. Let us briefly recall the definition:

Definition 1 (Holomorph). *Let G be a group with automorphism group* $\mathrm{Aut}(G)$. *The* semidirect product *of G by* $\mathrm{Aut}(G)$, *written* $G \rtimes \mathrm{Aut}(G)$, *is the set of ordered pairs from* $G \times \mathrm{Aut}(G)$ *equipped with multiplication defined by*

$$(g, \phi)(g', \psi) := (g\phi(g'), \phi \circ \psi)$$

where $\circ$ denotes function composition. We call this structure the holomorph *of G and denote it by* $\mathrm{Hol}(G)$.

By induction, one can verify that for $(g, \phi) \in \mathrm{Hol}(G)$ and $x \in \mathbb{N}$, we have

$$(g, \phi)^x = (\underbrace{g\phi(g) \dots \phi^{x-1}(g)}_{=:s_{g,\phi}(x)}, \phi^x),$$

and we can think of this as a function $s_{g,\phi} : \mathbb{Z} \to G$, mapping the exponent x to the projection onto the G-component of $(g, \phi)^x$. For finite groups G, the order of elements in $\mathrm{Hol}(G)$ is bounded above by $|G|$ (see [11]), so we may, without loss of generality, choose to restrict the domain of $s_{g,\phi}$ to a finite set.

Definition 2 (Semidirect Discrete Logarithm Problem). *Let G be a group and fix $(g, \phi) \in \mathrm{Hol}(G)$. Suppose $h = s_{g,\phi}(x)$ for some $x \in \mathbb{Z}$. We define $\mathsf{SDLP}(G, \phi, g, h)$ to be the set consisting of all the integers i such that $s_{g,\phi}(i) = h$. The* Semidirect Discrete Logarithm Problem *(SDLP) is to determine this set.*

Remark 1. It will be useful in some contexts for us to say "SDLP for G and ϕ", for a finite group G and one of its automorphisms ϕ. By this, we just mean an instance of SDLP where one recovers $\mathsf{SDLP}(G, \phi, g, h)$, without wishing to specify g and h.

Since $s_{g,\phi}(x)$ is the projection of a holomorph element onto one of its coordinates, the SDLP setup does not directly expose an element of G or $\mathrm{Aut}(G)$. The problem is therefore not trivially equivalent to a standard DLP. Thinking of $s_{g,\phi}$ in terms of a projection also tells us how to efficiently compute it: we can compute exponentiation in the holomorph using standard square-and-multiply techniques, and then project the result to obtain the desired value.

2.1 Essential Group Theory Notions

Let G be a group. A subgroup $N \leq G$ is said to be *normal* if for all $g \in G$ and $n \in N$, $gng^{-1} \in N$. We use $N \triangleleft G$ to denote that N is a normal subgroup of G. We can then define the *quotient group* G/N to be the set of left cosets of N in G. In other words, $G/N = \{gN \mid g \in G\}$. The group operation on G/N is induced by the group operation on G in the obvious way.

A group G is *simple* if it has no non-trivial proper normal subgroups, and we refer to a subgroup H of a group G as *characteristic* if $\phi(H) = H$ for every automorphism $\phi \in \mathrm{Aut}(G)$. The group G is said to be *characteristically simple* if it has no non-trivial proper characteristic subgroups. The example $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ illustrates that being characteristically simple is a strictly weaker property than being simple. A subnormal series $1 = H_m \triangleleft H_{m-1} \triangleleft \cdots \triangleleft H_1 \triangleleft H_0 = H$ of a group H is called a *composition series* if each quotient H_i/H_{i-1} is simple and called a *quasi-composition series* if each quotient is either abelian or non-abelian simple.

For technical reasons we require that any computational representation of a group G comes with two attributes $\mathsf{CS\_Abelian\_Flag}$, and $\mathsf{CS\_NonAbelianFlag}$, which are by default set to 0 (i.e., $G.\mathsf{CS\_Abelian\_Flag} = G.\mathsf{CS\_NonAbelian\_Flag} = 0$). One of our algorithms later on may update these values if it detects that the group is either of two special cases of characteristically simple.

A *linear representation* of a group G on a finite-dimensional vector space V is a group homomorphism

$$\psi : G \to \mathsf{GL}(V).$$

Here, $\mathsf{GL}(V)$ denotes the general linear group on V. We also consider *projective* linear representations, i.e., homomorphisms $G \to \mathbb{P}\mathsf{GL}(V)$, where $\mathbb{P}\mathsf{GL}(V) \cong \mathsf{GL}(V)/Z(\mathsf{GL}(V))$ contains the invertible linear maps acting on $\mathbb{P}(V)$ (since scalar matrices act trivially on $\mathbb{P}(V)$). If $\mathbf{A} \in \mathsf{GL}(V)$ we write $[\mathbf{A}]$ for the corresponding class in $\mathbb{P}\mathsf{GL}(V)$.

Black-Box Groups. The introduction of *black-box groups* can be traced back to Babai and Szemeredi [4] as a useful abstraction of computations in groups.

Definition 3 (Black-Box Group). *A **black-box group** $G \subset \{0,1\}^n$ is a group whose elements are bit strings of length n, endowed with an oracle that performs the group operations, multiplication and inversion, and can check if one element is the identity or not (this is equivalent to checking if two elements are equal or not).*

As an additional requirement, for technical reasons we will need our black-box groups to come equipped with a unique labelling; that is, a function λ on the bitstrings representing the group that is such that $\lambda(x) = \lambda(y)$ if and only if x and y represent the same group element.

The use of black-box oracles for groups is not new to cryptography. As an example, Shoup proved lower bounds for generic algorithms solving DLP using black-box groups [47]. This is a conservative computational model for cryptanalysis of SDLP-based cryptography, since any construction instantiated on a particular group will need to be able to perform operations on the base group G (and $\mathrm{Aut}(G)$) and test the equality of the resulting operations.

The Black-Box Group model is also of interest for computational group theorists as a tool to investigate the complexity of several group related problems such as the Hidden-Subgroup Problem [26], or in relation to "The computational matrix group project" [34, 40].

Of particular relevance is the **Constructive Recognition Problem**, proposed by Babai and Beals [1, Section 9.2], in which one is asked to find a computationally efficient isomorphism between a simple black-box group and an explicitly defined simple group. Observe that for the case of cyclic groups of prime order this problem reduces exactly to DLP since, given $\phi : G \xrightarrow{\sim} \mathbb{Z}/p\mathbb{Z}$, we can easily compute logarithms (divisions) in $\mathbb{Z}/p\mathbb{Z}$.

Several works [12, 13, 2, 27, 29, 30, 1] have investigated the constructive recognition problem for other families of simple groups; this is commonly done by reducing it to the case of $\mathbb{P}\mathsf{SL}(2,q)$ using so-called *number theory oracles*, i.e., oracles for solving discrete logarithm and factoring, to handle large finite-field computations [18, 2]. These algorithms thus run in quantum polynomial time [46].

2.2 Related Work and Known Results

Broadly speaking, there are two main categories of literature on SDLP: cryptographic constructions based on the Semidirect Product Key Exchange (SDPKE)

and the associated cryptanalysis, and algorithmic analysis of the underlying SDLP problem itself.

The first category of literature encompasses a decades-long cat-and-mouse game between papers suggesting parameters and choices of groups to instantiate SDPKE [24, 28, 41, 42, 23], and works cryptanalyzing the results [38, 43, 16, 37, 36]. These papers occur as responses to each other, in the sense that new proposals are patches to avoid the attacks of prior works. For a detailed review of the chronology see [8].

In the same way that the security of DHKE is not precisely equivalent to DLP, the security of SDPKE is not precisely equivalent to SDLP. The works mentioned above do not address the complexity of solving SDLP; the first result in this direction dates to 2022. This and subsequent such results form the second category of literature mentioned above, which also includes the present paper. Battarbee et al. [6] pointed out a connection to group actions and later exploited it [7] to give a subexponential quantum algorithm for SDLP.

Mendelsohn et al. [35] found faster methods for some small parameters. Most recently, Imran and Ivanyos [25] gave an efficient polynomial-time quantum algorithm to solve SDLP for solvable groups and matrix groups with certain associated endomorphisms. Our work is a generalization of this paper to all finite groups.

Imran and Ivanyos introduce two important notions, which we sketch here. The first is that, given a group G and a normal subgroup N, in order to solve SDLP in G, it suffices to solve SDLP in N and G/N. The second is that, if G is a matrix group, we can show that SDLP reduces to an instance of DLP after the application of some linear algebraic methods.[1] Suppose we can compute a composition series of an arbitrary group G; then, provided the composition factors are suitable matrix groups (or elementary abelian groups, in which SDLP is predictably easy), we can use the decomposition algorithm inductively to solve SDLP in the composition factors and to recover a solution of SDLP in the group that we started in. This breaks, among other things, all the finite solvable groups (which includes every group proposed for use with SDLP-based cryptography).

Our work can be seen as a more sophisticated version of this method. By refining the method of computing the appropriate subgroups we can reduce the solution to solving appropriate instances of SDLP in the simple groups. In addition, we construct a generalization of the reduction in a matrix group that turns out to be particularly effective for simple groups. Indeed, because we know that only the simple groups listed by the classification of simple groups can appear in this decomposition, and since we can show that each of these is vulnerable to some method of solving SDLP, we can show that SDLP is easy for any finite group, resolving a loose conjecture of [25].

