On the Semidirect Discrete Logarithm Problem in Finite Groups

Christopher Battarbee, Giacomo Borin, Ryann Cartor, Nadia Heninger, David Jao, Delaram Kahrobaei, Laura Maddison, Edoardo Persichetti, Angela Robinson, Daniel Smith-Tone, et al.

▶ **To cite this version:**

Christopher Battarbee, Giacomo Borin, Ryann Cartor, Nadia Heninger, David Jao, et al.. On the Semidirect Discrete Logarithm Problem in Finite Groups. 2024. hal-04663959

HAL Id: hal-04663959
https://hal.science/hal-04663959v1

Preprint submitted on 29 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the Semidirect Discrete Logarithm Problem in Finite Groups

Christopher Battarbee[1], Giacomo Borin[2], Ryann Cartor[3], 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 & University of Zurich
[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.

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 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 [Hab+13; KS16; RS22; RS21; GS19] since its introduction in 2013 by Habeeb et al. [Hab+13]. 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 [NIS17] 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.

The corresponding authors of this work, Christopher Batterbee and Giacomo Borin, can be reached at christopher.battarbee@lip6.fr and giacomo.borin@ibm.com, respectively.

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. [Bat+23b; Bat+23a] 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 [Hab+13; KS16; RS22; RS21; GS19]. Each of these proposals was later shown to be insecure due to some feature of the selected platform group [MR15; Rom15; BKL22; MM20; Mon21]. 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 [II24] 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 a generalization of the "decomposition" methods of [II24]. In particular, we will repeatedly use the "recursion tool" of [II24] 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 [II24]. 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 [II24], 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 [Hab+13], 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 [CI14] 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 1.1. *Let G be a finite black-box group. 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 1.2. *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 1.3. *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 1.3. 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 [II24] 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 2.1 (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 [Bor15]), so we may, without loss of generality, choose to restrict the domain of $s_{g,\phi}$ to a finite set.

Definition 2.2 (Semidirect Discrete Logarithm Problem). *Let G be a group and fix $(g,\phi) \in \mathrm{Hol}(G)$. Given an image $h := s_{g,\phi}(x)$, the* Semidirect Discrete Logarithm Problem *(*SDLP*) is to recover an x' such that $s_{g,\phi}(x') = h$. Given the group G and automorphism ϕ, we denote this problem by* $\mathsf{SDLP}(G,\phi)$*.*

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.

For technical reasons we require that any computational representation of a group G comes with an attribute CSFlag, which is by default set to 0 (i.e., $G.\mathsf{CSFlag} = 0$). One of our algorithms later on may update this value if it detects that the group is characteristically simple.

A *linear representation* [Ser77] of a group G on a finite-dimensional vector space V is simply 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., injective homomorphisms $G \to \mathbb{P}\mathsf{GL}(V)$, where $\mathbb{P}\mathsf{GL}(V)$ contains the invertible linear maps acting on $\mathbb{P}(V)$. It is immediate to identify these maps as $\mathsf{GL}(V)/Z(V)$, where $Z(V)$ is the (normal) subgroup of scalar matrices in $\mathsf{GL}(V)$, since scalar multiplication acts as the identity 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 [BS84] as a useful abstraction of computations in groups.

Definition 2.3 (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).*

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 [Sho97]. 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 [IMS01], or in relation to "The computational matrix group project" [Lee01; OBr11].

Of particular relevance is the **Constructive Recognition Problem**, proposed by Babai and Beals [BB99, 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 [Bro08; Bro03; BBS09; Jam+13; KM13; KM15; BB99] 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 [CL01; BBS09]. These algorithms thus run in quantum polynomial time [Sho94].

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 [Hab+13; KS16; RS22; RS21; GS19], and works cryptanalyzing the results [MR15; Rom15; BKL22; MM20; Mon21]. 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 [BKS23].

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. [Bat+23a] pointed out a connection to group actions and later exploited it [Bat+23b] to give a subexponential quantum algorithm for SDLP.

Mendelsohn et al. [MDL23] found faster methods for some small parameters. Most recently, Imran and Ivanyos [II24] 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 compute simple groups such that solving appropriate instances of SDLP in these simple groups suffices to derive a solution for the group we started in. 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 [II24].

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 [II24] 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 2.4 (SDLP(G, ϕ)). *For each g, h, determine an integer x for which*

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

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

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 [Bat+23a; Bat+23b]. 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 2.5. *Consider* $\mathsf{SDLP}(G, \phi)$ *for* $g, h \in G$*. There exists an integer* n_0 *(dependent on* g *and* ϕ*) such that* $\rho_{g,\phi}^{n_0}(1_G) = s_{g,\phi}(n_0) = 1_G$*, and the set*

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

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 algorithm, and that the solutions of* $\mathsf{SDLP}(G, \phi)$ *for* g *and* h *are 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 [II24] are given in detail in the main body of the present paper, we will just quote the fact given as [II24, 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 [II24] 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 [II24] 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 [II24], by using their method to completely reduce an arbitrary instance of SDLP to several instances of SDLP in a simple group. In particular, Theorem 3.6 demonstrates that, in order to solve some instance of $\mathsf{SDLP}(G, \phi)$, 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 [II24, 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 [II24, Theorem 3]. The main idea of the recursion tool is to notice that if we can find a normal subgroup N of G that is invariant under our automorphism, any solution of $\mathsf{SDLP}(G, \phi)$ must also be a solution of $\mathsf{SDLP}(G/N, \bar{\phi})$ for some automorphism $\bar{\phi}$. From this, we can infer certain information about the form of the solutions of $\mathsf{SDLP}(G, \phi)$. The remaining information required to give a complete description of these solutions can be obtained by solving SDLP in the quotient.

We will state and prove the result in full, in order to review ideas from its proof that are important in our reduction algorithms.

Theorem 3.1 (Recursion tool, [II24]). *For a finite group* G*, consider an automorphism* ϕ *of* G *and suppose we have a* ϕ*-invariant normal subgroup* N *of* G*. In order to solve* $\mathsf{SDLP}(G, \phi)$*, it is sufficient to solve* $\mathsf{SDLP}(G/N, \bar{\phi})$ *and* $\mathsf{SDLP}(N, \phi^{n_0})$*, for suitable choices of* G/N*, automorphism* $\bar{\phi}$*, and integer* n_0*.*

Proof. Let ψ be the quotient map from G to G/N. By the first isomorphism theorem, we know $\mathrm{Im}(\psi) \cong G/N$ (and as such we may write these two groups interchangeably). Suppose we know a map $\bar{\phi}$ satisfying $\psi \circ \phi = \bar{\phi} \circ \psi$. An easy induction shows that one must also have $\psi \circ \phi^i = \bar{\phi}^i \circ \psi$. It follows that every solution of $\mathsf{SDLP}(G, \phi)$ for g, h must also be a solution of $\mathsf{SDLP}(G/N, \bar{\phi})$ for $\psi(g), \psi(h)$, for if one has

$$g\phi(g) \dots \phi^{x-1}(g) = h$$

for some integer x, then

$$\begin{aligned}\psi(h) &= \psi(g\phi(g) \dots \phi^{t-1}(g)) \\ &= \psi(g)\psi(\phi(g)) \dots \psi(\phi_{t-1}(g)) \\ &= \psi(g)\bar{\phi}(\psi(g)) \dots \bar{\phi}^{t-1}(\psi(g)).\end{aligned}$$

Now, we know that the solutions of $\mathsf{SDLP}(G/N, \bar{\phi})$ for $\psi(g), \psi(h)$ form the set $\{t_0 + tn_0 : t \in \mathbb{Z}\}$, for some $0 \leq t_0 < n_0$, where

$$n_0 = |\{\rho^i_{(\psi(g),\bar{\phi})}(1_{G/N}) : i \in \mathbb{Z}\}|.$$

In other words, every solution of $\mathsf{SDLP}(G, \phi)$ for g, h is of the form $t_0 + tn_0$ for some $t \in \mathbb{Z}$. However, we cannot conclude that *every* $t \in \mathbb{Z}$ gives rise to a solution of $\mathsf{SDLP}(G, \phi)$ for g, h.

