Zero-knowledge proofs and isogeny-based cryptosystems.

Javier Silva Velón

December 2020

TESI DOCTORAL UPF / 2020
Supervisor: Carla Ràfols
Department of Information and Communication Technologies

As cliché as it sounds,
this one's for my parents.

Thanks

Quite a few people to mention here, so let's get the obvious ones out of the way, in no particular order. Huge thanks to Carla for all the hard work during the last four years. It's great to have a supervisor that one can talk to or ask for help anytime I get stuck. I know I'm not the easiest student to work with, and I have a tendency to get distracted or discuss silly details instead of doing the important work. So I really appreciate that she put up with my shenanigans.

Another easy one is Christophe. He was my dissertation supervisor during my master's, and the person who introduced me to cryptography research. We worked together a lot since. Thanks for including me in meaningful projects, for all the guidance and encouragement. Actually, it was him who sent me the posting for this PhD position. I was quite close to being his student instead of Carla's, and he can be considered an unofficial supervisor of this thesis.

Fortunately, not everything I did at UPF was research. I've been lucky enough to spend quite a lot of time teaching maths to undergraduate engineering students, an experience I really enjoyed, regardless of the messed-up education system that we have. So thanks to Vanesa, Victor and Alfonso for the opportunity. A special mention goes for Vanesa, for involving me in many teaching innovation projects.

During the third year of my PhD, I visited Christophe at the University of Birmingham, where I stayed for three months. I felt very welcomed at the department, and met some nice people there. So shout-out to Georgios, Jose and Zitai. I also enjoyed talking maths, crypto and basketball with Péter, and I really hope that we keep in touch.

Moving to more personal stuff, there's a lot of people who accompanied me since I started as a maths undergraduate in 2011. I have great memories of those first few years with Jorge, Gabi, Pedro and Cristian, discussing exercises and proofs, which at the time seemed impossible and are probably actually trivial, or wondering about why one of the lecturers had a clay brick in his office. I must also include Álvaro, who swapped maths for music, and Noel, who is resuming his own maths journey right now.

Obviously, I am mentioning my parents at some point. The biggest thanks go to them, Javier and Pili, for their unbounded support. Their son comes to the kitchen one morning and tells them that he wants to go to study a master's at a really expensive university overseas. No scholarship or anything. And they just say go, we'll pay for it. Not a single moment of hesitation. Not every parent is in a position to do that, and even less would be willing to do it. Specially because they didn't want me to leave. Yet, they encouraged me to do what I

really wanted to do. I am beyond grateful for that, and it is the reason why this work is dedicated to them.

Big thanks also to my flatmates Jose, Lucas and Mariana, who supported me during the time it took to complete this thesis, and made sure I strayed from sanity just the right amount. Here's to Asian food, board games and complaining about academia. Come to think of it, I've done a lot of complaining to a lot of people. For similar reasons, I also thank the guys back in Santiago: Jonatan, Andrés / Pablo, Óscar and all the crew at the ABSS.

And saved for last, for dramatic effect, the great people at 55.210: Zaira, Federico, Rasoul, Xavi, Aru, Alex, Sergi and Conor. Those of you who were around when I was starting will remember that I used to be on my own, not talking much at lunch and ignoring the group plans after work. But, for whatever reason, you kept insisting until I started to say yes. And I really thank you for that, and I'm happy now to be part of such a cool group. But, apologies to the rest of you, one of these people shines above the rest, and that is of course Zaira. We started our PhDs at the same time, and became good friends though the years. She's one of the greatest people I've ever met. I am not really the same person that arrived in Barcelona back then. And this is, in no small part, thanks to her.

Abstract

In this thesis, we present some public-key cryptographic schemes. This work is divided in two halves. The first half deals with zero-knowledge proofs in the classical setting and under falsifiable assumptions. In particular, we improve upon the efficiency of an argument for linear equations, and we present a proof of correct computation of a circuit that is of size logarithmic in the depth of the circuit. In the second half, we introduce a signature scheme, an encryption scheme and a trapdoor DDH scheme based on isogenies of supersingular elliptic curves. The signature and encryption schemes are secure against quantum adversaries.

Resum

En aquesta tesi presentem alguns esquemes criptogràfics de clau pública. Aquest treball consta de dues parts. La primera meitat tracta de proves de coneixement nul en el context clàssic i basades en hipòtesis falsificables. En particular, millorem l'eficiència d'un argument de coneixement nul per a equacions lineals i presentem una prova de computació correcte d'un circuit que té una mida logarítmica en la profunditat del circuit. A la segona meitat, introduïm un esquema de signatures, un esquema de xifratge i un esquema DDH de trampa basat en isogènies de corbes el·líptiques supersingulars. Els esquemes de signatura i xifrat són segurs contra adversaris quàntics.

Contents

Chapter 1

Introduction

One of the timeless problems in cryptography is the tug-of-war between security and efficiency. We want our schemes to be as secure as possible, but at the same time we do not want them getting in the way, so we also want them to be as fast as possible. Often, these two properties work against each other. This happens at different levels, the most obvious one being increasing parameter sizes to boost security.

A less-obvious ground in which these two properties pull in opposite directions is cryptographic assumptions. The security of a cryptographic scheme is often proven to depend on the computational hardness of a certain mathematical problem. One classical example is the discrete logarithm problem in a cyclic group $\mathbb{G}$, in which we are given a generator g and a random element g^a, and are asked to recover a. Number theorists have been attacking the problem for decades, with notable improvements but without ever finding a computationally efficient solution to the problem. Thus, although unproven, we have a reasonable confidence in the hardness of this problem, and so cryptographers will be happy to prove that their schemes are secure as long as the discrete logarithm problem remains hard.

Over the years, the discrete logarithm has spanned a huge tree of stronger assumptions [63, 81, 132]. That is, any of these can be broken by solving the discrete logarithm problem, while the other implication is not known to be true. In general, the stronger the assumption, the easier it is to reduce security to it. But, on the other hand, this also means that we are relying on an assumption that is less studied, and which potentially gives more power to an attacker trying to breach security. It is natural, then, than the most efficient schemes

are usually those relying on the strongest assumptions.

A good chunk of this work is concerned with studying zero-knowledge proofs, which actually provide a quite extreme example of how things change when we favor efficiency over security. Let us consider the problem of circuit satisfiability, which is an NP-complete problem, and the focus on a significant part of the research in zero-knowledge proofs. On the efficiency side of the spectrum, we have SNARKs [49, 79, 89, 92], which allow us to prove satisfiability of a circuit of any size by just sending a small constant number of group elements. However, SNARKs rely on so-called non-falsifiable assumptions. These receive their name because there is no way to efficiently check whether an adversary is breaking the assumption. Intuitively, these are not really assumptions about the hardness of some problem, but about *how* the adversary works internally. Thus, these are considered to be some of the strongest assumptions. On the other hand, an impossibility result by Gentry and Wichs [80] tells us that there is no hope for finding short non-interactive circuit satisfiability proofs under falsifiable assumptions.

Different lines of research span from the different positions in the security-efficiency spectrum, based on whether we completely favor one over the other, or try to find a reasonable middle ground. In this thesis, we look at various public-key schemes, with different functionalities and in different contexts. A common theme, however, is the focus on better assumptions, while keeping the efficiency of the schemes at a reasonable level. We can also look at the same idea in a different way: improving the efficiency without making the assumptions worse.

More precisely, this work is clearly divided in two halves, resulting from two independent lines of research. The first part deals with zero-knowledge proofs in the classical setting (i.e. adversaries do not have access to a quantum computer). Our main results are the following.

- *Chapter 3: proofs of same opening [146].* We look into zero-knowledge proofs of same opening of two commitments in a bilinear group, and manage to improve the efficiency of the state-of-the-art construction [85]. We also argue that, in its setting, our construction is optimal.

- *Chapter 4: verifiable computation.* We consider zero-knowledge proofs of circuit satisfiability under falsifiable assumptions. By combining ideas from different previous approaches [21, 86], we obtain a proof of size $O(\log n + \log d)$, where n is the size of the secret part of the input and d is the multiplicative depth of the circuit. The public parameters of our construction are circuit-independent, and the security is based on falsifiable assumptions and the random oracle model.

The second half of the thesis focuses on cryptosystems based on isogenies of supersingular elliptic curves. This is, comparatively, a new field, which has attracted some attention as a secure candidate in a post-quantum setting (i.e. adversaries have access to a quantum computer). We obtain the following results.

- *Chapter 5: isogeny signatures [75, 76].* We present two signature schemes based on isogenies of elliptic curves. The main result is the second scheme, which is the first signature scheme to achieve security based on the weakest (best understood) isogeny assumption.
- *Chapter 6: isogeny encryption [57].* We use the attacks of [138] against isogeny problems in a constructive way, obtaining a public-key encryption scheme based on different assumptions than previously known schemes. We argue that, in certain scenarios, our assumptions would be preferred.
- *Chapter 7: trapdoor DDH groups [121].* We revisit the lesser-known primitive of trapdoor DDH groups [59, 148], giving a new construction based on isogenies, which addresses some open problems left by previous works.[1]

The results of the first part lie mostly in the 'middle ground' between security and efficiency, while the second part leans more towards the the security end.

Organization. The thesis starts with a chapter on preliminaries and background knowledge that is used through the following chapters, but contains no new results. Afterwards, each chapter corresponds to one paper, some already published, some in preprint stage. The contents are essentially the same as in the papers, with only minor modifications to remove redundancies, unify notation and ensure a more cohesive document.

[1] We note that, despite being isogeny-based, this construction is only secure in the classical setting, since hardness of the CDH problem is required.

Chapter 2

Preliminaries

2.1 Provable security

Modern cryptography relies on mathematical proofs of security, in which we reduce the security properties of a scheme to the hardness of computational problems. In most cases, we model security properties and their underlying assumptions in terms of interactive games between algorithms, called *security games*, and probability events.

Our notation will be as follows. We represent the process of sampling an element x from a set S as $x \leftarrow \mathsf{S}$. Unless stated otherwise, we assume that the sampling occurs with respect to the uniform distribution on S. The notation Pr[sampling : event] means the probability of the event on the right happening, after the elements involved are produced as described on the left. For example,

$$\mathsf{Pr}\left[x \leftarrow \{0,1\} : x = 1\right]$$

denotes the probability of a uniformly random bit being 1.

We denote algorithms with uppercase calligraphic characters, e.g. $\mathcal{A}$. We write $y = \mathcal{A}(x)$ for the deterministic algorithm $\mathcal{A}$ outputting y, given input x. If the algorithm involves some randomness, we write $y \leftarrow \mathcal{A}(x)$ instead. Sometimes, we might want to emphasize the randomness explicitly. Then we can de-randomize it by including the randomness r as input: $y = \mathcal{A}(x; r)$. Also, quite often we will encounter algorithms that take some public parameters pp as input, which we might want to de-emphasize. In this case, we write $y \leftarrow \mathcal{A}_{pp}(x)$ instead of $y \leftarrow \mathcal{A}(pp, x)$. The same algorithm might appear at different stages of

a security game. We assume that such algorithms are *stateful*, that is, they keep a record of all the data they handled in previous stages (in particular previous input).

