SPDH-Sign: towards Efficient, Post-quantum Group-based Signatures

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

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

Abstract. In this paper, we present a new diverse class of post-quantum group-based Digital Signature Schemes (DSS). The approach is significantly different from previous examples of group-based digital signatures and adopts the framework of group action-based cryptography: we show that each finite group defines a group action relative to the semidirect product of the group by its automorphism group, and give security bounds on the resulting signature scheme in terms of the group-theoretic computational problem known as the Semidirect Discrete Logarithm Problem (SDLP). Crucially, we make progress towards being able to efficiently compute the novel group action, and give an example of a parameterised family of groups for which the group action can be computed for any parameters, thereby negating the need for expensive offline computation or inclusion of redundancy required in other schemes of this type.

Keywords: Group-based Signature · Post-quantum Signature · Group Action Based Cryptography · Post-quantum Group-based Cryptography

Introduction

Since the advent of Shor's algorithm and related quantum cryptanalysis, it has been a major concern to search for quantum-resistant alternatives to traditional public-key cryptosystems. The resultant field of study is known today as Post-Quantum Cryptography (PQC), and has received significant attention since the announcement of the NIST standardisation.

One of the goals of PQC is to develop a quantum-resistant Digital Signature Scheme (DSS), a widely applicable class of cryptographic scheme providing certain authenticity guarantees. Following multiple rounds of analysis, NIST have selected three such schemes for standardisation, two of which are based on the popular algebraic notion of a lattice. Nevertheless, stressing the importance of diversity amongst the post-quantum roster, a call for efficient DSS proposals not based on lattices was issued in 2022 [30]. A potential source of post-quantum hard computational problems come from group-based cryptography; for a comprehensive survey of the field including examples of DSSs, see the work of Kahrobaei et al in [18], [19].

Recall that a finite commutative group action consists of a finite abelian group G, a finite set X, and a function mapping pairs in $G \times X$ into G. Another promising framework for PQC has its origins in the so-called *Hard Homogenous Spaces* of Couveignes[6] [10]: one considers a family of group actions for which all the 'reasonable' operations - for example, evaluating the group action function, and sampling uniformly from the group - can be done efficiently, but a natural analogue of the discrete logarithm problem called the Vectorisation Problem is computationally difficult. Given such a group action, one can exploit the commutativity of the group operation to construct a generalisation of the Diffie-Hellman Key Exchange protocol based on the difficulty of the Vectorisation Problem, which is believed to be post-quantum hard.

As well as this analogue of Diffie-Hellman, the group action framework is used to construct an interactive proof of identity, which is effectively a standard three-pass identification scheme. In his doctoral thesis [36], Stolbunov uses this identification scheme to obtain a signature scheme by applying the standard Fiat-Shamir heuristic; we will here follow the convention of referring to this scheme as the CRS[7] Digital Signature Scheme (`CRS-DSS`). In order to specify a practical signature scheme it remains to specify a group action: very roughly, `CRS-DSS` uses the celebrated example, coming from the theory of isogenous elliptic curves, of a finite abelian group called the 'class group' acting on a set of elliptic curves.

`CRS-DSS` did not recieve much attention for a number of years, for two key reasons: first, it was demonstrated that the scheme admits an attack of quantum subexponential complexity [7] (in fact, this attack applies to all group-action based cryptography). This might in itself be tolerable; much more troubling is that the original version of `CRS-DSS` is unacceptably slow. There has, however, been a resurgence of interest in schemes similar to `CRS-DSS` following the discovery in [6] of a much faster isogeny-based group action; on the other hand, the computation of the class group is in general thought to be computationally difficult. In fact this is quite a significant problem: without random sampling the security proofs, which rely on group elements hiding secrets to have the appropriate distribution, break down. Two approaches to solving this problem have been suggested: in [12], one uses the 'Fiat-Shamir with aborts' technique developed by Lyubashevsky [22], at the cost of rendering the scheme considerably less space efficient; in [3], a state-of-the-art computation of a class group is performed and the resulting group action is used as the platform for `CRS-DSS`. However, it is important to note that here the computation of *a* class group is performed, and

[6] Similar notions were arrived at independently by Rostovstev and Stolbunov [34], [35].

[7] Couveignes, Rostovstev and Stolbunov.

so one is restricted in terms of tweaking parameters. In particular, the introduction of new parameters would require another extremely expensive offline class group computation.

A potential third solution is to dispense with the isogeny-based group action altogether, and search for different examples of group actions for which computing the appropriate group - and therefore uniform sampling - is efficient. Historically speaking, there has not been much research in this direction since non-trivial examples of cryptographically interesting group actions have not been available - though this work is predated by a general framework for actions by semigroups in [27], and an example semigroup action arising from semirings in [24]. In this paper we make an important step in the search for efficient group actions; in particular we show that every finite group gives rise to a group action on which `CRS-DSS`-type signatures can be constructed, and that the respective group is cyclic and has order dividing a known quantity. These group actions arise from the group-theoretic notion of the semidirect product, and were first studied in the context of a generalisation of Diffie-Hellman [15] - note, however, that it was not known at the time that the proposed framework was an example of a group action. Indeed, the link was only discovered rather recently [2], and prompted the isogeny-style renaming of the key exchange in [15] as **S**emidirect **P**roduct **D**iffie-**H**ellman, or SPDH (to be pronounced ‘spud’). With this in mind, in this paper we propose a hypothetical family of digital signature schemes which we christen `SPDH-Sign`.

It is important to note that we do not provide concrete security parameters, nor do we claim a security improvement over similar schemes: instead, the paper has two key contributions. First, we notify the community of a promising step towards efficient, scalable sampling in cryptographic group actions: our Theorem 4 shows that for each group action we construct there is quite a severe restriction on the possible sizes of the cyclic group acting. Since sampling from a cyclic group is trivial if we know its order, we have provided a large class of candidate group actions for which sampling is efficient. As such we also carry out the standard methodology of defining a resulting signature scheme, and give a security proof in the random oracle model that bounds the security of the signature scheme in terms of our central algorithmic problem in more explicit terms than comparable proofs.

The second key contribution is the proposal of a specific group as an example of a group in which one can efficiently sample in the resulting group action whilst maintaining resistance to related (but not known equivalent) cryptanalysis. Here we see an example of our Theorem 4 in action - the size of the crucial parameter needed for efficient sampling can be one of only 12 values, and we can check the validity of each of these values in logarithmic time.

Related Work

The following is a short note to emphasise the novelty of our contribution with respect to related areas of the literature.

The idea of defining cryptography based on the action of a semigroup on a set, and the resulting "semigroup action problem" (SAP), is proposed in Chris Monico's thesis [27], and is referenced by Han and Zhuang in their recent paper [16]. Certainly this idea of a semigroup action predates our establishment of a cryptographically relevant group action arising from topics in group theory. We therefore clarify that our contribution is not the novel proposal of a group action of this type, but the explicit connection between cryptographic group actions and the problems arising from semidirect product key exchange, which originally appears in [15].

In [16], SAP and the semidirect product key exchange are mentioned in the same breath in the introduction. This, however, does not constitute the explicit connection of the problem originally appearing in the discussion of semidirect product key exchange and cryptographic group actions - where this connection is one of the claimed novel aspects of our paper - but a list of problems related to the semigroup DLP. Moreover, none of the semidirect product key exchange-adjacent literature we are aware of mentions SAP, including proposals of semidirect product key exchange [15, 20, 31, 32] and cryptanalysis of the semidirect product key exchange authored by Monico himself [26, 25]. Accordingly, we believe that establishing the connection between semidirect product key exchange and group-action based cryptography is a novel contribution to the area.

1 Preliminaries

1.1 The Semidirect Product

The term 'semidirect' product refers, generally speaking, to a rather deep family of notions describing the structure of one group with respect to two other groups. For our purposes we are interested in a rather specific case of the semidirect product, defined as follows:

Definition 1. *Let G be a finite group and $Aut(G)$ its automorphism group. Suppose that the set $G \times Aut(G)$ is endowed with the following operation:*

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

where the multiplication is that of the underlying group G, and the automorphism $\phi'\phi$ is the automorphism obtained by first applying ϕ, and then ϕ'. We denote this group $G \ltimes Aut(G)$.

A few facts about this construction are standard.

Proposition 1. *Let G be a finite group and $\Phi \leq Aut(G)$ (where Φ can be any subgroup, including $Aut(G)$ itself). One has the following:*

1. *$G \ltimes \Phi$ is a finite group of size $|G||\Phi|$*
2. *Let $(g, \phi) \in G \ltimes \Phi$. One has*

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

1.2 Proofs of Knowledge and Identification Schemes

Roughly speaking, the idea of the Fiat-Shamir class of signatures is as follows: we interactively convince an 'honest' party that we possess a certain secret. We can then transform this interactive paradigm to a non-interactive digital signature scheme by applying the Fiat-Shamir transform. A primary motivation for this approach is that the resulting signature scheme inherits its security at rather low cost from security properties of the underlying interactive scheme - as such, it is necessary for us now to review some of these security notions.