We claim that, in order to find the integers that do yield a solution of $\mathsf{SDLP}(G, \phi)$ for g, h, it suffices to solve $\mathsf{SDLP}(N, \phi^{n_0})$ for suitably chosen values of $g', h' \in N$.

To prove this claim, let us first verify that $\rho^{n_0}_{(g,\phi)}(N) \subset N$; that is, for every $m \in N$, $\rho^{n_0}_{(g,\phi)}(m) \in N$. Again from Theorem 2.5, one has $\rho^{n_0}_{(\psi(g),\bar{\phi})}(1_{G/N}) = 1_{G/N}$, and so

$$\psi(g)\bar{\phi}(\psi(g))...\bar{\phi}^{n_0-1}(\psi(g)) = 1_{G/N}.$$

Following a similar argument to the computation of $\psi(h)$ above, it follows that $\psi(g\phi(g) \cdots \phi^{n_0-1}(g)) = 1_{G/N}$. By definition of the quotient map and $\rho_{(g,\phi)}$ we must therefore have $\rho^{n_0}_{(g,\phi)}(1_G) \in N$. Since for any $m \in N$ it holds that $\rho^{n_0}_{(g,\phi)}(m) = \rho^{n_0}_{(g,\phi)}(1_G)\phi^{n_0}(m)$, and because ϕ is N-invariant, we have that $\rho^{n_0}_{(g,\phi)}(m) \in N$, demonstrating the claim.

In fact, for any $t \in \mathbb{Z}$ we have $\rho^{tn_0}_{(g,\phi)}(N) \subset N$. Given our argument above, this follows by induction: suppose $\rho^{(t-1)n_0}_{(g,\phi)}(N) \subset N$, then clearly

$$\rho^{tn_0}_{(g,\phi)}(N) = \rho^{n_0}_{(g,\phi)}(\rho^{(t-1)n_0}_{(g,\phi)}(N)) \subset N.$$

Consider now an integer t such that $t_0 + tn_0$ is a solution to $\mathsf{SDLP}(G, \phi)$ for g, h. We have

$$\begin{aligned}h' &= \rho^{-t_0}_{(g,\phi)}(h) \\ &= \rho^{-t_0}_{(g,\phi)}(\rho^{t_0+tn_0}_{(g,\phi)}(1_G))) \\ &= \rho^{tn_0}_{(g,\phi)}(1_G).\end{aligned}$$

This equality demonstrates that $h' \in N$. Moreover, it is not too hard to see that $\rho^{tn_0}_{(g,\phi)}(1_G)$ with respect to the semidirect product $G \rtimes_\phi \mathbb{Z}$ is the same[2] as writing $\rho^t_{(g',1)}(1_N)$ with respect to the semidirect product $N \rtimes_{\phi^{n_0}} \mathbb{Z}$, where $g' = \rho^{n_0}_{(g,\phi)}(1_G)$ (with respect to $G \rtimes_\phi \mathbb{Z}$). In other words, every t such that $t_0 + tn_0$ is a solution of $\mathsf{SDLP}(N, \phi^{n_0})$ for the described g', h'. The claim of the theorem follows. □

[2] Note that every subgroup contains the group identity, so $1_N = 1_G$.

We can now use this tool to provide a reduction of the general case of SDLP to the case of SDLP in simple groups. Intuitively, since every finite group is "composed" of simple groups[3], we can imagine taking an instance of SDLP and producing two instances of SDLP in its composition factors. Iterating this process will eventually output several instances of SDLP in a simple group such that solving these instances of SDLP gives a solution to the input problem.

This strategy works provided we can compute the various objects used in the proof of Theorem 3.1. In particular, given $\mathsf{SDLP}(G, \phi)$ for $g, h \in G$ we need to be able to compute: a ϕ-invariant normal subgroup N of G; the quotient G/N, and the evaluation of the quotient map ψ; the induced map $\bar{\phi}$ on the quotient; and the integer n_0. We assume that, given the normal subgroup, computing the quotient and evaluating the quotient map can be done efficiently. Moreover, [II24] describes a general method of evaluating the induced map $\bar{\phi}$, so we consider this matter resolved as well. The computation of the integer n_0 can be done with a Shor-like algorithm, as discussed in Section 2.2.

The main remaining obstacle is the computation of the ϕ-invariant normal subgroup, which is the subject of the next section.

3.2 Computing an Invariant Normal Subgroup.

The purpose of this section is to describe an algorithm that computes the invariant subgroups we need for decomposition. First, we note that, if the group for which we wish to compute the ϕ-invariant normal subgroup has *no* characteristic subgroups, then we will not be able to proceed with the decomposition. However, these types of groups are well-understood: they are called "characteristically simple", and it is well-known (see [Wil09, 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. As we will see later, this classification allows us to proceed with a bespoke algorithm in the case of characteristic simplicity; so, for the time being, let us set this case aside.

We may then proceed as follows. By [IMS01, Theorem 4], it is possible to compute a composition series of an arbitrary black box group G in time quantum polynomial in $\log |G|$. For our purposes, we will take it for granted that it is possible to efficiently compute a maximal normal subgroup of G, as well as the composition factors of G[4]. However, for an arbitrary automorphism ϕ there is no reason to believe that the subgroup yielded by this method of computing a composition series will be ϕ-invariant. Indeed, we are not obviously guaranteed that such a normal subgroup exists. Our method consists of showing that either we can compute a ϕ-invariant normal subgroup, by using a maximal normal subgroup obtained by the method of [IMS01], or G is characteristically simple.

Now, a method of computing ϕ-invariant normal subgroups from an arbitrary maximal normal subgroup N is given in [II24], and works as follows. Take $N_1 = N$, $N_2 = N \cap \phi(N)$, and for $i \geq 3$ 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, say N_j, is non-trivial (certainly the trivial subgroup is ϕ-invariant). The intersection trick, however, will not terminate with the trivial subgroup if the 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. Since we can handle the case where there exist no non-trivial characteristic subgroups, it would suffice to demonstrate that a single characteristic subgroup forces every maximal normal subgroup to contain a non-trivial G-characteristic subgroup. In fact, we are able to provide this alternate classification of the characteristically simple groups, as shown below.

[3] The precise sense in which this is true is unimportant for our purposes, though the interested reader is advised to recall the famous Jordan-Hölder theorem.

[4] The full result requires knowledge of the set of primes dividing the order of the group, which we suppress since this can already be achieved by a quantum computer [Eke21].

Lemma 3.2. *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.

Consider the subgroup $\mathcal{J}(G)$ defined by the intersection of all maximal normal subgroups, known as the *Jacobson radical* of G. By definition, $\mathcal{J}(G)$ is contained in N. It is easy to see that $\mathcal{J}(G)$ is characteristic; therefore, if $\mathcal{J}(G)$ is non-trivial, the result follows. We are thus left with the case in which $\mathcal{J}(G)$ is trivial.

Baer established [Bae64, Remark 4.8] that $\mathcal{J}(G)$ is trivial if and only if G is isomorphic to a direct product of finitely many simple groups and G contains finitely many maximal normal subgroups. We therefore can express G as an internal direct product $S \times A$, where S is itself the internal direct product of non-abelian simple groups and A is abelian.

