An Isogeny-Based ID Protocol Using Structured Public Keys

Karim Baghery, Daniele Cozzo, and Robi Pedersen

imec-COSIC, KU Leuven, Leuven, Belgium.
karim.baghery@kuleuven.be, daniele.cozzo@kuleuven.be,
robi.pedersen@kuleuven.be

Abstract. Isogeny-based cryptography is known as one of the promising approaches to the emerging post-quantum public key cryptography. In cryptography, an IDentification (ID) protocol is a primitive that allows someone's identity to be confirmed. We present an efficient variation of the isogeny-based interactive ID scheme used in the base form of the CSI-FiSh signature [BKV19], which was initially proposed by Couveignes-Rostovtsev-Stolbunov [Cou06,RS06], to support a larger challenge space, and consequently achieve a better soundness error rate in each execution. To this end, we prolong the public key of the basic ID protocol with some *well-formed* elements that are generated by particular factors of the secret key. Due to the need for a well-formed (or structured) public key, the (secret and public) keys are generated by a trusted authority. Our analysis shows that, for a particular security parameter, by extending a public key of size 64 B to 2.1 MB, the prover and verifier of our ID protocol can be more than 14× faster than the basic ID protocol which has a binary challenge space, and moreover, the proof in our case will be about 13.5× shorter. Using standard techniques, we also turn the presented ID protocol into a signature scheme that is as efficient as the state-of-the-art CSI-FiSh signature, and is existentially unforgeable under chosen message attacks in the (quantum) random oracle model. However, in our signature scheme, a verifier should get the public key of a signer from a trusted authority, which is standard in a wide range of current uses of signatures. Finally, we show how to eliminate the need for a trusted authority in our proposed ID protocol.

Keywords: Isogeny-based Cryptography · Identification Protocols · Digital Signatures · Quantum Random Oracle Model

1 Introduction

An IDentification (ID) protocol is an interactive cryptographic protocol between two parties called Prover and Verifier, that allows to prove the identity of the former to the latter [Sch89]. At the end of a successful execution of an ID protocol, the Verifier is convinced that it is interacting with the Prover that knows the secret key sk corresponding to a particular public key pk. ID protocols are deployed in a wide range of cryptographic protocols and practical applications, and above all, they can be used to build digital signatures. Constructions like Schnorr's ID protocol and its corresponding signature [Sch89] are known for their simplicity and efficiency, but rely on the intractability of the discrete logarithm problem, which is known to be insecure against sufficiently powerful quantum computers [Sho94].

There are various research areas that are exploring post-quantum cryptographic techniques to design primitives and protocols that can remain secure in the presence of quantum computers. One of these is isogeny-based cryptography, which was independently proposed by Couveignes [Cou06] and by Rostovtsev and Stolbunov [RS06, Sto10]. The security of these isogeny-based constructions mainly relies on the difficulty of finding an explicit isogeny connecting two isogenous ordinary elliptic curves over a finite field, while the construction of such isogenies can be efficiently computed as the action of elements of the ideal-class group of the endomorphism ring of these elliptic curves. In these original works, the authors also independently proposed an isogeny-based interactive ID protocol. In his Ph.D. thesis, Stolbunov [Sto12] further mentioned how to convert the ID protocol to the first isogeny-based signature scheme using the Fiat–Shamir transform. However, these constructions have many drawbacks. First, they work with a binary challenge space, and therefore need to be repeated many times to achieve a reasonable soundness rate. Second, in order to allow the uniform sampling and efficiently computable canonical representations of elements in the class group needed in these protocols, the class group structure has to be known, which is a difficult problem for quadratic imaginary fieds [HM89]. Finally, a quantum attack by Childs, Jao, and Soukharev [CJS14] pushed the security parameter sizes of these schemes to an impractical scale. Even with current optimizations [DFKS18], these schemes are inefficient in practice.

Later works have tried to mitigate these shortcomings. In 2018, Castryck et al. [CLM$^+$18] proposed CSIDH (Commutative Supersingular Isogeny Diffie-Hellman) and showed that using supersingular curves over $\mathbb{F}_p$ instead of ordinary ones, combined with the action by $\mathbb{F}_p$-rational ideals, greatly increases the efficiency of isogeny computations and thus makes these schemes again usable in practice. De Feo and Galbraith [DFG19] used the tools of CSIDH to construct a signature scheme that does not need the knowledge of the class group, but rather uses rejection sampling. With later improvements by Decru, Panny, and Vercauteren [DPV19], Seasign signatures could be performed in a few minutes. Later that same year Beullens, Kleinjung and Vercauteren [BKV19] performed a record class-group computation for the CSIDH-512 parameter set (a class group of size $\approx 2^{257}$) that finally allowed class group elements to be uniformly sampled and efficiently represented, leading to a practical signature scheme, called CSI-FiSh. In its simplest version, using a binary challenge space, a CSI-FiSh signature takes slightly less than 3 seconds. Then, with further improvements, the authors managed to decrease it to a few hundred milliseconds by increasing the public-key size and using a different Σ-protocol which is an ID protocol for a different language, but supports a larger challenge space.

A very different approach to isogeny-based ID protocols and signatures was taken based on the SIDH scheme proposed by Jao and De Feo [JDF11], which uses supersingular elliptic curves over $\mathbb{F}_{p^2}$, where the endomorphism ring is isomorphic to an order in a quaternion algebra, rather than a quadratic imaginary field. The original paper also proposes an ID protocol, based on which later signature schemes have been proposed [YAJ$^+$17, GPS17], although not very practical. The work of Galbraith et. al [GPS17] however also introduced a signature scheme based on the KLPT algorithm [KLPT14], which uses the knowledge of the endomorphism ring of two supersingular elliptic curves over $\mathbb{F}_{p^2}$ to compute an isogeny connecting them. In 2020, De Feo et al. [DFKL$^+$20] showed that with further assumptions, this scheme can be made practical and proposed the signature scheme SQI-Sign. At the NIST security level 1, SQI-Sign runs in a few seconds and has public-key sizes a magnitude smaller than any other post-quantum secure signature scheme.

Our Contributions. Our main contribution is to extend the ID protocol used in the base form of CSI-FiSh signature [BKV19], which was initially proposed by Couveignes-Rostovtsev-Stolbunov [Cou06,RS06], to work with a larger challenge space rather than a binary space. By extending the challenge space, the proposed ID protocol achieves an arbitrarily small soundness error rate in each execution. To this end, we modify the ID protocol with binary challenge space [Cou06,RS06,BKV19] and prolong its public key with some new *structured* elements. Particularly, each new element in the public key is built from a distinct specific multiple of the secret key, where the coefficients are taken from a public *exceptional set* [BCPS18,DLSV20]. The latter is a crucial requirement in the security proof for knowledge soundness. Then, we show that using the *structured public key*, we can build an ID protocol that works with a larger challenge space, and consequently achieves a bigger soundness error rate in each run. Due to the need for a *well-formed* or structured public key, in the basic and more efficient version of our ID protocol, we assume that the (secret and public) keys are generated by a trusted authority and shared with parties. Our performance analysis shows that, in practice, for a particular security parameter, with an honestly generated public key of size 2.1 MB, the prover and verifier of our ID protocol can be more than 14× faster than using repetitions of the basic ID protocol with a binary challenge space and also the proof will be about 13.5× shorter. In order to apply further optimizations to the soundness security, we define *superexceptional sets* (in Definition 3.2) as a particular form of exceptional sets, which can be of independent interest.