First, let us define exactly what we mean by these interactive proof of knowledge protocols. The idea of communicating a 'secret' is neatly captured by the notion of a binary relation; that is, for two sets $\mathcal{W}$ and $\mathcal{S}$, consider a set $\mathcal{R} \subset \mathcal{W} \times \mathcal{S}$. Given a pair $(w, s) \in \mathcal{R}$, we say s is the *statement* and w is the *witness*. In general, for a given statement a party called the 'prover' wishes to demonstrate their knowledge of a valid witness (that is, given s we wish to prove that we possess a w such that $(w, s) \in \mathcal{R}$) to a party called the *verifier*. Of course, one can do this trivially by simply revealing the witness, so we add the crucial requirement that *no information about the witness is revealed*.

We refer more or less to this idea when discussing identification schemes, with the caveat that the prover should be able to compute an arbitrary pair of the binary relation. If the prover cannot generate an an arbitrary pair of the binary relation, and instead is to demonstrate his knowledge of some given element of the binary relation, we have instead a 'zero-knowledge proof'. A notable class of zero-knowledge proofs are the so-called 'sigma protocols'. One can always turn a zero-knowledge proof into an identification scheme by providing the prover with an algorithm capable of generating an arbitrary pair of the binary relation; our definition of identification schemes in fact refers only to those arrived at by transforming a sigma protocol into an identification scheme.

Notice that the idea of a binary relation serves as a neat generalisation of the usual notion of a public and private key pair. The algorithm used by the identification scheme to generate binary relation instances is therefore denoted by `KeyGen`, and produces a pair (sk, pk). We also require, in some sense to be made precise later, that recovering an appropriate witness from a statement is computationally difficult.

Definition 2 (Identification Scheme). *Let $\mathcal{R} \subset \mathcal{S} \times \mathcal{P}$ be a binary relation. An identification scheme is a triple of algorithms (*`KeyGen`*,*`P`*,*`V`*), where*

- `KeyGen` *takes as input a security parameter n and generates a pair $(sk, pk) \in \mathcal{R}$, publishes pk, and passes sk to* `P`
- `P` *is an interactive algorithm initialised with a pair $(sk, pk) \in \mathcal{R}$*
- `V` *is an interactive algorithm initialised with a statement $pk \in \mathcal{P}$. After the interaction,* `V` *outputs a decision 'Accept' or 'Reject'.*

The interaction of `P` *and* `V` *runs as follows:*

1. `P` *generates a random 'commitment' I from the space of all possible commitments $\mathcal{I}$ and sends it to* `V`

2. *Upon receipt of I,* V *chooses a 'challenge' c from the space of all possible challenges $\mathcal{C}$ at random and sends it to* P
3. P *responds with a 'response' p*
4. V *calculates an 'Accept' or 'Reject' response as a function of (I, c, p) and the statement pk.*

The interaction of P *and* V *is depicted in Figure 1.*

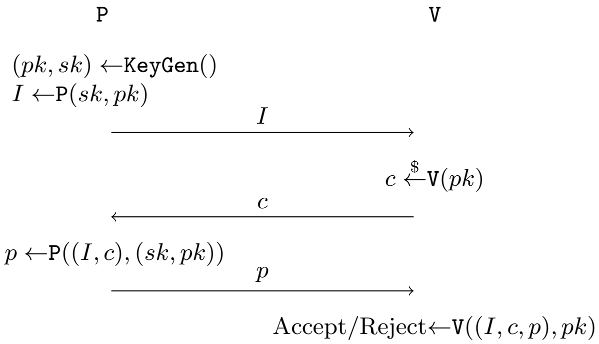

P
V
$(pk, sk) \leftarrow \mathtt{KeyGen}()$
$I \leftarrow \mathtt{P}(sk, pk)$
I
$c \overset{\$}{\leftarrow} \mathtt{V}(pk)$
c
$p \leftarrow \mathtt{P}((I, c), (sk, pk))$
p
Accept/Reject$\leftarrow \mathtt{V}((I, c, p), pk)$

Fig. 1. An identification scheme.

Definition 3. *Let (*KeyGen*,*P*,*V*) be an identification scheme. The triple (I, c, p) of exchanged values between* P *and* V *is called a 'transcript'; if a prover (resp. verifier) generates I, p (resp c) with the algorithm* P *(resp.* V*), they are called 'honest'. An identification scheme is 'complete' if a transcript generated by two honest parties is always accepted by the verifier.*

Turning our attention to the security of identification protocols, let us define the framework we wish to work with. As we will see later, it suffices for signature security to only consider identification schemes for which we have an honest verifier - in other words, it suffices to consider only a cheating prover. Let us do so in the form of the following attack games, which are [4, Attack Game 18.1] and [4, Attack Game 18.2] respectively.

Definition 4 (Direct Attack Game). *Let* ID*=(*KeyGen*,*P*,*V*) be an identification scheme and $\mathcal{A}$ be an adversary. Consider the following game:*

1. *The challenger obtains $(sk, pk) \leftarrow$*KeyGen *and passes pk to $\mathcal{A}$.*
2. *The adversary interacts with the challenger who generates responses with* V*. At the end, the challenger outputs 'Accept/Reject' as a function of the generated transcript and pk; the adversary wins the game if* V *outputs 'Accept'.*

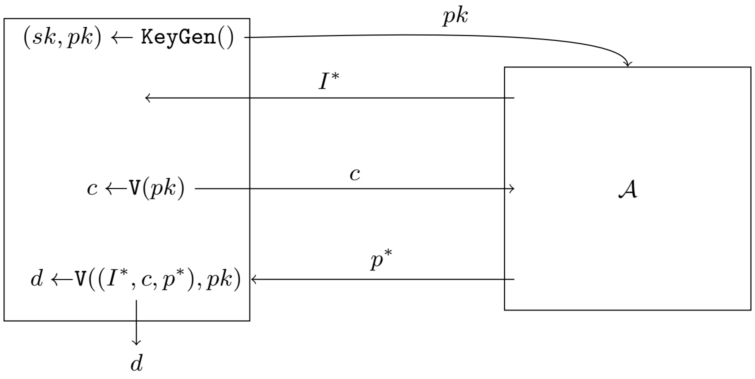

$(sk, pk) \leftarrow$ KeyGen()
pk
I^*
$c \leftarrow$V(pk)
c
$\mathcal{A}$
p^*
$d \leftarrow$V$((I^*, c, p^*), pk)$
d

Fig. 2. The direct attack game.

The Direct Attack game is depicted in Figure 2. We denote the advantage of the adversary in this game with `ID` *as the challenger by* `dir-adv`*($\mathcal{A}$,*`ID`*).*

Definition 5 (Eavesdropping Attack). *Let* `ID`*=(*`KeyGen`*,*`P`*,*`V`*) be an identification scheme and $\mathcal{A}$ be an adversary. Consider the following game:*

1. *The challenger obtains $(sk, pk) \leftarrow$*`KeyGen` *and passes pk to $\mathcal{A}$.*
2. *The adversary enters into an 'eavesdropping' phase, whereby they can request honestly-generated transcripts from a transcript oracle $\mathcal{T}$ possessing the same (sk, pk) pair generated in the previous step.*
3. *The adversary interacts with the challenger who generates responses with* `V`*. At the end, the challenger outputs 'Accept/Reject' as a function of the generated transcript and pk; the adversary wins the game if* `V` *outputs 'Accept'.*

The Eavesdropping Attack game is depicted in Figure 3. We denote the advantage of the adversary in this game with `ID` *as the challenger by* `eav-adv`*($\mathcal{A}$,*`ID`*).*

In practice, given a concrete identification scheme it is possible to bound the advantage of an adversary in these games provided one can prove the following two properties hold for the identification scheme:

Definition 6. *Let (*`KeyGen`*,*`P`*,*`V`*) be an identification scheme.*

- *The scheme has 'special soundness' if two transcripts with the same commitment and different challenges allow recovery of the witness sk; that is, if $(I, c, p), (I, c^*, p^*)$ are two transcripts generated with $(sk, pk) \leftarrow$*`KeyGen`*, there is an efficient algorithm taking these transcripts as input that returns sk.*
- *The scheme has 'special honest verifier zero knowledge' if, given a statement pk and a challenge c, there is an efficient algorithm to generate a passing transcript (I^*, c, p^*) with the same distribution as a legitimately generated transcript.*

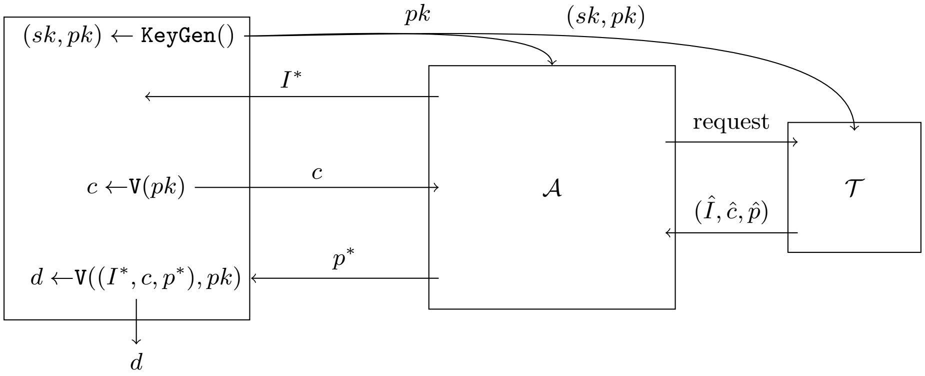

$(sk, pk) \leftarrow$ KeyGen()
pk
(sk, pk)
I^*
request
$c \leftarrow$V(pk)
c
$\mathcal{A}$
$\mathcal{T}$
$(\hat{I}, \hat{c}, \hat{p})$
p^*
$d \leftarrow$V$((I^*, c, p^*), pk)$
d

Fig. 3. The eavesdropping attack game.