First let us eliminate the case where G is abelian; that is, S is trivial and $G = A$. G must be the direct product of simple abelian factors; since G is assumed not to be characteristically simple, at least two of these have distinct prime orders, say p and q. Every maximal subgroup is normal and so has prime index, so N contains either the Sylow p-subgroup or the Sylow q-subgroup, both of which are characteristic.

Suppose, then, that S is non-trivial. We have two remaining cases, depending on whether or not A is trivial.

Suppose that A is trivial so that $G = S$. By the classification of characteristically simple groups, we can think of S as the direct product of characteristically simple factors G_i, where G_i does not share any direct factors with G_j for $i \neq j$. Certainly, these characteristically simple factors are normal subgroups of G; in fact, they are characteristic in G. To demonstrate this fact, we may write $G = G_i \times G_i^c$, where G_i has no common direct factors with G_i^c. By [BCM06, Theorem 3.1], the structure of any automorphism ψ of $G = G_i \times G_i^c$ is such that $\psi(g,h) = (r(g)s(h), t(g)u(h))$ with $r \in \mathrm{Aut}(G_i), s \in \mathrm{Hom}(G_i^c, Z(G_i)), t \in \mathrm{Hom}(G_i, Z(G_i^c)), u \in \mathrm{Aut}(G_i^c)$. Since both G_i and G_i^c have trivial centers, we see that every automorphism ψ leaves G_i invariant, and thus G_i is a G-characteristic subgroup.

Since G by assumption is not characteristically simple, there are at least two distinct characteristically simple direct factors, G_i and G_j with $i \neq j$. We now show that N contains at least one of these two characteristically simple direct factors, and thus contains a G-characteristic subgroup. Supposing that G_i is not contained in N, we have that $N \triangleleft G_i N = G$ due to the maximality condition. Since $G_j \leq G_i^c$, we have that $G_j \leq N$. Thus, in the case that A is trivial, the result holds.

Now, suppose A is not trivial; we further split our analysis into two sub-cases. If $A \leq N$, the result holds immediately. In fact, note that the center of a direct product is the direct product of its centers. Since the center of each simple non-abelian component is trivial (the center is a normal subgroup), and the center of each abelian component is the component itself, we have $Z(G) = A$. Any maximal normal subgroup containing A therefore contains the center of G, which is characteristic in G.

In the second sub-case, A is non-trivial and N does not contain A. With this in mind, we note that $N \triangleleft NA = G$ by the maximality condition, and since $G = S \times A$, we have that $S \leq N$. Finally, note that $S = [G, G]$ since S is perfect and A is abelian. Therefore, S is G-characteristic due to the fact that the commutator of G is characteristic. Thus N contains a non-trivial G-characteristic subgroup in the final case as well. □

We can now describe our algorithm computing a ϕ-invariant normal subgroup. Indeed, the equivalence of the conditions set out above gives us the additional ability to *detect* characteristic simplicity, whence we can handle this case separately.

Theorem 3.3. *Let G be a finite black-box group, and suppose ϕ is an automorphism of G. Algorithm 1 either computes a non-trivial ϕ-invariant subgroup of G, or detects that G is characteristically simple (and outputs itself). In either case the algorithm finishes in time quantum polynomial in* $\log |G|$.

Proof. Obtain a maximal normal subgroup N of G using the quantum algorithm of [IMS01, Theorem 4]. 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 identity.

If it does terminate with the identity, then the subgroup N we started with did not contain a characteristic subgroup by the contrapositive. It is therefore not true that every maximal normal subgroup of G contains a non-trivial characteristic subgroup, so by Lemma 3.2 G has no non-trivial characteristic subgroups. The CSFlag is therefore set correctly. □

Algorithm 1 (***Inv***): Computing ϕ-invariant normal subgroups, or detecting characteristically simple groups.

Input: G, ϕ
Output: ϕ-invariant $N \triangleleft G$ or G

$N \leftarrow$ max normal subgroup obtained from [IMS01] algorithm
$N_1 \leftarrow N$
$N_2 \leftarrow \phi(N)$
$j \leftarrow 2$
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
    G.CSFlag $\leftarrow 1$
    return G
end if

Before moving on to the full reduction, let us see how to induce instances of SDLP in a simple group when the input group is characteristically simple.

Lemma 3.4. *Let G be a characteristically simple group. Then any instance of $\mathsf{SDLP}(G, \psi)$ can be solved in polynomial time with polynomially many accesses to an oracle solving $\mathsf{SDLP}(S, \phi)$ for some finite simple group S.*

Proof. The classification of characteristically simple groups is known, see [Wil09, Lemma 2.8]. Specifically, G is characteristically simple if and only if G is the direct product of k isomorphic copies of a finite simple group S.

If G is abelian then $G = \mathbb{Z}_p^k$ for some prime p, and we are done by [II24, Theorem 6]. We may therefore henceforth assume that G is non-abelian (and therefore composed of non-abelian, simple factors).

Suppose that $G \approx S^k$ and let V denote a linear representation of S of minimal representation dimension n. Then G has a linear representation $V^n \approx \bigoplus_{i=1}^k V$ of dimension nk. We also note that $\mathrm{Aut}(G) \approx \mathrm{Aut}(S) \wr S_k$, i.e. the wreath product of $\mathrm{Aut}(S)$ and the symmetric group S_k. Note that k is logarithmic in the size of G.

We may now define a reduction for SDLP on S^k to SDLP on S. First, we have a linear bound in k on both cycle length and the number of disjoint cycles for any element of S_k.

Let $\psi \in \mathrm{Aut}(S) \wr S_k$ and let $\sigma \in S_k$ be the permutation on the coordinates of S^k such that $\sigma \circ \psi$ acts coordinate-wise on S^k. Further, let σ have the disjoint cycle decomposition $\sigma = \alpha_1 \cdots \alpha_t$ and let $r_i = |\alpha_i|$ denote the cycle length of α_i. Then ψ^{r_i} acts coordinate-wise in at least r_i coordinates.

We now outline a process by which we can recover r_i and make progress toward solving the SDLP instance. Note that since r_i is bounded by k, we may merely try all of the small values of r_i at each step i, introducing

only a polynomial factor, specifically, no more than $\binom{r_i+1}{2}$, in the total number of oracle calls required to find r_i and recover the step solution.

Let α_i be one of the disjoint cycles in the decomposition of σ. We may consider the projection

$$(s_{g,\psi}(r_i), \psi^{r_i}) \mapsto (s_{g,\psi}(r_i)_j, \psi_j^{r_i}),$$

where j is a symbol in α_i, and this projection is onto the jth coordinate of S^k. Note that, given an instance $\mathsf{SDLP}_G(G, \psi)$ for g and $h = s_{g,\psi}(x)$, that one among the instances $\mathsf{SDLP}(G, \psi)$ for $s_{g,\psi}(r_i)$ and $s_{g,\psi}(x)$, for $s_{g,\psi}(r_i)$ and $s_{g,\psi}(x-1)$, ..., or for $s_{g,\psi}(r_i)$ and $s_{g,\psi}(x-r_i+1)$, has a solution. In particular, since the jth coordinate is stable under $\rho_{(g,\phi)}^{r_i}$, there is a solution x_j to the instance $\mathsf{SDLP}(S, \psi^{r_i})$ for $s_{g,\psi}(r_i)_j$ and $s_{g,\psi}(x)_j$ among the instances $\mathsf{SDLP}(S, \psi^{r_i})$ for $s_{g,\psi}(r_i)_j$ and $s_{g,\psi}(x)_j$, $s_{g,\psi}(r_i)_j$ and $s_{g,\psi}(x-1)_j$, ..., and $s_{g,\psi}(r_i)_j$ and $s_{g,\psi}(x-r_i+1)_j$. Suppose without loss of generality that the instance with solution x_j is $\mathsf{SDLP}(S, \psi^{r_i})$ for $s_{g,\psi}(r_i)_j$ and $s_{g,\psi}(x-t_j)_j$; call this solution the step solution. Since we have an instance of SDLP in one of the co-ordinates of S^k we have SDLP in the simple group S, so we may apply the oracle to recover the step solution. Moreover, this step solution x_j satisfies $x = r_i x_j + t_j$ modulo the order of $(s_{g,\psi}(r_i)_j, \psi_j^{r_i}) \in S \rtimes \mathrm{Aut}(S)$. Here we use at most r_i calls for the exponent r_i, and thus a total of no more than $\binom{r_i+1}{2}$ oracle calls to both recover r_i and the step solution x_j.