As our second contribution, we use standard techniques to turn the proposed ID protocol into a signature scheme that has the same efficiency as the state-of-the-art isogeny-based signature scheme CSI-FiSh [BKV19], constructed to work in the CSIDH setting. In our signature scheme, the verifier needs to get the public key of the signer from a trusted authority rather than from the signer itself, which is standard in applications like public key certificates. Similar to CSI-FiSh, our signature scheme would allow to generate and verify a signature of size less than 400 bytes in less than 0.5 seconds.

In our basic ID protocol, to guarantee the well-formedness of the public keys, we assume that these are generated by a trusted authority. As the next contribution of the paper, we show how this trust can be eliminated by letting the prover generate the key pair themselves, while appending a proof of well-formedness to the public key. We also show that in order to increase the efficiency, this proof can be incrementally generated, i.e. that the correctness of the i-th public key element can be proven more efficiently by using the fact that elements $1, \ldots, i-1$ have already been proven.

Organization. Section 2 presents some preliminaries used in the paper. In Section 3, we present our ID protocol in the setting where a trusted authority has generated the keys. In Section 4, we detail the corresponding signature scheme. In Section 5, we propose two protocols to eliminate the trust on the key generation and also discuss some applications. We present some benchmarks in Section 6. Finally, we conclude the paper in Section 7.

2 Preliminaries

We denote by $\mathbb{Z}_N = \mathbb{Z}/N\mathbb{Z}$ the integers modulo N, where we assume that N is a composite number of known prime factorisation $N = \prod_{i=1}^{m} q_i^{r_i}$ with $q_1 < \cdots < q_m$ primes and all $r_i \in \mathbb{N}$. We further say that a function $\mu(x)$ is a negligible function of x, if for any constant c, there exists x_0, such that for all $x > x_0$, we have $\mu(x) < \frac{1}{x^c}$.

2.1 ID protocols

Sigma-Protocols. Let λ be a security parameter and let $X = X(\lambda)$ and $W = W(\lambda)$ be sets. Let $\mathcal{R}$ be a relation on $X \times W$ that defines a language $\mathbf{L} = \{\mathsf{x} \in X : \exists \mathsf{w} \in W, \mathcal{R}(\mathsf{x}, \mathsf{w}) = 1\}$. Given $\mathsf{x} \in \mathbf{L}$, an element $\mathsf{w} \in W$ such that $\mathcal{R}(\mathsf{x}, \mathsf{w}) = 1$ is called a witness. Let R be a PPT algorithm such that $\mathsf{R}(1^\lambda)$ outputs pairs (x, w) such that $\mathcal{R}(\mathsf{x}, \mathsf{w}) = 1$.

A sigma-protocol (Σ-protocol) for the relation $\mathcal{R}$ is a 3-round interactive protocol between two PPT algorithms: a prover P and a verifier V. P holds a witness w for $\mathsf{x} \in \mathbf{L}$ and V is given x. P first sends a value a to V, and then V answers with a challenge c , and finally P answers with z. V accepts or rejects the proof. The triple $\mathsf{trans} = (a, c, z)$ is called a transcript of the Σ-protocol. A Σ-protocol is supposed to satisfy *Completeness*, *Honest Verifier Zero-Knowledge* (HVZK), and *Special Soundness* defined below.

Definition 2.1 (Completeness). *A Σ-protocol Π with parties* (P, V) *is* complete for R*, if for all* $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}$*, the honest* V *will always accept the honest* P*.*

Definition 2.2 (HVZK). *A Σ-protocol satisfies* HVZK for R*, if there exists a PPT algorithm* Sim *that given* $\mathsf{x} \in X$*, can simulate the* trans *of the scheme, s.t. for all* $\mathsf{x} \in \mathbf{L}$*,* $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}$*,*

$$\mathsf{trans}(\mathsf{P}(\mathcal{R}, \mathsf{x}, \mathsf{w}) \leftrightarrow \mathsf{V}(\mathcal{R}, \mathsf{x})) \approx \mathsf{trans}(\mathsf{Sim}(\mathcal{R}, \mathsf{x}) \leftrightarrow \mathsf{V}(\mathcal{R}, \mathsf{x}))$$

where $\mathsf{trans}(\mathsf{P}(\cdot) \leftrightarrow \mathsf{V}(\cdot))$ *indicates the transcript of Π with* (P, V)*, and $\approx$ denotes the indistinguishability of transcripts.*

Definition 2.3 (Special Soundness). *The Σ-protocol Π with parties* (P, V) *is* special sound for R*, if there exists a PPT extractor* Ext*, such that for any* $\mathsf{x} \in \mathbf{L}$*, given two valid transcripts (a, c, z) and (a, c', z') for the same message a but $c \neq c'$, then* $\mathsf{Ext}(a, c, z, c', z')$ *outputs a witness* w *for the relation* $\mathcal{R}$*.*

Identification Protocols. An ID protocol is a special case of a Σ-protocol between two parties (P, V), with respect to a hard relation defined by a key generator KGen, as $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$, where one thinks of sk as a witness for the public key pk.

2.2 Building NIZK ID Protocols and Signatures.

An HVZK Σ-protocol Π can be transformed to a Non-Interactive Zero-Knowledge (NIZK) argument Π_{NIZK} in the Random Oracle Model (ROM) via the Fiat–Shamir (FS) transformation [FS87]. The transformation also allows to build signatures from an ID protocol [AABN02]; we describe this procedure in Appendix A. Next we define strong existential unforgeability under chosen message attacks, the primary security notion for signatures.

Definition 2.4 (Strong Existential Unforgeability under Chosen Message Attacks). *A signature scheme* $\Pi_{\mathsf{Sign}} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf})$ *is said to be strong Existentially Unforgeable under adaptive Chosen-Message Attacks (sEU-CMA) if for all PPT adversaries $\mathcal{A}$,*

$$\left| \Pr \begin{bmatrix} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda),\ \sigma_i \leftarrow \mathsf{Sign}(\mathsf{sk}, m_i) \textit{ for } 1 \leq i \leq k; \\ (m, \sigma) \leftarrow \mathcal{A}^{\mathsf{Sign}(.)}(\mathsf{pk}, (m_i, \sigma_i)_{i=1}^{k}) : \mathsf{Vf}(m, \sigma, \mathsf{pk}) = 1 \wedge (m, \sigma) \notin Q \end{bmatrix} \right|$$

is negligible in the security parameter λ, where $Q := \{(m_1, \sigma_1) \cdots, (m_k, \sigma_k)\}$ is the set of the messages requested by $\mathcal{A}$ and the signatures returned from the signing oracle.