[1] Interestingly, this method is somewhat similar to the "linear decomposition" attacks presented in the analysis of SDPKE.

For the purpose of describing our algorithms let us recall some of the known results relating to the structure of SDLP.

Prior Results. One of the main ideas of [25] is to reframe SDLP as an orbit problem. For each pair (g, ϕ) in the holomorph of G consider the function $\rho_{(g,\phi)}$ defined by $\rho_{(g,\phi)}(h) = g\phi(h)$. It is not difficult to check by induction that $\rho^x_{(g,\phi)}(h) = g\phi(g)\cdots\phi^{x-1}(g)\phi^x(h)$. We therefore get the following equivalent definition of SDLP.

Definition 4 (SDLP(G, ϕ, g, h)). *Let G be a finite group, and $\phi \in \mathrm{Aut}(G)$ be one of its automorphisms. Suppose $h = \rho^x_{(g,\phi)}(1_G)$ for some $x \in \mathbb{N}$. We define the set* SDLP(G, ϕ, g, h) *to be the set of integers i satisfying*

$$h = \rho^i_{(g,\phi)}(1_G).$$

The Semidirect Discrete Logarithm Problem, or SDLP*, is to determine this set.*

We will use both variants interchangeably. Let us also recall some of the results on the set of solutions to SDLP: the following is a synthesis of ideas found in [6, 7]. In the following, the symbol 1 refers to the integer value 1, and 1_G denotes the group identity; these are (clearly) not the same.

Theorem 3. *Let G be a finite group and ϕ one of its automorphisms. Consider* SDLP *for $g, h \in G$. There exists an integer n_0 (dependent on g and ϕ) such that $\rho^{n_0}_{g,\phi}(1_G) = s_{g,\phi}(n_0) = 1_G$, and the set*

$$\{1_G, s_{(g,\phi)}(1), ..., s_{(g,\phi)}(n_0 - 1)\} = \{1_G, \rho_{(g,\phi)}(1_G), ..., \rho^{n_0-1}_{(g,\phi)}(1_G)\}$$

has size n_0, and is exactly the codomain of $s_{(g,\phi)}$. We have that one can compute n_0 in quantum polynomial time with a Shor-like period-finding algorithm, and that the solution set SDLP(G, ϕ, g, h) *is of the form*

$$\{t_0 + tn_0 : t \in \mathbb{Z}\}$$

where $0 \leq t_0 < n_0$.

Finally, although some of the ideas of [25] are given in detail in the main body of the present paper, we will just quote the fact given as [25, Theorem 6] that one can solve SDLP in an elementary abelian group in time polynomial in the input size of the group. This will be necessary since several of the results on simple groups will require that the simple group is non-abelian, and finite cyclic groups of prime order are the only abelian simple groups. Note also that, although our more general ideas capture the result of [25] for solving SDLP in solvable groups, their specific methods may be slightly more efficient in practice for this particular case.

3 The Main Reduction

Recall from the discussion in the previous section that Imran and Ivanyos [25] provide a solution for SDLP in solvable groups by descending a composition series (using Theorem 3 in their paper), at each step encountering an easy variant of SDLP in an elementary abelian group. In this section, we significantly generalize the results of [25], by using their method to completely reduce an arbitrary instance of SDLP to several instances of SDLP in a simple group. In particular, Theorem 6 demonstrates that if we know how to compute maximal normal subgroups, in order to solve some instance of SDLP in a finite group G, it suffices to solve at most $\log|G|$ instances of SDLP in a simple group. The data describing each of these instances of SDLP can be obtained in time quantum polynomial in $\log|G|$.

We will defer the proof of this result to the end of the section. We begin by developing more sophisticated techniques for computing the subgroups required for [25, Theorem 3], and devise a contingency for the case in which no such subgroups exist.

3.1 Reduction to SDLP in Simple Groups

Let us review the central "recursion tool" of Imran-Ivanyos [25, Theorem 3]. The main idea of the recursion tool is to demonstrate that if we can find a normal subgroup N of G that is invariant under our automorphism, solving $\mathsf{SDLP}(G,\phi)$ can be reduced to solving $\mathsf{SDLP}(N,(\phi|_N)^{n_0})$ and $\mathsf{SDLP}(G/N,\bar{\phi})$ for some n_0, automorphism $\bar{\phi}$ and suitable elements.

We will state and prove the result in full, in order to review ideas from its proof that are important in our reduction algorithms. For these purposes, we first provide the following lemma concerning powers of $\rho_{(g,\phi)}$.

Lemma 1. *Let $g\in G$, $\phi\in\mathrm{Aut}(G)$. For any integer x, then $\rho^{-x}_{(g,\phi)}(h):=(\rho^{x}_{(g,\phi)})^{-1}(h)=\rho^{x}_{(\phi^{-1}(g^{-1}),\phi^{-1})}(h)$. Additionally, for any $m,n\in\mathbb{Z}$, then $\rho^{mn}_{(g,\phi)}=\rho^{m}_{(\rho^{n}_{(g,\phi)}(1_G),\phi^n)}$.*

Proof. The first statement follows from the observation that $(\rho_{(g,\phi)})^{-1}(f)=\phi^{-1}(g^{-1}f)=\rho_{(\phi^{-1}(g^{-1}),\phi^{-1})}(h)$.

As $\rho^{x}_{(g,\phi)}(h)=\rho^{x}_{(g,\phi)}(1_G)\phi^x(h)$ it suffices to prove the statement for $h=1_G$. Assume first m is positive. If n is also positive then

$$\rho^{m}_{(\rho^{n}_{(g,\phi)}(1_G),\phi^n)}(1_G)=\prod_{i=0}^{m-1}(\phi^n)^i\left(\rho^{n}_{(g,\phi)}(1_G)\right)=\rho^{mn}_{g,\phi}(1_G).$$

While for negative n applying the formula for $\rho^{-x}_{(g,\phi)}$ above yields

$$\rho^{mn}_{(g,\phi)}(1_G)=\rho^{m(-n)}_{(\phi^{-1}(g^{-1}),\phi^{-1})}(1_G)=\rho^{m}_{(\rho^{-n}_{(\phi^{-1}(g^{-1}),\phi^{-1})}(1_G),\phi^n)}(1_G)=\rho^{m}_{(\rho^{n}_{(g,\phi)}(1_G),\phi^n)}(1_G).$$

On the other hand, if m is negative then

$$\rho^{mn}_{(g,\phi)} = \left(\rho^{-mn}_{g,\phi}(1_G)\right)^{-1}(1_G) = \left(\rho^{-m}_{(\rho^n_{(g,\phi)}(1_G),\phi^n)}\right)^{-1}(1_G) = \rho^{m}_{(\rho^n_{(g,\phi)}(1_G),\phi^n)}(1_G).$$

□

Theorem 4 (Recursion tool, [25]). *Let G be a finite group, $\phi \in \mathrm{Aut}(G)$ and $g, h \in G$. Given a ϕ-invariant normal subgroup N, set $\overline{\phi}$ to be the induced automorphism on G/N and $\phi|_N$ the induced automorphism on N. Then*

$$\mathrm{SDLP}(G, \phi, g, h) = (t_0 + t_1 n_0) + (n_1 n_0)\mathbb{Z},$$

where

$$\mathrm{SDLP}(G/N, \overline{\phi}, gN, hN) = t_0 + n_0\mathbb{Z}$$

and

$$\mathrm{SDLP}(N, (\phi|_N)^{n_0}, \rho^{n_0}_{(g,\phi)}(1_G), (\rho^{t_0}_{(g,\phi)})^{-1}(h)) = t_1 + n_1\mathbb{Z}$$

.

Proof. As N is ϕ-invariant it follows that $\phi|_N \in \mathrm{Aut}(N)$ and $\overline{\phi}(gN) := \phi(g)N$ is a well defined automorphism of G/N. In the group G/N, for any $f \in G$, it follows that

$$(\rho_{(g,\phi)}(f))N = (g\phi(f))N = (gN)\phi(fN) = \rho_{(gN,\overline{\phi})}(fN)$$

and thus inductively it can be shown that $(\rho^x_{(g,\phi)}(f))N = \rho^x_{(gN,\overline{\phi})}(fN)$ for any integer x.

Assume $h = \rho^x_{(g,\phi)}(1_G)$ for some x. Then $hN = \rho^x_{(g,\phi)}(1_G)N = \rho^x_{(gN,\overline{\phi})}(1_{G/N})$. In other words $x \in \mathrm{SDLP}(G/N, \overline{\phi}, gN, hN) = t_0 + n_0\mathbb{Z}$. Hence it suffices to compute the set of all t such that $h = \rho^{t_0+tn_0}_{(g,\phi)}(1_G)$.