We may now consider another disjoint cycle α_ℓ of length r_ℓ containing the symbol k. We repeat the initial process to solve an instance of $\mathsf{SDLP}(G, s_{g,\psi}(r_i))$ with a strategy similar to the above step, and related to the subgroup of $S \rtimes \mathrm{Aut}(S)$ generated by $(s_{g,\psi}(r_i r_\ell)_j, \psi_j^{r_i r_\ell})$. Since $\left\langle (s_{g,\psi}(r_i r_\ell)_j, \psi_j^{r_i r_\ell}) \right\rangle$ is a subgroup of $\left\langle (s_{g,\psi}(r_i)_j, \psi_j^{r_i}) \right\rangle$, and we have previously discovered which coset of $\left\langle (s_{g,\psi}(r_i)_j, \psi_j^{r_i}) \right\rangle$ contains an element whose first coordinate is $s_{g,\psi}(x)_j$, we need only consider the r_ℓ simultaneous $\mathsf{SDLP}(S, \psi^{r_i r_\ell})$ instances $s_{g,\psi}(r_i r_\ell)_t$ and $s_{g,\psi}(x-t_j)_t$, $s_{g,\psi}(r_i r_\ell)_t$ and $s_{g,\psi}(x-r_i-t_j)_t$, ..., and $s_{g,\psi}(r_i r_\ell)_t$ and $s_{g,\psi}(x-(r_\ell-1)r_i-t_j)_t$ for t a symbol in α_i or α_ℓ to recover a step solution x_ℓ satisfying SDLP on all such coordinates modulo the least common multiple of the orders of $(s_{g,\psi}(r_i r_\ell)_t, \psi_t^{r_i r_\ell})$. This step requires at most r_ℓ calls to the oracle at exponent r_ℓ, and thus at most $\binom{r_\ell+1}{2}$ iterations to recover r_ℓ and the step solution x_ℓ.

Since at each step we require a number of SDLP instances linear in the cycle length, and each cycle length is bounded by k, the total number of oracle calls to recover a solution to $\mathsf{SDLP}(G, \psi)$ for g and $h = s_{g,\psi}(x)$ is bounded by k^3, and therefore the number of SDLP instances over the simple group S required to solve the problem is polylogarithmic in $|G|$. Thus, given an oracle solving $\mathsf{SDLP}(S, \phi)$, we may solve $\mathsf{SDLP}(S^k, \psi)$ with polynomially many branches. □

We summarize the method outlined above as follows:

Definition 3.5. *Let G be a characteristically simple group. The algorithm CharSimp assumes access to an SDLP oracle for simple groups Θ, and takes as input G, ϕ, g, h. The algorithm outputs a solution of $\mathsf{SDLP}(G, \phi)$ for g and h after applying the procedure described in the proof of Lemma 3.4.*

3.3 The Decomposition Algorithm

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

Theorem 3.6. *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 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 is 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 2, where ϕ, n_0, g', h', $\bar{\phi}$ and ψ have the same meaning as in the proof of Theorem 3.1.*

Algorithm 2 Solve(G, ϕ, g, h)

Input: (G, ϕ, g, h)
Output: Solutions of $SDLP(G, \phi)$ for g, h

Solutions $\leftarrow$ {}
if G is simple **then**
    Solutions $\leftarrow$ Solutions $\cup$ $\Theta(G, \phi, g, h)$
else
    $N \leftarrow Inv(G, \phi)$
    if $N.\mathsf{CSFlag} == 1$ **then**
        Solutions $\leftarrow$ Solutions $\cup$ $CharSimp(G, \phi, g, h)$
    else
        q_solns $\leftarrow$ Solve$(G/N, \bar{\phi}, \psi(g), \psi(h))$
        n_0 smallest positive element of q_solns
        s_solns $\leftarrow$ Solve(N, ϕ^{n_0}, g', h')
        Solutions $\leftarrow$ Solutions $\cup$ q_solns $\cup$ s_solns
    end if
end if
$X \leftarrow$ linear combinations of elements of Solutions
return X

Proof. We first verify that the algorithm terminates. 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; when N is computed its CSFlag attribute is set to 1 by Inv, and we will output $N = G$. In this case we are done by applying the $CharSimp$ algorithm of Definition 3.5. If not, we compute a ϕ-invariant subgroup N and run Solve() on the two induced problems defined in N and G/N. For these groups, if they are not simple, repeat the procedure for an appropriate subgroup, and so on.

This gives rise to a tree graph defined inductively. Define the original node to be the group G; if G is simple or characteristically simple we stop, otherwise there is a ϕ-invariant normal subgroup N and G/N that are defined as children of G. We can repeat this process for N and G/N. If the algorithm does not terminate, there is an infinite sequence of groups in which the algorithm checks for simplicity and characteristic simplicity and, having failed this test, runs itself on another instance of SDLP in another group. In other words, the algorithm failing to terminate implies the presence of at least one infinite path in the graph defined above that never reaches a simple or characteristically simple group. The basic strategy of the proof is to consider the graph above such that *none* of its nodes are simple or characteristically simple groups (so there are infinitely many infinite paths). An infinite path corresponding to the algorithm failing to terminate would be contained within this graph, so we can use its properties to extract a contradiction.

In this direction, let us recall the third isomorphism theorem, which we consider the source[5] of the following two facts: first, for a quotient group A/B, the proper normal subgroups of A/B are exactly the subgroups of the form C/B, where C is a normal subgroup of A strictly containing B; and second, that for these normal subgroups one has $(A/B)/(C/B) \cong A/C$. In other words, if we encounter a quotient group A/B on the graph, since it is not simple or characteristically simple we can compute a ϕ-invariant normal subgroup. By the discussion above, there exists $C \subset G$ such that $B \subsetneq C \subsetneq A$, and the children of A/B are C/B and A/C.

We call "level" of a node the distance of a path from the node to G; note that this is unambiguously defined, since clearly there is a unique path from each node to G. The set of nodes whose level is i for some $i \in \mathbb{N}$ can be said to be "at level i". At level 1 the process has generated the group $N \subsetneq G$, and we have to solve $\mathsf{SDLP}(G)$ and $\mathsf{SDLP}(G/N)$. Suppose at level i the process has generated $2^i - 1$ groups, say

$$N_1 \subsetneq N_1 \subsetneq \ldots \subsetneq N_{2^i-1} \subsetneq G$$

[5] Some sources call the first part of this theorem the *correspondence theorem*.

and the nodes are N_j/N_{j-1} for $j \in \{0, \ldots, 2^i\}$ (where $G = N_{2^i}$). At level $i+1$, then, there must be subgroups N_i' such that $N_i \subsetneq N'i \subsetneq N_{i+1}$, and the nodes of the next level are N_i'/N_i and $N_{i+1}/N'i$. In other words we have a chain of subsets of G

$$N_1 \subsetneq N_1' \subsetneq N_2 \subsetneq \ldots \subsetneq N_{2^i-1}' \subsetneq G$$

It follows by induction that at level i we describe 2^{i-1} subgroups distinct from all the subgroups described in the previous levels. Since an infinite path in the graph defined in the first paragraph would occur as a subgraph of the graph containing no simple groups, and such a path would describe infinitely many distinct subsets of G, we have a contradiction. The algorithm must therefore terminate.