Asymptotic security. Security is measured asymptotically, through a *security parameter* $\lambda \in \mathbb{N}$. This is a global parameter, which quantifies the security we can expect from a given scheme. The precise meaning of security is different for each primitive, and will be discussed later for each case. It is standard to measure efficiency against security, by assessing the costs of the scheme (e.g. the computations of each party, the communication between them or the size of the public parameters) for any given λ. More precisely, we say that a function $f : \mathbb{N} \to [0, 1]$ is *negligible* if

$$f(\lambda) = O\left(\frac{1}{p(\lambda)}\right),$$

for any positive polynomial p. We denote a negligible function of λ by $\mathsf{negl}(\lambda)$. We also make use of the opposite definition. We say that f is *overwhelming* if $1 - f(\lambda)$ is negligible.

Since we deal with probability distributions over finite sets, a cheating player (whatever that means in the context of each scheme) might succeed by just guessing the answer at random. We use this as a baseline for measuring how 'effective' an adversary is at breaking a certain scheme. The distance between the success probability of an adversary $\mathcal{A}$ and the success of the random guessing approach is what we call the *advantage* of the adversary $\mathcal{A}$, denoted by $\mathsf{Adv}\mathcal{A}$.

Then, very generally, we will consider a scheme secure if the advantage of any malicious player can be bounded by a function $\mathsf{negl}(\lambda)$. We usually distinguish between three flavors of security:

- *Computational:* the pool of potential adversaries is restricted to those that run in probabilistic polynomial time (PPT) in λ.
- *Statistical:* the negligible advantage still holds for unbounded adversaries.
- *Perfect:* the advantage is not only negligible, but exactly 0.

Additionally, adversaries can be either classical or quantum. The former is only allowed to run classical algorithms, while the latter can run quantum algorithms and make queries in superposition, and thus has the potential to run stronger attacks. For example, a classical PPT adversary cannot solve a

discrete logarithm instance, since there is not known PPT algorithm for the discrete logarithm problem, but a quantum PPT adversary can easily break it by using Shor's algorithm [149].

Although seemingly meaningless in practice, the difference between computational and statistical security might have some powerful implications. A case that illustrates this distinction is the Fiat–Shamir transformation, which takes a zero-knowledge proof of knowledge and derives a signature scheme from it. Unruh [157] proved, for quantum adversaries, that the security of the transform depends on the *statistical* security of the former against cheating provers, even if the end goal is just computational security.

Tigthness. Let $\mathcal{A}$ be any PPT adversary against a certain scheme, which we wish to prove secure. Ideally, we want to prove the existence of another PPT algorithm $\mathcal{B}$ against a hard mathematical problem, such that $\mathcal{B}$ runs in roughly the same time as $\mathcal{A}$, and

$$\mathsf{Adv}\mathcal{A}(1^\lambda) \leq \mathsf{Adv}\mathcal{B}(1^\lambda).$$

This proves that, if we can break the scheme with a certain probability, then with at least the same probability, and in the same time, we can solve an instance of the mathematical problem of the same size. Thus, if the problem is assumed to be hard, then necessarily $\mathsf{Adv}\mathcal{A}$ is negligible in λ.

However, in practice the situation is not always like this. Maybe $\mathcal{B}$ is slower than $\mathcal{A}$, often due to the fact that $\mathcal{B}$ has to run $\mathcal{A}$ many times, or maybe $\mathcal{B}$ fails sometimes, even when $\mathcal{A}$ succeeds, and our bound is more like

$$\mathsf{Adv}\mathcal{A}(1^\lambda) \leq q \cdot \mathsf{Adv}\mathcal{B}(1^\lambda),$$

for some $q \in \mathbb{N}$. In both these cases, it is clear that we are not getting the same security guarantee than in the previous case. The notion of *tigthness* is introduced to deal with these differences. Assume that the algorithm $\mathcal{A}$ has advantage $\varepsilon_{\mathcal{A}}$ and runs in time $t_{\mathcal{A}}$, and the algorithm $\mathcal{B}$ has advantage $\varepsilon_{\mathcal{B}}$ and runs in time $t_{\mathcal{B}}$. Then, the *tightness loss* is defined as the factor μ such that

$$\frac{t_{\mathcal{B}}}{\varepsilon_{\mathcal{B}}} = \mu \frac{t_{\mathcal{A}}}{\varepsilon_{\mathcal{A}}}.$$

Informally, we say that a scheme is tight when the tightness loss coefficient is small. Tightness loss gives us a measure of how 'bad' a security reduction is, which in turn tells us how much we need to increase the parameter sizes of our scheme to compensate for this loss, and attain the desired security level.

Oracles. Often in security definitions we want to give some extra power to an attacker $\mathcal{A}$, like access to the outputs of an algorithm $\mathcal{O}$ involved in our cryptographic scheme, but without giving them the secret information required to operate $\mathcal{O}$ by themselves.

For example, in an encryption scheme, we might want to give an attacker the ability to obtain decryptions of any ciphertext c, except the ciphertext c^* that they actually want to decrypt. We do so by giving $\mathcal{A}$ black-box access to an algorithm $\mathcal{O}$ that receives c and, if $\mathsf{c} \neq \mathsf{c}^*$, outputs the decryption of c. In this case, we say that $\mathcal{A}$ has *oracle access* to $\mathcal{O}$, or that $\mathcal{O}$ is an *oracle* for $\mathcal{A}$.

The random oracle model. The random oracle model [12] assumes the existence of a public oracle $\mathcal{H}$ which produces an uniformly random output. That is, $\mathcal{H}$ is a publicly computable random function treated as a black box. When implementing a scheme that involves random oracles, $\mathcal{H}$ is replaced by a suitable hash function. If the security of the scheme was argued in the random oracle model, we cannot conclude the security in the standard model. Nevertheless, almost no attacks are known for schemes proven to be secure in the random oracle model, except for a few examples [34, 136] explicitly constructed to prove this separation. Thus, a proof in this model is still a reasonable security guarantee.

The random oracle model has proven to be very useful for constructing cryptographic primitives. In particular, the Fujisaki–Okamoto transformation [69] can be used to strengthen the security of any encryption scheme, and the Fiat–Shamir [67] transformation can be used to remove interaction or produce signature schemes from many zero-knowledge proofs. Both transforms are only known to be secure in the random oracle model.

2.2 Cryptographic primitives

This section reviews some very well-known cryptographic primitives, namely encryption, commitment schemes, zero-knowledge proofs and signatures. In the honest execution of any of these schemes, there are two parties involved that communicate with each other. Their names vary slightly depending on the primitive, but we will consistently denote them as $\mathcal{P}$ and $\mathcal{V}$, with the former taking the role of the 'sender' and the latter being the 'receiver'. Moreover, we will assume the existence of some public data that has not been tampered with and can be accessed by both parties. Formally, each of these is a PPT algorithm.

2.2.1 Commitments

Suppose that $\mathcal{P}$ wants to make a prediction about something that will happen later. On one hand, $\mathcal{P}$ wants his prediction to be hidden from $\mathcal{V}$ until they decide to reveal it, but also $\mathcal{V}$ wants to be sure that $\mathcal{P}$ cannot change their prediction after the fact. A solution is that $\mathcal{P}$ writes their prediction and puts it into a safe that is given to $\mathcal{V}$, while $\mathcal{P}$ keeps the key. This way, $\mathcal{V}$ cannot see the prediction yet, but $\mathcal{P}$ cannot alter the content of the safe.

Commitment schemes are the cryptographic analogue of the safe in the example. More precisely, a *commitment scheme* is a triple of algorithms

$$(\mathsf{Setup}, \mathsf{Com}, \mathsf{Verify})$$

that works as follows.

- A trusted party runs the Setup algorithm, which takes as input the security parameter and produces a public string ck, called the *commitment key*.
- $\mathcal{P}$ runs Com, which takes as input ck and a message m and produces $\mathsf{c} = \mathsf{Com}_{ck}(m; r)$, where r is some randomness chosen by $\mathcal{P}$.
- Given an *opening* (m, r) of a commitment c, $\mathcal{V}$ runs the algorithm Verify, which checks that $\mathsf{c} = \mathsf{Com}_{ck}(m; r)$.

A commitment scheme verifies the following two properties. Let R be the randomness space.

Definition 1 (Hiding)**.** *A commitment scheme is* hiding *if, for all PPT adversaries $\mathcal{A}$,*

$$\left| \mathsf{Pr} \left[\begin{array}{l} ck \leftarrow \mathsf{Setup}(1^\lambda), m_0, m_1 \leftarrow \mathcal{A}(1^\lambda, ck), \\ b \leftarrow \{0,1\}, r \leftarrow \mathsf{R}, \mathsf{c} \leftarrow \mathsf{Com}_{ck}(m_b; r), \tilde{b} \leftarrow \mathcal{A}(\mathsf{c}) \end{array} : \tilde{b} = b \right] - \frac{1}{2} \right|$$

is negligible in λ.

Definition 2 (Binding)**.** *A commitment scheme is* binding *if, for all PPT adversaries $\mathcal{A}$,*

$$\mathsf{Pr} \left[\begin{array}{l} ck \leftarrow \mathsf{Setup}(1^\lambda) \\ m_0, m_1, r_0, r_1 \leftarrow \mathcal{A}(1^\lambda, ck) \end{array} : \begin{array}{c} m_0 \neq m_1 \\ \mathsf{Com}_{ck}(m_0; r_0) = \mathsf{Com}_{ck}(m_1; r_1) \end{array} \right]$$

is negligible in λ.

Intuitively, the hiding property means that it is hard to tell which message a commitment contains, and the binding property means that it is hard to produce two openings of the same commitment, ensuring that $\mathcal{P}$ cannot change their mind after committing.

We say that a commitment scheme is *homomorphic* if

$$\mathsf{Com}_{ck}(m_0; r_0) + \mathsf{Com}_{ck}(m_1; r_1) = \mathsf{Com}_{ck}(m_0 + m_1; r_0 + r_1).$$

for all commitment keys, messages and randomness, where in each case + represents a certain operation in the corresponding space. There is a second property, sometimes called *homomorphy with respect to the commitment key*, which means that the message can 'absorb' the commitment key:

$$\mathsf{Com}_{ck+x}(m) = \mathsf{Com}_{ck}(m + x).$$

A classical example of homomorphic commitments is the *Pedersen commitment scheme*. Let $\mathbb{G}$ be a multiplicative group and let $ck = \{g, h\}$ be fixed group elements. To commit to a message m with randomness r, $\mathcal{P}$ computes $\mathsf{c} = h^r g^m$. This scheme is perfectly hiding and computationally binding based on the discrete logarithm assumption, and it is easy to see that it is homomorphic, in both senses described above. It is easy to generalize to multiple messages as

$$\mathsf{Com}_{ck}(m_1, \ldots, m_n; r) = h^r \prod_{i=1}^{n} g_i^{m_i},$$

where $ck = (g_1, \ldots, g_n, h)$. This version is called *Pedersen multi-commitment scheme*.

When there is no ambiguity with respect to the commitment key ck, we write $\mathsf{Com}_{ck}(m; r)$ as $\mathsf{Com}(m; r)$ for simplicity of the presentation. Homomorphic commitment schemes are often used as a building block for zero-knowledge proofs.

2.2.2 Encryption

An encryption scheme is a cryptographic primitive which ensures that only the intended recipient $\mathcal{V}$ of a message is able to read its contents. A public-key encryption scheme can be seen as a commitment scheme, with a decryption mechanism that requires a secret key owned by $\mathcal{V}$. More precisely, an *encryption scheme* is a triple of algorithms

$$(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$$

that works as follows.