By applying the second property from Lemma 1,

$$h = \rho^{t_0+tn_0}_{(g,\phi)}(1_G) \iff (\rho^{t_0}_{(g,\phi)})^{-1}(h) = \rho^{tn_0}_{(g,\phi)}(1_G) = \rho^{t}_{(\rho^{n_0}_{(g,\phi)}(1_G),\phi^{n_0})}(1_G).$$

Moreover, by Theorem 2.5, the definition of n_0 implies that $\rho^{n_0}_{(g,\phi)}(1_G)N = \rho^{n_0}_{(gN,\overline{\phi})}(1_{G/N}) = 1_{G/N}$. In other words $\rho^{n_0}_{(g,\phi)}(1_G) \in N$. Thus $h = \rho^{t_0+tn_0}_{(g,\phi)}(1_G)$ if and only if $t \in \mathrm{SDLP}(N, (\phi|_N)^{n_0}, \rho^{n_0}_{(g,\phi)}(1_G), (\rho^{t_0}_{(g,\phi)})^{-1}(h))$. In particular,

$$\mathrm{SDLP}(G, \phi, g, h) = t_0 + n_0(t_1 + n_1\mathbb{Z}) = t_0 + n_0 t_1 + n_0 n_1\mathbb{Z}.$$

□

We can now consider applying this tool to reduce the general case of SDLP, via a composition series, to the case of SDLP in simple groups.

To determine $\mathsf{SDLP}(G, \phi, g, h)$ via the application of Theorem 4, we need to construct the following: a ϕ-invariant normal subgroup N of G; the quotient G/N; the induced map $\overline{\phi}$ on the quotient; and the integer n_0. We assume that given $N \lhd G$, constructing G/N can be done efficiently. Moreover, [25] describes a general method of evaluating the induced map $\overline{\phi}$. The computation of the integer n_0 can be done with a Shor-like algorithm by Theorem 3. Thus the main obstacle is the computation of the ϕ-invariant normal subgroup.

3.2 Computing automorphism invariant Normal Subgroups.

The purpose of this section is to describe an algorithm that computes the invariant subgroups. The technique can be understood as building a machine taking as input some maximal normal subgroup of the group in which we wish to address SDLP, and outputting a ϕ-invariant subgroup of the maximal normal subgroup. The techniques for computing maximal normal subgroups in arbitrary finite groups are taken from the literature, which does not appear to be entirely resolved on this subject. For now, we assume we have an oracle $\Gamma()$, that on input of a black-box description of a group G, outputs a black-box description of one of its maximal normal subgroups. Discussion of the methods in the literature for implementing such an oracle are delayed to Appendix A.

Our method consists of showing that either we can compute a ϕ-invariant normal subgroup from an arbitrary maximal normal subgroup, or G has no characteristic subgroups (that is, it is "characteristically simple") - and it is well known (see [48, Lemma 2.8]) that a group is characteristically simple if and only if it is isomorphic to S^k, where S is a simple group. In this latter case we have two sub-cases: either G is abelian, or ϕ acts transitively on the k factors of G, allowing a bespoke method of reduction.[2]

A method of computing ϕ-invariant normal subgroups from an arbitrary maximal normal subgroup N is given in [25], and works as follows. Set $N_1 = N$ and for $i \geq 2$ define $N_i = N_{i-1} \cap \phi^{i-1}(N)$. This sequence must eventually stabilize, say for some integer $j \in \mathbb{N}$: it is not difficult to show that N_j is ϕ-invariant, and that, since each intersection is a subgroup, we arrive at this stabilization within $\log |G|$ steps. For brevity we will refer to this method as the "intersection trick".

Notice that we are not *a priori* guaranteed that the output of the intersection trick is non-trivial (certainly the trivial subgroup is ϕ-invariant). The intersection trick, however, will not terminate with the trivial subgroup if the maximal normal subgroup we started with contains a G-characteristic subgroup, since such a G-characteristic subgroup is also contained in the image of N under any automorphism, by definition. It would therefore suffice to demonstrate that a non-characteristically simple group is such that every maximal normal subgroup contains a characteristic subgroup in G. In fact, we are able to provide this alternate classification of the characteristically simple groups, as shown below.

Lemma 2. *Let G be a finite group. G possesses a non-trivial G-characteristic subgroup if and only if every maximal normal subgroup N of G contains a non-trivial G-characteristic subgroup.*

Proof. The reverse direction is trivial. Assume then that G is not characteristically simple and contains a maximal normal subgroup N. We show that N contains a nontrivial characteristic subgroup of G.

[2] The situation is actually slightly more complicated than this, as we will see.

Consider the subgroup $\mathcal{J}(G)$ defined as the intersection of all maximal normal subgroups, known as the "Jacobson radical" of G. By definition, $\mathcal{J}(G)$ is contained in N and $\mathcal{J}(G)$ is characteristic. Hence it can be assumed that $\mathcal{J}(G)$ is trivial, which implies G is a direct product of simple groups by [5, Remark 4.8]

Set $G = S_1^{a_1} \times \dots S_n^{a_n}$, with $S_i \ncong S_j$ for $i \neq j$. As G is not characteristically simple, $n \geq 2$. Assume $S_1, \dots, S_m$ are non-abelian and $S_{m+1}, \dots, S_n$ are abelian, so that the centre of G is given by $Z(G) = \prod_{i=m+1}^{n} S_i^{a_i}$. Additionally, write each factor as $S_i^{a_i} = S_{i,1} \times \dots \times S_{i,a_i}$.

If N is a maximal normal subgroup, then there exists some pair (i, j) such that $S_{i,j} \not\subseteq N$. By normality, $[N, S_{i,j}] \leq N \cap S_{i,j} = 1$. It follows that $G \cong N \times S_{i,j}$. Hence for any $k \neq i$ it follows that $S_k^{a_k} \subseteq N$, as otherwise there is some l such that $G \cong N \times S_{k,l}$ implying that $S_{k,l} \cong S_{i,j}$. To prove the statement, it thus suffices to show that for each $1 \leq k \leq n$ the subgroup $S_k^{a_k}$ is characteristic in G.

Assume first $k \leq m$ and $\phi \in \mathrm{Aut}(G)$. Let $1 \leq j, j' \leq a_k$. If $\phi(S_{k,j}) \cap S_{k,j'} = 1$ then $[\phi(S_{k,j}), S_{k,j'}] = 1$. Thus if $\phi(S_{k,j}) \cap S_{k,j'} = 1$ for all j', then $\phi(S_{k,j}) \leq C_G(S_k^{a_k}) = \prod_{j \neq k} S_j^{a_j}$; which yields a contradiction as $C_G(S_k^{a_k})$ has no composition factor isomorphic to S_k. Thus there must exist some j' such that $\phi(S_{k,j}) = S_{k,j'}$ and so $\phi(S_k^{a_k}) = S_k^{a_k}$.

Finally consider $k \geq m + 1$. As the S_k are non-isomorphic groups, each $S_k^{a_k}$ must be the unique Sylow p_k subgroup of $Z(G)$ for some prime p_k. Therefore $S_k^{a_k}$ is characteristic in G as being characteristic is transitive. □

Notice that if the intersection trick terminates with the identity, by Lemma 2, G is characteristically simple. However, there are situations where a maximal normal subgroup of a characteristically simple group contains a ϕ-invariant normal subgroup. Whether or not this happens, in the non-abelian case, is related to the the automorphism ϕ. In particular, for S a non-abelian simple group, $Aut(S^k) \cong Aut(S)^k \wr Sym(\{1, ..., k\})$; in other words, every automorphism in $Aut(S^k)$ can be thought of as possessing a unique permutation component.

Lemma 3. *Let G be a non-abelian finite group, and ϕ one of its automorphisms. The intersection trick for determining a ϕ-invariant normal subgroup from a maximal normal subgroup of G terminates in the trivial subgroup if and only if the group $G \cong S^k$ for some non-abelian simple group S with $k \in \mathbb{N}$, and the permutation component of ϕ is a k-cycle.*

Proof. First, we note that the normal subgroups of a non-abelian characteristically simple group S^k are exactly the subgroups $\prod_{j=1}^{l} S_{i_j}$, where $\{i_1, ..., i_l\} \subset \{1, ..., k\}$. In other words, every normal subgroup of S^k corresponds uniquely with a subset of $\{1, ..., k\}$. Clearly, the maximal normal subgroups of S^k correspond to the subsets of $\{1, ..., k\}$ of size $k - 1$.

Set $G = S^k$ for S a non-abelian simple group and suppose the permutation component of ϕ is a k-cycle. Since the maximal normal subgroup N we give as input to the intersection trick is of the form $\prod_{i=1, i \neq j}^{k} S_i$ for some $j \in \{1, ..., k\}$,

we have that

$$\{\phi^i(N) : i \in \mathbb{N}\} = \left\{ \prod_{i=1, i\neq j}^{k} S_i : 1 \leq j \leq k \right\}$$

The intersection of all these subgroups is trivial, and so we are done in this direction.

Now suppose that the intersection trick terminated in the trivial subgroup. We have already seen that the group G must, in this case, be characteristically simple, and so without loss of generality is of the form S^k, where S is a non-abelian simple group. Consider a maximal normal subgroup N of G. We are going to argue that if the permutation component of ϕ, which we will denote σ_ϕ, is not a k-cycle, then N will contain a non-trivial, ϕ-invariant normal subgroup, and so the intersection trick could not have had as output the trivial subgroup - a contradiction.