2.3 CSI-FiSh

The digital signature scheme CSI-FiSh [BKV19] is based on the ID protocol with *binary challenge space* initially proposed by Couveignes-Rostovtsev-Stolbunov [Cou06, RS06], that closely follows the lines of the Schnorr identification protocol as introduced in [Sch89]. We will introduce it in the notation of hard homogeneous spaces, a notion introduced by Couveignes [Cou06], which generalizes group actions that contain hard computational problems.

Definition 2.5 (Hard homogeneous space [Cou06]). *A* Hard Homogeneous Space *(HHS) is a pair of a finite Abelian group $\mathcal{G}$ acting on a finite set $\mathcal{E}$ with a free and transitive map $\star : \mathcal{G} \times \mathcal{E} \to \mathcal{E}$, that is efficiently computable. Furthermore, operations, sampling and membership checks in $\mathcal{G}$, as well as membership and equality checks in $\mathcal{E}$ are efficiently computable. Given an element of $\mathcal{G}$, one can also efficiently compute a unique representation. The following are hard algorithmic problems:*

- Vectorization*: Given $E_1, E_2 \in \mathcal{E}$, find $\mathfrak{a} \in \mathcal{G}$, such that $\mathfrak{a} \star E_1 = E_2$.*
- Parallelization*: Given $E_1, E_2, F_1 \in \mathcal{E}$ with $E_2 = \mathfrak{a} \star E_1$, compute $F_2 = \mathfrak{a} \star F_1$.*

When $\mathcal{G}$ is cyclic of order N and $\mathfrak{g}$ is a given generator of $\mathcal{G}$, we can also define the group action $[\ ] : \mathbb{Z}_N \times \mathcal{E} \to \mathcal{E}$ as $[a]E = \mathfrak{g}^a \star E$ for $a \in \mathbb{Z}_N, E \in \mathcal{E}$. It holds $[a][b]E = [a+b]E$.

The ID protocol underlying CSI-FiSh allows to prove knowledge of a secret group action $[a]$ connecting two given set elements $(E_0, E_1 = [a]E_0)$, where $E_0 \in \mathcal{E}$ is a public starting element. Similar to the Schnorr protocol, the prover first commits to a random $b \in \mathbb{Z}_N$ via $E_b = [b]E_0$, then after receiving a random bit c from the verifier, sends the response $r = b - ca \bmod N$. The verifier checks whether $[r]E_c = E_b$. While in Schnorr protocols, the soundness error can be increased by choosing challenges as bit-strings of length k, computing $[r]E_c = [r][ca]E_0$ for non-binary c is not directly possible in the more restrictive HHS setting, since there is no way for the verifier to compute the action of ca without knowing a.

In order to decrease the soundness error of their ID protocol, the authors rather increase the challenge space by using larger keys: the secret key is a set $a_1, \ldots, a_{S-1}$ which defines the corresponding public key $E_1, \ldots, E_{S-1}$. Then the prover proves knowledge of any isogeny connecting two elements of its public key, which results in a Σ-protocol with soundness error rate $\frac{1}{S}$. We note at this point, that this protocol cannot be used as an identification protocol for the knowledge of the secret key, in that an extractor can only extract a difference $a_i - a_j$ of secret keys. The purpose of the next sections is to construct such an identification protocol.

The authors of [BKV19] instantiate the HHS by identifying $\mathcal{E}$ with the set of supersingular elliptic curves defined over a prime field $\mathbb{F}_p$ with $\log_2 p \approx 512$. The class group $\mathsf{Cl}(\mathcal{O})$ of the $\mathbb{F}_p$-rational endomorphism ring $\mathcal{O}$ acts freely and transitively on these elements by isogenies, which allows the identification $\mathcal{G}$ with

$\mathsf{Cl}(\mathcal{O})$. The full class group structure has also been determined in [BKV19]. It has size

$$\begin{aligned}\#\mathsf{Cl}(\mathcal{O}) = 3 \cdot 37 \cdot 1407181 \cdot 51593604295295867744293584889 \\ \cdot 31599414504681995853008278745587832204909\end{aligned}$$

and is cyclic with generator $\mathfrak{g} = (3, \pi - 1)$. The starting element $E_0 : y^2 = x^3 + x$ enjoys the special symmetry, that the twist of $[a]E_0$ is $[-a]E_0$. Since twisting can be performed efficiently, the authors implicitly include twists in the public key and thus double the challenge space, reducing the soundness error rate to $\frac{1}{2S-1}$. For the sake of generality, we also describe this concept for HHS by introducing the following notion of a *symmetric* HHS. In this setting, we generally use the index notation to identify the "twists", i.e. we write $E_a = [a]E_0$ and $E_{-a} = [-a]E_0$ for the twists.

Definition 2.6 (Symmetric hard homogeneous space). *We call a hard homogeneous space* symmetric around $E_0 \in \mathcal{E}$*, if, given an element* $\mathfrak{a} \star E_0$*, one can efficently compute* $\mathfrak{a}^{-1} \star E_0$ *without any extra information.*

3 An Efficient ID Protocol

Next, we generalize the ID scheme with binary challenge space used in the basic version of CSI-FiSh [BKV19] to support a larger challenge space. Their protocol allows to prove the knowledge of secret key x for the public key $E_1 = [x]E_0$, but works with a binary challenge space. As a consequence, this construction requires a large number of parallel executions and large communication to achieve a reasonable soundness error rate. In order to extend the ID protocol to support a larger challenge space, we assume that there exists a trusted authority in the protocol that generates the pair of secret and (structured) public keys. The trusted authority sends both keys to the prover, while only the public key to the verifier. We later discuss how to eliminate the need for a trusted authority.

3.1 Construction and Security Proofs

To efficiently prove the knowledge of x in $E_1 = [x]E_0$, our key idea is to ask a trusted authority to generate $k - 2$ new curves $E_2, E_3, \cdots, E_{k-1}$ using other multiples of x, say $E_i = [c_i x]E_0$ for $i = 2, \cdots, k-1$, where c_i are public integers.

The issue with composite N. In order to achieve special soundness and build an efficient extraction algorithm that can extract the witness from two acceptable transcripts of our construction, we need to assume that the difference of any two challenge values is always invertible. Since N can be composite, we need to define the challenge set to only contain elements, whose pairwise difference is invertible. To this end we use *exceptional sets* [BCPS18, DLSV20].

Definition 3.1 (Exceptional set). *An* exceptional set (modulo N) *is a set* $\mathbb{C} = \{c_0, \ldots, c_{k-1}\} \subseteq \mathbb{Z}_N$*, where the pairwise difference* $c_i - c_j$ *of all elements* $c_i \neq c_j$ *is invertible modulo* N*.*

Given k and particular N with smallest prime factor $q_1 \geq k$, there exists an efficient algorithm XSGen that outputs an exceptional set of size k with integer elements, $\mathbb{C} = \{c_0 = 0, c_1 = 1, c_2, \cdots, c_{k-1}\}$.[1] In order for the exceptional set to have a specific target size $k \geq q_1$, we need to work in a subgroup $\mathbb{Z}_{N'}$, where $N' \mid N$ has smallest prime factor $q_1' \geq k$. To do this we factor out the smaller primes. The only restriction this puts on N is that it is not k-smooth, which is a reasonable assumption for arbitrary composite numbers and $k \ll N$.