Before moving on there is one final security notion to explore. Notice that if the underlying binary relation of an identification scheme is such that one can easily recover a valid witness from the public statement, an adversary can easily succeed in either of the above games simply by honestly generating the proof p with the appropriate value of sk. We have loosely discussed the notion that recovering a witness should therefore be difficult; it is nevertheless so far not clear how precisely this difficulty is accounted for. In fact, there are a number of ways to get round this. For our purposes, and in our application of the Fiat-Shamir transform, we will invoke the system outlined in [4, Section 19.6]. The idea is basically thus: provided the properties in Definition 6 hold, it is possible to set up the security proof such that all the difficulty of recovering a witness is 'priced in' to the key generation algorithm. Again, we will need a precise definition to make this rigorous later on: the following is [4, Attack Game 19.2]

Definition 7 (Inversion Attack Game). *Let* `KeyGen` *be a key generation algorithm for a binary relation* $\mathcal{R} \subset \mathcal{S} \times \mathcal{P}$ *and* $\mathcal{A}$ *be an adversary. Consider the following game:*

1. *A pair* (sk, pk) *is generated by running* `KeyGen`*, and the value* pk *is passed to the adversary* $\mathcal{A}$.
2. $\mathcal{A}$ *outputs some* $\hat{sk} \in \mathcal{S}$. *The adversary wins if* $(\hat{sk}, pk) \in \mathcal{R}$.

We denote the advantage of the adversary in this game with `kg` *as the challenger by* `inv-adv`*(*$\mathcal{A}$*,*`kg`*).*

1.3 Signature Schemes

Recall that a 'signature scheme' is a triple of algorithms (`KeyGen, Sg, Vf`), where `KeyGen()` outputs a private-public key pair (sk, pk) upon input of a security parameter. For some space of messages $\mathcal{M}$, `Sg` takes as input sk and some $m \in \mathcal{M}$, producing a 'signature' σ. `Vf` takes as input pk and a pair (m, σ), and outputs either 'Accept' or 'Reject'. We have the obvious correctness requirement that for a key pair (sk, pk) generated by `KeyGen` we can expect, for any $m \in \mathcal{M}$, that one has

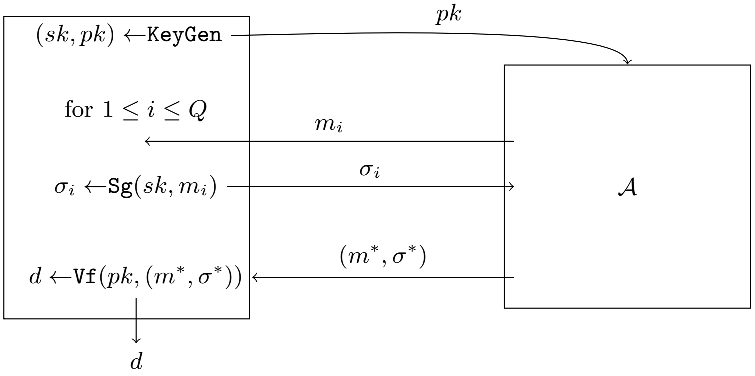
$$\mathtt{Vf}(pk, (m, \mathtt{Sg}(sk, m))) = \text{Accept}$$

The security of a signature scheme is defined with respect to the following attack game, which is [4, Attack Game 13.1] (but is widely available).

Definition 8 (Chosen Message Attack). *Let* S=(KeyGen,Sg,Vf) *be a signature scheme and $\mathcal{A}$ be an adversary. Consider the following game:*

1. *The challenger obtains $(sk, pk) \leftarrow$*KeyGen *and passes pk to $\mathcal{A}$.*
2. *The adversary enters into an 'querying' phase, whereby they can obtain signatures $\sigma_i = \mathtt{Sg}(sk, m_i)$ from the challenger, for the adversary's choice of message m_i. The total number of messages queried is denoted Q.*
3. *The adversary submits their attempted forgery - a message-signature pair (m^*, σ^*) - to the challenger. The challenger outputs $\mathtt{Vf}(pk, (m^*, \sigma^*))$; the adversary wins if this output is 'Accept'.*

The Chosen Message Attack game is depicted in Figure 4. We denote the advantage of the adversary in this game with S *as the challenger by* cma-adv*($\mathcal{A}$,*S*).*



$(sk, pk) \leftarrow$KeyGen
pk
for $1 \leq i \leq Q$
m_i
$\sigma_i \leftarrow \mathtt{Sg}(sk, m_i)$
σ_i
$\mathcal{A}$
(m^*, σ^*)
$d \leftarrow \mathtt{Vf}(pk, (m^*, \sigma^*))$
d

Fig. 4. The chosen message attack game.

A signature scheme S for which cma-adv($\mathcal{A}$,S) is bounded favourably[8] from above for any efficient adversary $\mathcal{A}$ is sometimes called euf-cma secure, or 'existentially unforgeable under chosen message attacks'.

It remains to briefly define the well-known notion of the Fiat-Shamir transform, initially presented in [14]:

[8] 'Favourably' here usually means as a negligible function of a security parameter.

Definition 9 (Fiat-Shamir). *Let* ID*=(*KeyGen*,*P*,*V*) be an identity scheme with commitment space $\mathcal{I}$ and $\mathcal{C}$. We define a signature scheme FS(*ID*)=(*KeyGen*,*Sg*,*Vf*) on the message space $\mathcal{M}$ given access to a public function $H : \mathcal{M} \times \mathcal{I} \to \mathcal{C}$:*

1. KeyGen *is exactly the key generation algorithm of* ID *and outputs a pair (sk, pk), where pk is made public*
2. Sg *takes as input $m \in \mathcal{M}$ and the key pair (pk, sk) and outputs a signature (σ_1, σ_2):*
   ```
   $I \leftarrow$P$((sk, pk))$
   $c \leftarrow H(m, I)$
   $p \leftarrow$P$((I, c), (sk, pk))$
   $(\sigma_1, \sigma_2) \leftarrow (I, p)$
   return $(\sigma_1, \sigma_2)$
   ```
3. Vf *takes as input a message-signature pair $(m, (\sigma_1, \sigma_2))$ and outputs a decision d, which is 'Accept' or 'Reject':*
   ```
   $c \leftarrow H(I, \sigma_1)$
   $d \leftarrow$V$((\sigma_1, c, \sigma_2), pk)$
   return $d$
   ```

Intuitively, we can see that Sg is simulating an interactive protocol non-interactively with a call to the function H; in order to inherit the security properties of the identification scheme, this function H should have randomly distributed outputs on fresh queries and should be computationally binding - that is, it should be difficult to find a value $I' \neq I$ such that $H(m, I) = H(m, I')$; and given a commitment $c \in \mathcal{C}$ it should be difficult to find a message m and commitment $I \in \mathcal{I}$ such that $H(m, I) = c$. On the other hand, for correctness we need H to be deterministic on previously queried inputs. Such a function is modelled by a hash function under the random oracle model: in this model, it was famously demonstrated in [1] that a relatively modest security notion for the underlying identification scheme gives strong security proofs for the resulting signature scheme. In our own security proof we use the slightly more textbook exposition presented in [4].

2 A Novel Connection to a Group Action

Our first task is to demonstrate the existence of the claimed group action, for any finite group. A very similar structure was outlined in [2] - with the important distinction that *semi*groups are insisted upon. Indeed, it turns out that allowing invertibility changes the structure in a way that we shall outline below.

Definition 10. *Let G be a finite group, and $\Phi \leq Aut(G)$. Fix some $(g, \phi) \in G \ltimes \Phi$. For any $x \in \mathbb{Z}$, the function $s_{g,\phi} : \mathbb{Z} \to G$ is defined as the group element such that*

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

The group action of interest arises from the study of the set $\{s_{g,\phi}(i) : i \in \mathbb{Z}\}$. Certainly $1 \in \{s_{g,\phi}(i) : i \in \mathbb{Z}\}$, since there is some $n \in \mathbb{N}$ such that $(s_{g,\phi}(n), \phi^n) = (g, \phi)^n = (1, id)$, but one cannot immediately deduce that this is the smallest integer for which $s_{g,\phi}$ is 1. Indeed, even if the order n of (g, ϕ) is the smallest integer such that $s_{g,\phi}(n) = 1$, we are not necessarily guaranteed that every integer up to n is mapped to a distinct elements of G by $s_{g,\phi}$. Before resolving these questions let us introduce some terminology.