To see this, consider the orbits of the permutation σ_ϕ (that is, the distinct subsets of $\{1, ..., k\}$ that are invariant under σ_ϕ). Of course, σ_ϕ is a k-cycle if and only if it has a single orbit; suppose that it has strictly more than one. Denote by $\mathcal{I}_N$ the size $k-1$ subset of $\{1, ..., k\}$ corresponding to N under the bijection alluded to above. Because $\mathcal{I}_N$ has size $k-1$ and there are two or more orbits, $\mathcal{I}_N$ must contain one of the orbits. Consider the normal subgroup corresponding to this orbit; since the orbit is fixed under the permutation σ_ϕ, the corresponding subgroup, say N', is fixed under ϕ. Now, the ϕ-invariant normal subgroup N' is contained in N, so the intersection trick will terminate in a subgroup no smaller than N'. In particular, the intersection trick did not terminate in the trivial subgroup, giving the desired contradiction. □

We are now ready to give the algorithm computing ϕ-invariant normal subgroups, given a maximal normal subgroup. In the case that no ϕ-invariant normal subgroup can be found, our algorithm outputs its input as a characteristically simple group, and determines whether this characteristically simple group is abelian or not.

Theorem 5. *Let G be a finite black-box group, and suppose ϕ is an automorphism of G. Given an oracle computing maximal normal subgroups, Algorithm 1 either computes a non-trivial ϕ-invariant subgroup of G, or detects that G is characteristically simple. If characteristic simplicity is detected, the algorithm also detects whether the group was abelian or not. In any case the algorithm finishes in time quantum polynomial in* $\log |G|$.

Proof. Let N be a maximal normal subgroup of G obtained from the oracle Γ. If N contains a non-trivial characteristic subgroup of G then, since this characteristic subgroup will also be contained in $\phi^i(N)$ for every $i \in \mathbb{N}$, the intersection trick will not terminate with the trivial subgroup.

If it does terminate with the trivial subgroup, we have already seen that the group we started with must be characteristically simple. If it is abelian, then, it is elementary abelian, and there are efficient quantum methods of recognising

elementary abelian groups. If this test is failed we indicate instead that we have a non-abelian characteristically simple group. □

Algorithm 1 (***Inv***): Computing ϕ-invariant normal subgroups, or detecting either flavour of characteristically simple group.

Input: G, ϕ, oracle Γ computing maximal normal subgroups
Output: ϕ-invariant $N \triangleleft G$ or G

$N \leftarrow \Gamma(G)$
$N_1 \leftarrow N$
$N_2 \leftarrow \phi(N)$
$j \leftarrow 2$
while $N_j \neq N_{j-1}$ **do**
    $j \leftarrow j + 1$
    $N_{j+1} \leftarrow N_j \cap \phi^{j-1}(N)$
end while
if $N_j \neq \{1\}$ **then**
    return N_j
else if G abelian **then**
    G.CS_Abelian_Flag $\leftarrow 1$ **return** G
else
    G.CS_NonAbelian_Flag $\leftarrow 1$ **return** G
end if

Before moving on to the full reduction, we note that in the case that G is abelian and characteristically simple, the structure of the automorphisms is more complicated than the structure described in Lemma 3, that is, $Aut((\mathbb{Z}/p\mathbb{Z})^n) \cong \mathrm{GL}_n(\mathbb{F}_p)$. In order to avoid dealing with this algebraically, we can now simply outsource the abelian case to the method of [25] for solving SDLP in an elementary abelian group. Otherwise, the group and automorphism we started with have the form described in Lemma 3. We develop an algorithm for handling this case below.

Lemma 4. *Suppose G is a finite, non-abelian, characteristically simple group and ϕ is one of its automorphisms. We have that $G = S^k$ for S some non-abelian simple group and $k \in \mathbb{N}$; suppose moreover that the permutation component of ϕ is a k-cycle. Denote by $[g]_i$ the i-th coordinate of an element in the direct product group. Provided access to an oracle Θ for solving* SDLP *in simple groups, Algorithm 2 solves* SDLP(G, ϕ) *efficiently, with at most k^2 calls to the oracle.*

Proof. First note that by [1, Theorem 5.1], we can decompose G into its non-abelian simple factors. As such we can talk about projections of G onto its co-ordinates, and assume knowledge both of the integer k and black box representation of the simple factor S.

We know that the permutation component of ϕ is a k-cycle, so ϕ^k must consist only of co-ordinate-wise application of automorphisms in $Aut(S)$. Call

these permutations to be applied co-ordinate wise $\phi^k = (\phi_1, ..., \phi_k)$. Set also $(g_1, ..., g_k) = \rho^k_{(g,\phi)}(1_G)$.

We wish to find the integers x such that $h = \rho^x_{(g,\phi)}(1_G)$. Of course, any such integer is of the form $i + kt$ for $0 \leq i < k$. Defining $(h_{i,1}, ..., h_{i,j}) = \rho^{-i}_{(g,\phi)}(h)$, for any $i \in \{0, ..., k-1\}$, have

$$\begin{aligned} h = \rho^x_{g,\phi}(1_G) &\iff \rho^{-i}(h) = \rho^{kt}_{(g,\phi)}(1_G) \\ &\iff (h_{i,1}, ..., h_{i,k}) = \rho^t_{(\rho^k_{g,\phi}(1_G),\phi^k)}(1_S, ..., 1_S) \\ &\iff (h_{i,1}, ..., h_{i,k}) = (\rho^t_{(g_1,\phi_1)}(1_S), ..., \rho^t_{(g_k,\phi_k)}(1_S)) \end{aligned}$$

In other words, given an $i \in \{0, ..., k-1\}$, we get k instances of SDLP in S that we can input to the SDLP oracle Θ. Any value t that solves all k of these instances is such that $x = i + kt$ has $h = \rho^x_{(g,\phi)}(1_G)$. In order to find the solutions of this latter instance of SDLP, then, it suffices to check the k problem instances defined by all k choices of i, giving k^2 total calls to the oracle. This procedure is outlined in Algorithm 2. □

Algorithm 2 (***CSimple***): Solving particular instances of SDLP in non-abelian, characteristically simple groups.

Input: G, ϕ, g, h
Output: Element of solution set of $\mathsf{SDLP}(G, \phi)$ for g, h

$S \leftarrow$ non-abelian simple factor of G
$k \leftarrow$ number of copies of S
$(\phi_1, ..., \phi_k) \leftarrow \phi^k$
Solutions $\leftarrow \{\}$
for i **from** 0 **to** $k-1$ **do**
    $h_{i,1} \leftarrow [\rho^{-i}_{(g,\phi)}(h)]_1$
    SubSolutions $\leftarrow \Theta(S, \phi_1, g_1, h_{i,1})$
    for j **from** 2 **to** k **do**
        $h_{i,j} \leftarrow [\rho^{-i}_{(g,\phi)}(h)]_j$
        SubSolutions $\leftarrow$SubSolutions $\cap\ \Theta(S, \phi_j, g_j, h_{i,j})$
    end for
    Solutions $\leftarrow$ Solutions $\cup\ \{i + k \cdot$ SubSolutions$\}$
end for

3.3 The Decomposition Algorithm

We are now ready to provide our reduction to simple groups.

Theorem 6. *Consider* $\mathsf{SDLP}(G, \phi)$ *for some finite group* G*, one of its automorphisms* ϕ*, and group elements* g, h*. Suppose we have an oracle* Γ *computing maximal normal subgroups of* G*. Suppose, moreover, that we have an oracle* Θ

that, on input of the data S, ν, g, h for S a simple group, ν one of its automorphisms, and $g, h \in S$, outputs the set of solutions of $\mathsf{SDLP}(S, \psi)$ for g, h. There exists an algorithm Solve() *that has the following properties: the algorithm terminates in time polynomial in $\log|G|$, having made logarithmically many calls to Θ; and outputs a solution of $\mathsf{SDLP}(G, \phi)$. The algorithm* Solve() *is defined as in Algorithm 3, where ϕ, n_0, g', h', $\bar{\phi}$ and ψ have the same meaning as in the proof of Theorem 4.*

Algorithm 3 Solve(G, ϕ, g, h)

Input: (G, ϕ, g, h), oracles Γ, Θ
Output: (t, n) such that $SDLP(G, \phi, g, h) = t + n\mathbb{Z}$

1: $N \leftarrow Inv(G, \phi)$ ▷ Algorithm 1
2: **if** N.CS_Abelian_Flag $== 1$ **then**
3: $(t, n) \leftarrow$ solutions obtained from [25] method of solving SDLP in elementary abelian groups
4: **else if** N.CS_NonAbelian_Flag $== 1$ **then**
5: $(t, n) \leftarrow$ CSimple(G, ϕ, g, h) ▷ CSimple (Algorithm 2) can access Θ
6: **else**
7: $(t_0, n_0) \leftarrow$ Solve$(G/N, \bar{\phi}, \psi(g), \psi(h))$
8: $(t_1, n_1) \leftarrow$ Solve(N, ϕ^{n_0}, g', h')
9: $(t, n) \leftarrow (t_0 + t_1 n_0, n_0 n_1)$
10: **end if**
11: **return** (t, n)

Proof. We verify that the algorithm terminates after at most $\log|G| - 1$ internal repetitions of Solve(). Start with G: if it is not simple, there are two cases. If the group is characteristically simple, this is detected by the algorithm Inv defined in Algorithm 1 (which implicitly calls Γ) , and there are two sub-cases. Either the CS_Abelian_Flag attribute is set to 1 by Inv, and we can solve the problem instance by applying the method of [25] for solving SDLP in an elementary abelian group; or CS_NonAbelian_Flag is set to 1, and we solve the problem instance with Algorithm 2. If characteristic simplicity is not detected, Algorithm 1 computes a ϕ-invariant subgroup N, and we run Solve() on the two induced problems defined in N and G/N. For these groups, if they are not simple, repeat the procedure, and so on.