The ID-protocol. We now describe the steps of our ID-protocol. Given a security parameter and the system parameters, the trusted authority samples a secret key $x \leftarrow \mathbb{Z}_N$, generates an exceptional set $\mathbb{C} = \{c_0 = 0, c_1 = 1, c_2, \cdots, c_{k-1}\}$ using XSGen and then generates the public key $(E_0, E_1, \ldots, E_{k-1})$, where $E_i = [c_i x]E_0$ for $i = 1 \ldots, k-1$. Note that we see E_0 as part of the public-key for simplicity and that $[0]$ denotes the neutral element of the group action. The trusted authority then sends the secret key to the prover, and the public key to both the prover and the verifier. Then, the prover can use the Σ-protocol in the figure below to convince the verifier about its *knowledge* of the secret key x.

The Main ID Protocol Under a Trusted Setup with an Arbitrary N

Trusted Authority

Use XSGen to obtain $\mathbb{C}$, where

$\mathbb{C} = \{c_0 = 0, c_1 = 1, c_2, \cdots, c_{k-1}\}$

$x \leftarrow \mathbb{Z}_N$, set $\{E_i = [c_i x]E_0\}_{i=1}^{k-1}$

$(x, (E_0, E_1, \cdots, E_{k-1}))$		$(E_0, E_1, \cdots, E_{k-1})$
Prover:		**Verifier:**
$b \leftarrow \mathbb{Z}_N,\ E_b \leftarrow [b]E_0$	$\xrightarrow{\quad E_b \quad}$	
	$\xleftarrow{\quad d \quad}$	$d \leftarrow \{0, \ldots, k-1\}$
$r \leftarrow b - c_d \cdot x \mod N$	$\xrightarrow{\quad r \quad}$	return $E_b \overset{?}{=} [r]E_d$

The following theorem proves the security of the proposed ID protocol.

Theorem 3.1. *Assuming the existence of an exceptional set* $\mathbb{C} = \{c_0 = 0, c_1 = 1, c_2, c_3, \cdots, c_{k-1}\}$*, the described ID-protocol is complete, HVZK, and special sound with soundness error rate* $\frac{1}{k}$*.*

[1] An easy approach is just to generate $k-2$ distinct elements from $\{2, \ldots, q_1\}$. In its simplest form, we have $\mathbb{C} = \{0, 1, 2, \cdots, k-1\}$.

Proof. For the completeness, the honest prover follows the protocol and additionally knows a secret x such that $E_i = [c_i x]E_0$ for $i = 1, \dots, k-1$. The honest verifier checks whether $E_b = [r]E_d = [b - c_d x]E_d = [b - c_d x][c_d x]E_0 = [b]E_0$ which holds given the assumptions on the prover.

For the HVZK, we construct a simulator that given the honestly generated challenge d, samples r randomly from $\mathbb{Z}_N$, then sets $E_b = [r]E_d$ and returns the transcript (E_b, d, r). In both the real and the simulated transcripts, r and E_b are sampled uniformly at random, yielding indistinguishable distributions.

For special soundness, given two valid transcripts of the protocol, we build an efficient extraction algorithm that extracts the witness x. Let (E_b, d, r) and (E_b, d', r') be two acceptable transcripts of the protocol, where $d \neq d'$, consequently $r \neq r'$ (for non-zero x). From the verification equation, one can conclude that $[r]E_d = [r']E_{d'}$, and from the (trusted) key generation we know that $E_i = [c_i x]E_0$ for $i = 1, \dots, k-1$. These imply that we have $[r][c_d x]E_0 = [r'][c_{d'} x]E_0$, which implies that $r - r' \equiv x(c_{d'} - c_d) \pmod{N}$. Considering the fact that both c_d and $c_{d'}$ are sampled from the exceptional set $\mathbb{C}$, $c_{d'} - c_d$ is invertible modulo N, this allows the extraction of x as $x = \frac{r - r'}{c_{d'} - c_d} \mod N$. □

Soundness error rate. In its current form, our protocol has soundness error rate $1/k$. To achieve a target soundness error of $2^{-\lambda}$ for a given security parameter λ, we therefore have to repeat our protocol at least $\lceil \lambda \log_k 2 \rceil$ times.

Making the construction non-interactive. The described ID protocol is a public-coin Σ-protocol, therefore can be turned into a non-interactive ID protocol using the Fiat–Shamir transform [FS87]. To do so, let $t = t(k) = \lceil \lambda \log_k 2 \rceil$. The prover generates t distinct elements $b_1, \dots, b_t \leftarrow \mathbb{Z}_N$ and commits to t elliptic curves $E_{b_i} = [b_i]E_0$ for $i = 1, \dots, t$. Then the challenge is determined by hashing the commitments and the statements using a hash function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^{t\lceil \log_2 k \rceil}$, modeled as a random oracle, and parsing it into t challenges:

$$d = d_1 || \dots || d_t = \mathsf{H}(E_0, \dots, E_{k-1} || E_{b_1}, \dots, E_{b_t})\,.$$

The response is given as $r = (r_1, \dots, r_t) \equiv (b_1 - c_{d_1} x, \dots, b_t - c_{d_t} x) \pmod{N}$. The prover publishes (d, r) as its proof. The verifier then checks, whether

$$\mathsf{H}(E_0, \dots, E_{k-1} || [r_1]E_{d_1}, \dots, [r_t]E_{d_t}) \stackrel{?}{=} d\,.$$

Lemma 3.1. *The non-interactive version of our ID-protocol is a NIZK quantum proof of knowledge in the quantum random oracle model.*

Proof. The freeness of the group action implies that, if $[b]E_0 = [b']E_0$, then $b = b'$. This immediately implies that our scheme has unique responses. Furthermore, the freeness of the group action also implies superlogarithmic collision-entropy of the commitments, since commitments will only collide if they are generated using the same b, which is a negligible function of the security parameter. Finally, the challenge space is of size $2^{t\lceil \log_2 k \rceil} \geq 2^{\lambda}$, thus superpolynomial in λ. Using our results for completeness, special soundness and HVZK from Theorem 3.1 this implies that our protocol is a quantum proof of knowledge using [DFMS19, Th. 25] and zero-knowledge against quantum adversaries [Unr17]. □

3.2 Optimizations and Efficiency

Similar to the proposal used in CSI-FiSh [BKV19], we can double our challenge space using twists. To this end, we assume that the underlying HHS is *symmetric* as by Definition 2.6. Defining $E_{-i} = [c_{-i}x]E_0 = [-c_i x]E_0$ allows challenges to be sampled from the set $d \leftarrow \{-(k-1), \ldots, k-1\}$ of size $2k-1$, while the response and verification steps proceed in exactly the same way as in the ID-protocol: In the case $d < 0$, the response is simply $r = b - c_{-d}x = b + c_d x$ and for the verification step, the verifier needs to compute $E_{-d} = [-c_d x]E_0$ via the efficient map from E_d, and check if $E_b = [r]E_{-d} = [b + c_d x][-c_d x]E_0$.