Definition 11. *Let G be a finite group, and $\Phi \leq Aut(G)$. Fix some $(g, \phi) \in G \ltimes \Phi$. The set*

$$\mathcal{X}_{g,\phi} := \{s_{g,\phi}(i) : i \in \mathbb{Z}\}$$

is called the cycle of (g, ϕ), and its size is called the period of (g, ϕ).

In the interest of brevity we will also assume henceforth that by (g, ϕ) we mean some pair occurring in a semidirect product group as described above. For any such pair (g, ϕ), note that $\mathcal{X}_{g,\phi}$ is not necessarily closed under the group operation - we can, nevertheless, implement addition in the argument of $s_{g,\phi}$ as follows:

Theorem 1. *Let $i, j \in \mathbb{Z}$ and suppose $(g, \phi) \in G \ltimes \Phi$ in the usual way. One has that*

$$\phi^j(s_{g,\phi}(i))s_{g,\phi}(j) = s_{g,\phi}(i + j)$$

Proof. Following the definitions one has

$$\begin{aligned}(s_{g,\phi}(i + j), \phi^{i+j}) &= (g, \phi)^{i+j} \\ &= (g, \phi)^i (g, \phi)^j \\ &= (s_{g,\phi}(i), \phi^i)(s_{g,\phi}(j), \phi^j) \\ &= (\phi^j(s_{g,\phi}(i))s_{g,\phi}(j), \phi^{i+j})\end{aligned}$$

□

Put another way, we can use integers to map $\mathcal{X}_{g,\phi}$ to itself. This idea is sufficiently important to earn its own notation:

Definition 12. *Let $i \in \mathbb{Z}$. The function $* : \mathbb{Z} \times \mathcal{X}_{g,\phi} \to \mathcal{X}_{g,\phi}$ is given by*

$$i * s_{g,\phi}(j) := \phi^j(s_{g,\phi}(i))s_{g,\phi}(j)$$

We have seen that $i * s_{g,\phi}(j) = s_{g,\phi}(i + j)$; accordingly, we pronounce the $*$ symbol as 'step'. An immediate consequence is the presence of some degree of 'looping' behaviour; that is, supposing $s_{g,\phi}(n) = 1$ for some $n \in \mathbb{Z}$, one has

$$\begin{aligned}s_{g,\phi}(n + 1) = 1 * s_{g,\phi}(n) &= 1 * 1 \\ &= \phi(1)s_{g,\phi}(1) \\ &= s_{g,\phi}(1)\end{aligned}$$

Generalising this idea we get a more complete picture of the structure of the cycle.

Theorem 2. *Let G be a finite group and $\Phi \leq Aut(G)$ an automorphism subgroup. Fix $(g, \phi) \in G \ltimes Aut(G)$, and let n be the smallest positive integer for which $s_{g,\phi}(n) = 1$. One has that $|\mathcal{X}_{g,\phi}| = n$, and*

$$\mathcal{X}_{g,\phi} = \{1, g, ..., s_{g,\phi}(n-1)\}$$

Proof. First, let us demonstrate that the values $1 = s_{g,\phi}(0), s_{g,\phi}(1), ..., s_{g,\phi}(n-1)$ are all distinct. Suppose to the contrary that there exists $0 \leq i < j \leq n-1$ such that $s_{g,\phi}(i) = s_{g,\phi}(j)$; then some positive $k < n$ must be such that $i + k = j$. In other words:

$$\begin{aligned} i * s_{g,\phi}(k) = s_{g,\phi}(j) &\Rightarrow \phi^i(s_{g,\phi}(k))s_{g,\phi}(i) = s_{g,\phi}(j) \\ &\Rightarrow \phi^i(s_{g,\phi}(k)) = 1 \\ &\Rightarrow s_{g,\phi}(k) = 1 \end{aligned}$$

which is a contradiction, since $k < n$. It remains to show that every integer is mapped by $s_{g,\phi}$ to one of these n distinct values - but this is trivial, since we can write any integer i as $kn + j$ for some integer k and $0 \leq j < n$. It follows that

$$s_{g,\phi}(i) = s_{g,\phi}(j)$$

where $s_{g,\phi}(j)$ is one of the n distinct values. □

It follows that we can write $i * s_{g,\phi}(j) = s_{g,\phi}(i + j \mod n)$. In fact, the latter part of the above argument demonstrates something slightly stronger: not only is every integer mapped to one of n distinct values by $s_{g,\phi}$, but every member of a distinct residue class modulo n is mapped to the *same* distinct value. It is this basic idea that gives us our group action.

Theorem 3. *Let G be a finite group and $\Phi \leq Aut(G)$. Fix a pair $(g, \phi) \in G \ltimes Aut(G)$, and let n be the smallest positive integer such that $s_{g,\phi}(n) = 1$. Define the function as*

$$\begin{aligned} \circledast : \quad & \mathbb{Z}_n \times \mathcal{X}_{g,\phi} \to \mathcal{X}_{g,\phi} \\ & [i]_n \circledast s_{g,\phi}(j) = i * s_{g,\phi}(j) \end{aligned}$$

The tuple $(\mathbb{Z}_n, \mathcal{X}_{g,\phi}, \circledast)$ is a free, transitive group action.

Proof. First, let us see that $\circledast$ is well-defined. Suppose $i \cong j \mod n$, then $i = j + kn$ for some $k \in \mathbb{Z}$. For some arbitrary $\mathcal{X}_{g,\phi}$, say $s_{g,\phi}(l)$ for $0 \leq l < n$, one has

$$\begin{aligned} i * s_{g,\phi}(l) &= (j + kn) * s_{g,\phi}(l) \\ &= j * s_{g,\phi}(l + kn) \\ &= j * s_{g,\phi}(l) \end{aligned}$$

We also need to verify that the claimed tuple is indeed a group action. In order to check that the identity in $\mathbb{Z}_n$ fixes each $\mathcal{X}_{g,\phi}$, by the well-definedness just demonstrated, it suffices to check that $0 * s_{g,\phi}(l) = s_{g,\phi}(l)$ for each $0 \leq l < n$ - which indeed is the case. For the compatibility of the action with modular addition, note that for $0 \leq i, j, k < n - 1$ one has

$$\begin{aligned} [k]_n \circledast ([j]_n \circledast s_{g,\phi}(i)) &= [k]_n \circledast s_{g,\phi}(i + j \mod n) \\ &= s_{g,\phi}(i + j + k \mod n) \\ &= [j + k]_n \circledast s_{g,\phi}(i) \end{aligned}$$

as required. It remains to check that the action is free and transitive. First, suppose $[i]_n \in \mathbb{Z}_n$ fixes each $s_{g,\phi}(j) \in \mathcal{X}_{g,\phi}$. By the above we can assume without loss of generality that $0 \leq i < n - 1$, and we have $\phi^j(s_{g,\phi}(i))s_{g,\phi}(j) = s_{g,\phi}(j)$. It follows that $s_{g,\phi}(i) = 1$, so we must have $i = 0$ as required. For transitivity, for any pair $s_{g,\phi}(i), s_{g,\phi}(j)$ we have $[j - i]_n \circledast s_{g,\phi}(i) = s_{g,\phi}(j)$, and we are done. □

Recalling that the set $\mathcal{X}_{g,\phi}$ and the period n are a function of the pair (g, ϕ), we have actually shown the existence of a large family of group actions. Nevertheless, we have only really shown the existence of the crucial parameter n - it is not necessarily clear how this value should be calculated. With this in mind let us conclude the section with a step in this direction:

Theorem 4. *Fix a pair $(g, \phi) \in G \ltimes Aut(G)$. Let n be the smallest integer such that $s_{g,\phi}(n) = 1$, then n divides the order of the pair (g, ϕ) as a group element in $G \ltimes Aut(G)$.*

Proof. Suppose $m = ord((g, \phi))$. Certainly $s_{g,\phi}(m) = 1$, and by definition one has $m \geq n$. We can therefore write $m = kn + l$, for $k \in \mathbb{N}$ and $0 \leq l < n$. It is not too difficult to verify that $s_{g,\phi}(x) = \phi^{x-1}(g)...\phi(g)g$ for any $x \in \mathbb{N}$. It follows that

$$s_{g,\phi}(m) = \phi^{kn}(s_{g,\phi}(l))\phi^{(k-1)n}(s_{g,\phi}(n))...\phi^n(s_{g,\phi}(n))s_{g,\phi}(n)$$

Since $s_{g,\phi}(m) = s_{g,\phi}(n) = 1$, we must have $s_{g,\phi}(l) = 1$. But $l < n$ and so $l = 0$ by the minimality of n, which in turn implies that $n|m$ as required. □

2.1 Semidirect Discrete Logarithm Problem

Given a group G and a pair $(g, \phi) \in G \ltimes Aut(G)$, observe that as a consequence of Theorem 1 and Definition 12, for any two integers $i, j \in \mathbb{N}$ we have that $s_{g,\phi}(i + j) = j * s_{g,\phi}(i) = i * s_{g,\phi}(j)$. A Diffie-Hellman style key exchange immediately follows[9]; indeed, a key exchange based on this idea first appears in [15], and is known as Semidirect Product Key Exchange. In the same way that the security of Diffie-Hellman key exchange is related to the security of the Discrete Logarithm Problem, to understand the security of Semidirect Product Key Exchange we should like to study the difficulty of the following task:

[9] Historically speaking, the key exchange predates the more abstract treatment in this paper.