- $\mathcal{V}$ runs the Setup algorithm, which takes as input the security parameter and produces a tuple (pk, sk), where pk is a *public key*, known by everyone, and sk is a *secret key*, known only by $\mathcal{V}$.
- $\mathcal{P}$ runs Enc, which takes as input pk and a *plaintext* m and produces the *ciphertext* $\mathsf{c} = \mathsf{Enc}_{pk}(m; r)$, where r is some randomness chosen by $\mathcal{P}$.
- $\mathcal{V}$ runs the algorithm Dec, which takes as input c and sk, and computes $\tilde{m} = \mathsf{Dec}(sk, \mathsf{c})$.

We say that an encryption scheme is *complete* if $\tilde{m} = m$ with overwhelming probability. What follows is a sequence of increasingly hard security definitions for public-key encryption schemes.

Definition 3. *Let* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme, and let* M *be the message space.*

- *We say that the encryption scheme is* secure against key recovery *if, for any PPT adversary* $\mathcal{A}$*,*

$$\mathsf{Pr}\left[\begin{matrix}(pk, sk) \leftarrow \mathsf{Setup}(1^\lambda) \\ sk' \leftarrow \mathcal{A}(pk)\end{matrix} : \forall m \in \mathsf{M}, \mathsf{Dec}_{sk'}\left(\mathsf{Enc}_{pk}(m)\right) = m\right]$$

  is negligible in λ*.*

- *We say that the encryption scheme is* secure against one-way chosen plaintext attacks (OW-CPA secure) *if, for any PPT adversary* $\mathcal{A}$*,*

$$\mathsf{Pr}\left[\begin{matrix}(pk, sk) \leftarrow \mathsf{Setup}(1^\lambda), m^* \leftarrow \mathsf{M}, \\ \mathsf{c}^* \leftarrow \mathsf{Enc}_{pk}(m^*), \tilde{m} \leftarrow \mathcal{A}(pk, \mathsf{c}^*)\end{matrix} : \tilde{m} = m^*\right]$$

  is negligible in λ*.*

- *We say that the encryption scheme is* secure against indistinguishable chosen-plaintext attack (IND-CPA secure) *if, for any PPT adversary* $\mathcal{A}$*,*

$$\left|\mathsf{Pr}\left[\begin{matrix}(pk, sk) \leftarrow \mathsf{Setup}(1^\lambda), m_0, m_1 \leftarrow \mathcal{A}(pk), \\ b \leftarrow \{0,1\}, \mathsf{c}^* \leftarrow \mathsf{Enc}_{pk}(m_b), \tilde{b} \leftarrow \mathcal{A}(pk, \mathsf{c}^*),\end{matrix} : \tilde{b} = b\right] - \frac{1}{2}\right|$$

  is negligible in λ*.*

- *We say that the encryption scheme is* secure against indistinguishable chosen-ciphertext attack (IND-CCA secure) *if, for any PPT adversary $\mathcal{A}$,*

$$\left| \mathsf{Pr} \begin{bmatrix} (pk, sk) \leftarrow \mathsf{Setup}(1^\lambda), m_0, m_1 \leftarrow \mathcal{A}^{\mathcal{O}(\cdot)}(pk), \\ b \leftarrow \{0,1\}, \mathsf{c}^* \leftarrow \mathsf{Enc}_{pk}(m_b), \tilde{b} \leftarrow \mathcal{A}^{\mathcal{O}(\cdot)}(pk, \mathsf{c}^*), \end{bmatrix} : \tilde{b} = b \right] - \frac{1}{2} \right|$$

is negligible in λ, where $\mathcal{O}(\mathsf{c})$ returns $\mathsf{Dec}_{sk}(\mathsf{c})$ for $\mathsf{c} \neq \mathsf{c}^*$.

A related notion is that of a *key encapsulation mechanism (KEM)*, which is essentially an encryption scheme with a large enough message space that is used to securely send a symmetric key. This key will later be used for a symmetric encryption scheme. More precisely, the encapsulation algorithm Enc, with pk as input, produces a symmetric key $K \in \mathsf{K}$ and a ciphertext c for the message K. Upon getting c, the receiver runs the decapsulation algorithm $\mathsf{Dec}(\mathsf{c})$, recovering K.

The security definitions for KEMs are almost the same, with the subtlety that the adversary of indistinguishability games no longer chooses the two messages, since the symmetric keys are random elements of a certain large space.

Definition 4. *Let $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ be a KEM, and let K be the symmetric key space.*

- *We say that the KEM is* secure against indistinguishable chosen-plaintext attack (IND-CPA secure) *if, for any PPT adversary $\mathcal{A}$,*

$$\left| \mathsf{Pr} \begin{bmatrix} (pk, sk) \leftarrow \mathsf{Setup}(1^\lambda), b \leftarrow \{0,1\}, \\ (K_0^*, \mathsf{c}^*) \leftarrow \mathsf{Enc}(pk), K_1^* \leftarrow \mathsf{K}, \tilde{b} \leftarrow \mathcal{A}(pk, K_b^*, \mathsf{c}^*), \end{bmatrix} : \tilde{b} = b \right] - \frac{1}{2} \right|$$

is negligible in λ.

- *We say that the KEM is* secure against indistinguishable chosen-ciphertext attack (IND-CCA secure) *if, for any PPT adversary $\mathcal{A}$,*

$$\left| \mathsf{Pr} \begin{bmatrix} (pk, sk) \leftarrow \mathsf{Setup}(1^\lambda), b \leftarrow \{0,1\}, \\ (K_0^*, \mathsf{c}^*) \leftarrow \mathsf{Enc}(pk), K_1^* \leftarrow \mathsf{K}, \tilde{b} \leftarrow \mathcal{A}^{\mathcal{O}(\cdot)}(pk, K_b^*, \mathsf{c}^*), \end{bmatrix} : \tilde{b} = b \right] - \frac{1}{2} \right|$$

is negligible in λ, where $\mathcal{O}(\mathsf{c})$ returns $\mathsf{Dec}_{sk}(\mathsf{c})$ for $\mathsf{c} \neq \mathsf{c}^*$.

2.2.3 Signatures

A signature scheme is a cryptographic primitive that aims to provide integrity of a message, i.e. that the message has not been modified in transit, and authenticity of the sender, i.e. the sender is not being impersonated. The procedure is very similar to encryption, except that the secret key is now used by the sender and the public key by the receiver. More precisely, a signature scheme is a tuple of algorithms $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify})$ that work as follows.

- $\mathcal{P}$ runs the Setup algorithm, which takes as input the security parameter and produces a tuple (pk, sk), where pk is a *public key* and sk is a *secret key*, known only by $\mathcal{P}$.
- $\mathcal{P}$ runs Sign, which takes as input sk and a *message* m and produces the *signature* $\sigma = \mathsf{Sign}_{sk}(m; r)$, where r is some randomness chosen by $\mathcal{P}$.
- $\mathcal{V}$ runs the algorithm Verify, which takes as input m, σ and pk, and outputs either 0 (reject) or 1 (accept).

We say that an encryption scheme is *complete* if an honest signature of a message is accepted with overwhelming probability.

Definition 5. *A signature scheme* $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify})$ *is* existentially unforgeable against chosen-message attacks (EUF-CMA) *if, for any PPT adversary* $\mathcal{A}$*,*

$$\mathsf{Pr}\left[\begin{matrix}(pk, sk) \leftarrow \mathsf{Setup}(1^\lambda) \\ (m, \sigma) \leftarrow \mathcal{A}^{\mathsf{Sign}_{sk}(\cdot)}(pk)\end{matrix} : 1 \leftarrow \mathsf{Verify}_{pk}(m, \sigma) \wedge m \notin \mathsf{Q}\right]$$

is negligible in λ*, where* Q *is the set of messages that* $\mathcal{A}$ *has queried to the* $\mathsf{Sign}_{sk}(\cdot)$ *oracle.*

Intuitively, this means that an adversary cannot produce a message and a valid signature for it, even after seeing signatures of other messages of their choice.

2.2.4 Proof systems

We now turn our attention to a cryptographic primitive that will be ubiquitous through this work: (zero-knowledge) proof systems, and their many flavors. We start by discussing proof systems.

A proof system is a cryptographic primitive in which a prover $\mathcal{P}$ wishes to prove to a verifier $\mathcal{V}$ that a *statement* u is in a certain language $\mathcal{L}$. The prover is

in possession of a *witness*, which allows to check efficiently that the statement is in the language. We denote the corresponding relation by R, that is, $(u, w) \in R$ if and only if w is a valid witness for $u \in \mathcal{L}$. At the end of the process, the verifier either accepts the proof (outputs 1) or rejects it (outputs 0).

For example, let $\mathbb{G}$ be a group of prime order p and let $g \in \mathbb{G}$ be a generator. Then we can consider the language of DDH tuples

$$\mathcal{L} = \left\{ \left(g, g^a, g^b, g^{ab}\right) \mid a, b \in \mathbb{F}_p \right\}.$$

In this case, either a or b are valid witnesses for the statement (g, g^a, g^b, g^c), since any of them allows to efficiently check that the tuple is in the language.

A proof system is *interactive* if it requires input from the verifier to proceed, and *non-interactive* it it consists of just one message from the prover, which the verifier can check offline. A relation R is called a *hard relation* if it is computationally hard to produce a valid witness w for a random statement $u \in \mathcal{L}$.

Formally, a proof system is described by the three algorithms $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$, which verify the following properties. We denote the transcript of the interaction between $\mathcal{P}$ and $\mathcal{V}$ as $tr \leftarrow (\mathcal{P} \rightleftarrows \mathcal{V})$, and we assume the existence of a *common reference string (CRS)* that is publicly available.

Definition 6. *A proof system is* complete *if, for any* $(u, w) \in R$*,*

$$\mathsf{Pr}[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), tr \leftarrow (\mathcal{P}(\mathsf{crs}, u, w) \rightleftarrows \mathcal{V}(\mathsf{crs}, u)) : 0 \leftarrow \mathcal{V}(tr)]$$

is negligible in λ*.*

Definition 7. *A proof system is* sound *if, for any PPT adversary* $\mathcal{A}$

$$\mathsf{Pr}[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), u \leftarrow \mathcal{A}(\mathsf{crs}), tr \leftarrow (\mathcal{A}(u) \rightleftarrows \mathcal{V}(\mathsf{crs}, u)), 1 \leftarrow \mathcal{V}(tr) : u \notin \mathcal{L}]$$

is negligible in λ*.*

Intuitively, the first property states that an honest prover must succeed, and the second states that a cheating prover, trying to prove a false statement, must fail. We often make the distinction between proof, when Definition 7 holds for unbounded adversaries, and argument, when it only holds for PPT adversaries. If one does not want to emphasize the flavour of soundness, the word proof is used generically for both.

A stronger notion of soundness is that of knowledge soundness, which means that the prover is not only proving that the statement is true, but is also proving knowledge of the corresponding witness. This is formalized as follows:

Definition 8. *A proof system is* knowledge sound *if there exists an extractor $\mathcal{E}$ such that for any PPT prover $\tilde{\mathcal{P}}$ that convinces $\mathcal{V}$ with non-negligible probability*

$$\left|\mathsf{Pr}\left[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), u \leftarrow \tilde{\mathcal{P}}(\mathsf{crs}); w \leftarrow \mathcal{E}(\mathsf{crs}, u)^{\mathcal{P}\tilde{(\cdot)}} : (u, w) \in R\right] - 1\right|$$

is negligible in λ, where the extractor has oracle access to $\tilde{\mathcal{P}}$.