As each normal subgroup of G/N is of the form M/N and $(G/N)/(M/N) \cong G/M$, it follows that the internal repetitions of Solve() reduces the problem to solving instances Solve$(N_i/N_{i-1}, \phi_i, g_i, h_i)$ for a subnormal series $1 \lhd N_1 \lhd \cdots \lhd N_n = G$ such that N_i/N_{i-1} is either abelian or has no ϕ_i-invariant subgroup for suitable automorphisms ϕ_i and elements g_i and h_i. Moreover, as each N_i/N_{i-1} has order at least 2 it follows that $n \leq \log|G|$ and thus Solve() must terminate after at most $\log|G|$ internal repetitions.

□

It now remains to develop methods for solving SDLP in simple groups. The rest of the paper will be devoted to this effort.

4 Reduction to Matrix Power Problem

In this section, we present a rather generic method of solving SDLP—indeed, it is defined for any group. We build on the ideas of [25, Theorem 8], which provides a reduction of SDLP in some finite group G, to the matrix power problem in the case that the group G is a matrix group over a field. Our observation is that, by looking at the linear representations of an arbitrary group, there is a sense in which *every* group is a matrix group over a field. Moreover, in the case where ϕ is inner, we are able to compute a linear map that "mimics" the effect of $\rho_{(g,\phi)}$, thereby allowing us to apply the same techniques given by [25, Theorem 8]. It turns out that simple groups are well-suited to the application of this method, because the outer automorphism group of a simple group in general remains quite small.

Let us first outline the intuition behind the method: first, by Cayley's theorem, we know that every finite group G admits a faithful linear representation[3]; that is, an injective group homomorphism $G \to \mathsf{GL}_n(K)$ for some field K. Now, $\mathsf{GL}_n(K)$ lives in the ambient space $M_n(K)$, the matrix algebra of all $n \times n$ matrices with entries in the field K. We can think of this space as an n^2-dimensional vector space equipped with the natural addition and scalar multiplication, so we can imagine that we have a linear map T on this vector space. Suppose that this map T is such that $T \circ \psi = \psi \circ \rho_{(g,\phi)}$; we then immediately have that $T^i \circ \psi = \psi \circ \rho^i_{(g,\phi)}$. It follows that, in order to solve the SDLP instance, it suffices to find an integer x such that $T^x \cdot \psi(1_G) = \psi(h)$, where $\psi(1_G)$ is a vector in the n^2-dimensional vector space, and $\cdot$ refers to the usual notion of multiplication of a matrix by a vector. We have arrived at an instance of the so-called *matrix power problem*; when the matrices are invertible we have the same reduction to the period-finding routine of Shor's algorithm as one has for the standard discrete logarithm problem, and so we have a solution in quantum polynomial time.

If instead we have a projective linear representation, i.e., an injective homomorphism $G \to \mathbb{P}\mathsf{GL}_n(K)$ the same reduction can be applied to projective matrices in $\mathbb{P}\mathsf{GL}_{n^2}(K)$.

Lemma 5. *Let G be a finite group, and $\psi : G \to \mathsf{GL}_n(K)$ and $\overline{\psi} : G \to \mathbb{P}\mathsf{GL}_n(K)$ a (projective) linear representation. Given an instance of SDLP for G and ϕ, where ϕ is an inner automorphism, i.e., $\phi(g) = mgm^{-1}$ for some $m \in G$, define the linear map $\mathbf{T} : M_n(K) \to M_n(K), M \mapsto \psi(gm)M\psi(m^{-1})$. Then $\mathbf{T}$ descends to a map $\overline{\mathbf{T}} \colon \mathbb{P}M_n(K) \to \mathbb{P}M_n(K)$ and*

$$\mathbf{T} \circ \psi = \psi \circ \rho_{(g,\phi)} \text{ and } \overline{\mathbf{T}} \circ \overline{\psi} = \overline{\psi} \circ \rho_{(g,\phi)} \tag{1}$$

[3] Note that the dimension of the representation implied by Cayley's theorem is rather large. For the groups we are interested in we will have to work harder than this to find lower-dimensional linear representations.

Proof. Since $\mathbf{T}$ is linear, it clearly descends to a map $\overline{\mathbf{T}}$ as described. Let $h \in G$, then by definition

$$(\mathbf{T} \circ \psi)(h) = \psi(gm)\psi(h)\psi(m^{-1}) = \psi(gmhm^{-1}) = \psi(g\phi(h)) = (\psi \circ \rho_{(g,\phi)})(h)$$

The projective case follows immediately. □

We delay the discussion of the case in which the automorphism ϕ is outer. Armed with $\mathbf{T}$, the reduction to the matrix power problem works as follows.

Lemma 6. *Given a finite group G together with an efficiently computable injective (projective) linear representation $\psi : G \to (\mathbb{P})\mathsf{GL}_n(K)$, if ϕ is an inner automorphism, then we can reduce any* SDLP *instance to an instance of the matrix power problem in time polynomial in n.*

Proof. First suppose ψ is a linear representation. Given $h \in G$, we want to find $x \in \mathbb{N}$ such that $\rho^x_{(g,\phi)}(1_G) = h$. By Lemma 5, if ψ is faithful, this is equivalent to finding $x \in \mathbb{N}$ such that $T^x(\mathbf{a}) = \mathbf{b}$ where $\mathbf{a} = 1_{n\times n}$ (the $n \times n$ identity matrix) and $\mathbf{b} = \psi(h)$.

Let $W := \operatorname{span}_K(T^i(\mathbf{a}) \mid i \geq 0)$, which is a K-linear subspace of $M_n(K)$. We define the K-linear map

$$\mathbf{S} \colon W \to W,\ \mathbf{v} \mapsto \mathbf{T}^x\mathbf{v}$$

Note that even though we do not know x, we can compute $\mathbf{S}$ on W in polynomial time since we know $S(\mathbf{a}) = \mathbf{b}$. Note that $(\mathbf{T}|_W)^x = \mathbf{S}$, and since both $\mathbf{S}$ and $\mathbf{T}|_W$ are known we can find x by solving the matrix power problem in $\mathsf{GL}_{n^2}(K)$ (noting that $\mathbf{S}$ and $\mathbf{T}|_W$ can be regarded as elements in $\mathsf{GL}_{n^2}(K)$ after a choice of basis).

In the case that ψ is an injective projective representation, the result follows similarly, reducing SDLP to the matrix power problem in $\mathbb{P}\mathsf{GL}_{n^2}(K)$. □

Recall also that we did not have a method of computing the crucial map $\mathbf{T}$, should the automorphism in question not be inner. However, by [25, Proposition 2], we do have the option of taking the smallest power of the automorphism that is inner, say y, and instead solving at most y instances of SDLP for G and ϕ^y. It turns out, due to a result of Kohl [33, Theorem 1] that for simple groups one can expect this power to be small.

Theorem 7 (Kohl). *If G is a non-abelian finite simple group, then*

$$|\operatorname{Out}(G)| < \log_2 |G|.$$

Since $\operatorname{Out}(G) \cong \operatorname{Aut}(G)/\operatorname{Inn}(G)$ it follows that for any outer automorphism ϕ of a non-abelian finite simple group G there exists an integer x such that $\phi^x \in \operatorname{Inn}(G)$; and crucially that this x is no larger than $\log_2 |G|$. We conclude the following.

Corollary 2. *Let G be a non-abelian finite simple group, and suppose we have an efficiently computable non-trivial (projective) linear representation $\psi : G \to (\mathbb{P})\mathsf{GL}_n(K)$. Then we can solve* SDLP *in G, for any $\phi \in \mathrm{Aut}(G)$, on a quantum computer in probabilistic polynomial time in $\log |G|$.*

Remark 2. Note that we did not have to insist in the above that the linear representation was faithful. In fact, any non-trivial representation of a simple group is faithful, since if the map were not injective it would have non-trivial kernel and therefore imply a proper normal subgroup of a simple group.

5 SDLP in Simple Groups

Now that we have an efficient reduction of the general case of SDLP to SDLP in simple groups, and a method of solving SDLP in simple groups whose complexity is a function of the faithful dimension in simple groups, let us review the known results in this area.