Definition 13 (Semidirect Discrete Logarithm Problem). *Let G be a finite group, and let $(g, \phi) \in G \times Aut(G)$. Suppose, for some $x \in \mathbb{N}$, that one is given $(g, \phi), s_{g,\phi}(x)$; the Semidirect Discrete Logarithm Problem (SDLP) with respect to (g, ϕ) is to recover the integer x.*

The complexity of SDLP is relatively well understood, in large part due to the connection with group actions highlighted above. We will see later on that the security game advantages for our identification and signature schemes can be bounded in terms of the advantage of an adversary in solving SDLP; indeed, for the SDLP attack game defined in the obvious way, we write the advantage of an adversary $\texttt{sdlp-adv}(\mathcal{A},(g, \phi))$.

Before we move on to study the signature schemes resulting from each group action we note that the convention in the area is to restrict a finite group G to be a finite, *non-abelian* group G. This was in part to preclude a trivial attack on the related key exchange for a specific choice of ϕ - nevertheless, throughout the rest of the paper we adopt this convention.

3 `SPDH-Sign`

3.1 An Identification Scheme

Recall that our strategy is to set up an honest-verifier identification scheme, to which we can apply the well-known Fiat-Shamir heuristic and obtain strong security guarantees in the ROM. The central idea of this identification scheme is as follows: suppose we wish to prove knowledge of some secret $\mathbb{Z}_n$ element, say $[s]_n$. We can select an arbitrary element of $\mathcal{X}_{g,\phi}$, say X_0, and publish the pair $X_0, X_1 := [s]_n \circledast X_0$. An honest party wishing to verify our knowledge of the secret $[s]_n$ might invite us to commit to some group element $[t]_n$, for $[t]_n$ sampled uniformly at random from $\mathbb{Z}_n$. We can do this by sending the element $I = [t]_n \circledast X_0$ - note that as a consequence of the free and transitive properties, $[t]_n$ is the unique group element such that $I = [t]_n \circledast X_0$. However, with our knowledge of the secret $[s]_n$ and the commitment $[t]_n$, we can calculate the element $[p]_n = [t - s]_n$ such that $[p]_n \circledast X_1 = I$, where this equation holds by the group action axioms: one has $[t - s]_n \circledast ([s]_n \circledast X_0) = [t]_n \circledast X_0 = I$.

$$\begin{tikzcd}
X_0 \arrow[r, "{[t]_n \circledast}"] \arrow[d, "{[s]_n \circledast}"'] & I \\
X_1 \arrow[ur, dashed, "{[t - s]_n \circledast}"'] &
\end{tikzcd}$$

Fig. 5. Paths to the commitment.

Interpreted graph-theoretically (as depicted in Figure 5), an honest verifier can ask to see one of two paths to the commitment value. Consider a dishonest party attempting to convince the verifier that they possess the secret $[s]_n$. In attempting to impersonate the honest prover, our dishonest party can generate their own value of $[t]_n$, and so can certainly provide the correct path in one of the two scenarios. Assuming, however, that recovering the appropriate group element is difficult, without knowledge of the secret $[s]_n$ this party succeeds in their deception with low probability.

This intuition gives us the following non-rigorous argument of security in the framework described in Section 1.2. First, recall that we are in the honest verifier scenario, and so a challenge bit c will be 0 with probability 1/2, in which case a cheating prover succeeds with probability 1. Supposing that ε is the probability of successfully recovering the value $[t-s]_n$, it follows that a cheating prover succeeds with probability $(1+\varepsilon)/2$ - that is, with probability larger than 1/2. We can quite easily counter this by requiring that N instances are run at the same time. In this case, if N zeroes are selected the prover wins with probability 1 by revealing their dishonestly generated values of $[t]_n$ - otherwise, they are required to recover at least 1 value of $[t-s]_n$. Assuming for simplicity that the probability of doing so remains consistent regardless of the number of times such a value is to be recovered, since the honest verifier selects their challenges uniformly at random the cheating prover succeeds with probability

$$\frac{1}{2^N} + \sum_{i=1}^{2^N-1} \frac{\varepsilon}{2^N} = \frac{1}{2^N} + \varepsilon\frac{2^N-1}{2^N}$$

which tends to ε as $N \to \infty$.

The actual proof of security operates within the security games defined in the preliminaries. As a step towards this formalisation, we need to specify the binary relation our identification scheme is based on. Choose some finite non-abelian group G: given a fixed pair $(g,\phi) \in G \times Aut(G)$ we are interested, by Theorem 3, in a subset $\mathcal{R}$ of $\mathbb{Z}_n, \mathcal{X}_{g,\phi}$, where n is the smallest integer such that $s_{g,\phi}(n) = 1$. In fact, legislating for N parallel executions of the proof of knowledge, to each tuple $(X_1, ..., X_N)$ is associated a binary relation

$$\mathcal{R} \subset \mathbb{Z}_n^N \times \mathcal{X}_{g,\phi}^N$$

where $(([s_1]_n, ..., [s_N]_n), (Y_1, ..., Y_N)) \in \mathcal{R}$ exactly when $(Y_1, ..., Y_N) = ([s_1]_n * X_1, ..., [s_N]_n * X_N)$.

With all this in mind let us define our identification scheme. The more rigorous presentation should not distract from the intuition that we describe N parallel executions of the game in Figure 5.

Protocol 1. Let G be a finite non-abelian group and $(g,\phi) \in G \ltimes Aut(G)$. Suppose also that $n \in \mathbb{N}$ is the smallest integer such that $s_{g,\phi}(n) = 1$. The identification scheme $\texttt{SPDH-ID}_{g,\phi}(N)$ is a triple of algorithms

$$(\texttt{KeyGen}_{g,\phi}, \texttt{P}_{g,\phi}, \texttt{V}_{g,\phi})$$

such that

1. $\texttt{KeyGen}_{g,\phi}$ takes as input some $N \in \mathbb{N}$.
   $(X_1, ..., X_N) \leftarrow \mathcal{X}^N_{g,\phi}$
   $([s_1]_n, ..., [s_N]_n) \leftarrow \mathbb{Z}^N_n$
   $(Y_1, ..., Y_N) \leftarrow ([s_1]_n \circledast X_1, ..., [s_N]_n \circledast X_N)$
   $\texttt{KeyGen}_{g,\phi}$ outputs the public key $((X_1, ..., X_N), (Y_1, ..., Y_N))$ and passes the secret key $([s_1]_n, ..., [s_N]_n)$ to the prover $\texttt{P}_{g,\phi}$. The public key and the value of N used is published.
2. $\texttt{P}_{g,\phi}$ and $\texttt{V}_{g,\phi}$ are interactive algorithms that work as depicted in Figure 6:

Security In this section we demonstrate that `SPDH-ID` is secure against eavesdropping attacks in the following sense: the advantage of an adversary in the eavesdropping attack game can be bounded by that of the adversary in the SDLP game. First, let us check that the desirable properties of an identification scheme hold:

Theorem 5. `SPDH-ID` *has the following properties:*

1. *Completeness*
2. *Special soundness*
3. *Special honest-verifier zero knowledge.*

Proof. Note that in order to prove each of these properties on the N-tuples comprising the transcripts generated by `SPDH-ID`, we need to prove that the properties hold for each component of the tuple; but since each component is independent of all the others, it suffices to demonstrate the stated properties for a single arbitrary component. In other words, we show that the stated properties hold when $N = 1$, and the general case immediately follows.

1. If $b = 0$ then $[p]_n = [t]_n$, and trivially we are done. If $b = 1$ then $[p]_n = [t-s]_n$; doing the bookkeeping we get that
$$\begin{aligned} [p]_n \circledast S_1 &= [p]_n \circledast ([s]_n \circledast S_0) \\ &= ([t-s]_n[s]_n) \circledast S_0 \\ &= ([s]_n \circledast S_0) = I \end{aligned}$$
2. Two passing transcripts with the same commitment are $(I, 0, [t]_n)$ and $(I, 1, [t-s]_n)$. Labelling the two responses x^{p_1}, x^{p_2}, we recover the secret as $(x^{p_2})^{-1}(x^{p_1})$.
3. It suffices to show that one can produce passing transcripts with the same distribution as legitimate transcripts, but without knowledge of $[s]_n$. We have already discussed how to produce these transcripts; if a simulator samples $[t]_n$ uniformly at random, then the transcript $([t]_n \circledast S_b, b, [t]_n)$ is valid regardless of the value of b. Moreover, if $b = 0$, trivially the transcripts have the same distribution; if $b = 1$, since $[s]_n$ is fixed and $[t]_n$ is sampled uniformly at random, the distribution of a legitimate passing transcript is also uniformly random.

```
$\mathtt{P}_{g,\phi}$                                        $\mathtt{V}_{g,\phi}$

for $i \leftarrow 1, N$ do
    $[t_i]_n \xleftarrow{\$} \mathbb{Z}_n$
    $I_i \leftarrow [t_i]_n \circledast X_i$
end for
$I \leftarrow (I_1, ..., I_N)$
                         $\xrightarrow{I}$
                                        for $i \leftarrow 1, N$ do
                                            $c_i \xleftarrow{\$} \{0, 1\}$
                                        end for
                                        $c \leftarrow (c_1, ..., c_N)$
                         $\xleftarrow{c}$
for $i \leftarrow 1, N$ do
    if $c_i = 0$ then
        $[p_i]_n \leftarrow [t_i]_n$
    else
        $[p_i]_n \leftarrow [t_i - s_i]_n$
    end if
end for
$p \leftarrow ([p_1]_n, ..., [p_N]_n)$
                         $\xrightarrow{p}$
                                        for $i \leftarrow 1, N$ do
                                            if $c_i = 0$ then
                                                $V_i \leftarrow [p_i]_n \circledast X_i$
                                            else
                                                $V_i \leftarrow [p_i]_n \circledast Y_i$
                                            end if
                                        end for
                                        $V \leftarrow (V_1, ..., V_N)$
                                        $d \leftarrow I \overset{?}{=} V$
                                        return $d$
```

Fig. 6. `SPDH-ID`

□

We are now ready to bound on the security of our identification scheme.