The rationale of this definition is that we can think of what someone 'knows' as what it is able to compute in PPT. Thus, the knowledge of $\tilde{\mathcal{P}}$ about the witness that can be deduced from its interaction with the verifier can be formalized as the knowledge that we can extract in PPT from watching $\tilde{\mathcal{P}}$ convince the verifier. Hence, if the verifier accepts the proof, it means that the prover 'knows' the secret.

For example, Schnorr proposed a proof of knowledge for the discrete logarithm problem, in which the prover tries to convince the verifier that he knows x such that $g^x = h$ for some g, h elements of a prime order group $\mathbb{G}$. Note that given any $h \in \mathbb{G}$, the statement is trivially true, so there is no point in producing a proof for it. However, proving knowledge of a witness x is a stronger claim, which makes the problem non-trivial.

Another property desired in many contexts is *zero-knowledge*, which means that the interaction between the two parties does not reveal anything about the statement apart from the fact that the statement is true, even for a dishonest verifier. To formalize this, we say that there exists a simulator $\mathcal{S}$ such that $\mathcal{S}$ can produce transcripts indistinguishable from real ones only knowing the statement and any prior information than the verifier might have. Note that the simulator does not have access to the witness.

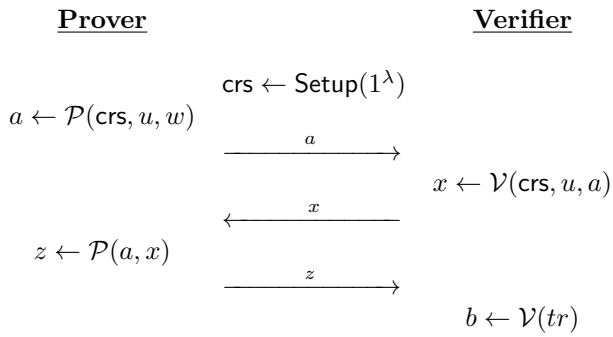
Definition 9 (Zero-knowledge)**.** *A proof system is* zero-knowledge *if there exists a PPT algorithm $\mathcal{S}$ such that, for any PPT verifier $\tilde{\mathcal{V}}$ and any string $y \in \{0,1\}^*$ of prior information,*

$$\left|\begin{array}{l}\mathsf{Pr}\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda),\\ (u, w) \leftarrow \tilde{\mathcal{V}}(\mathsf{crs}), \qquad\qquad : (u, w) \in R \wedge \tilde{\mathcal{V}}(tr) = 1\\ tr \leftarrow (\mathcal{P}(\mathsf{crs}, u, w) \rightleftarrows \mathcal{V}(u, y))\end{array}\right] -\\ -\ \mathsf{Pr}\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda),\\ (u, w) \leftarrow \tilde{\mathcal{V}}(\mathsf{crs}), : (u, w) \in R \wedge \tilde{\mathcal{V}}(tr) = 1\\ tr \leftarrow \mathcal{S}(\mathsf{crs}, u, y)\end{array}\right]\end{array}\right|$$

is negligible in λ.

Sigma protocols

So far, we have not specified how $\mathcal{P}$ and $\mathcal{V}$ interact. One particular very common form of interaction are *Sigma protocols*, which follow a specific structure. In its simplest form, a Sigma protocol follows this pattern:



This is a *three-move* interactive protocol. The first message a sent by $\mathcal{P}$ is usually referred to as the *initial message*, whereas x is called the *challenge* and z is called the *response*. Moreover, the challenge is sampled from the uniform distribution corresponding to the challenge space. When messages from the adversary are uniformly random and independent from each other, we say that the scheme is *public coin*.

A Sigma protocol is said to be *non-trivial* if the bit-length of the challenge is at least λ. Note that we can make a Sigma protocol non-trivial through parallel repetition. A Sigma protocol is said to be *recoverable* if there is a deterministic polynomial time algorithm Rec such that, if (a, x, z) is the transcript of an honest execution of the protocol for the statement u, then $\mathsf{Rec}_{\mathsf{crs}}(u, x, z) = a$.

In the context of Sigma protocols, the definition of soundness is often replaced by the following. It says that an extractor is able to produce a witness, given n accepting transcripts with the same initial message and different challenges.

Definition 10 (n-special soundness)**.** *For $n \in \mathbb{N}$, a Sigma protocol is* n-special sound *if there exists a PPT extractor $\mathcal{E}$ such that for any PPT prover $\tilde{\mathcal{P}}$ that convinces $\mathcal{V}$ with non-negligible probability*

$$\mathsf{Pr}\left[\begin{matrix}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), (u, a, x_1, \ldots x_n, z_1, \ldots, z_n) \leftarrow \tilde{\mathcal{P}}(\mathsf{crs}), \\ w \leftarrow \mathcal{E}(\mathsf{crs}, T)\end{matrix} : (u, w) \in R\right]$$

is overwhelming in λ, where $T = (u, a, x_1, \ldots x_n, z_1, \ldots, z_n)$ is such that for all $i = 1, \ldots, n$, we have that $\mathcal{V}(u, (a, x_i, z_i)) = 1$.

Note that special soundness is a form of knowledge soundness. It is easy to generalize a Sigma protocol to a $(2m + 1)$-move protocol, in which an initial message is sent by the prover, and then verifier and prover exchange uniformly random challenges $x^{(i)}$ and responses $z^{(i)}$, for $i = 1, \ldots, m$. To generalize the soundness property, we consider a tree of accepting transcripts with the following structure. We start from the initial message a, from which we have n_1 different challenges for the first move of the verifier. From each of those we have n_2 different challenges for the second move, and so on. At the end of each path $a, x^{(1)}, \ldots, x^{(m)}$, we attach a valid tuple of responses $(z^{(1)}, \ldots z^{(m)})$.

Definition 11 ($(n_1, \ldots, n_m)$-special soundness)**.** *For $n_1, \ldots, n_m \in \mathbb{N}$, a Sigma protocol is* $(n_1, \ldots, n_m)$-special sound *if there exists a PPT extractor $\mathcal{E}$ such that for any PPT prover $\tilde{\mathcal{P}}$ that convinces $\mathcal{V}$ with non-negligible probability*

$$\left|\mathsf{Pr}\left[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), T \leftarrow \tilde{\mathcal{P}}(\mathsf{crs}); w \leftarrow \mathcal{E}(\mathsf{crs}, T) : (u, w) \in R\right] - 1\right|$$

is negligible in λ, where T is a $(n_1, \ldots, n_m)$-tree of accepting transcripts as described above.

For Sigma protocols, we can also consider a slightly different definition of zero-knowledge.

Definition 12 (Special honest verifier zero-knowledge (SHVZK))**.** *A Sigma protocol is* special honest verifier zero-knowledge *if there exists a PPT algorithm $\mathcal{S}$ such that for any PPT adversary $\tilde{\mathcal{A}}$ and any string $y \in \{0,1\}^*$ of prior information*

$$\left|\begin{array}{l} \mathsf{Pr}\left[\begin{array}{ll} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), & \\ (u, w, x_1, \ldots, x_m) \leftarrow \tilde{\mathcal{A}}(\mathsf{crs}), & : (u, w) \in R \text{ and } \tilde{\mathcal{A}}(tr) = 1 \\ tr \leftarrow (\mathcal{P}(\mathsf{crs}, u, w) \rightleftarrows \mathcal{V}(\mathsf{crs}, u, y)) & \end{array}\right] - \\ - \mathsf{Pr}\left[\begin{array}{ll} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), & \\ (u, w) \leftarrow \tilde{\mathcal{A}}(\mathsf{crs}), & : (u, w) \in R \text{ and } \tilde{\mathcal{A}}(tr) = 1 \\ tr \leftarrow \mathcal{S}(\mathsf{crs}, u, x_1, \ldots, x_m, y) & \end{array}\right] \end{array}\right|$$

is negligible in λ, where $\mathcal{V}$ sends the challenge x_i in round i, and the transcripts of both cases contain the challenges x_i for all $i = 1, \ldots, m$.

While SHVZK is a weaker property than full zero-knowledge as described in Definition 9, efficient transformations are known to obtain full zero-knowledge arguments from SHVZK arguments in the CRS model [77].

Generic transformations

A standard technique to transform Sigma protocols into non-interactive protocols is the *Fiat–Shamir transformation* [67]. Essentially, the Fiat–Shamir transformation exploits the fact that the protocol is public coin, so the challenges sent by the verifier are uniformly random, and thus can be replaced by a hash of the statement and the previous messages without affecting the distribution of the transcripts. Let $H : \{0,1\}^* \to \{0,1\}^c$ be a hash function, modelled as a random oracle, and let $(\mathcal{P}, \mathcal{V})$ be the prover and verifier of a non-trivial Sigma protocol with challenge of length c, as described above. The Fiat–Shamir transform is as follows.

$\overline{\mathcal{P}}(u, w)$:	$\overline{\mathcal{V}}(u, \pi)$:
$a \leftarrow \mathcal{P}(\mathsf{crs}, u, w)$	$x = H(a)$
$x = H(a)$	$b \leftarrow \mathcal{V}(u, a, x, z)$
$z \leftarrow \mathcal{P}(a, x)$	**return** b
return $\pi = (a, z)$	

Figure 2.1: Fiat–Shamir transform

Theorem 1. *Let $(\mathcal{P}, \mathcal{V})$ be a Sigma protocol that is complete, knowledge sound and zero-knowledge. Then $(\overline{\mathcal{P}}, \overline{\mathcal{V}})$, obtained as in Figure 2.1, is a complete, knowledge sound and zero-knowledge non-interactive zero-knowledge proof, in the random oracle model.*

If the Sigma protocol is recoverable, one can send the challenge x instead of the initial message a, since the verifier can recompute a and then verify that the challenge is correct. This saves some space, as a is usually larger than x.

The security of the Fiat–Shamir transform was initially proven by Pointcheval and Stern [142,143] for three-move Sigma protocols, in the random oracle model and against classical adversaries. The key idea, known as the *forking lemma*, is that a PPT adversary $\mathcal{A}$ that can, with non-negligible probability, produce an accepting transcript without knowing the witness can be rewinded to produce more transcripts with the same initial message with non-negligible probability,

thus breaking the n-special soundness property of the interactive protocol. The forking lemma can be generalized [1, 11], and in particular we can prove the security (in the ROM, against classical adversaries) of the Fiat–Shamir transform applied to the general definition of Sigma protocols [24].

A slight modification of the transformation allows us to produce a signature scheme from a Sigma protocol for a hard relation. Indeed, let $(u, w) \in R$. We set $pk = u$ and $sk = w$. Then signing and verification of a message m are as in Figure 2.1, except that we include m as input to the hash function. On a high level, the unforgeability follows from these facts. The adversary cannot recover sk directly, because of the hard relation, nor from the signing oracle, because of the zero-knowledge property. Signing queries do not provide any useful information to the adversary, since they are not related to each other because of the random oracle and the zero-knowledge property. Finally, by rewinding an adversary that breaks unforgeability without the oracle, we can recover additional signatures for the same message, which allow us to extract the witness because of the n-special soundness property.

Theorem 2. *Let $(\mathcal{P}, \mathcal{V})$ be a non-trivial Sigma protocol for a hard relation that is complete, knowledge sound and zero-knowledge. Then $(\overline{\mathcal{P}}, \overline{\mathcal{V}})$, obtained from the Fiat–Shamir transformation for signatures, as described above, is a signature scheme that is complete and EUF-CMA (5), in the random oracle model.*

The security of the Fiat–Shamir transform against quantum adversaries is a topic of ongoing research. Notable progress has been made in this direction, and in particular the security in the quantum random oracle model (QROM) has been proven, provided that the underlying Sigma protocol verifies some extra property [157]. But so far there is no direct analogue for the result in the classical setting.