By this extension, our protocol achieves soundness error rate $\frac{1}{2k-1}$, and thus has to be repeated $t(2k-1) = \lceil \lambda \log_{2k-1} 2 \rceil$ times to achieve a target soundness error of at least $2^{-\lambda}$. Note that in the non-interactive case, the hash function needs to be redefined to have the output domain $\{0,1\}^{t(2k-1)\lceil \log_2(2k-1) \rceil}$.

However, there is another problem. To guarantee the special soundness proven in Theorem 3.1, we used exceptional sets (Definition 3.1), that guarantee that any pair of challenges allows the extraction of the secret x by an extractor. Since we are implicitly extending our challenge space to also include negative values of the factors c_i, we have to guarantee that their pairwise sums are invertible too. We therefore define the notion of superexceptional sets.

Definition 3.2 (Superexceptional set). *A* superexceptional set (modulo N) *is a set $\mathbb{C} = \{c_0, \ldots, c_{k-1}\}$, where the pairwise difference $c_i - c_j$ of all distinct elements $c_i \neq c_j$ and the pairwise sum $c_i + c_j$ of all elements c_i, c_j (including $c_i = c_j$) is invertible modulo N.*

Similarly to exceptional sets, we can define an efficient algorithm SXSGen for generating superexceptional sets modulo N of size $k \leq \frac{1}{2}(q_1 + 1)$. By letting the trusted authority in our ID-protocol generate a superexceptional set instead of an exceptional one, and by assuming the underlying hard homogeneous space is symmetric around E_0, we have the following lemma.

Lemma 3.2. *Assuming the existence of a superexceptional set $\mathbb{C} = \{c_0 = 0, c_1 = 1, c_2, \ldots, c_{k-1}\}$, the described ID-protocol is complete, HVZK, and special sound with soundness error rate $\frac{1}{2k-1}$.*

Proof. We have already shown completeness. HVZK and special soundness closely follow the proof in Theorem 3.1. Note that because we also allow negative challenges, we can end up with three different scenarios for challenges d, d': They can either be both positive, both negative or one positive and one negative. In the first two cases, the extractor will need to invert an element of the form $\pm(c_{|d'|} - c_{|d|}) \mod N$, which is guaranteed to be possible in exceptional sets. In the third case, the extractor will end up with needing to invert an element of the form $\pm(c_{|d'|} + c_{|d|}) \mod N$, which is only guaranteed to be possible by using a superexceptional set $\mathbb{C}$. □

Computational cost. We establish the computational costs in terms of Group Actions (GAs) of our proposed protocol in the standard and in the symmetric case. We assume that we want to reach a target soundness error of $2^{-\lambda}$. Given a soundness error rate of $1/s$ per round, we need to repeat the underlying protocol $t(s) = \lceil \lambda \log_s 2 \rceil$ times. In both protocols, the prover and the verifier only need to compute a single GA per step thus, for both, the total cost in GAs is also expressed by $t(s)$. We find the following total costs:

- Standard ID-protocol: $t(k) = \lceil \lambda / \log_2 k \rceil$ GAs,
- Symmetric ID-protocol: $t(2k-1) = \lceil \lambda / \log_2(2k-1) \rceil$ GAs.

Assuming $k = 2^\kappa$, this implies $t(2) \approx \kappa t(k) \approx (\kappa + 1) t(2k-1)$.

Public key size. Instead of a single set element, the public key now consists of $k-1$ set elements, generated using the secret key x and elements of the exceptional set $\mathbb{C}$.

Proof size. We further establish the proof size of the non-interactive version of the ID-protocol in the standard and symmetric cases. To that end, we realize that the prover publishes the challenge-response pair (d, r). The total challenge size is simply the size of the output domain of the hash function, which is $t(s)\lceil \log_2 s \rceil$ bits. The responses are $t(s)$ elements in $\mathbb{Z}_N$, thus have total size at most $t(s)\lceil \log_2 N \rceil$. This gives the total proof size of

- Standard ID-protocol: $\lceil \lambda / \log_2 k \rceil (\lceil \log_2 k \rceil + \lceil \log_2 N \rceil)$ bits,
- Symmetric ID-protocol: $\lceil \lambda / \log_2(2k-1) \rceil (\lceil \log_2(2k-1) \rceil + \lceil \log_2 N \rceil)$ bits.

4 Signatures from the Proposed ID Protocol

The ID protocol in Section 3 can be turned into a signature scheme using the Fiat–Shamir transform [FS87]. Let again $t = t(s) = \lceil \lambda \log_s 2 \rceil$, then the challenges are obtained by hashing the commitments $E_{b_1}, \ldots, E_{b_t}$ and the message m to sign using a hash function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^{t\lceil \log_2 s \rceil}$, modeled as a random oracle. The challenge is obtained as $d = d_1 \parallel \cdots \parallel d_t = \mathsf{H}(E_{b_1}, \ldots, E_{b_t} \parallel m)$.

The signature on m consists of $(m; (r_1, d_1), \ldots, (r_t, d_t))$. The verifier recomputes the $E'_{b_i} = [r_i]E_i$ and checks that indeed $d = \mathsf{H}(E'_{b_1}, \ldots, E'_{b_t} \parallel m)$. The description of the trusted key generation, signing and verification of the signature scheme is presented in Figure 1.

Theorem 4.1. *When the hash function* H *is modelled as a (quantum) random oracle, then the signature scheme in Figure 1 is sEUF-CMA secure.*

Proof. In Lemma 3.1, we proved that the ID-protocol from Section 3 has special soundness and unique responses. Then by Theorem 25 of [DFMS19] the protocol

```
The signature scheme based on our standard (resp. symmetric) ID protocol

$\mathsf{KGen}(1^\lambda)$: To generate the keys, a *trusted authority* acts as follows,
    1. Sample $x \leftarrow \mathbb{Z}/N\mathbb{Z}$.
    2. Run algorithm $\mathsf{XSGen}$ (resp. $\mathsf{SXSGen}$) and obtain an exceptional (resp. superexceptional) set $\mathbb{C} = \{0, c_1 = 1, c_2, \cdots, c_{k-1}\}$
    3. For each $c_i \in \mathbb{C}$ set $E_i = [c_i x]E_0$.
    4. Return $\mathsf{sk} = x$, $\mathsf{pk} = (E_0, E_1, \ldots, E_{k-1})$.
$\mathsf{Sign}(\mathsf{sk}, m)$: To sign a message $m$, the signer performs
    1. For $i = 1, 2, \ldots, t$:
        (a) $b_i \leftarrow \mathbb{Z}/N\mathbb{Z}$.
        (b) $E_{b_i} = [b_i]E_0$.
    2. Set $(d_1, \ldots, d_t) = \mathsf{H}(E_{b_1}, \ldots, E_{b_t} \parallel m)$.
    3. For $i = 1, 2, \ldots, t$:
        (a) set $r_i = b_i - c_{d_i} \cdot x \pmod{N}$.
    4. Return $\{(r_i, d_i)\}_{i=1}^t$.
$\mathsf{Vf}(\{(r_i, d_i)\}_{i=1}^t, m, \mathsf{pk})$: To verify a signature $\{(r_i, d_i)\}_{i=1}^t$ on $m$, one performs:
    1. For $i = 1, 2, \ldots, t$: compute $E'_{b_i} = [r_i]E_i$.
    2. $(d'_1, d'_2, \ldots, d'_t) = \mathsf{H}(E'_{b_1}, \ldots, E'_{b_t} \| m)$.
    3. If $(d_1, d_2, \ldots, d_t) = (d'_1, d'_2, \cdots, d'_t)$ then return $\mathsf{valid}$, else output $\mathsf{invalid}$.
```