Theorem 6. *Let G be a finite abelian group and let $(g, \phi) \in G \ltimes Aut(G)$. For some $N \in \mathbb{N}$, consider the identification scheme* $\texttt{SPDH-ID}_{g,\phi}(N)$ *and an efficient adversary $\mathcal{A}$. There exists an efficient adversary $\mathcal{B}$ with $\mathcal{A}$ as a subroutine, such that with* $\varepsilon = \texttt{sdlp-adv}(\mathcal{B}, (g, \phi))$, *we have*

$$\texttt{eav-adv}(\mathcal{A}, \texttt{SPDH-ID}_{g,\phi}(N)) \leq \sqrt{\varepsilon} + \frac{1}{2^N}$$

Proof. This is just a straightforward application of two results in [4]. By [4, Theorem 19.14], since $\texttt{SPDH-ID}_{g,\phi}(N)$ has honest verifier zero knowledge, there exists an efficient adversary $\mathcal{B}'$ with $\mathcal{A}$ as a subroutine such that

$$\texttt{eav-adv}(\mathcal{A}, \texttt{SPDH-ID}_{g,\phi}(N)) = \texttt{dir-adv}(\mathcal{B}', \texttt{SPDH-ID}_{g,\phi}(N))$$

Moreover, let

$$\delta = \texttt{inv-adv}(\mathcal{B}', \texttt{KeyGen}_{g,\phi})$$

Since $\texttt{SPDH-ID}_{g,\phi}(N)$ has special soundness, [4, Theorem 19.13] gives

$$\texttt{dir-adv}(\mathcal{B}, \texttt{SPDH-ID}_{g,\phi}(N)) \leq \sqrt{\delta} + \frac{1}{M}$$

where M is the size of the challenge space. It is easy to see that $M = 2^N$; it remains to relate the quantities ε and δ. We do so eschewing some of the detail since the argument is straightforward; note that by definition of the binary relation underpinning $\texttt{KeyGen}_{g,\phi}$, we can think of the inversion attack game as a security game in which one solves N independent SDLP instances in parallel. Call the advantage in this game $\texttt{N-sdlp-adv}(\mathcal{B}', (g, \phi))$, and suppose an adversary $\mathcal{B}$ in the standard SDLP attack game runs $\mathcal{B}'$ as an adversary. $\mathcal{B}$ can simply provide $\mathcal{B}'$ with N copies of its challenge SDLP instance, and succeeds whenever $\mathcal{B}'$ does. It follows that $\delta \leq \varepsilon$, and we are done. □

3.2 A Digital Signature Scheme

It remains now to apply the Fiat-Shamir transform to our identification scheme. Doing so yields the signature scheme claimed in the title of this paper.

Protocol 2 ($\texttt{SPDH-Sign}$)**.** Let G be a finite non-abelian group and let $(g, \phi) \in G \times Aut(G)$ be such that n is the smallest integer for which $s_{g,\phi}(n)$=1. For any $N \in \mathbb{N}$ and message space $\mathcal{M}$, suppose we are provided a hash function $H : \mathcal{X}_{g,\phi}^N \times \mathcal{M} \to \{0, 1\}^N$. We define the signature scheme

$$\texttt{SPDH-Sign}_{g,\phi}(N) = (\texttt{KeyGen}, \texttt{Sg}, \texttt{Vf})$$

as in Figure 7.

```
KeyGen($N$):
    for $i \leftarrow 1, N$ do
        $X_i \xleftarrow{\$} \mathcal{X}_{g,\phi}$
        $[s_i]_n \xleftarrow{\$} \mathbb{Z}_n$
        $Y_i \leftarrow [s_i]_n \circledast X_i$
    end for
    $sk \leftarrow ([s_1]_n, ..., [s_N]_n)$
    $pk \leftarrow ((X_1, ..., X_N), (Y_1, ..., Y_N))$
    return $(sk, pk)$
```

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

```
Vf($m, (\sigma_1, \sigma_2), pk$):
    $c \leftarrow H(\sigma_1, m)$
    for $i \leftarrow 1, N$ do
        if $c_i = 0$ then
            $V_i \leftarrow p_i \circledast X_i$
        else
            $V_i \leftarrow p_i \circledast Y_i$
        end if
    end for
    $V \leftarrow (V_1, ..., V_N)$
    $d \leftarrow V \overset{?}{=} I$
    return $d$
```

Fig. 7. SPDH-Sign

It is easy to see that given the identification scheme $\texttt{SPDH-ID}_{g,\phi}(N)$, the signature scheme $\texttt{SPDH-Sign}_{g,\phi}(N)$ is exactly FS($\texttt{SPDH-ID}_{g,\phi}(N)$). Before we can use this fact to prove the security of the signature, we require that the hash function gives outputs distributed at 'random', in some sense. This is accounted for by the 'Random Oracle Model': every time we wish to compute the hash function H, we suppose that an oracle function of the appropriate dimension selected at random is queried. Any party can query the random oracle at any time, and the number of these queries is kept track of. We also note that we do not in this paper account for the quantum-accessible random oracle model required for post-quantum security - equivalent security proofs in the quantum-accessible random oracle model are provided, for example, in [3].

With this heuristic in place we can prove the security of our signature scheme relative to SDLP with a simple application of [4, Theorem 19.15] and its corollaries:

Theorem 7. *Let G be a finite non-abelian group; $(g, \phi) \in G \ltimes Aut(G)$; and $n \in \mathbb{N}$ be the smallest integer such that $s_{g,\phi}(n) = 1$. Consider the chosen message attack game in the random oracle model, where Q_s is the number of signing queries made and Q_{ro} is the number of random oracle queries. For any efficient adversary $\mathcal{A}$ and $N \in \mathbb{N}$, there exists an efficient adversary $\mathcal{B}$ running $\mathcal{A}$ as a subroutine such that the signature scheme $\texttt{SPDH-Sign}_{g,\phi}(N)$ has*

$$\delta \leq \frac{Q_s}{n}(Q_s + Q_{ro} + 1) + \frac{Q_{ro}}{2^N} + \sqrt{(Q_{ro} + 1)\texttt{sdlp-adv}(\mathcal{B}, (g, \phi))}$$

where $\delta = \texttt{cma-adv}^{\texttt{ro}}(\texttt{SPDH-Sign}_{g,\phi}(N), \mathcal{A})$ is the advantage of the signature scheme in the random oracle model version of the chosen message attack game.

Proof. Applying [4, Theorem 19.15] and [4, Equation 19.21], since the underlying identification scheme has honest verifier zero knowledge there is an efficient adversary $\mathcal{B}'$ running $\mathcal{A}$ as a subroutine such that

$$\delta \leq \gamma Q_s(Q_s + Q_{ro} + 1) + \frac{Q_{ro}}{|\mathcal{C}|} + \sqrt{(Q_{ro}+1)\texttt{inv-adv}(\mathcal{B}, \texttt{KeyGen}_{g,\phi})}$$

where γ is the probability that a given commitment value appears in a transcript, and $\texttt{KeyGen}_{g,\phi}$ is the key generation algorithm of the underlying identification scheme. Since choosing a random group element corresponds to choosing a random element of $\mathcal{X}_{g,\phi}$, each commitment value in $\mathcal{X}_{g,\phi}$ has probability $1/|\mathcal{X}_{g,\phi}| = 1/n$ of being selected. We have already seen in the proof of Theorem 6 that the advantage of an adversary in the inversion attack game against this key generation algorithm is bounded by the advantage in an SDLP attack game, and the result follows. □

The above theorem provides a concrete estimate on the advantage of an adversary in the chosen message attack game; nevertheless, a plain English rephrasing is a useful reflection on these results. Essentially, we now know that the **`euf-cma`** security of our signature scheme is reliant on the integer n corresponding to the pair (g, ϕ), the size of N, and the difficulty of SDLP relative to the pair (g, ϕ). We can discount the reliance on N, which can be 'artificially' inflated as we please; note also that we can intuitively expect the size of n and the difficulty of SDLP for (g, ϕ) to be at least somewhat correlated, since a small value of n trivially renders the associated SDLP instance easy by brute force. In essence, then, we have shown that we can expect the signature scheme corresponding to (g, ϕ) to be secure provided the associated SDLP instance is difficult.