This motivated the introduction of the *Unruh transformation* [83,156], which is more expensive but can be applied to many schemes. Let $(\mathcal{P}, \mathcal{V})$ be a Sigma protocol for a hard relation, with challenges of bit-length c, and let t be such that $ct \geq \lambda$. Let Z be the set of possible responses. Let $H : \{0,1\}^* \to \{0,1\}^{ct}$ and $G : \mathsf{Z} \to \mathsf{Z}$ be hash functions, modelled as random oracles, and such that G is injective or it has at most polinomially many preimages for each point. For $n \in \mathbb{N}$, we denote by $[n]_2$ the binary string representation of n.

Theorem 3. *Let $(\mathcal{P}, \mathcal{V})$ be a non-trivial Sigma protocol for a hard relation that is complete, knowledge sound and zero-knowledge. Then $(\overline{\mathcal{P}}, \overline{\mathcal{V}})$, obtained as in Figure 2.2 is a signature scheme that is complete and EUF-CMA (5), in the quantum random oracle model.*

$\overline{\mathcal{P}}(u, w)$:
for $i = 1, \ldots, t$:
  $a^{(i)} \leftarrow \mathcal{P}(\mathsf{crs}, u, w)$
  for $j = 1, \ldots, 2^c$:
    $x^{(i,j)} = [j]_2$
    $z^{(i,j)} \leftarrow \mathcal{P}(a^{(i)}, x^{(i,j)})$
    $g^{(i,j)} = G(z^{(i,j)})$
$T = \left(\{a^{(i)}\}_{i=1}^{t}, \{x^{(i,j)}, g^{(i,j)}\}_{i,j=1}^{t,2^c}\right)$
$x = H(u, T)$
parse $x = ([j_1]_2, \ldots, [j_t]_2)$
return $\pi = (T, \{z^{(i,j_i)}\}_{i=1}^{t})$

$\overline{\mathcal{V}}(u, \pi)$:
$x = H(u, T)$
parse $x = ([j_1]_2, \ldots, [j_t]_2)$
$b \leftarrow \mathcal{V}(u, a, x, z)$
for $i = 1, \ldots, t$:
  for $j = 1, \ldots, 2^c$:
    check $x^{(i,j)} = [j]_2$
    check $g^{(i,j)} = G(z^{(i,j)})$
  check $\mathcal{V}(u, a^{(i)}, x^{(i,j_i)}, z^{(i,j_i)})$
if all checks return 1 **return** 1

Figure 2.2: Unruh transform

On the output length of the hash functions. The question of the output length of the hash function depends on the security requirements. The conservative choice in the classical setting is 2λ, to avoid generic collision attacks. However, in the Fiat–Shamir transform for signatures the hash value is $h = H(m, a)$. To construct an existential forgery when given a signing oracle (or to break non-repudiation) it is sufficient to generate a random a and then find a collision in the hash function $H'(x) = H(x, a)$. For a chosen-message forgery or non-repudiation it is necessary, given a chosen message m, to find a second message m' with $H(m, a) = H(m', a)$, which is essentially computing a second-preimage in the hash function. As a result, in most practical settings and if H behaves like a random function, then one can use a hash of output length λ. This optimisation was already mentioned in the original paper on Schnorr signatures, and has been discussed in detail by Neven-Smart-Warinschi [134].

The correct choice of hash length in the quantum setting is still a subject of active research. The first question is to what extent quantum algorithms speed up collision finding. The second question is to consider a concrete analysis of the security proof for Unruh's transform, and any other factors in the security reduction that may be influenced by the hash output size. One conservative option is to assume that Grover's algorithm gives the maximal speedup for quantum algorithms, in which case one could take length 3λ to ensure collision-resistance. Bernstein [16] has questioned the practicality of quantum collision-finding algorithms. Following his arguments, Goldfeder, Chase and Zaverucha [83] use 2λ,

and a similar choice was made in Yoo et al. [167]. On the other hand, Beals et al. [9] suggest there may be a quantum speedup that would require increasing the hash length.

In our discussion, we will keep this value as a parameter, which can be adjusted as more information comes to light.

Quasi-adaptive proofs

A particular type of non-interactive proofs are quasi-adaptive non-interactive zero-knowledge (QA-NIZK) arguments, in which the CRS is allowed to depend on specific language from a parametrized family being proven. This allows for greater flexibility in building proofs for certain statements. In particular, the QA-NIZK approach has been highly successful for proving membership in linear spaces over a group.

More precisely, let ρ be a parameter sampled from a distribution D over the parameter language $\mathcal{L}_{\mathsf{par}}$, and let ω be a witness that allows to efficiently check that $\rho \in \mathcal{L}_{\mathsf{par}}$. We say that D is *witness sampleable* if the pairs (ρ, ω) can be efficiently sampled.

We consider a family of languages $\{\mathcal{L}_\rho\}$ parametrized by ρ, and denote the corresponding relations by R_ρ. A quasi-adaptive proof system for this family is a tuple of algorithms $(\mathsf{Setup}_0, \mathsf{Setup}_1, \mathcal{P}, \mathcal{V})$ verifying the following properties. Note that the setup is split into two parts, one that is universal and one that will depend on ρ. Likewise, we will have a CRS split in two parts, each generated by one of these algorithms: $\mathsf{crs} = (\mathsf{crs}_0, \mathsf{crs}_\rho)$.

Definition 13. *A QA-NIZK proof is* complete *if, for* $\rho \leftarrow \mathsf{D}$ *and* $(u, w) \in R_\rho$,

$$\mathsf{Pr}\left[\begin{array}{l} \mathsf{crs}_0 \leftarrow \mathsf{Setup}_0(1^\lambda), \\ \mathsf{crs}_\rho \leftarrow \mathsf{Setup}_1(\mathsf{crs}_0, \rho), \\ \pi \leftarrow \mathcal{P}(\mathsf{crs}, u, w) \end{array} : 0 \leftarrow \mathcal{V}(\mathsf{crs}, u, \pi)\right]$$

is negligible in λ.

Definition 14. *A QA-NIZK proof is* sound *if, for any PPT adversary* $\mathcal{A}$,

$$\mathsf{Pr}\left[\begin{array}{l} \mathsf{crs}_0 \leftarrow \mathsf{Setup}_0(1^\lambda); \rho \leftarrow \mathsf{D}; \\ \mathsf{crs}_\rho \leftarrow \mathsf{Setup}_1(\mathsf{crs}_0, \rho); (u, \pi) \leftarrow \mathcal{A}(\mathsf{crs}) \end{array} : \begin{array}{c} 1 \leftarrow \mathcal{V}(\mathsf{crs}, u, \pi) \\ \wedge u \notin \mathcal{L}_\rho \end{array}\right]$$

is negligible in λ.

Definition 15. *A QA-NIZK proof is* zero-knowledge *if there exists a PPT simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that, for any PPT adversary $\mathcal{A}$,*

$$\left| \mathsf{Pr}\left[\begin{array}{l} \mathsf{crs}_0 \leftarrow \mathsf{Setup}_0(1^\lambda); \rho \leftarrow \mathsf{D}, \mathsf{crs}_\rho \leftarrow \mathsf{Setup}_1(\mathsf{crs}_0, \rho), \\ (u, w) \leftarrow \mathcal{A}(\mathsf{crs}), \pi \leftarrow \mathcal{P}(\mathsf{crs}, u, w) \end{array} : \begin{array}{r} (u, w) \in R_\rho \\ \wedge 1 \leftarrow \mathcal{A}(\pi) \end{array} \right] - \right.$$
$$\left. - \mathsf{Pr}\left[\begin{array}{l} \mathsf{crs}_0 \leftarrow \mathsf{Setup}_0(1^\lambda); \rho \leftarrow \mathsf{D}, (\mathsf{crs}_\rho, \tau) \leftarrow \mathcal{S}_1(\mathsf{crs}_0, \rho), \\ (u, w) \leftarrow \mathcal{A}(\mathsf{crs}), \pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, u) \end{array} : \begin{array}{r} (u, w) \in R_\rho \\ \wedge 1 \leftarrow \mathcal{A}(\pi) \end{array} \right] \right|$$

is negligible in λ.

Observe that, unlike in the previous definitions of zero-knowledge, here the second part of the CRS generation is also simulated in the second case. Moreover, $\mathcal{S}_1$ produces a *simulation trapdoor* τ that is later used by $\mathcal{S}_2$ to simulate the proof. This is necessary because, if the simulator was able to produce honest-looking proofs without any secret information, then it could be used to break soundness. The reason why we did not need it in the previous sections is that, in the interactive setting, the fact that the verifier takes part in the proof prevents this 'attack'.

We also consider the notion of F-knowledge soundness, which we define in the context of witness sampleable distributions. Intuitively, F-knowledge means that, with access to some extraction key, it is possible to extract a function F of the witness from the statement and the proof. We note that our definition differs from the definition in [62], as we give the extraction key generator access to the witness ω that proves membership of ρ in $\mathcal{L}_{\mathsf{par}}$ (in practice, this means that it has access to the discrete logarithms of the commitment key) and allow to extract information from not only the statement, but also the proof.

Definition 16. *Given a function F, a QA-NIZK proof for a witness-sampleable distribution is* F-knowledge sound *if there exist a soundness PPT extraction key generator $\mathcal{E}_1$ and a DPT extractor $\mathcal{E}_2$ such that, for any non-uniform PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\left[\begin{array}{l} \mathsf{crs}_0 \leftarrow \mathsf{KeyGen}_0(1^\lambda); \rho \leftarrow \mathsf{D}; \\ (\mathsf{crs}_1, ek) \leftarrow \mathcal{E}_1(\mathsf{crs}_0, (\rho, \omega)); \\ (u, \pi) \leftarrow \mathcal{A}(\mathsf{crs}) \end{array} : \begin{array}{l} \mathcal{V}(\mathsf{crs}, u, \pi) = 1 \text{ and} \\ \mathcal{E}_{2ek}(u, \pi) \neq F(u, w) \end{array} \right]$$

is negligible in λ, where the distributions of crs_ρ produced by KeyGen_1 and $\mathcal{E}_1$ are indistinguishable.

We also define a stronger notion of zero-knowledge, called composable zero-knowledge [87]. Essentially, this means that real and simulated proofs are indistinguishable even when the simulation trapdoor is known.

Definition 17. *A QA-NIZK proof is* composable zero-knowledge *if there exists a PPT simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that, for any PPT adversary $\mathcal{A}$,*

$$\left| \mathsf{Pr}\left[\begin{array}{l} \mathsf{crs}_0 \leftarrow \mathsf{Setup}_0(1^\lambda); \rho \leftarrow \mathsf{D}, (\mathsf{crs}_\rho, \tau) \leftarrow \mathcal{S}_1(\mathsf{crs}_0, \rho), \\ (u, w) \leftarrow \mathcal{A}(\mathsf{crs}, \tau), \pi \leftarrow \mathcal{P}(\mathsf{crs}, u, w) \end{array} : 1 \leftarrow \mathcal{A}(\pi) \right] - \right.$$
$$\left. - \mathsf{Pr}\left[\begin{array}{l} \mathsf{crs}_0 \leftarrow \mathsf{Setup}_0(1^\lambda); \rho \leftarrow \mathsf{D}, (\mathsf{crs}_\rho, \tau) \leftarrow \mathcal{S}_1(\mathsf{crs}_0, \rho), \\ (u, w) \leftarrow \mathcal{A}(\mathsf{crs}, \tau), \pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, u) \end{array} : 1 \leftarrow \mathcal{A}(\pi) \right] \right|$$

is negligible in λ, where the distributions of crs_ρ produced by KeyGen_1 and $\mathcal{S}_1$ are indistinguishable.