Figure 1. The signature scheme based on our standard (resp. symmetric) ID protocol

enjoys the Quantum Proof of Knowledge property. This along with the fact that the protocol has λ bits of min entropy (Lemma 3.1) impies by Theorem 22 of [DFMS19] that the resuting signature scheme obtained via Fiat–Shamir is sEUF-CMA in the QROM. □

Computational cost and signature size. We notice that the number of group actions to be performed in the signature and verification process are the same as in the proof and verification of the non-interactive ID protocol, respectively. Similarly, the size of the signature on m is given by the size of the output domain of the hash function, which depends on inverse the soundness error rate s and is therefore also equal to the proof size of the non-interactive ID protocol.

5 Eliminating the Trusted Setup

In the presented ID protocol (in Section 3), the need for a trusted authority mainly was for ensuring the *well-formedness* of the public key pk. We call a public key $\mathsf{pk} := (E_0, E_1, \ldots, E_{k-1})$ *well-formed*, if for a secret key $x \in \mathbb{Z}_N$ and a set $\mathbb{C} = \{c_0, \ldots, c_{k-1}\}$ it holds that $E_i = [c_i x]E_0$ for $i = 1, \ldots, k-1$ and that $\mathbb{C}$ is a (super-)exceptional set for the case of a (symmetric) HHS.

The proof of special soundness in the main protocol relies on the fact that the elements of pk are well-formed and each one contains a particular multiple of sk. In practice, this trust can be eliminated if the prover generates the keys and proves their *well-formedness*. This proof[2] needs to be generated only once, and a verifier can eliminate the need for a trusted party by verifying it and checking that $\mathbb{C}$ is a (super-)exceptional set (which can be done in polynomial time).

We present two Σ-protocols for a well-formedness proof. The first protocol is more general and proves that a given pk has the correct structure simply by showing that a single commitment-response pair applies to all elements of it. The second protocol, on the other hand, uses an incremental approach, where the correctness of an element E_i of the pk is proven by using elements $E_0, \ldots, E_{i-1}$. By starting from (E_0, E_1), which is well-formed by definition, we can then prove the well-formedness of the entire key incrementally. This approach will turn out to be much more efficient, but only works for exceptional sets of the form $\{0, 1, 2, \ldots, k-1\}$. This protocol also allows for a pk to be upgraded, i.e. to add a new element to a pk with a short proof, that the element is also well-formed.

Both protocols can be made non-interactive using the Fiat–Shamir transform.

5.1 First Approach: General well-formedness proof

We present a Σ-protocol of the following *well-formedness* (WF) relation for a given E_0 and a particular k.

$$\mathbf{L}_{k-1}^{WF} := \{((E_0, E_1, \ldots, E_{k-1}), x, \mathbb{C} = \{c_1, \ldots, c_{k-1}\}) : \bigwedge_{i=1}^{k-1} E_i = [c_i x]E_0\}.$$

Namely, P needs to prove in zero-knowledge that all the elements of the pk are computed using the same secret key x but with different public coefficients $c_1, \ldots, c_{k-1}$. This can be achieved in a straightforward fashion by sampling $b \leftarrow \mathbb{Z}_N$ and commiting to $\hat{E}_i = [c_i b]E_0$ for $i = 1, \ldots, k-1$. The challenge d is binary, or ternary if we assume a symmetric HHS, and the prover can respond with $r = b - dx \mod N$. Finally, the verifier checks if all $\hat{E}_i \overset{?}{=} [c_i r]E_{di}$.

Theorem 5.1. *The above Σ-protocol is correct, HVZK, and special sound with soundness error rate $\frac{1}{3}$.*

Proof. For completeness, we simply realize that $[c_i r]E_{di} = [c_i b - d c_i x][d c_i x]E_0 = [c_i b]E_0 = \hat{E}_i$, which shows that the honest verifier will return *accept*.

For special soundness, given two transcripts $((\hat{E}_1, \ldots, \hat{E}_{k-1}), d, r)$ and $((\hat{E}_1, \ldots, \hat{E}_{k-1}), d', r')$ where $d \neq d'$, and consequently $r \neq r'$ (for non-zero x), we have $[c_i r]E_{di} = [c_i r']E_{d'i}$ for all $i = 1, \ldots, k-1$. Thus an extractor can extract the secret by computing $x = \frac{r - r'}{d' - d} \mod N$.

For the HVZK, given d, a simulator samples $r \leftarrow \mathbb{Z}_N$, then for $i = 1, \ldots, k-1$ sets $\hat{E}_i = [c_i r]E_{di}$. In both the real and the simulated transcripts, r and $\hat{E}_i$ are sampled uniformly at random, leading to indistinguishable distributions. □

[2] Note that the proof does not need to be a proof of knowledge, rather a *sound* proof. Our presented protocol achieves special soundness, which is stronger than what we need. We consider constructing a sound proof system based on isogenies as an interesting future research direction.

5.2 Second Approach: Incremental well-formedness proof

We present our second approach as an algorithm for upgrading a well-formed public key: To this end, assume a prover holds a well-formed public-key $PK_{k-1} = (E_0, E_1, \dots, E_{k-1})$ of size k, where $E_c = [cx]E_0$ for $c = 1, \dots, k-1$.[3] Now, assume the prover wants to add a new element $E_k = [kx]E_0$ to upgrade its public-key to $PK_k = (E_0, E_1, \dots, E_{k-1}, E_k)$. Instead of repeating the full well-formedness proof of Section 5 for PK_k, the prover can create the following proof increment to show, that indeed $E_k = [kx]E_0$. Throughout this section, we denote $\mathbb{C}_k = \{0, \dots, k\}$. We define the language of correct public-key increments

$$\mathbf{L}_k^{Incr.} = \{(PK_{k-1}, E_k) : \text{the new set } \{PK_{k-1} \cup E_k\} \text{ is well-formed}\}.$$