The classification of finite simple groups [48] says any finite simple group is isomorphic to one of the following:

1. A **cyclic group** of prime order p;
2. A group of even permutations of a finite set of cardinality $n \geq 5$, also called **alternating group** Alt_n;
3. A **classical group** of Lie Type:

Linear:	$A_{n-1}(q) \cong \mathbb{P}\mathsf{SL}_n(q), n \geqslant 2$, except $\mathbb{P}\mathsf{SL}_2(2)$ and $\mathbb{P}\mathsf{SL}_2(3)$;
Unitary:	${}^2A_{n-1}(q)\mathbb{P}\mathsf{SU}_n(q), n \geqslant 3$, except $\mathbb{P}\mathsf{SU}_3(2)$;
Symplectic:	$C_n(q) \cong \mathbb{P}\mathsf{Sp}_{2n}(q), n \geqslant 2$, except $\mathbb{P}\mathsf{Sp}_4(2)$;
Orthogonal:	$B_n(q) \cong \mathbb{P}\Omega_{2n+1}(q), n \geqslant 3, q$ odd;
	$D_n(q) \cong \mathbb{P}\Omega^+_{2n}(q), n \geqslant 4$;
	${}^2D_n(q) \cong \mathbb{P}\Omega^-_{2n}(q), n \geqslant 4$

   where q is a power p^a of a prime p;
4. An **exceptional group** of Lie type:

   $$G_2(q), q \geqslant 3; F_4(q); E_6(q); {}^2E_6(q); {}^3D_4(q); E_7(q); E_8(q)$$

   where q is a prime power, or

   $${}^2B_2\left(2^{2n+1}\right), n \geqslant 1; {}^2G_2\left(3^{2n+1}\right), n \geqslant 1; {}^2F_4\left(2^{2n+1}\right), n \geqslant 1$$

   or the Tits group ${}^2F_4(2)'$
5. One of 26 **sporadic simple groups**.

For cyclic groups, SDLP is known to be equivalent to classical DLP, so we need to focus on the other families of groups. Our main tool for the infinite families is to show the existence of a linear representation to use Corollary 2, while for the sporadic groups (and the Tits group) we have a separate discussion in Section 5.2.

5.1 Infinite Families

For alternating groups and groups of Lie type, we show that they have a known efficient linear representation. Thus, if we have them in their "natural representation" (the explicit representation used in their textbook definitions), by Corollary 2 there is a quantum polynomial-time algorithm to solve SDLP.

However, it is possible that, even if we know the isomorphism class of a simple group, an isomorphism to the natural representation of the simple group may still be unknown or hard to compute. A classical example of this is elliptic curves of prime order, which are known to be cyclic groups but require difficult discrete logarithm computations to actually map points to modular integers in a homomorphic way.

This is known in the group theory literature as the **Constructive Recognition Problem** [1, Section 9.2]; hence, for each family, we will discuss how to go from a simple black-box group G to an efficient linear representation. By *efficient* we mean that the complexity is polynomial in the string length of the black-box group elements and in the logarithm of the target group cardinality.

Alternating Groups. Alternating groups are the group of even permutations of a finite set of cardinality n. Since these are permutations, they act on any n-dimensional vector space by permuting the entries, and thus can be represented in $\mathsf{GL}_n(K)$. Additionally, thanks to [27, Theorem 1], there is a probabilistic algorithm in time $O(n \log^2(n) N)$ to compute an isomorphism from any black-box group to the permutation representation of Alt_n, where N is the string length of the black-box group and a maximal n is provided. As a consequence of Corollary 2, we have the following result.

Lemma 7. *If G is a simple black-box group isomorphic to any alternating group Alt_n, for some known maximal n, we can solve* SDLP *for G in probabilistic polynomial time in $n \log |G|$ on a quantum computer.*

Groups of Lie type. Following [22, Section 2], if S is a finite simple group of Lie type, then there exists an algebraic group $\mathbf{H} \leq \mathrm{GL}_n(\overline{\mathbb{F}})$ over an algebraically closed field $\overline{\mathbb{F}}$ and a Steinberg endomorphism σ of $\mathbf{H}$ such that $S \cong C_{\mathbf{H}}(\sigma)/Z(C_{\mathbf{H}}(\sigma))$. Note that there are 8 small cases where this group is not simple, however the only new non-abelian simple group which arises from these cases is the Tit's group ${}^2F_4(2)$ (see [22, Definition 2.2.8 and Theorem 2.2.10]), which will be considered alongside the sporadic simple groups. Given a family of simple groups of Lie type ${}^{\epsilon}\Gamma_m(q)$ for any suitable ϵ and prime power q, the dimension n of the underlying algebraic group $\mathbf{H} \leq \mathsf{GL}_n(\overline{\mathbb{F}}_q)$ is determined by Γ_m:

Γ_m	A_m	B_m	C_m	D_m	G_2	F_4	E_6	E_7	E_8
n	$m+1$	$2m+1$	$2m$	$2m$	14	52	78	133	248

Thus they are naturally described as subgroups of $\mathbb{P}\mathsf{GL}_n(\mathbb{F}_q)$ (or $\mathsf{GL}_n(\mathbb{F}_q)$ if the centre is trivial). This means that we can solve SDLP for such groups using a quantum computer as a consequence of Corollary 2.

Sadly, in contrast to the case of alternating groups, there is no plain polynomial-time algorithm to solve the constructive recognition problem, even if extensive literature has been written on it.

A series of works of Brooksbank and Kantor have proven that for all the families of classical groups (linear [15], unitary [13], symplectic [12] and orthogonal [14]), summarized in [21], we can efficiently compute isomorphisms to the natural representations of the groups under the availability of:

1. So called *number theory oracles*, computing discrete logarithms and factoring in polynomial time;
2. An oracle that, for any input black-box group G isomorphic either to $\mathsf{SL}(2,q)$ or $\mathbb{P}\mathsf{SL}(2,q)$, produces in time polynomial in $\log(q)$ an effective isomorphism $\mathsf{SL}(2,q) \to G$.

Similarly, in [29, 30] the authors show how to compute, in polynomial time, isomorphisms for groups of exceptional Lie type, with the exception of large Ree groups ${}^2F_4\left(2^{2n+1}\right)$ and even characteristic Steinberg triality groups of type ${}^3D_4(2^e)$, assuming the availability of number theory oracles and $\mathsf{SL}(2,q)$ oracles as for classical types.

Since, thanks to Shor's algorithm [46], we know that quantum computers can implement efficient *number theory oracles*, we can combine the previous results in the following lemma.

Lemma 8. *On a quantum computer, if G is a simple black-box group isomorphic to any group of Lie Type of characteristic q and dimension n, with the exception of ${}^2F_4\left(2^{2n+1}\right)$ and ${}^3D_4(2^e)$, we can reduce SDLP for G in probabilistic polynomial time in n and $\log(q)$ to the constructive recognition problem for the group $\mathsf{SL}(2,q)$.*

Constructive Recognition of $\mathsf{SL}(2,q)$ Given its relevance for the general formulation of the problem, several works have studied $\mathsf{SL}(2,q)$. For instance, the authors in [19] show how to compute an efficient isomorphism when the black-box group is a subgroup of the general linear group $\mathsf{GL}_d(q^i)$, given discrete logarithm oracles.

In [2, Lemma 2.10], the authors are able to generalize the result even further, for the much wider class of black-box groups of quotients of matrix groups by recognizable normal subgroups, showing that $\mathsf{SL}(2,q)$ can be constructively recognized in polynomial time having access to number theory oracles.

For general black-box groups, the problem has been solved in [31] for even characteristic and in [9] for the case of small characteristic $p \equiv 1 \mod 4$. For

a general field, the research is partially open: actually, in the preprint [10], the authors show how to compute an isomorphism in polynomial time between the black-box group and $\mathsf{SL}_2(\mathsf{K})$, where K is black-box field isomorphic to $\mathbf{F}_q$, this last isomorphism can be clearly computed via the solution of discrete logarithms over K. Although these last results would suffice to solve the problem, we await further review of these results among the community before drawing this conclusion definitively.

5.2 Sporadic Groups

There are 26 finite simple groups that do not fall into one of the infinite families and the Tits Group ${}^2F_4(2)'$. By the definition of $\rho_{(g,\phi)}$, it suffices to find $x \leq \max_{g\in G}(\mathrm{ord}(g))\cdot\max_{\phi\in\mathrm{Aut}(G)}(\mathrm{ord}(\phi))$. The ATLAS of finite groups [20] provides a complete list of element orders for sporadic groups and their automophism group. In particular, it follows that $\max_{g\in G}(\mathrm{ord}(g)), \max_{g\in G}(\mathrm{ord}(g)) \leq 119 < 2^7$ for each of these 27 groups.

Lemma 9. *For any sporadic finite simple group G and automorphism $\phi \in \mathrm{Aut}(G)$, there is a brute force algorithm to solve* SDLP *for G, ϕ with at most 2^{14} multiplications in the holomorph of G.*

Adapting Shanks' Baby-Step Giant-Step algorithm Adjusting Shanks' Baby-Step Giant-Step (BSGS) algorithm [45] to our setting is a reasonably simple task. Knowing a modest-size upper bound N for the possible values of x, this can be a practical way to find x. Algorithm 4 shows the SDLP variant of the BSGS algorithm, and it is easy to verify that the algorithm stores $O(\sqrt{N})$ elements in the holomorph $G \rtimes \mathrm{Aut}(G)$ and recovers the secret exponent x in $O(\sqrt{N})$ operations in $G \rtimes \mathrm{Aut}(G)$.

Algorithm 4 Baby-step giant-step algorithm in $G \rtimes \mathrm{Aut}(G)$.

Input: $(g, \phi) \in G \rtimes \mathrm{Aut}(G)$, $h = (g, \phi)^x$, $N \in \mathbb{N}$ with $x \leq N$;
Output: the solution of x of the input SDLP instance.