2.3 The discrete logarithm setting

The discrete logarithm problem is one of the cornerstones of classical cryptography. In this section, we discuss the family of problems spanned from it.

We start by introducing some notation. Let $\mathbb{G}$ be a cyclic group. We will write group elements in *implicit notation* with respect to a fixed generator $g \in \mathbb{G}$. That is, we will write $[x]$ for g^x. The group operation then becomes $g^x g^{x'} = [x + x']$. This will allow us to highlight the linear algebra happening "in the exponent".

Let $\mathcal{G}$ be a PPT algorithm that, on input the security parameter λ, outputs $gk = (\mathbb{G}, g)$, where $\mathbb{G}$ is a description of a group of order $n = \Theta(2^\lambda)$, and $\mathbb{G} = \langle g \rangle$. We consider the following assumptions with respect to $\mathcal{G}$.

Assumption 1 (discrete logarithm)**.** *For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\left[gk \leftarrow \mathcal{G}(1^\lambda), x \leftarrow \mathbb{Z}_n, \tilde{x} \leftarrow \mathcal{A}(gk, [x]) : \tilde{x} = x\right]$$

is negligible in λ.

Typically, we will consider this assumption in groups of prime order p. From a security point of view, the best choice is a subgroup of a well-chosen elliptic curve, for which only generic discrete logarithm algorithms are known. These algorithms have a running time of at best $O(\sqrt{p})$.

Two very well-known assumptions derived from it are the *computational Diffie–Hellman (CDH)* and *decisional Diffie–Hellman (DDH)* assumptions.

Assumption 2 (CDH)**.** *For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\left[gk \leftarrow \mathcal{G}(1^\lambda), a, b \leftarrow \mathbb{Z}_n, \tilde{c} \leftarrow \mathcal{A}(gk, [a], [b]) : \tilde{c} = ab\right]$$

is negligible in λ.

Assumption 3 (DDH)**.** *For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\begin{bmatrix} gk \leftarrow \mathcal{G}(1^\lambda), a, b, c \leftarrow \mathbb{Z}_n, z_0 = ab, \\ z_1 = c, \beta \leftarrow \{0,1\}, \tilde{\beta} \leftarrow \mathcal{A}(gk, [a], [b], [z_\beta]) \end{bmatrix} : \tilde{\beta} = \beta\Bigg]$$

is negligible in λ.

The following is a direct generalization of the DDH assumption. We introduce the family of *matrix decisional Diffie–Hellman (MDDH)* [64] assumptions in a prime order group, since it is the only case that is relevant to this work. We call $\mathsf{D}_{\ell,k}$ a *matrix distribution* if it outputs (in PPT, with overwhelming probability) matrices in $\mathbb{Z}_p^{\ell \times k}$. We also define $\mathsf{D}_k := \mathsf{D}_{k+1,k}$.

Assumption 4 (MDDH)**.** *For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\begin{bmatrix} gk \leftarrow \mathcal{G}(1^\lambda), \mathbf{A} \leftarrow \mathsf{D}_{\ell,k}, \boldsymbol{w} \leftarrow \mathbb{Z}_p^k, \boldsymbol{v}_0 = \mathbf{A}\boldsymbol{w}, \\ [\boldsymbol{v}_1] \leftarrow \mathbb{G}^\ell, b \leftarrow \{0,1\}, \tilde{b} \leftarrow \mathcal{A}(gk, [\mathbf{A}, \boldsymbol{v}_b]) \end{bmatrix} : \tilde{b} = b\Bigg]$$

is negligible in λ.

Intuitively, the $\mathsf{D}_{\ell,k}$-MDDH assumption means that it is hard to decide whether a vector is in the image space of a matrix or it is a random vector, where the matrix is drawn from $\mathsf{D}_{\ell,k}$ and given in the exponent. In this work we will refer to the following matrix distributions:

$$\mathcal{L}_k : \mathbf{A} = \begin{pmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_k \\ 1 & 1 & \dots & 1 \end{pmatrix}, \qquad \mathcal{RL}_k : \mathbf{A} = \begin{pmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_k \\ r_1 & r_2 & \dots & r_k \end{pmatrix},$$

where $a_i, r_i \leftarrow \mathbb{Z}_p$ for $i = 1, \dots, k$. The $\mathcal{L}_k$-MDDH Assumption is the k-linear family of Decisional Assumptions and corresponds to the DDH assumption when $k = 1$.

2.3.1 Pairings

Pairings (or *bilinear maps*) were introduced in cryptography as a cryptanalytic tool, aimed at reducing the discrete logarithm problem in an elliptic curve to the discrete logarithm problem in a finite field, where subexponential time algorithms are available. Starting from [106], pairings have proven to be a powerful tool for building public-key schemes, ranging from identity-based encryption schemes, to pseudo-random functions, to countless zero-knowledge proofs.

Definition 18. *Let $\mathbb{G}_1 = \langle g_1 \rangle, \mathbb{G}_2 = \langle g_2 \rangle, \mathbb{G}_T$ be cyclic groups of order n. A* pairing *is an efficiently computable map*

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$$

such that, for any $a, b \in \mathbb{Z}_n$, we have that

$$e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}.$$

The groups $\mathbb{G}_1, \mathbb{G}_2$ are called the *source groups*, whereas $\mathbb{G}_T$ is called the *target group*. Given fixed generators $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$, the tuple $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ is often called a *bilinear group*. When $\mathbb{G}_1 = \mathbb{G}_2$, we say that the pairing is *symmetric*, otherwise we say that it is *asymmetric*. Note that a symmetric pairing allows to easily solve the DDH problem in $\mathbb{G}_1$.

We extend our implicit notation to this setting. For $\gamma \in \{1, 2\}$, we define $[x]_\gamma = g_\gamma^x$. For the target group, we define $[x]_T = e(g_1, g_2)^x$. Observe that with this notation it is immediate to see that the pairing behaves as a multiplication in the exponent. That is,

$$e([x]_1, [y]_2) = [xy]_T.$$

For this reason, we often just write $[x]_1[y]_2$ for $e([x]_1, [y]_2)$. This notation extends naturally to vectors and matrices of group elements. Moreover, we will write $[x]_{1,2}$ for the tuple $([x]_1, [x]_2)$.

The assumptions introduced in the previous section can easily be generalized to the pairing setting. Additionally, we will be using the *Kernel Diffie–Hellman (KerMDH)* [132] family of computational assumptions, which are the computational counterparts of the MDDH assumptions. Let $\mathcal{G}$ be a bilinear group generator, which takes as input the security parameter and outputs a bilinear group of order $p = \Theta(2^\lambda)$, and let $\gamma \in \{1, 2\}$.

Assumption 5 (KerMDH in $\mathbb{G}_\gamma$)**.** *For any PPT adversary $\mathcal{A}$:*

$$\mathsf{Pr}\left[gk \leftarrow \mathcal{G}(1^\lambda), [\boldsymbol{x}]_{3-\gamma} \leftarrow \mathcal{A}(gk, [\mathbf{A}]_\gamma) : \boldsymbol{x} \neq 0 \wedge \boldsymbol{x}^\top \mathbf{A} = \mathbf{0}\right]$$

is negligible in λ.

Intuitively, this means that it is hard to find non-trivial elements in the co-kernel of a matrix given in the exponent. The $\mathsf{D}_{\ell,k}$-$\mathsf{KerMDH}_{\mathbb{G}_\gamma}$ Assumption is not stronger than the $\mathsf{D}_{\ell,k}$-$\mathsf{MDDH}_{\mathbb{G}_\gamma}$ Assumption, since a solution to the former allows to decide membership in $\mathrm{Im}[\mathbf{A}]_\gamma$. A natural variant of this assumption is the *split kernel Diffie–Hellman assumption (SKerMDH)* [85], in which the solution is partially given in each of the source groups.

Assumption 6 (SKerMDH)**.** *For any PPT adversary* $\mathcal{A}$*:*

$$\mathsf{Pr}\left[gk \leftarrow \mathcal{G}(1^\lambda), \mathbf{A} \leftarrow \mathsf{D}_{\ell,k}, [\boldsymbol{r}]_1, [\boldsymbol{s}]_2 \leftarrow \mathcal{A}(gk, [\mathbf{A}]_{1,2}) : \boldsymbol{r} \neq \boldsymbol{s} \wedge \boldsymbol{r}^\top \mathbf{A} = \boldsymbol{s}^\top \mathbf{A}\right]$$

is negligible in λ.

In their weakest and most efficient instatiations, KerMDH is weaker than DDH, and SKerMDH is weaker than 2-Lin. Some other (less standard) assumptions will be used in chapter 4, and will be introduced in section 4.2.2.

2.4 Mathematical background

Note: the remainder of this chapter is only relevant to the second half of the thesis, that is, chapters 5, 6 and 7.

2.4.1 Elliptic curves

We recall some elementary definitions and results from the theory of elliptic curves. See [72, 150, 164] for a more detailed exposition. Through this section and unless stated otherwise, $p > 3$ is a prime number and q is some power of p. We denote by $\mathbb{F}_q$ the finite field of order q, and its algebraic closure by $\overline{\mathbb{F}}_q$.

An *elliptic curve* E *over* $\mathbb{F}_q$ (denoted $E/\mathbb{F}_q$) is a smooth algebraic curve of genus 1, defined by the projective solutions of the equation

$$Y^2Z = X^3 + aXZ^2 + bZ^3,$$

where $a, b \in \mathbb{F}_q$, and $\Delta = -16(4a^3 + 27b^2) \neq 0$. This amount is called the *discriminant* of E, and the condition $\Delta \neq 0$ ensures that the curve is smooth. When $Z = 0$, the only solution is $[0 : 1 : 0]$. Thus, an elliptic curve can also be seen as the set of solutions over $\overline{\mathbb{F}}_q$ of the equation

$$Y^2 = X^3 + aX + b,$$

plus an additional point ∞, called the *point at infinity*, corresponding to $[0:1:0]$. We will usually work with this second definition, and think of points in the curve as points in $\left(\overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q\right) \cup \{\infty\}$.

An elliptic curve has an abelian group structure, with ∞ being the neutral element of the operation. We write the group operation with additive notation, so the addition of two points P, Q is $P + Q$, and for $m \in \mathbb{N}$, we write

$$mP = \underbrace{P + \cdots + P}_{m}.$$

Given a field $K \supset \mathbb{F}_q$, we define the set of *K-rational points* of E as

$$E(K) = \{(x, y) \in K^2 \mid y^2 = x^3 + ax + b\} \cup \{\infty\}.$$

This is actually a subgroup of the curve E. We will mostly focus on $E(\mathbb{F}_q)$ and $E = E(\overline{\mathbb{F}}_q)$. The former verifies Hasse's bound, which states that

$$|\#E(\mathbb{F}_q) - (q+1)| < 2\sqrt{q}.$$

The number $t = \#E(\mathbb{F}_q) - (q+1)$ is usually called the *trace of Frobenius* of E. When $p \mid t$, we say that the curve is *supersingular*, otherwise we say that it is *ordinary*. Note that, since we can count the number of points of an elliptic curve in polynomial time, we can determine whether a curve is ordinary or supersingular. In this work, we will focus on supersingular elliptic curves.