We can refine this argument a little: supposing that no simple groups are encountered, reviewing the argument above we must actually have that the 2^{i-1} subgroups described at level i of the graph are a chain of subgroups; that is, one has $N_1 \triangleleft \ldots \triangleleft N_{2^i-1} \triangleleft G$. Since the order of the subgroup divides that of its parent subgroup each N_i has size at most half of N_{i+1}, so we can describe at most $\log_2 |G|$ subgroups in this way. We therefore have to make at most $\log_2 |G|$ calls to Ω. A similar argument shows that the level at which the algorithm terminates does not exceed $\log_2 |G|$; since applications of Algorithm 1 and *CharSimp* run in time quantum polynomial in $\log |G|$, the complexity claim of theorem follows. Finally, it follows directly from the proof of [II24, Theorem 3] (recorded in this paper as Theorem 3.1) that there is a linear combination of the elements of the set Solutions that returns the solution of $\mathsf{SDLP}(G, \phi)$. We eschew the details of the precise form of such a linear combination. □

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 [II24, Theorem 8], which provides a reduction of $\mathsf{SDLP}(G, \phi)$ 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 [II24, 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[6]; 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 $T^i \circ \psi = \psi \circ \rho_{(g,\phi)}^i$. It follows that, in order to solve $\mathsf{SDLP}(G, \phi)$, 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 $G \to \mathbb{P}\mathsf{GL}_n(K)$ we show that the same reduction can be applied to projective matrices in $\mathbb{P}\mathsf{GL}_{n^2}(K)$.

Before presenting this reduction and discussing its efficiency, let us see that it is indeed possible to compute the crucial matrix $\mathbf{T}$. In order to do this, we will have to re-introduce a small amount of technicality which

[6] 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.

was suppressed in the outline above: to be able to think of elements of $M_n(K)$ as concrete n^2-dimensional column vectors, we have to choose a basis in which to represent them. We just pick the basis defined by stacking the columns of an $n \times n$ matrix to obtain an n^2-dimensional vector; in other words there is a function $\mathsf{vec} : M_n(K) \to K^{n^2}$ defined by $\mathsf{vec}(M)_{in+j} = M_{j,i}$. If we are dealing with projective matrices we use instead $\mathbb{P}\mathsf{vec} : \mathbb{P}\mathsf{GL}_n(K) \to \mathbb{P}\left(K^{n^2}\right)$ which takes a representative of a projective class and associates the class of the image through vec. Since both the classes are defined up to scalar multiplication of elements in K^*, the function is well defined.

Lemma 4.1. *Let G be a finite group, and $\psi : G \to (\mathbb{P})\mathsf{GL}_n(K)$ a (projective) linear representation. Given an instance of $\mathsf{SDLP}(G, \phi)$, where ϕ is an inner automorphism, i.e., $\phi(g) = mgm^{-1}$ for some $m \in G$. Define $\mathbf{T} := \psi(gm)^t \otimes \psi(m^{-1}) \in (\mathbb{P})\mathsf{GL}_{n^2}(K)$, then for any $h \in G$*

$$\mathbf{T} \circ ((\mathbb{P})\mathsf{vec} \circ \psi) = ((\mathbb{P})\mathsf{vec} \circ \psi) \circ \rho_{(g,\phi)} \ . \tag{1}$$

Proof. Since ψ is a homomorphism we have:

$$\psi(\rho_{(g,\phi)}(h)) = \psi(g \cdot mhm^{-1}) = \psi(gm) \cdot \psi(h) \cdot \psi(m^{-1}) \ ;$$

thus, by using a basic property of the Kronecker Product, we have:

$$\mathsf{vec}\left(\psi(gm) \cdot \psi(h) \cdot \psi(m^{-1})\right) = \left(\psi(gm)^t \otimes \psi(m^{-1})\right) \cdot \mathsf{vec}\left(\psi(h)\right) \ .$$

By the property of Kronecker product $\mathsf{rank}(\mathbf{T}) = \mathsf{rank}(\psi(gm))\mathsf{rank}(\psi(m^{-1})) = n \cdot n$, so $\mathbf{T}$ is also invertible. To prove the projective case we just have to consider the representative in $\mathsf{GL}_n(K)$ of the matrices during the application of $\mathbb{P}\mathsf{vec}$. □

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

Proof. By Lemma 4.1 we can compute a (projective) linear map $\mathbf{T}$ such that $\mathbf{T} \circ ((\mathbb{P})\mathsf{vec} \circ \psi) = ((\mathbb{P})\mathsf{vec} \circ \psi) \circ \rho_{(g,\phi)}$. Note that this implies that for all $i \in \mathbb{N}$ we have

$$\mathbf{T}^i \circ (\mathbb{P})\mathsf{vec} \circ \psi = (\mathbb{P})\mathsf{vec} \circ \psi \circ \rho^i_{(g,\phi)} \ .$$

We are tasked with finding $x \in \mathbb{N}$ such that $\rho^x_{(g,\phi)}(1_G) = h$, for some $h \in G$. Applying ψ to each side of this equation we have to find $x \in \mathbb{N}$ such that

$$(\mathbb{P})\mathsf{vec}(\psi(h)) = \mathbf{T}^x \cdot (\mathbb{P})\mathsf{vec}(\psi(1_G))$$

Let us rename the vectors in play here: define $\mathbf{a} = (\mathbb{P})\mathsf{vec}(\psi(1_G))$ and $\mathbf{b} = (\mathbb{P})\mathsf{vec}(\psi(h))$.[7]

The reduction to the matrix power follows the proof of Theorem 8 of [II24] (that is an adaptation to finite fields of [KL86, Theorem 1]), to then be adapted to projective matrices.

In the non-projective case, consider the subspace W of the (vector space) K^{n^2} spanned by the vectors $\{\mathbf{T}^i\mathbf{a} \mid i \geq 0\}$. First, check if $\{\mathbf{a}, \mathbf{Ta}\}$ is linearly independent by means of Gaussian elimination. If not, check if $\{\mathbf{a}, \mathbf{Ta}, \mathbf{T}^2\mathbf{a}\}$ is linearly independent - since the vector space is of dimension n^2 eventually we arrive at some $k \leq n^2$ such that $\{\mathbf{a}, ..., \mathbf{T}^{k-1}\mathbf{a}\}$ is linearly independent, but $\{\mathbf{a}, ..., \mathbf{T}^{k-1}\mathbf{x}, \mathbf{T}^k\mathbf{x}\}$ is not. In fact the set $\{\mathbf{a}, ..., \mathbf{T}^{k-1}\mathbf{a}\}$ is a basis for W, which we can see by induction. Without loss of generality we may write

[7] Note that $\psi(1_G)$ is the identity matrix, which gets sent to some sparsely populated vector of 1s and 0s. In other words $\mathsf{vec}(\psi(1_G))$ does not act as an "identity" element.