$n \leftarrow \left\lceil \sqrt{N} \right\rceil$
$(s, t) \leftarrow ((g, \phi)^n, (1, id))$
$T \leftarrow [(0, t)]$ ▷ Initialize table
for $(j \leftarrow 1; j \leq n; j++)$
    $t \leftarrow t \cdot s$ ▷ Giant step
    Store (t, j) in T.
end for
$(y, i) \leftarrow (h, 0)$.
while $(y, \_)$ is not in T **do**
    $(y, i) \leftarrow (y \cdot (g, \phi)^{-1}, i + 1)$ ▷ Baby step
end while
return $jn - i$ where (y, j) is in T.

We illustrate the algorithm with SDLP over $\mathbb{M}$.

Example 1. We implemented our BSGS algorithm in approximately 30 lines of Python using the `mmgroup` Python library [44], which offers an efficient implementation of $\mathbb{M}$. In all of our experiments, the running time did not exceed 5 seconds on a 2022 Macbook Air with 16 GB of RAM.

5.3 Determining the isomorphism type of a black box simple group

Note that for the alternating group, the recognition algorithm requires as input a maximum n such that G could be isomorprhic to Alt_n, while the recognition algorithms for groups of Lie type require the isomorphism type of the black box group. Therefore an important step to apply Lemma 7 and Lemma 8 requires finding out which recognition algorithm needs to be implemented on a given black box simple group. It turns out that nearly all simple groups (and characteristically simple) are characterised by their order.

Theorem 8. *[32, Theorem 6.1] Let S and T be non-isomorphic finite simple groups. If $|S^a| = |T^b|$ for some natural numbers a and b, then $a = b$ and S and T either are $A_2(4)$ and $A_3(2)$ or are $B_n(q)$ and $C_n(q)$ for some $n \geq 3$ and some odd prime power q.*

Once a black box simple groups order $|S|$ is known, it can be tested to which simple group does it coincide and then run the corresponding recognition algorithm, while if there is a collision it is only between two groups and thus both corresponding recognition algorithms could be run. For sporadic simple groups (and the Tits group), this is a direct test against 27 fixed values and for alternating groups this requires finding n such that $|S| = n!$. For finite groups of Lie type their orders are of the form $\frac{q^N}{m}\prod_{i=1}^{n}(q^{d_i} - \epsilon_i)$ and thus it suffices to find the valid values for q, N, n, m, d_i and ϵ_i. In particular, determining the simple groups with order equal to that of a given black box group is polynomial in $\log|S|$.

6 Conclusion

We conclude by giving a comprehensive overview of our results, and discussing the consequences for SDLP. We have also summarized the flow of our argument visually in Figure 1; one can take this diagram as a map of the paper.

Consider a finite, black-box group G. Then, in quantum polynomial time (in $\log|G|$), we can reduce any SDLP in G instance to at most $\log|G|$ instances of SDLP in a simple group by using Section 3. As a corollary of the Classification of Finite Simple Groups, once the isomorphism type is known we can efficiently study each possible instance separately, employing two main attack tools: for infinite families, the results from Section 4; and for sporadic groups, an adapted version of the *Baby-Step Giant-Step algorithm* (Algorithm 4).

We see that, if the groups are given in their natural representations we can find linear representations and apply Corollary 2 to produce a solution to SDLP in the corresponding simple group S in quantum polynomial time in $\log |S|$, so SDLP on simple groups is no harder than the problem of computing an efficient linear representation starting from a black-box group. Even if not conclusive, the extensive group theory literature on the solution of the constructive recognition problem in probabilistic quantum polynomial time is enough evidence to conclude that SDLP on finite groups is not a reliable candidate for the construction of quantum resistant primitives.

We highlight that, from Figure 1, we could get also constructive quantum probabilistic polynomial-time algorithms for solving SDLP in a finite, black-box group G if we solve these last open questions:

1. Provide constructive recognition algorithms for large Ree groups ${}^2F_4\left(2^{2n+1}\right)$ and even characteristic Steinberg triality groups of type ${}^3D_4(2^e)$;
2. Have a clean, peer-reviewed discussion of the Constructive Recognition problem for $\mathsf{SL}(2, q)$ on quantum computers.
3. Resolve the gaps in the literature on the computation of maximal normal subgroups (discussed in the appendix).

We close with some high-level remarks. It is perhaps not too surprising, that an arbitrary instance of SDLP reduces to SDLP instances in finite simple groups. However, the fact that *all* of these finite simple groups admit efficient methods of solving SDLP relies on the property that simple groups have low dimension and very small outer-automorphism groups. Recalling that the method of decomposition into finite simple groups could only fail when no characteristic subgroups were present, it is also rather unfortunate that this scenario coincides with the group being a direct product of simple groups, from which a different method of reduction is possible. The insecurity of SDLP in finite groups, in other words, does not appear to result from some error in cryptographic design, but instead from fundamental properties of the finite groups themselves.

Acknowledgments. This collaboration was initiated during the "Post-Quantum Group-Based Cryptography" workshop at the American Institute of Mathematics (AIM), April 29-May 3, 2024. The authors are indebted to the workshop organizers Delaram Kahrobaei and Ludovic Perret and the AIM team for bringing this group together and creating a stimulating and collaborative atmosphere.

We want to thank Ray Perlner for spotting problems in the reasoning of an earlier version of this paper, and bringing those to our attention. We would also like to Gábor Ivanyos, with whom we had helpful correspondence. We also would like to acknowledge support by the following organizations: CB is supported by ONR Grant 62909-24-1-2002. GB is supported by SNSF Consolidator Grant CryptonIs 213766. DCST was partially supported by a grant from the Simons Foundation (712530, DCST). DJ is supported by an NSERC Alliance Consortia

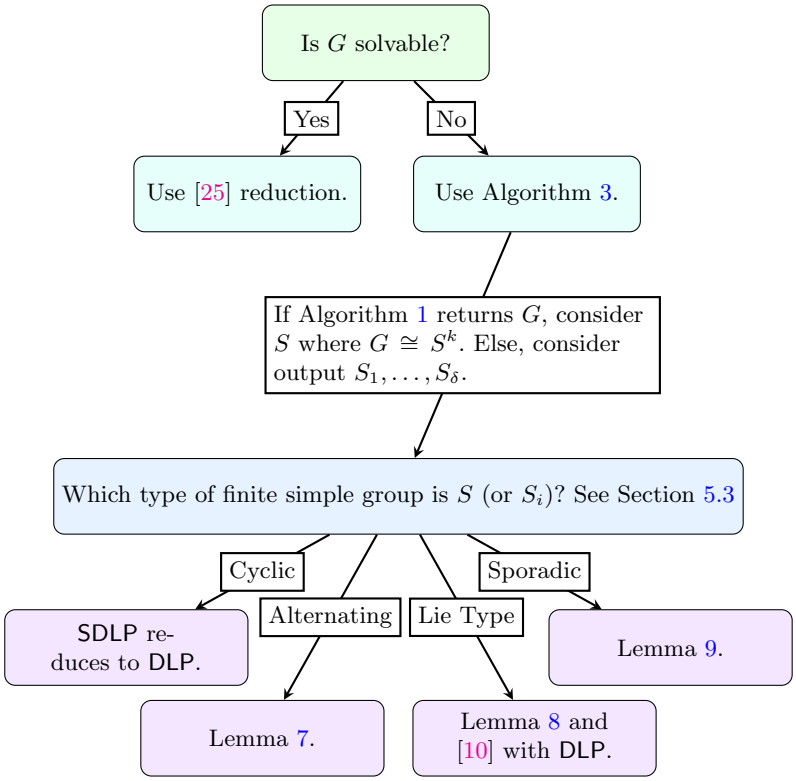

Fig. 1. Visual summary of a possible roadmap for a general SDLP instance over a finite group.

Quantum Grant (ALLRP 578463 – 22). LM is supported by an NSERC Canada Graduate Scholarship (Master's). NH is supported by a gift from Google. RS is supported by NATO SPS project G5985. EP is supported by NCAE grant H98230-22-1-0328.

References

[1] László Babai and Robert Beals. "A polynomial-time theory of black box groups I". In: *London Mathematical Society Lecture Note Series* (1999), pp. 30–64.

[2] László Babai, Robert Beals, and Ákos Seress. "Polynomial-time theory of matrix groups". In: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. STOC '09. Bethesda, MD, USA: Association for Computing Machinery, 2009, pp. 55–64.

[3] László Babai, Gene Cooperman, Larry Finkelstein, Eugene Luks, and Ákos Seress. "Fast Monte Carlo algorithms for permutation groups". In: *Proceedings of the twenty-third annual ACM symposium on Theory of computing*. 1991, pp. 90–100.

[4] László Babai and Endre Szemerédi. "On the complexity of matrix group problems I". In: *25th Annual Symposium onFoundations of Computer Science, 1984*. IEEE. 1984, pp. 229–240.

[5] Reinhold Baer. "Der reduzierte Rang einer Gruppe". In: *Journal für die reine und angewandte Mathematik* 0214_0215 (1964), pp. 146–173. URL: http://eudml.org/doc/150612.

[6] Christopher Battarbee, Delaram Kahrobaei, Ludovic Perret, and Siamak F. Shahandashti. *A Subexponential Quantum Algorithm for the Semidirect Discrete Logarithm Problem*. 2023. arXiv: 2209.02814 [cs.CR].

[7] Christopher Battarbee, Delaram Kahrobaei, Ludovic Perret, and Siamak F. Shahandashti. "SPDH-Sign: Towards Efficient, Post-quantum Group-Based Signatures". In: *Post-Quantum Cryptography*. Ed. by Thomas Johansson and Daniel Smith-Tone. Cham: Springer Nature Switzerland, 2023, pp. 113–138.