For any $m \in \mathbb{N}$, we define the *m-torsion* of the curve E as

$$E[m] = \{P \in E \mid mP = \infty\}.$$

If $p \nmid m$, we have that $E[m] = \mathbb{Z}_m \times \mathbb{Z}_m$. Observe that the m-torsion of a curve $E/\mathbb{F}_q$ is not necessarily contained in $\mathbb{F}_q^2$.

The m-torsion of an elliptic curve is closely related to the *division polynomials* $\Psi_m(X, Y) \in \mathbb{F}_{p^2}[X, Y]$, which has the following properties:

- If m is odd, then Ψ_m is a polynomial in X only, of degree $(n^2-1)/2$, and $(x, y) \in E[m]$ if and only if $\Psi_m(x) = 0$.
- If m is even, then Ψ_m is the product of Y and a polynomial in X of degree $(n^2-4)/2$, and $(x, y) \in E[m]$ if and only if $\Psi_m(x, y) = 0$.

Given an elliptic curve E with coefficients a, b, we define the *j-invariant* of E as

$$j = j(E) = 1728\frac{4a^3}{4a^3 + 27b^2}.$$

This quantity is invariant through isomorphism. We will use the j-invariant to label isomorphism classes. For supersingular elliptic curves, every isomorphism class has a representative over $\mathbb{F}_{p^2}$, and thus we have $j \in \mathbb{F}_{p^2}$. Moreover, observe that Hasse's bound allows only for $t \in \{0, \pm p, \pm 2p\}$. For the first three, there are only two possible j-invariants for each, and most of the supersingular j-invariants correspond to curves with $t = \pm 2p$.

Given a curve $E/\mathbb{F}_{p^2}$ defined by the equation

$$E : Y^2 = X^3 + aX + b,$$

we let $d \in \mathbb{F}_{p^2}$ be a quadratic nonresidue in $\mathbb{F}_{p^2}$. Then, the *quadratic twist* of E is the curve defined by the equation

$$\tilde{E} : Y^2 = X^3 + ad^2X + bd^3.$$

The curves E and $\tilde{E}$ are not isomorphic over $\mathbb{F}_{p^2}$, but they are isomorphic over $\overline{\mathbb{F}}_{p^2}$. Also, if E has trace t, then $\tilde{E}$ has trace $-t$.

2.4.2 Isogenies

We turn our attention to maps between elliptic curves. An *isogeny* $\varphi : E \to E'$ is a non-constant rational map between two elliptic curves E, E' that preserves the point at infinity. An equivalent definition, from the point of view of group theory, is that an isogeny is a surjective homomorphism. We remark that the surjective property refers to the full-domain isogeny $\varphi : E \to E'$, not to $\varphi : E(\mathbb{F}_q) \to E(\mathbb{F}_q)$.

An isogeny is defined over K if its coefficients as a rational function are in K. It is important to notice that we can have two curves defined over $\mathbb{F}_q$, and an isogeny between them that is only defined over an extension of $\mathbb{F}_q$.

An isogeny $\varphi : E \to E'$ induces an injection of function fields

$$\begin{array}{rccc} \varphi^* : & K(E') & \longrightarrow & K(E) \\ & f & \longmapsto & f \circ \varphi. \end{array}$$

We define the *degree* of φ as the degree of the corresponding field extension,

$$\deg \varphi = [K(E) : \varphi^* K(E')] .$$

We say that the isogeny is *separable* if this extension is separable, otherwise we say that it is *inseparable*. For separable isogenies, we have that

$$\# \ker \varphi = \deg \varphi.$$

An isogeny of degree ℓ is called an *ℓ-isogeny*. The composition of two isogenies of degrees ℓ_1, ℓ_2 is an isogeny of degree $\ell_1\ell_2$. Likewise, an isogeny of composite degree can be factored into a sequence of prime-degree components.

The simplest isogenies are *multiplication by m* maps, $[m] : E \to E$ defined as $P \mapsto mP$. Note that the kernel of $[m]$ is precisely the torsion $E[m]$. Thus, if $p \nmid m$, we have that $\deg[m] = m^2$. We say that two curves are *isogenous* if there exists an isogeny between them. Given an isogeny $\varphi : E \to E'$, there exists another isogeny $\hat{\varphi} : E' \to E$ of the same degree, called the *dual isogeny*, such that

$$\hat{\varphi} \circ \varphi = [\deg \varphi]_E, \qquad \text{and} \qquad \varphi \circ \hat{\varphi} = [\deg \varphi]_{E'}.$$

Thus, being isogenous is an equivalence relation. Given an isogeny, its dual can be efficiently computed. Tate's theorem states that two curves $E/\mathbb{F}_q, E'/\mathbb{F}_q$ are isogenous over $\mathbb{F}_q$ if and only if $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q)$.

Given a prime $\ell \neq p$, the *ℓ-th modular polynomial* is a polynomial $\Phi_\ell(X, Y) \in \mathbb{F}_{p^2}[X, Y]$ of degree $\ell + 1$ in each variable, and such that

$$\Phi_\ell(j, j') = 0$$

if and only if j and j' are the j-invariants of two ℓ-isogenous elliptic curves.

The endomorphism ring. Isogenies from a curve to itself are called *endomorphisms*. The set of endomorphisms $\mathrm{End}(E)$ of an elliptic curve, together with the zero map, form a ring with the operations point-wise addition and function composition. Since multiplication maps are endomorphisms, we can embed $\mathbb{Z} \hookrightarrow \mathrm{End}(E)$. Over finite fields, this inclusion is always strict.

We define the *q-power Frobenius endomorphism* as $\pi : (x, y) \mapsto (x^q, y^q)$. Note that this is the identity on $E(\mathbb{F}_q)$, but not on $E(\overline{\mathbb{F}}_q)$. If t is the trace of Frobenius, then π satisfies

$$\pi^2 - [t]\pi + [q] = 0. \tag{2.1}$$

The Frobenius endomorphism has degree q and is inseparable, and any isogeny can be decomposed into a power of π and a separable isogeny. If E is ordinary, then $\pi \notin \mathbb{Z}$ and $\mathrm{End}(E) = \mathbb{Z}[\pi]$. If E is supersingular, the endomorphism ring is larger.

Isogeny kernels. A separable isogeny can be identified with its kernel. Given a subgroup $H \subseteq E$, Vélu's formulas provide an explicit way to build a curve $E' \cong E/H$ and an isogeny $\varphi : E \to E'$ such that $\ker \varphi = H$. Vélu's formulas

involve sums over every point in H, and thus the isogeny computation depends linearly on the size of H, and on the degree of the extension field that contains the coefficients. A large-degree isogeny can be efficiently computed as long as its order is smooth enough, by decomposing it into prime-order steps and applying Vélu's formulas to them.

Moreover, the subgroup also tells us over which field extension the corresponding isogeny is defined. We say that $H \subseteq E$ is *defined over* $\mathbb{F}_q$ if H is fixed by the q-power Frobenius endomorphism. Note that this holds for (but is not restricted to) subgroups contained in $E(\mathbb{F}_q)$. We have that if H is defined over $\mathbb{F}_q$, then the isogeny with kernel H is defined over $\mathbb{F}_q$. A particular case are most supersingular curves over $\mathbb{F}_{p^2}$.

Proposition 4. *Let $E/\mathbb{F}_{p^2}$ be a supersingular elliptic curve, with $\#E(\mathbb{F}_{p^2}) = (p^2+1) \pm 2p$, and let $\ell \neq p$ be a prime number. Then every ℓ-isogeny φ from E is defined over $\mathbb{F}_{p^2}$.*

Proof. The p^2-power Frobenius π verifies

$$\pi^2 \mp [2p]\pi + [p^2] = 0 \Longrightarrow (\pi \mp [p])^2 = 0 \Longrightarrow \pi = [\pm p],$$

so the Frobenius is just a multiplication map. Moreover, since $\#\ker\varphi = \ell$ and p is coprime to ℓ, the maps $[\pm p]$ are permutations of $\ker\varphi$, leaving it fixed. □

The Weil pairing. Elliptic curves provide the only known instantiations of pairings suitable for cryptography. The best-known case is the *Weil pairing*

$$e : E[m] \times E[m] \to \mu_m,$$

where μ_m is the group of m-th roots of unity. The concrete definition of e is quite involved, and we refer the interested reader to [150]. The smallest integer k such that $\mu_m \subseteq \mathbb{F}_q^k$ is called the *embedding degree* of the pairing. The Weil pairing is degenerate, meaning that $e(P,P) = 1$ for any $P \in E[m]$, so by itself it is not useful for cryptography. This is solved by using a *distortion map*, that is, an endomorphism $\phi : E \to E$ such that $\phi(P) \notin \langle P \rangle$. We then define a new pairing as

$$\hat{e}(P,Q) = e(P, \phi(Q)),$$

which is used instead of the Weil pairing.

Pairings interact with isogenies in the following way. Let $P \in E, P' \in E'$. Then, given an isogeny $\varphi : E \to E'$, we have that

$$e(P, \hat{\varphi}(P')) = e'(\varphi(P), P').$$

A direct consequence of this is that, for $P, Q \in E$,

$$e(\varphi(P), \varphi(Q)) = e(P, Q)^{\deg \varphi}.$$

2.4.3 Supersingular isogeny graphs

We start by briefly reviewing some notions from graph theory. Let $\mathsf{G} = (V, E)$ be a graph, where V is the set of vertices and E is the set of edges.

For any $U \subseteq V$, we define the *boundary* of U as

$$\Gamma(U) = \{v \in V \setminus U \mid \exists u \in U \text{ s. t. } (v, u) \in E\}.$$

Definition 19. *Fix $k \in \mathbb{N}$. Let $\{\mathsf{G}_n = (V_n, E_n)\}_{n \in \mathsf{S}}$ be a sequence of graphs, with $\mathsf{S} \subseteq \mathbb{N}$, and where G_n is a k-regular graph, where $\#V_n = n$. We say that $\{\mathsf{G}_n\}_{n \in \mathsf{S}}$ is a family of* expander *graphs with* expansion constant $c > 0$ *if, for all $n \in \mathsf{S}$, for any $U \subseteq V_n$ of size at most $n/2$, we have*

$$\#\Gamma(U) \geq c(\#U)$$

Intuitively, this means that the graph has high connectivity, and there is a relatively short path between any two given vertices. A particular type of expander graphs are Ramanujan graphs.

Definition 20. *Let G be a k-regular graph, and let $k, \lambda_2, \cdots, \lambda_r$ be the eigenvalues of the adjacency matrix sorted by decreasing order of the absolute value. Then G is a* Ramanujan graph *if*

$$\lambda_2 \leq 2\sqrt{k-1}.$$

This is optimal by the Alon–Boppana bound, which states that, given a family $\{\mathsf{G}_n\}_{n \in \mathsf{S}}$ of k-regular graphs as above, and denoting by $\lambda_{2,n}$ the corresponding second eigenvalue of each graph G_n, we have

$$\liminf_{n \to \infty} \lambda_{2,n} \geq 2\sqrt{k-1}.$$

Given two different primes p and ℓ, we consider the graph $\mathsf{G}_\ell(\overline{\mathbb{F}}_{p^2}) = (V, E)$, which we call the *$\ell$-supersingular isogeny graph*: the set of vertices V is the set of j-invariants of supersingular elliptic curves over $\mathbb{F}_{p^2}$, each representing an isomorphism class. Given two vertices j, j', there is an edge between them if there exist elliptic curves $E/\mathbb{F}_{p^2}, E'/\mathbb{F}_{p^2}$, such that $j = j(E), j' = j(E')$, and

there is an ℓ-isogeny $\varphi : E \to E'$. Because of the existence of the dual isogeny $\hat{\varphi} : E' \to E$, the graph is undirected. We have that

$$\#V = p/12 + \epsilon,$$

where $\epsilon = 0, 1, 1, 2$ for $p = 1, 5, 7, 11 \bmod 12$, respectively. Also, since $E[\ell]$ has $\ell + 1$ subgroups of order ℓ, we have that the graph is $(\ell + 1)$-regular. Moreover, the graph $\mathsf{G}_\ell(\overline{\mathbb{F}}_{p^2})$ is Ramanujan [141].