$\mathbf{T}^k\mathbf{a} = \sum_{i=0}^{k-1} \lambda_i \mathbf{T}^i\mathbf{a}$ with each $\lambda_i \in K$, so $\mathbf{T}^{k+1} = \sum_{i=1}^{k-1} \lambda_i \mathbf{T}^i\mathbf{a} + \mathbf{T}^k\mathbf{a}$. However, since we have seen that $\mathbf{T}^k\mathbf{a}$ has a suitable linear decomposition, it follows that $\mathbf{T}^{k+1}$ does too - and the rest of the claim follows by induction.

Consider now the $k \times k$ matrices $\mathbf{C}$ and $\mathbf{D}$ whose columns are $\mathbf{a}, ..., \mathbf{T}^{k-1}\mathbf{a}$ and $\mathbf{b}, ..., \mathbf{T}^{k-1}\mathbf{b}$, respectively. If $k = n^2$, these matrices are $n^2 \times n^2$; otherwise, since $\mathbf{b} = \mathbf{T}^x\mathbf{a}$, we have that $\mathbf{b} \in W$, and indeed that $\mathbf{T}^i\mathbf{b} \in W$ for each $i \in \mathbb{N}$. We may therefore write the vectors $\mathbf{a}, ..., \mathbf{T}^{k-1}\mathbf{a}$ and $\mathbf{b}, ..., \mathbf{T}^{k-1}\mathbf{b}$ as height-k column vectors with respect to the basis of W we found (after computing the restriction of $\mathbf{T}$ to the subspace W). In other words, the matrices that we consider are all square. We have

$$\begin{aligned}\mathbf{T}^x\mathbf{C} &= \mathbf{T}^x\{\mathbf{a}|\mathbf{T}\mathbf{a}|...|\mathbf{T}^{k-1}\mathbf{a}\} \\ &= \{(\mathbf{T}^x\mathbf{a})|\mathbf{T}(\mathbf{T}^x\mathbf{a})|...|\mathbf{T}^{k-1}(\mathbf{T}^x\mathbf{a})\} \\ &= \{\mathbf{b}|\mathbf{T}\mathbf{b}|...|\mathbf{T}^{k-1}\mathbf{b}\} \\ &= \mathbf{D}.\end{aligned}$$

We have computed matrices $\mathbf{T}$, $\mathbf{C}$, and $\mathbf{D}$ such that, in order to find the $x \in \mathbb{N}$ such that $h = \rho^x_{(g,\phi)}(1_G)$, it suffices to find the $x \in \mathbb{N}$ such that $\mathbf{T}^x = \mathbf{C}\mathbf{D}^{-1}$. The result follows by noting that the complexity of this method is dominated by the Gaussian elimination required to compute the basis of the subspace W, requiring at least one computation of complexity $\mathcal{O}(n^2k^2)$. Since k is bounded above by n^2, we are done.

In the projective case, we have projective vectors $[\mathbf{a}]$ and $[\mathbf{b}]$ such that a projective matrix $[\mathbf{T}]$ is such that $[\mathbf{T}^x \cdot \mathbf{a}] = [\mathbf{b}]$. We can therefore just pick representatives of the projective class; that is, there are vectors $\mathbf{a}, \mathbf{b}$ and a linear matrix $\mathbf{T}$ such that for some scalar λ in the underlying field we have

$$\mathbf{b} = \lambda\mathbf{T}^x\mathbf{a}$$

This time, we just have to compute a basis of the subspace W spanned by $\{\lambda\mathbf{T}^i\mathbf{a} : i \in \mathbb{N}\}$, which we can do just by picking arbitrary representatives of the appropriate projective classes (since the span is the same under scalar multiplication). The $k \times k$ matrices $\mathbf{C}$ and $\mathbf{D}$ as defined above are such that $\lambda\mathbf{T}^x\mathbf{C} = \mathbf{D}$, so $\mathbf{T}^{\mathbf{x}} = \lambda^{-1}\mathbf{C}\mathbf{D}^{-1}$. Projecting back down we have $[\mathbf{T}]^x = [\mathbf{T}^{\mathbf{x}}] = [\mathbf{C}\mathbf{D}^{-1}]$, thereby inducing a matrix power problem in the projective space. □

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 [II24, 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 $\mathsf{SDLP}(G, \phi^y)$. It turns out, due to a result of Kohl [Koh03, Theorem 1] that for simple groups one can expect this power to be small.

Theorem 4.3 (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 4.4. *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 $\mathsf{SDLP}(G, \phi)$ for any $\phi \in \operatorname{Aut}(G)$ on a quantum computer in probabilistic polynomial time in $\log |G|$.*

Remark 4.5. 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 key advantage of the reduction to the simple groups from Theorem 3.6 is that we have access to the famous classification of finite simple groups. For this result we take as reference the book *The Finite Simple Groups* of Robert Wilson [Wil09], and further insight on the topic can be found in *The Atlas of Finite Groups* [CW98]. Summarizing the pivotal results, we know that 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** $\mathbb{A}_n$;
3. A **classical group** of Lie Type:

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

   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**.

Since we have a complete (and quite short) list of what all the finite simple groups are, we can analyze the hardness of solving SDLP separately for each of them.

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 4.4, while for the sporadic groups (and the Tits group) we have a separate discussion in Section 5.2.

5.1 Infinite Families

For each of the non-sporadic groups, 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 4.4 there is a quantum polynomial-time algorithm to solve $\mathsf{SDLP}(G, \phi)$.

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** [BB99, 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. An alternating group is 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 $\mathsf{GL}_n(K)$.

Also, thanks to [Jam+13, 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 $\mathbb{A}_n$, where N is the string length of the black-box group. As a consequence of Corollary 4.4, we have the following result.

Lemma 5.1. *If G is a simple black-box group isomorphic to any alternating group $\mathbb{A}_n$ we can solve* SDLP *for G in probabilistic polynomial time in $n \log |G|$ on a quantum computer.*

Classical Groups. The classical groups of Lie Type are the groups of linear, unitary, symplectic and orthogonal projective matrices, so they are all naturally described as well-defined subgroups of $\mathbb{P}\mathsf{GL}_n(\mathbb{F}_q)$, i.e., we can use the inclusion as a projective linear representation. Again, this means that we can solve SDLP such groups using a quantum computer as a consequence of Corollary 4.4. Observe that here it is important to have a reduction that also works for projective representations.

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 [BK99], unitary [Bro03], symplectic [Bro08] and orthogonal [BK06]), summarized in [DLO15], 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 $\mathsf{SL}(2,q) \to G$.

Since, thanks to Shor's algorithm [Sho94], we know that quantum computers can implement efficient *number theory oracles*, we can combine the previous results in the following lemma.

Lemma 5.2. *On a quantum computer, if G is a simple black-box group isomorphic to any classical group of Lie Type of characteristic q and dimension n, we can reduce* SDLP *for G in probabilistic polynomial time in n and $\log(q)$ to the constructive recognition problem for the group $\mathsf{SL}(2,q)$.*

We tackle this problem in a separate section after the discussion on exceptional groups.

Exceptional Groups. As for the classical groups, we start by showing that an efficient linear representation is known, then we discuss the difficulty of computing an isomorphism starting from a black-box group. For our cryptographic context, the Tits group is more reasonable to be treated with the sporadic ones.

We start immediately from the groups of untwisted type; thanks to the arguments in [Wil09, Section 4.12], we have the following relationship between the families

$$\mathtt{G}_2 < \mathtt{F}_4 < \mathtt{E}_6 < \mathtt{E}_7 < \mathtt{E}_8 \ .$$

It follows that, since $E_8(q)$ can be represented as automorphisms of a Lie algebra of dimension 248 (see again [Wil09, Section 4.12]), we have an efficient representation for all of them.