[8] Christopher Battarbee, Delaram Kahrobaei, and Siamak F Shahandashti. "Semidirect product key exchange: The state of play". In: *Journal of Algebra and Its Applications* (2023), p. 2550066.

[9] Alexandre Borovik and Sukru Yalcinkaya. *Steinberg presentations of black box classical groups in small characteristics*. 2013. arXiv: 1302.3059 [math.GR].

[10] Alexandre Borovik and Şükrü Yalçınkaya. *Natural representations of black box groups encrypting $SL_2(\mathbb{F}_q)$*. 2020. arXiv: 2001.10292 [math.GR].

[11] Alexander Bors. *A bound on element orders in the holomorph of a finite group*. 2015. arXiv: 1510.02014 [math.GR].

[12] Peter A. Brooksbank. "Fast constructive recognition of black box symplectic groups". In: *Journal of Algebra* 320.2 (2008). Computational Algebra, pp. 885–909. ISSN: 0021-8693.

[13] Peter A. Brooksbank. "Fast constructive recognition of black-box unitary groups". In: *LMS Journal of Computation and Mathematics* 6 (2003), pp. 162–197.

[14] Peter A. Brooksbank and William M. Kantor. "Fast constructive recognition of black box orthogonal groups". In: *Journal of Algebra* 300.1 (2006), pp. 256–288.

[15] Peter A. Brooksbank and William M. Kantor. "On constructive recognition of a black box PSL (d, q)". In: *Groups and computation* 3 (1999), pp. 95–111.

[16] Daniel Brown, Neal Koblitz, and Jason Legrow. "Cryptanalysis of 'MAKE'". In: *J. Math. Cryptol.* 16.1 (2022), pp. 98–102.

[17] Andrew M. Childs and Gábor Ivanyos. "Quantum computation of discrete logarithms in semigroups". In: *J. Math. Cryptol.* 8.4 (2014), pp. 405–416.

[18] Marston Conder and Charles R. Leedham-Green. "Fast recognition of classical groups over large fields". In: *Groups and computation, III (Columbus, OH, 1999)* 8 (2001), pp. 113–121.

[19] Marston Conder, Charles R. Leedham-Green, and Eamonn O'Brien. "Constructive recognition of $\mathsf{PSL}(2, q)$". In: *Trans. Amer. Math. Soc.* 358.3 (2006), pp. 1203–1221.

[20] John H. Conway, Robert T. Curtis, Simon P. Norton, Richard A. Parker, and Robert A. Wilson. *Atlas of finite groups*. Oxford University Press, Eynsham, 1985.

[21] Heiko Dietrich, Charles R. Leedham-Green, and Eamonn A. O'Brien. "Effective black-box constructive recognition of classical groups". In: *Journal of Algebra* 421 (2015), pp. 460–492.

[22] Daniel Gorenstein, Richard. Lyons, and Ron Solomon. *The classification of finite simple groups. Number 3. Part I*. American Mathematical Society, Providence, RI, 1998.

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

[24] 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.

[25] Muhammad Imran and Gábor Ivanyos. "Efficient quantum algorithms for some instances of the semidirect discrete logarithm problem". In: *Designs, Codes and Cryptography* (May 2024).

[26] Gábor Ivanyos, Frédéric Magniez, and Miklos Santha. "Efficient quantum algorithms for some instances of the non-abelian hidden subgroup problem". In: *Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures* (2001), pp. 263–270.

[27] Sebastian Jambor, Martin Leuner, Alice C Niemeyer, and Wilhelm Plesken. "Fast recognition of alternating groups of unknown degree". In: *Journal of Algebra* 392 (2013), pp. 315–335.

[28] Delaram Kahrobaei and Vladimir Shpilrain. "Using semidirect product of (semi) groups in public key cryptography". In: *Pursuit of the Universal*. Ed. by Arnold Beckmann, Laurent Bienvenu, and Nataša Jonoska. Cham: Springer International Publishing, 2016, pp. 132–141.

[29] W. M. Kantor and K. Magaard. "Black box exceptional groups of Lie type". In: *Trans. Amer. Math. Soc.* 365.9 (2013), pp. 4895–4931.

[30] W. M. Kantor and K. Magaard. "Black box exceptional groups of Lie type II". In: *Journal of Algebra* 421 (2015), pp. 524–540.

[31] William M. Kantor and Martin Kassabov. "Black box groups isomorphic to PGL (2, 2e)". In: *Journal of Algebra* 421 (2015), pp. 16–26.

[32] Wolfgang Kimmerle, Richard Lyons, Robert Sandling, and David N. Teague. "Composition factors from the group ring and Artin's theorem on orders of simple groups". In: *Proceedings of the London Mathematical Society* 3.1 (1990), pp. 89–122.

[33] Stefan Kohl. *A bound on the order of the outer automorphism group of a finite simple group of given order*. Available at https://stefan-kohl.github.io/preprints/outbound.pdf. 2003.

[34] Charles R. Leedham-Green. "The computational matrix group project". In: *Groups and computation* 3 (2001), pp. 229–248.

[35] Andrew Mendelsohn, Edmund Dable-Heath, and Cong Ling. *A Small Serving of Mash: (Quantum) Algorithms for SPDH-Sign with Small Parameters*. Cryptology ePrint Archive, Paper 2023/1963. 2023. URL: https://eprint.iacr.org/2023/1963.

[36] Chris Monico. *Remarks on MOBS and cryptosystems using semidirect products*. 2021. arXiv: 2109.11426 [cs.CR].

[37] Chris Monico and Ayan Mahalanobis. *A remark on MAKE – a Matrix Action Key Exchange*. 2020. arXiv: 2012.00283 [cs.CR].

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

[39] NIST. *Post-Quantum Cryptography Standardization*. URL: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography. 2017.

[40] Eamonn A O'Brien. "Algorithms for matrix groups". In: *London Math. Soc. Lecture Note Ser* 388 (2011), pp. 297–323.

[41] Nael Rahman and Vladimir Shpilrain. "MAKE: A matrix action key exchange". In: *J. Math. Cryptol.* 16.1 (2022), pp. 64–72.

[42] Nael Rahman and Vladimir Shpilrain. *MOBS (Matrices Over Bit Strings) public key exchange*. Cryptology ePrint Archive, Paper 2021 /560. 2021. URL: https://eprint.iacr.org/2021/560.

[43] Vitaliĭ Roman'kov. *Linear decomposition attack on public key exchange protocols using semidirect products of (semi) groups*. 2015. arXiv: 1501.01152 [cs.CR].

[44] Martin Seysen. *Python implementation of the monster group*. GitHub repository. 2024. URL: https://github.com/Martin-Seysen/mmgroup.

[45] Daniel Shanks. "Class number, a theory of factorization, and genera". In: *Proceedings of Symposia in Pure Mathematics*. 1971.

[46] Peter W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134.
[47] Victor Shoup. "Lower Bounds for Discrete Logarithms and Related Problems". In: *Advances in Cryptology — EUROCRYPT '97*. Ed. by Walter Fumy. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 256–266.
[48] Robert A. Wilson. *The Finite Simple Groups*. Vol. 251. Graduate Texts in Mathematics. Springer, 2009.

A Appendix: Finding Maximal Normal Subgroups

The task of finding a maximal normal subgroup depends on the particular implementation of the black-box group G. In general, if we know the particular structure of the group G, we may be able to recover them immediately from it. This can be done even with little knowledge, since from any subgroup S we can construct the smallest normal subgroup containing it via computing the normal closure $\langle S^G \rangle$ in linear time as explained in [3].

In the literature, several techniques are known to solve this task more systematically, via computing a composition series, in this way the first element in the series (starting from G) is our desired normal subgroup. However, this branch of literature typically wishes to achieve much stronger results, in particular without using quantum computers - we do not impose this limitation upon ourselves. To perform this calculation, aided by a quantum computer, we can:

- Use [25] if every non-Abelian composition factor of G possesses a faithful permutation representation of degree polynomial in the input size;
- Otherwise, [1, Theorem 1.1] gives us a quasi-composition series for G. Note that [1] requires a superset of the primes dividing the order of the group $|G|$ to solve the problem of computing order of group elements, with a quantum computer we can solve both these tasks. This result provides a quasi-composition chain $\{1\} \triangleleft G_{m-1} \triangleleft \cdots \triangleleft G_1 \triangleleft G$, and tells us if G/G_1 is abelian, or simple and nonabelian. In the latter case, we have found a maximal normal subgroup $N = G_1$. In the former case, if $A = G/G_1$ has the unique encoding property, we can use [26, Theorem 6] on it, since abelian groups are solvable, i.e. $\nu(G) = 1$, and the procedure runs in quantum polynomial time. In this way we get the maximal normal subgroup $A_1 \triangleleft A$ from the composition series, and $A_1 G_1$ will be a maximal normal in G by the correspondence theoren. However, the general results from [1], does not immediately imply the unique-encoding property requested, so additional work may be required to solve this problem for the general case, even if in more concrete cases this may be practical.

In general, we do not expect that these problems should be of some fundamental computational difficulty. We leave the full resolution of the computation of maximal normal subgroups to further work.