A Σ-protocol for $\mathbf{L}_k^{Incr.}$

$$PK_{k-1} = (E_0, E_1, \cdots, E_{k-1}), E_k$$

$$\begin{array}{lcr} \textbf{Prover}, x & & \textbf{Verifier} \\ b \leftarrow \mathbb{Z}/N\mathbb{Z}, & & \\ \hat{E}_0 \leftarrow [-b]E_0, \quad \hat{E}_k \leftarrow [b]E_k & \xrightarrow{\hat{E}_0, \hat{E}_k} & \\ & \xleftarrow{d} & d \leftarrow \mathbb{C}_k \\ r \leftarrow b + (k-d) \cdot x \mod N & \xrightarrow{r} & \textbf{return: } [-r]\hat{E}_k \overset{?}{=} E_d \\ & & \wedge \quad [r]\hat{E}_0 \overset{?}{=} E_{k-d} \end{array}$$

Theorem 5.2. *The above Σ-protocol is correct, HVZK, and special sound with soundness error rate $\frac{1}{k}$.*

Proof. For completeness, we have that $[-r]\hat{E}_k = [-b - (k-d)x + b + kx]E_0 = [dx]E_0 = E_d$ and $[r]\hat{E}_0 = [b + (k-d)x - b] = [(k-d)x]E_0 = E_{k-d}$.

For special soundness, given two accepting transcripts $((\hat{E}_0, \hat{E}_k), d, r)$ and $((\hat{E}_0, \hat{E}_k), d', r')$ with $d \neq d'$, and consequently $r \neq r'$, we have $[r]E_d = [r']E_{d'}$ and $[-r]E_{k-d} = [-r']E_{k-d'}$, which implies that we can extract $x = \frac{r-r'}{d'-d} \mod N$ from either equation.

Finally, for the HVZK, given a honestly generated $d \leftarrow \mathbb{C}_k$, the simulator samples $r \leftarrow \mathbb{Z}_N$, then computes $\hat{E}_k = [r]E_d$ and $\hat{E}_0 = [-r]E_{k-d}$. Finally, it outputs $((\hat{E}_0, \hat{E}_k), d, r)$ as a simulated transcript. □

5.3 Efficiency and Applications

In order to reach a soundness error of $\leq 2^{-\lambda}$, a protocol with soundness error $1/s$ needs to be repeated at least $t(s) = \lceil \lambda \log_s 2 \rceil$ times.

[3] Note that this protocol does not work for general exceptional sets, only for sets of the form $\{0, 1, \dots, k\}$.

- The first protocol has soundness error $1/3$. For a public key PK_k, at each step, both the prover and verifier compute k group actions, so that the full protocol results in $C_b(k, \lambda) = kt(3)$ group actions per party.
- The second protocol has soundness error $1/(k+1)$. At every step, the prover and verifier have to compute 2 group actions, yielding the total cost $c_I(k, \lambda) = 2t(k+1)$ for the proof of the increment $PK_{k-1} \rightarrow PK_k$. If we want to create the well-formedness proof using only incremented public keys, we find the total cost $C_I(k, \lambda) = \sum_{j=2}^{k} c_I(j, \lambda)$.

It is easy to see, that $c_I(2, \lambda) = C_b(2, \lambda)$ and that $c_I(k, \lambda) < C_b(k, \lambda)$ for $k > 2$. Numerically, we also find, that $C_I(k, \lambda) < C_b(k, \lambda)$ for $k > 16$, independent of λ. Finally, we can optimize well-formedness proofs by combining the two approaches and finding $l < k$, such that a combination of the full well-formedness proof and the incremental proof has minimal cost $C(k, l, \lambda) := C_b(l, \lambda) + \sum_{j=l+1}^{k} c_I(j, \lambda)$. Numerically, we find $l = 7, 8$ to be optimal. Note that this is independent of λ. For $k < 7$, $l = k$ is optimal and equal to $C_b(k, \lambda)$. Asymptotically for $k \rightarrow \infty$, we have $C(k, l = 7, \lambda) \approx C_I(k, \lambda)$.

Applications. We realize that the cost of the well-formedness proofs established in the previous section are quite high for large public keys, which would allow a more efficient ID protocol as presented in Section 3.1. Note that the the well-formedness proofs are not meant to be added to the ID-protocol at every invocation, since this would completely defeat the purpose of having a large public key to increase the efficiency in the first place.

Rather, the idea is to reduce the trust in comparison to our initial proposal in Section 3.1. There are many applications, where having a third party generating your private key is not an option. In such a case, a prover could simply generate its own key pair and send a proof of well-formedness to the trusted party. The trusted party verifies it and can then publish, that the well-formedness is accepted for that particular public key, by e.g. signing it. Thus, the expensive proof and verification have to be performed only once. An example of such an application could for instance be in TLS, where a certificate authority could verify the well-formedness of the public key, before issuing a certificate.

6 Instantiation with CSIDH-512

We instantiate our protocol using the known class-group and relation lattice of the CSIDH-512 parameter set, established in [BKV19]. In order to allow public-keys with more than 36 elements ($k \geq 37$), we work in the subgroup generated by $\mathfrak{g}^{111}$ and identify $N = \#\mathsf{Cl}(O)/111$, which has smallest prime divisor $q_1 = 1407181$ (cf. Section 2.3). We note that, $\log_2(q_1) \approx 2^{20.4}$, which allows our public key sizes to have that same size in case we work with exceptional sets, or up to $\approx 2^{19.4}$, if we work with superexceptional sets. Since the CSIDH-512 parameters set provides an instantiation of a symmetric HHS, we can choose the latter.

Table 1 summarizes different computational and communication costs related to our ID protocol. We use the complexity results established in Section 3.2. In our instantiation, we have the parameters $\lceil \log_2 N \rceil = 251$, $\lceil \log_2 p \rceil = 511$ and choose $\lambda = 128$. For simplicity, we bound the elements in $\mathbb{C}$ by q_1 and can express the public-key size as $(k-1)\lceil \log_2 p \rceil + (k-2)\lceil \log_2 q_1 \rceil = 532k - 553$. In order to give more descriptive examples for the runtime of our protocol, we further use the estimate of 35 ms per GA from [BDPV20], which uses the optimizations from [MR18].