The twisted group ${}^3D_4(q)$ is well known, thanks to its relation with the orthogonal family, to have a linear representation in dimension 8, see [Wil09, Section 4.6]. Also, the twisted group of type ${}^2E_6(q) < E_6(q^2)$ [Wil09, Section 4.11] can be represented using the linear representation of $E_6(q^2)$.

For the more exceptional ones, with fields of characteristic 2 or 3, there are efficient representations known in the literature:

– The Suzuki groups ${}^2B_2\left(2^{2n+1}\right)$ are defined in [Suz60] as subgroups of $\mathsf{SL}_4(\mathbb{F}_{2^{2n+1}}) \leq \mathsf{GL}_4(\mathbb{F}_{2^{2n+1}})$;

– The family of small Ree groups ${}^2G_2\left(3^{2n+1}\right)$, are described in [Wil10b] as groups of 7×7 matrices over $\mathbb{F}_{3^{2n+1}}$;
– The family of large Ree groups ${}^2F_4\left(2^{2n+1}\right)$ is described in [Wil10a] as symmetries of a 26 dimensional vector space over $\mathbb{F}_{2^{2n+1}}$.

The Tits Group can also be represented via matrices, being a subgroup of ${}^2F_4\left(2^{2n+1}\right)$, however, it being a standalone group, we leave the discussion on its utility to Section 5.2.

With respect to the constructive recognition problem, in [KM13; KM15] 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 the classical groups of Lie type discussed above, so we have

Lemma 5.3. *On a quantum computer, if G is a simple black-box group isomorphic to any exceptional group of Lie Type defined on the finite field K, 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 $\log|K|$ to the constructive recognition problem for the group $\mathsf{SL}(2,q)$.*

We can finally enter the discussion for constructive recognition problem of 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 [CLO06] 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 [BBS09, 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 [KK15] for even characteristic and in [BY13] for the case of small characteristic $p \equiv 1 \mod 4$. For a general field, the research is partially open: actually, in the preprint [BY20], the authors show how to compute an isomorphism in polynomial time between the black-box group and $\mathsf{SL}_2()$, 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 are not part of the infinite families discussed earlier, plus the Tits Group ${}^2F_4(2)'$. It is clear by the definition of semidirect product that instead of choosing $x \in \mathbb{N}$ in the definition of the SDLP in G, we can restrict without loss of generality to $x \leq \max_{g\in G}(\mathrm{ord}(g)) \cdot \max_{\phi\in \mathrm{Aut}(G)}(\mathrm{ord}(\phi))$.

The largest of the 26 *sporadic* groups is the Fischer-Griess monster group $\mathbb{M}$, which has no outer automorphisms [Lyo11], i. e., $\mathrm{Aut}(\mathbb{M}) \simeq \mathbb{M}$. Consequently, the value of x is upper-bounded by $119^2 < 2^{14}$ [BSW22, Table 14], placing SDLP in $\mathbb{M}$ well in reach of an exhaustive search. With the exception of six *pariahs*, all sporadic groups are part of the *happy family*, i. e., they are subquotients of $\mathbb{M}$ [Gri82]. Additionally, the Tits group ${}^2F_4(2)'$ can be considered as part of this family since it is a maximal subgroup of the Fischer Group Fi_{22} [Wil09, Section 5.7.2]. Moreover, for all sporadic groups G, the outer automorphism group has order at most 2. Therefore, the order of an automorphism of a sporadic group in the happy family is upper-bounded by $2 \cdot 119$, yielding an upper bound $2 \cdot 119^2 < 2^{15}$ for x. Consequently, SDLP over members of the happy family is firmly within reach of an exhaustive search.

Using, for example, the computer algebra system GAP [GAP24], one can verify that the maximal element order in the six pariahs and in their automorphism groups is upper-bounded by 67. Thus, for SDLP in the

pariahs, we can upper bound x with $67^2 < 2^{13}$, which is within reach of an exhaustive search, too. We summarize this as follows.

Lemma 5.4. *For any sporadic finite simple group G and automorphism $\phi \in \mathrm{Aut}(G)$, there is a brute force algorithm to solve $\mathsf{SDLP}(G, \phi)$ with at most 2^{14} multiplications in the holomorph of G.*

5.3 Adapting Shanks' Baby-Step Giant-Step algorithm

Adjusting Shanks' Baby-Step Giant-Step (BSGS) algorithm [Sha71] 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 3 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 3 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.

$n \leftarrow \lceil \sqrt{N} \rceil$
$(s, t) \leftarrow ((g, \phi)^n, (1, id))$
$T \leftarrow [(0, t)]$ ▷ 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 5.5. We implemented our BSGS algorithm in approximately 30 lines of Python using the `mmgroup` Python library [Sey24], 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.

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, 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 3).

We see that, if the groups are given in their natural representations we can find linear representations and apply Corollary 4.4 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.

We close with some high-level remarks. It is perhaps not too surprising, given the existing rich theory of finite group decomposition, that we could reduce an arbitrary instance of SDLP to SDLP in finite simple groups. However, the fact that *all* of these finite simple groups admit efficient methods of solving SDLP—in particular, the fact that all the infinite families of simple groups have low faithful dimension—is quite unexpected. 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 Julian Brough, Tobias Hemmert, and Ray Perlner for spotting problems in the reasoning of an earlier version of this paper, and bringing those to our attention. 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 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.