4 On the Difficulty of SDLP

For any finite non-abelian group G, we have shown the existence of signature scheme for any pair $(g, \phi) \in G \times Aut(G)$. It is now clear from Theorem 7 that if the signature is defined with respect to a pair (g, ϕ), SDLP with respect to (g, ϕ) should be difficult. In this section we discuss sensible choices of G with respect to this criterion.

As alluded to in the title of this paper we are interested in post-quantum hard instances of SDLP; that is, if an instance of SDLP has a known reduction to a quantum-vulnerable problem we should consider it to be easy.

There are three key strategies in the literature for addressing SDLP. Two of them, at face value, appear to solve a problem instead related to SDLP: let us explore the gap between the problems below.

4.1 Dihedral Hidden Subgroup Problem

It should first be noted that, as with all group action-based cryptography, the Dihredral Hidden Subgroup Problem will be highly relevant. Indeed, we can bound the complexity of SDLP above by appealing to Kuperberg's celebrated quantum algorithm for the Abelian Hidden Shift Problem [21], defined as follows:

Definition 14. *Let A be an abelian group and S be a set. Consider two injective functions $f, g : A \to S$ such that for some $h \in A$, we have $f(a) = g(a + h)$ for all $a \in A$. We say that the functions f, g 'hide' h, and the Abelian Hidden Shift Problem is to recover h via queries to f, g.*

Adapting an argument seen throughout the literature, but first codified in its modern sense in [7], gives us the following result.

Theorem 8. *Let G be a finite non-abelian group and let $(g, \phi) \in G \ltimes Aut(G)$; and $n \in \mathbb{N}$ be the smallest integer such that $s_{g,\phi}(n) = 1$. Given (g, ϕ) and a group element $s_{g,\phi}(x)$, there is a quantum algorithm that recovers x in time $2^{\mathcal{O}(\sqrt{\log n})}$.*

Proof. If the relevant abelian group has size n we have the claimed complexity for an abelian hidden shift problem by [21, Proposition 6.1]. It suffices to show that one can solve SDLP provided one can solve the abelian hidden shift problem - the argument goes as follows. Define $f, g : \mathbb{Z}_n \to \mathcal{X}_{g,\phi}$ by

$$f([z]_n) = [z]_n \circledast s_{g,\phi}(x) \quad g([z]_n) = [z]_n \circledast s_{g,\phi}(1)$$

We have for all $[z]_n \in \mathbb{Z}_n$ that

$$\begin{aligned} f([z]_n) &= [z]_n \circledast s_{g,\phi}(x) \\ &= [z]_n \circledast ([x-1]_n \circledast s_{g,\phi}(1)) \\ &= ([z]_n + [x-1]_n) \circledast s_{g,\phi}(1) \\ &= g([z]_n + [x-1]_n) \circledast s_{g,\phi}(1) \end{aligned}$$

so f and g hide $[x-1]_n$, from which $x \in \mathbb{N}$ can be recovered trivially. □

A small amount of detail is suppressed in the above proof: namely, that we have tacitly assumed knowledge of the quantity n. Since the best algorithm for the abelian hidden shift problem is quantum anyway, we need not be reticent to compute n with a quantum algorithm - and since the function $s_{g,\phi}$ is periodic in n, certainly such Shor-like techniques are available, such as [8, Algorithm 5]. On the other hand, the ability to compute n efficiently and classically is both desirable and addressed later in this paper.

4.2 Semidirect Computational Diffie-Hellman

The other major body of work related to the analysis of SDLP addresses the following related problem:

Definition 15 (Semidirect Computational Diffie-Hellman). *Let G be a finite abelian group, and let $(g, \phi) \in G \ltimes Aut(G)$. Let $x, y \in \mathbb{N}$ and suppose we are given the data $(g, \phi), s_{g,\phi}(x)$ and $s_{g,\phi}(y)$. The Semidirect Computational Diffie-Hellman problem (SCDH) is to compute the value $s_{g,\phi}(x + y)$.*

Recall our discussion of Semidirect Product Key Exchange in Section 2.1. Notice that SCDH is, similarly to the role of the classic CDH, precisely the problem of key recovery in Semidirect Product Key Exchange, and moreover that the relationship between SCDH and SDLP is not immediately obvious. Of course, one can solve SCDH if one can solve SDLP, but the converse does not follow *a priori.*

There are two general approaches for solving SCDH:

The Dimension Attack. The general form of this argument appears in [29]; we prefer the slightly more purpose-built exposition of [33]. The idea is basically that if our group G can be embedded as a multiplicative subgroup of a finite-dimensional algebra over a field, and if the automorphism ϕ can be extended to preserve addition on this algebra, we can solve SCDH for some pair (g, ϕ) using Gaussian elimination.

The Telescoping Attack. In [5], it is noticed that $1 * s_{g,\phi}(x) = \phi^x(g) s_{g,\phi}(x)$. Since we know $s_{g,\phi}(x)$ we can calculate $1 * s_{g,\phi}(x)$ and solve for $\phi^x(g)$. In some cases - notably, in the additive structure given in [31] - this suffices for recovery of $s_{g,\phi}(x + y)$.

We comment that a method of efficiently converting an SCDH solver to an SDLP solver is not currently known. On the other hand, a recent result of Montgomery and Zhandry [28] shows that a computational problem underpinning SDLP and a computational problem underpinning SCDH[10] are (surprisingly) quantum equivalent. We therefore cautiously conjecture that there exists some efficient quantum method of converting an SCDH solver to an SDLP solver.

5 A Candidate Group

We propose the following group of order p^3, where p is an odd prime, for use with `SPDH-Sign`.

Definition 16. *Let p be an odd prime. The group G_p is defined by*

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

As discussed in [9], this group is one of two non-abelian groups of order p^3 for an odd prime up to isomorphism. It has presentation

$$G_p = \langle x, y : y^p = 1, [x, y] = x^p =: z \in Z(G_p), z^p = 1 \rangle$$

as described in [23]; moreover, its automorphism group is known and has size $(p-1)p^3$ by [11, Theorem 3.1].

With respect to the various matters discussed in this paper, we briefly present the advantages of employing such a group.

[10] More precisely, the Vectorisation and Parallelisation problems of Couveignes [10], respectively.

Sampling. Recall that our security proof for `SPDH-Sign` relied heavily on the underlying identification scheme being honest-verifier zero knowledge, which in turn relied on the 'fake' transcripts to have the same distribution as honestly generated transcripts. For a pair (g, ϕ), it is therefore important to be able to sample uniformly at random from the group $\mathbb{Z}_n$, where n is the smallest integer for which $s_{g,\phi}(n) = 1$ - in our case, to do so it clearly suffices to compute n.

Here we recall Theorem 4, which tells us basically that, thinking of (g, ϕ) as a member of the semidirect product group $G \ltimes Aut(G)$, n must divide the order of (g, ϕ). We therefore have the following

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

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

Proof. By Theorem 4 we know that $n | ord((g, \phi))$, and it is standard that

$$ord((g, \phi)) \quad | \quad |G_p \ltimes Aut(G)|$$

. We know from the discussion at the outset of this section that $|G_p| = p^3$ and $|Aut(G_p)| = p^3(p-1)$. It follows that $n|p^3p^3(p-1)$. Since p is prime, and assuming that (g, ϕ) is not the identity, the claimed set is a complete list of divisors of $p^6(p-1)$ - excluding $p^6(p-1)$ itself, since this would imply $G_p \ltimes Aut(G_p)$ is cyclic.

It follows that for an arbitrary pair (g, ϕ) in $G_p \ltimes Aut(G_p)$, in order to compute the smallest n for which $s_{g,\phi}(n) = 1$, and therefore the group $\mathbb{Z}_n$, one has to compute $s_{g,\phi}(i)$ for at most 12 values of i. Moreover, by square-and-multiply each such computation requires $\mathcal{O}(\log p)$ applications of the group operation in the semidirect product group. In other words, we can compute a complete description of $\mathbb{Z}_n$ efficiently.

SDLP. By Theorem 8 and Theorem 9 we know SDLP in $G_p \ltimes Aut(G_p)$ has time complexity at most $2^{\mathcal{O}(\sqrt{\log poly(p)}} = 2^{\mathcal{O}(\sqrt{\log p})}$. Taking the security parameter to be the length of an input, we can represent a pair $(g, \phi) \in G_p \ltimes Aut(G_p)$ with a bitstring of length $\mathcal{O}(\log p^2) = \mathcal{O}(\log p)$. Asymptotically, then, with k as the security parameter we estimate the time complexity of the main quantum attack on SDLP as $2^{\mathcal{O}(\sqrt{k})}$. On the other hand, in order to derive a concrete estimate for specific security parameters - say, those required by NIST - one would have to check the associated constants much more carefully. Although this is outside the scope of this paper, we refer the reader to [6, Section 7.2 'Subexponential vs Practical'] for an idea of type of spirited research carried out in pursuit of a satisfactory resolution to deriving concrete security estimates - one should note, however, that this exposition deals with specific artefacts of the isogeny framework.