Table 1. Public-key size, non-interactive proof size (or signature size), computational cost and estimated time of proof generation and verification, and computational cost of the optimal well-formedness proof established in Section 5 for various values of k. The row with $k = 2$, shows the efficiency of the basic ID protocol which has a binary challenge space. Runtimes are expressed in Group Actions (GA) and also using the estimate that each GA takes 35 ms for demonstration purposes.

k	PK size	Proof size	Cost	Run time	Well-formedness proof	
2^1	64 B	2552 B	81 GA	2835 ms	—	—
2^2	197 B	1455 B	46 GA	1610 ms	566 GA	19.8 sec
2^5	2.0 KB	704 B	22 GA	770 ms	2082 GA	72.8 sec
2^8	16.6 KB	486 B	15 GA	525 ms	10377 GA	6.1 min
2^{10}	66.4 KB	392 B	12 GA	420 ms	31761 GA	18.5 min
2^{12}	265.9 KB	329 B	10 GA	350 ms	101996 GA	59.5 min
2^{15}	2.1 MB	299 B	9 GA	315 ms	628528 GA	6.1 h
2^{18}	16.6 MB	235 B	7 GA	245 ms	4093141 GA	1.7 days

7 Conclusion

The ID protocol underlying CSI-FiSh [BKV19] allows one to prove knowledge of a secret isogeny, but suffers from a low constant soundness error rate. We were able to arbitrarily decrease the soundness error per round by sampling challenges from exceptional sets, namely sets having certain algebraic properties needed for the extraction. At the same time, this came at the cost of introducing new (structured) public keys that are indexed by the elements of the exceptional set. In the basic form of the protocol, we assumed that both the (structured) public key and the exceptional set were honestly generated by e.g. a trusted authority. We showed that with a 2.1MB public key, this ID protocol generates proofs of size 299 bytes, and its prover and verifier can generate and verify a proof both in 315 milliseconds. Our ID protocol would allow to prove knowledge of the secret key sk of any CSIDH-based primitive with public-key $\mathsf{pk} := (E_0, E_1)$, where $E_1 := [\mathsf{sk}]E_0$.

We also showed how to get rid of the need for a trusted authority by presenting a protocol that allows the prover to convince the verifier that the keys have the required form. This proof takes a combined approach, by first proving the well-formedness of a small subset of the public key, and then iteratively using this to more efficiently prove the well-formedness of further elements.

We also presented the NIZK version of our ID protocol along with the resulting signature scheme obtained by the Fiat–Shamir transform [FS87]. We devote future work to improve the efficiency of the proof of well-formedness of the public keys, as this is the main bottleneck of the trustless version of our protocol. A possible improvement might come from designing *sound-only* proofs as this would not impose strong algebraic conditions on the challenge space for extraction.

Acknowledgments

We would like to thank the anonymous reviewers for their comments. This work has been supported in part by ERC Advanced Grant ERC-2015-AdG-IMPaCT, by the Defense Advanced Research Projects Agency (DARPA) under contract No. HR001120C0085, by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (Grant agreement No. 101020788 - Adv-ERC-ISOCRYPT), by Research Council KU Leuven grant C14/18/067, and by CyberSecurity Research Flanders with reference number VR20192203.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ERC, DARPA, the US Government, Cyber Security Research Flanders. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

AABN02. 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 *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 418–433. Springer, 2002.

BCPS18. Anurag Bishnoi, Pete L Clark, Aditya Potukuchi, and John R Schmitt. On zeros of a polynomial in a finite grid. *Combinatorics, Probability and Computing*, 27(3):310–333, 2018.

BDPV20. Ward Beullens, Lucas Disson, Robi Pedersen, and Frederik Vercauteren. CSI-RAShi: Distributed key generation for CSIDH. *IACR Cryptol. ePrint Arch.*, 2020:1323, 2020.

BKV19. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 227–247, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany.

CJS14. Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.

CLM$^+$18. 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*, pages 395–427. Springer, 2018.

Cou06. Jean Marc Couveignes. Hard homogeneous spaces. *IACR Cryptol. ePrint Arch.*, 2006:291, 2006.

DFG19. 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*, pages 759–789. Springer, 2019.

DFKL$^+$20. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: compact post-quantum signatures from quaternions and isogenies. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 64–93. Springer, 2020.

DFKS18. Luca De Feo, Jean Kieffer, and Benjamin Smith. Towards practical key exchange from ordinary isogeny graphs. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 365–394. Springer, 2018.

DFMS19. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In *Annual International Cryptology Conference*, pages 356–383. Springer, 2019.

DLSV20. Anders Dalskov, Eysa Lee, and Eduardo Soria-Vazquez. Circuit amortization friendly encodings and their application to statistically secure multiparty computation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 213–243. Springer, 2020.

DPV19. Thomas Decru, Lorenz Panny, and Frederik Vercauteren. Faster seasign signatures through improved rejection sampling. In *International Conference on Post-Quantum Cryptography*, pages 271–285. Springer, 2019.

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany.

GPS17. Steven D Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 3–33. Springer, 2017.

HM89. James L Hafner and Kevin S McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of the American mathematical society*, 2(4):837–850, 1989.

JDF11. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International Workshop on Post-Quantum Cryptography*, pages 19–34. Springer, 2011.

KLPT14. David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17(A):418–432, 2014.

MR18. Michael Meyer and Steffen Reith. A faster way to the CSIDH. In *International Conference on Cryptology in India*, pages 137–152. Springer, 2018.

RS06. Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptol. ePrint Arch.*, 2006:145, 2006.

Sch89. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Conference on the Theory and Application of Cryptology*, pages 239–252. Springer, 1989.

Sho94. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

Sto10. Anton Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Advances in Mathematics of Communications*, 4(2):215, 2010.

Sto12. Anton Stolbunov. Cryptographic schemes based on isogenies. 2012.

Unr17. Dominique Unruh. Post-quantum security of Fiat-Shamir. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 65–95. Springer, 2017.

YAJ$^+$17. Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A post-quantum digital signature scheme based on supersingular isogenies. In *International Conference on Financial Cryptography and Data Security*, pages 163–181. Springer, 2017.

A Building Signatures from ID protocols

In order to build a signature scheme from a secure ID protocol, the Fiat–Shamir transformation [FS87] acts as follows. In nutshell, it makes an interactive ID protocol $\Pi = (\mathsf{KGen}, \mathsf{P}, \mathsf{V})$ with c-bit challenges for some integer $c \geq 1$, non-interactive using an RO to generate the challenges. Assume the ID protocol must be run in parallel t times to achieve the soundness error rate $\frac{1}{2^{tc}}$. Let H be an RO that outputs a bit string of length c. Then, the resulting signature can be expressed as follows,

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: as in the setup phase of the ID protocol, given the security parameter, the key generation algorithm KGen returns the public key and secret key.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$: given the secret key sk and a message m to be signed, the signing algorithm Sign first computes the commitments $a_i \leftarrow \mathsf{P}(\mathsf{sk}, r_i)$ for $1 \leq i \leq t$. Then computes $h = H(m, a_1, \cdots, a_t)$. Parses h as the t values $c_i \in \{0,1\}^c$. Computes $z_i \leftarrow \mathsf{P}(\mathsf{sk}, r_i, a_i, c_i)$ for $1 \leq i \leq t$. Outputs the signature $\sigma = (a_1, \cdots, a_2, z_1, \cdots, z_t)$.
- $\{1,0\} \leftarrow \mathsf{Vf}(m, \sigma, \mathsf{pk})$: Given a signature, a message and the public key, it compute $h = H(m, a_1, \cdots, a_t)$. Parse h as the t values $c_i \in \{0,1\}^c$. Using the verifier of the ID protocol, checks that $\mathsf{V}(\mathsf{pk}, a_i, c_i, z_i) = 1$ for all $1 \leq i \leq t$. If V returns 1 for all i then outputs 1, else outputs 0.

It is proven that, starting from a secure ID protocol, the above signature scheme derived by the Fiat–Shamir transform, is unforgeable against chosen-message attacks in the ROM [AABN02].