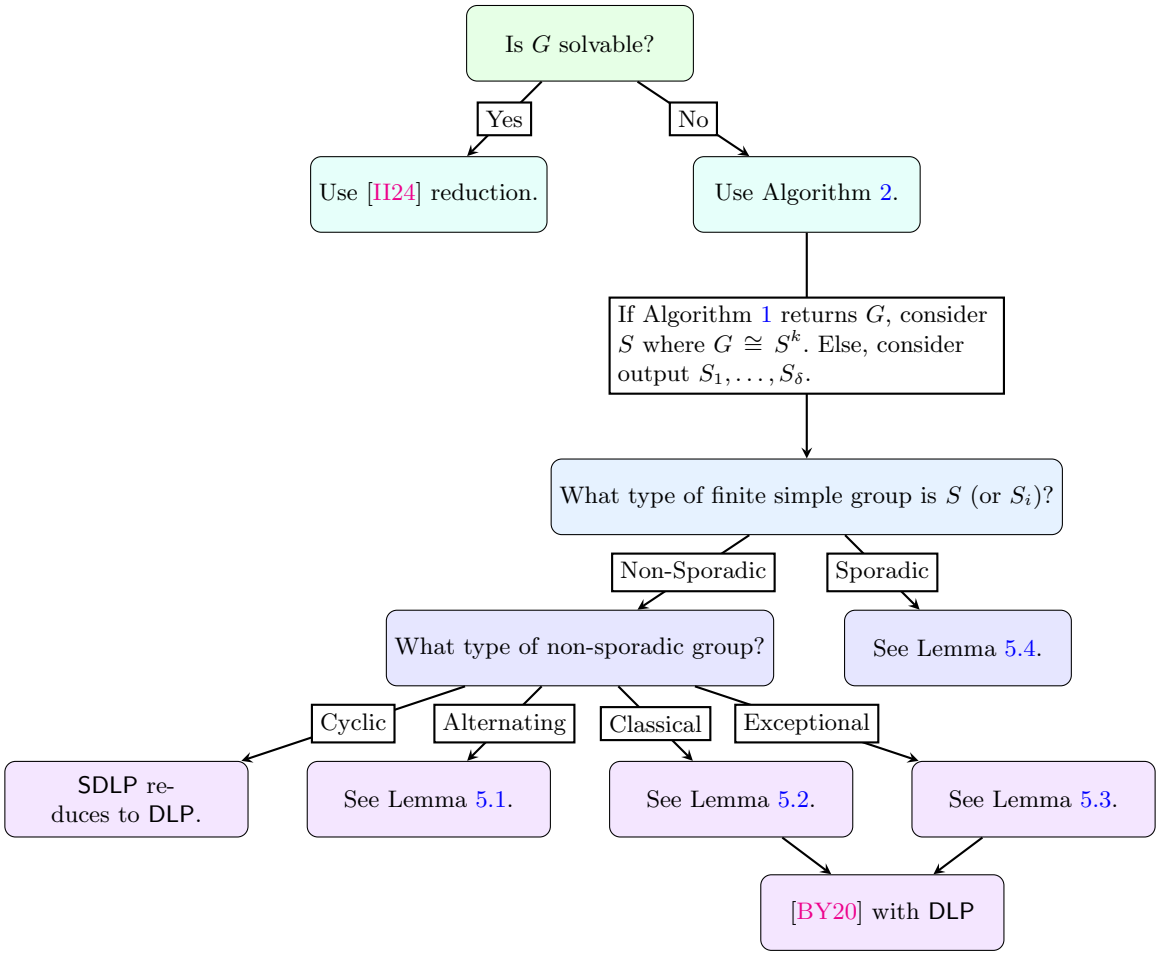

Fig. 1. Visual summary of a possible roadmap for a general SDLP instance over a finite group.

References

[Bae64] 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.

[Bat+23a] 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].

[Bat+23b] 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.

[BB99] 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.

[BBS09] 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.

[BCM06] Jonathan NS Bidwell, M John Curran, and Dennis J McCaughan. "Automorphisms of direct products of finite groups". In: *Archiv der Mathematik* 86 (2006), pp. 481–489.

[BK06] 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.

[BK99] 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.

[BKL22] Daniel Brown, Neal Koblitz, and Jason Legrow. "Cryptanalysis of 'MAKE'". In: *J. Math. Cryptol.* 16.1 (2022), pp. 98–102.

[BKS23] 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.

[Bor15] Alexander Bors. *A bound on element orders in the holomorph of a finite group*. 2015. arXiv: 1510.02014 [math.GR].

[Bro03] Peter A. Brooksbank. "Fast constructive recognition of black-box unitary groups". In: *LMS Journal of Computation and Mathematics* 6 (2003), pp. 162–197.

[Bro08] 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.

[BS84] 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.

[BSW22] Daniela Bubboloni, Pablo Spiga, and Thomas Weigel. *Normal 2-coverings of the finite simple groups and their generalizations*. 2022. arXiv: 2208.08756 [math.GR].

[BY13] Alexandre Borovik and Sukru Yalcinkaya. *Steinberg presentations of black box classical groups in small characteristics*. 2013. arXiv: 1302.3059 [math.GR].

[BY20] Alexandre Borovik and Şükrü Yalçınkaya. *Natural representations of black box groups encrypting $SL_2(\mathbb{F}_q)$*. 2020. arXiv: 2001.10292 [math.GR].

[CI14] Andrew M. Childs and Gábor Ivanyos. "Quantum computation of discrete logarithms in semigroups". In: *J. Math. Cryptol.* 8.4 (2014), pp. 405–416.

[CL01] 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.

[CLO06] 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.

[CW98] Robert Curtis and Robert A. Wilson, eds. *The Atlas of Finite Groups - Ten Years On*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1998.

[DLO15] 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.

[Eke21] Martin Ekerå. "On completely factoring any integer efficiently in a single run of an order-finding algorithm". In: *Quantum Information Processing* 20.6 (2021), p. 205.

[GAP24] GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.13.0*. 2024. URL: https://www.gap-system.org.

[Gri82] Robert L. Griess Jr. "The Friendly Giant." In: *Inventiones Mathematicae* 69 (1982), pp. 1–102.

[GS19] Dima Grigoriev and Vladimir Shpilrain. "Tropical cryptography II: extensions by homomorphisms". In: *Communications in Algebra* 47.10 (2019), pp. 4224–4229.

[Hab+13] 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.

[II24] Muhammad Imran and Gábor Ivanyos. "Efficient quantum algorithms for some instances of the semidirect discrete logarithm problem". In: *Designs, Codes and Cryptography* (May 2024).

[IMS01] 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.

[Jam+13] 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.

[KK15] William M. Kantor and Martin Kassabov. "Black box groups isomorphic to PGL (2, 2e)". In: *Journal of Algebra* 421 (2015), pp. 16–26.

[KL86] Ravindran Kannan and Richard J. Lipton. "Polynomial-time algorithm for the orbit problem". In: *Journal of the ACM (JACM)* 33.4 (1986), pp. 808–821.

[KM13] W. M. Kantor and K. Magaard. "Black box exceptional groups of Lie type". In: *Trans. Amer. Math. Soc.* 365.9 (2013), pp. 4895–4931.

[KM15] W. M. Kantor and K. Magaard. "Black box exceptional groups of Lie type II". In: *Journal of Algebra* 421 (2015), pp. 524–540.

[Koh03] 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.

[KS16] 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.

[Lee01] Charles R. Leedham-Green. "The computational matrix group project". In: *Groups and computation* 3 (2001), pp. 229–248.

[Lyo11] Richard Lyons. *Automorphism groups of sporadic groups*. 2011. arXiv: 1106.3760 [math.GR].

[MDL23] 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.

[MM20] Chris Monico and Ayan Mahalanobis. *A remark on MAKE – a Matrix Action Key Exchange*. 2020. arXiv: 2012.00283 [cs.CR].

[Mon21] Chris Monico. *Remarks on MOBS and cryptosystems using semidirect products*. 2021. arXiv: 2109.11426 [cs.CR].

[MR15] Alexei Myasnikov and Vitaliĭ Roman'kov. "A linear decomposition attack". In: *Groups Complexity Cryptology* 7.1 (2015), pp. 81–94.

[NIS17] NIST. *Post-Quantum Cryptography Standardization*. URL: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography. 2017.

[OBr11] Eamonn A O'Brien. "Algorithms for matrix groups". In: *London Math. Soc. Lecture Note Ser* 388 (2011), pp. 297–323.

[Rom15] Vitaliĭ Roman'kov. *Linear decomposition attack on public key exchange protocols using semidirect products of (semi) groups*. 2015. arXiv: 1501.01152 [cs.CR].

[RS21] 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.

[RS22] Nael Rahman and Vladimir Shpilrain. "MAKE: A matrix action key exchange". In: *J. Math. Cryptol.* 16.1 (2022), pp. 64–72.

[Ser77] Jean-Pierre Serre. *Linear Representations of Finite Groups*. Vol. 42. Graduate Texts in Mathematics. Springer, 1977.

[Sey24] Martin Seysen. *Python implementation of the monster group*. GitHub repository. 2024. URL: https://github.com/Martin-Seysen/mmgroup.

[Sha71] Daniel Shanks. "Class number, a theory of factorization, and genera". In: *Proceedings of Symposia in Pure Mathematics*. 1971.

[Sho94] 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.

[Sho97] 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.

[Suz60] Michio Suzuki. "A New Type of Simple Groups of Finite Order". In: *PNAS* 46.6 (1960), pp. 868–870.

[Wil09] Robert A. Wilson. *The Finite Simple Groups*. Vol. 251. Graduate Texts in Mathematics. Springer, 2009.

[Wil10a] Robert A. Wilson. "A simple construction of the Ree groups of type 2F4". In: *Journal of Algebra* 323.5 (2010), pp. 1468–1481. ISSN: 0021-8693.

[Wil10b] Robert A. Wilson. "Another new approach to the small Ree groups". In: *Archiv der Mathematik* 94.6 (2010), pp. 501–510.