The Dimension Attack. Supposing an efficient method of converting an SCDH solver to an SDLP solver can be found, one solves SDLP efficiently provided one can efficiently embed G_p in an algebra over a field. However, as argued in [20], the following result of Janusz [17] limits the effectiveness of such an approach: the smallest dimension of an algebra over a field in which a p-group with an element of order p^n can be embedded is $1+p^{n-1}$. In our case, certainly G_p has an element of order p^2, and so since the attack relies on Gaussian elimination we expect the dimension attack for G_p to have complexity polynomial in $(p+1)^3 = \mathcal{O}(p^3)$. Since the G_p elements can be represented by a bitstring of order $4 \log p^2 = 8 \log p$, with k the security parameter the dimension attack runs in time $\mathcal{O}(2^{3k/8})$.

The Telescoping Attack. In general, the explicit method of deducing $s_{g,\phi}(x+y)$ from $s_{g,\phi}(y)$ and $\phi^x(g)$ relies on the group G being the abelian group of a matrix alegbra over a field under addition. In particular, an extension outside of this linear context is not known - we would expect, however, that such an extension would rely on equation solving techniques available only in an algebra over a field, rather than over a ring, and therefore that arguments on the efficiency of a representation discussed above would also apply.

Efficiency. Multiplication in G_p consists of 8 multiplication operations and 4 addition operations in $\mathbb{Z}_{p^2}$, for a total of $\mathcal{O}(8 \log p^2) = \mathcal{O}(\log p)$ operations. Assuming that applying an automorphism ϕ has about the same complexity as multiplication[11]. It follows by standard square-and-multiply techniques that calculating $s_{g,\phi}$ and evaluating the group action is very roughly of complexity $\mathcal{O}((\log p)^2)$.

The signatures are also rather short, consisting of N elements of $\mathcal{X}_{g,\phi}$ and N elements of $\mathbb{Z}_n$. Since $\mathcal{X}_{g,\phi} \subset G_p$ we can represent $\mathcal{X}_{g,\phi}$ elements as bitstrings of length $4 \log(p^2) = 8 \log p$; and since $n = p^i(p-1)^j$ for some $1 \leq i \leq 5$ and $0 \leq j \leq 1$, $\mathbb{Z}_n$ elements can be represented by bitstrings of length $\log p^i(p-1)^j$. It follows that we get signatures of length

$$N((8+i)\log p + j \log(p-1))$$

6 Conclusion

We have given a constructive proof that a few elementary definitions give rise to a free, transitive group action; such a group action naturally gives rise to an identification scheme and a signature scheme. Moreover, well-known tools allow us to phrase the security of this signature scheme in terms of the semidirect discrete logarithm problem, which is itself a special case of Couveignes' Vectorisation Problem.

Our main contributions are as follows: firstly, the generality of the construction gives an unusually diverse family of signature schemes - indeed, a signature scheme of the `SPDH-Sign` type is defined for each finite group. Much further

[11] This is indeed the case if the automorphism is inner.

study on the relative merits of different choices of finite non-abelian group in different use cases is required to fully realise the potential of this diversity.

Second, our Theorem 4 essentially gives us information about how to compute the group in our group action. In Theorem 9, we saw one particular case where the result was enough to completely describe how to efficiently compute the group, thereby yielding an example of a group-action based key exchange in which efficient sampling is possible from the whole group, without appealing to techniques inducing additional overhead, most notably the 'Fiat-Shamir with aborts' technique of Lyubashevsky.

The paper notably does not address concrete security estimates or recommend parameter sizes for a signature scheme. In order to do so we would need to carefully check the constants in the asymptotic security estimates - we consider the scale of this task, along with that of providing an implementation of the scheme, as sufficient to merit a separate paper.

At a late stage of the preparation of this manuscript the authors were made aware of work in [13] discussing the security of group action-induced computational problems, particularly in a quantum sense. The arguments therein should be addressed when discussing the difficulty of SDLP in subsequent work.

Acknowledgements

We would like to thank the anonymous reviewers who provided useful feedback on this manuscript.

References

[1] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. "From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security". In: *Advances in Cryptology—EUROCRYPT 2002: Amsterdam, The Netherlands, April 28–May 2, 2002*. Springer. 2002, pp. 418–433.

[2] Christopher Battarbee, Delaram Kahrobaei, Ludovic Perret, and Siamak F Shahandashti. "A Subexponential Quantum Algorithm for the Semdirect Discrete Logarithm Problem". In: (2022), pp. 1–27. URL: https://csrc.nist.gov/csrc/media/Events/2022/fourth-pqc-standardization-conference/documents/papers/a-subexpoenential-quantum-algorithm-pqc2022.pdf.

[3] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. "CSI-FiSh: efficient isogeny based signatures through class group computations". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2019, pp. 227–247.

[4] Dan Boneh and Victor Shoup. "A graduate course in applied cryptography". In: *Draft 0.5* (2020). URL: https://crypto.stanford.edu/~dabo/cryptobook/BonehShoup_0_4.pdf.

[5] Daniel Brown, Neal Koblitz, and Jason Legrow. "Cryptanalysis of 'MAKE'". In: *Journal of Mathematical Cryptology* 16.1 (2022), pp. 98–102.

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

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

[8] Andrew M Childs and Wim Van Dam. "Quantum algorithms for algebraic problems". In: *Reviews of Modern Physics* 82.1 (2010), p. 1.

[9] Keith Conrad. *Groups of Order p^3*. URL: `https://kconrad.math.uconn.edu/blurbs/grouptheory/groupsp3.pdf`.

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

[11] MJ Curran. "The automorphism group of a nonsplit metacyclic p-group". In: *Archiv der Mathematik* 90 (2008), pp. 483–489.

[12] Luca De Feo and Steven D Galbraith. "SeaSign: compact isogeny signatures from class group actions". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2019, pp. 759–789.

[13] Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. "Generic Models for Group Actions". In: *Cryptology ePrint Archive* (2023). URL: `https://eprint.iacr.org/2022/1230`.

[14] Amos Fiat and Adi Shamir. "How to prove yourself: Practical solutions to identification and signature problems". In: *Advances in Cryptology—CRYPTO'86: Proceedings 6*. Springer. 1987, pp. 186–194.

[15] 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.

[16] Jiao Han and Jincheng Zhuang. "DLP in semigroups: Algorithms and lower bounds". In: *Journal of Mathematical Cryptology* 16.1 (2022), pp. 278–288.

[17] GJ Janusz. "Faithful Representations of p-Groups at Characteristic p". In: *Representation Theory of Finite Groups and Related Topics* 21 (1971), p. 89.

[18] Delaram Kahrobaei, Ramon Flores, and Marialaura Noce. "Group-based Cryptography in the Quantum Era". In: *Notices of the American Mathematical Society* 70.5 (2023), pp. 752–763.

[19] Delaram Kahrobaei, Ramon Flores, Marialaura Noce, Maggie Habeeb, and Christopher Battarbee. *Applications of Group Theory in Cryptography: Post-quantum Group-based Cryptography*. The Mathematical Surveys and Monographs series of the American Mathematical Society, forthcoming, 2023.

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

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

[22] Vadim Lyubashevsky. “Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2009, pp. 598–616.

[23] Ayan Mahalanobis. “The MOR cryptosystem and extra-special p-groups”. In: *Journal of Discrete Mathematical Sciences and Cryptography* 18 (2015), pp. 201–208.

[24] Gérard Maze, Chris Monico, and Joachim Rosenthal. “Public key cryptography based on semigroup actions”. In: *arXiv preprint cs/0501017* (2005).

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

[26] Chris Monico and Ayan Mahalanobis. “A remark on MAKE–a Matrix Action Key Exchange”. In: *arXiv preprint arXiv:2012.00283* (2020).

[27] Christopher J Monico. *Semirings and semigroup actions in public-key cryptography*. University of Notre Dame, 2002.

[28] Hart Montgomery and Mark Zhandry. “Full Quantum Equivalence of Group Action DLog and CDH, and More”. In: *Advances in Cryptology – ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part I*. Taipei, Taiwan: Springer-Verlag, 2023, pp. 3–32. ISBN: 978-3-031-22962-6. DOI: `10.1007/978-3-031-22963-3_1`. URL: `https://doi.org/10.1007/978-3-031-22963-3_1`.

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

[30] *PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates*. URL: `https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4#new-call`.

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

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

[33] Vitaliĭ Roman’kov. “Linear decomposition attack on public key exchange protocols using semidirect products of (semi) groups”. In: *arXiv preprint arXiv:1501.01152* (2015).

[34] Alexander Rostovtsev and Anton Stolbunov. “Public-key cryptosystem based on isogenies”. In: *Cryptology ePrint Archive* (2006). URL: `https://eprint.iacr.org/2006/145`.

[35] Anton Stolbunov. "Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves". In: *Advances in Mathematics of Communications* 4.2 (2010), pp. 215–235.

[36] Anton Stolbunov. "Cryptographic Schemes Based on Isogenies". PhD thesis. Jan. 2012. DOI: 10.13140/RG.2.2.20826.44488.