We also consider the graph $\mathsf{G}_\ell(\mathbb{F}_{p^2})$ where vertices are defined by classes of $\mathbb{F}_{p^2}$-isomorphisms, instead of classes of $\overline{\mathbb{F}}_{p^2}$-isomorphisms, and edges are the ℓ-isogenies defined over $\mathbb{F}_{p^2}$. This graph has five connected components, corresponding to supersingular curves of orders $p^2 + 1 + t$, where $t \in \{0, \pm p, \pm 2p\}$. While the first three graphs are small, each of the graphs corresponding to $\pm 2p$ is isomorphic to $\mathsf{G}_\ell(\overline{\mathbb{F}}_{p^2})$. Thus, we can also think of these as representations of the ℓ-supersingular isogeny graph.

2.4.4 Quaternion algebras

We summarize the required background on quaternion algebras. For a more detailed exposition of the theory, see [117, 160, 161].

Let $p = 3 \bmod 4$ be a prime number. A *quaternion algebra* over $\mathbb{Q}$ is an algebra of the form

$$\mathbb{Q}\langle \mathbf{i}, \mathbf{j} \rangle = \mathbb{Q} + \mathbf{i}\mathbb{Q} + \mathbf{j}\mathbb{Q} + \mathbf{k}\mathbb{Q}, \tag{2.2}$$

where $\mathbf{i}^2, \mathbf{j}^2 \in \mathbb{Q}, \mathbf{i}^2, \mathbf{j}^2 < 0$, and $\mathbf{k} = \mathbf{ij} = -\mathbf{ji}$.

We can also consider quaternion algebras over fields of p-adic numbers $\mathbb{Q}_p$, or $\mathbb{R}$, by replacing the field in equation 2.2. Given a prime p, we say that $\mathbb{Q}\langle \mathbf{i}, \mathbf{j} \rangle$ *splits at* p if $\mathbb{Q}_p\langle \mathbf{i}, \mathbf{j} \rangle$ has divisors of zero. Otherwise, we say that it *ramifies at* p. Similarly, we say that $\mathbb{Q}\langle \mathbf{i}, \mathbf{j} \rangle$ *splits at* ∞ if $\mathbb{R}\langle \mathbf{i}, \mathbf{j} \rangle$ has divisors of zero, and that it *ramifies at* ∞ otherwise.

The quaternion algebra over $\mathbb{Q}$ that ramifies at ∞ and a prime p, and splits at every other prime, is unique up to isomorphism, and it can be represented by

$$\mathbf{B}_{p,\infty} = \mathbb{Q} + \mathbf{i}\mathbb{Q} + \mathbf{j}\mathbb{Q} + \mathbf{k}\mathbb{Q}, \tag{2.3}$$

for $\mathbf{i}^2 = -1, \mathbf{j}^2 = -p$. Given $\alpha = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \in \mathbf{B}_{p,\infty}$, we define the *canonical involution* $\overline{\alpha}$, *reduced trace* $\mathrm{Trd}(\alpha)$ and *reduced norm* $\mathrm{Nrd}(\alpha)$ as follows:

$$\begin{aligned} \overline{\alpha} &= a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k} \\ \mathrm{Trd}(\alpha) &= \alpha + \overline{\alpha} = 2a \\ \mathrm{Nrd}(\alpha) &= \alpha\overline{\alpha} = a^2 + b^2 + pc^2 + pd^2 \end{aligned}$$

The reduced trace is linear, i.e.

$$\mathrm{Trd}(\alpha+\beta) = \mathrm{Trd}(\alpha) + \mathrm{Trd}(\beta),$$

and the reduced norm is multiplicative, i.e.,

$$\mathrm{Nrd}(\alpha\beta) = \mathrm{Nrd}(\alpha)\,\mathrm{Nrd}(\beta).$$

An *ideal*[1] I of $\mathbf{B}_{p,\infty}$ is a $\mathbb{Z}$-module in $\mathbf{B}_{p,\infty}$ containing a $\mathbb{Q}$-basis for $\mathbf{B}_{p,\infty}$. Thus, it can be seen as a lattice of rank 4 in $\mathbf{B}_{p,\infty}$. We define the *reduced norm* $\mathrm{Nrd}(I)$ of I as the greatest common divisor of the reduced norms of its elements. Two ideals I, J such that $\mathrm{Nrd}(I) = \mathrm{Nrd}(J)$ and $J \subseteq I$ are necessarily equal.

An *order* of $\mathbf{B}_{p,\infty}$ is an ideal that is also a subring of $\mathbf{B}_{p,\infty}$. Elements in an order have integer reduced trace and norm. An order $\mathcal{O}$ is *maximal* if it is not strictly contained in any other order.

Given an ideal I, we define the *left order* of I as

$$\mathcal{O}_{\mathsf{L}}(I) = \{h \in \mathbf{B}_{p,\infty} \mid hI \subseteq I\}.$$

For an order $\mathcal{O}$, we say that I is a *left $\mathcal{O}$-ideal* if $\mathcal{O} \subseteq \mathcal{O}_{\mathsf{L}}(I)$. Clearly I is a left $\mathcal{O}_{\mathsf{L}}(I)$-ideal. If I is a left $\mathcal{O}$-ideal, we have that $\mathcal{O}n \subseteq I$. The product IJ of two left $\mathcal{O}$-ideals I, J is defined as the ideal spanned by

$$\{\alpha\beta \mid \alpha \in I, \beta \in J\}.$$

It is immediate to see that $IJ \subseteq I, J$. The definitions of right $\mathcal{O}$-ideal and right order $\mathcal{O}_{\mathsf{R}}(I)$ are analogous, and the same results apply. For any order $\mathcal{O}$ and any prime $\ell \neq p$, there are $\ell + 1$ left $\mathcal{O}$-ideals with norm ℓ.

Proposition 5. *Let $\mathcal{O}$ be an order in $\mathbf{B}_{p,\infty}$, and let I be a left $\mathcal{O}$-ideal of prime norm n. Then there exists $\alpha \in I$ such that $I = \mathcal{O}n + \mathcal{O}\alpha$.*

Proof. We have that $\mathcal{O}n \subseteq I$. Let $\alpha \in I$ such that $\gcd(\mathrm{Nrd}(\alpha), n^2) = n$, and let $J = \mathcal{O}n + \mathcal{O}\alpha \subseteq I$. Note that $\mathrm{Nrd}(\mathcal{O}n) = n^2$, so $\alpha \notin \mathcal{O}n$ and thus J is strictly larger than $\mathcal{O}n$. Then necessarily $\mathrm{Nrd}(J) = n$, and since it is contained in I, we conclude that they are the same. □

Since orders and ideals in $\mathbf{B}_{p,\infty}$ are lattices, we can also represent them by means of a $\mathbb{Z}$-basis $\omega_0, \omega_1, \omega_2, \omega_3$. In the case of orders, we can always take $\omega_0 = 1$. In particular, $\mathbf{B}_{p,\infty}$ contains the maximal order

$$\mathcal{O}_0 = \left\langle 1, \mathbf{i}, \frac{1+\mathbf{k}}{2}, \frac{\mathbf{i}+\mathbf{j}}{2} \right\rangle.$$

[1] Sometimes called a *fractional ideal*, or simply *lattice*.

We establish the following equivalence relations on orders and their left ideals.

$$\begin{aligned} \mathcal{O}_1 \equiv \mathcal{O}_2 &\iff \exists q \in \mathbf{B}_{p,\infty}^* \mid \mathcal{O}_1 = q^{-1}\mathcal{O}_2 q, \\ I_1 \equiv I_2 &\iff \exists q \in \mathbf{B}_{p,\infty}^* \mid I_1 = I_2 q. \end{aligned} \tag{2.4}$$

These equivalence classes are compatible in the sense that the left ideals I_1 and I_2 are equivalent if and only if their right orders are equivalent. The number of equivalence classes is independent of $\mathcal{O}$ and is called the class number.

Quaternion algebras and supersingular elliptic curves. If $E/\mathbb{F}_q$ is an ordinary curve, then $\mathrm{End}(E)$ is an order in an imaginary quadratic field. If E is supersingular, then $\mathrm{End}(E)$ is a maximal order in the quaternion algebra $\mathbf{B}_{p,\infty}$. Moreover, a theorem by Deuring [60] gives an equivalence of categories between the j-invariants of supersingular elliptic curves over $\mathbb{F}_{p^2}$ up to Galois conjugacy in $\mathbb{F}_{p^2}$, and the maximal orders in $\mathbf{B}_{p,\infty}$ up to the equivalence relation given in equation (2.4).

We make the correspondence more explicit: let $E/\mathbb{F}_q$ be a supersingular elliptic curve with j-invariant j, and let $\varphi : E \to E'$ be an isogeny. We define the *kernel ideal* of φ as the left $\mathrm{End}(E)$-ideal I such that

$$E[I] = \{P \in E \mid \alpha(P) = 0 \quad \forall \alpha \in I\} = \ker \varphi.$$

Then, the *Deuring correspondence* assigns:

$$j \mapsto \mathcal{O} \cong \mathrm{End}(E), \qquad \varphi \mapsto I,$$

where I is the kernel ideal of φ.

Let $\ell_1, \ldots, \ell_r$ be prime numbers different from p. For $j = 1, \ldots, r$, let $\varphi_j : E_{j-1} \to E_j$ be an isogeny of degree $\ell_j^{e_j}$. Let $I_0 = \mathrm{End}(E_0)$, and let I_i be the kernel ideal of the isogeny $\varphi_i \circ \cdots \circ \varphi_1 : E_0 \to E_i$. Then, very similarly to Proposition 5, we can write

$$I_i = I_{i-1}\ell_i^{e_i} + I_{i-1}\alpha, \tag{2.5}$$

where $\alpha \in I_0$ is such that

$$\ker(\varphi) \cap E_0[\ell_i^{e_i}] \subseteq \ker(\alpha), \qquad \gcd(\deg(\alpha), \ell_i^{e_i+1}) = \ell_i^{e_i},$$

where $\varphi = \varphi_r \circ \cdots \circ \varphi_1$.

2.4.5 Isogeny problems

We now present some hardness assumptions related to supersingular elliptic curves, and discuss the related algebraic problems in light of the Deuring correspondence. Assume the existence of a prime generator $\mathcal{G}$ that, on input the security parameter, produces a prime $p = 3 \bmod 4$ of length λ bits. We denote by $\mathbf{E}_{(p+1)^2}$ a set of representatives of j-invariants in the graph $\mathsf{G}_\ell(\overline{\mathbb{F}}_{p^2})$ with $(p+1)^2$ points. We denote by $\mathbf{Iso}(E, E')$ the set of isogenies from E to E'.

Remark 1. *It is not immediately clear how to sample uniformly random curves from $\mathbf{E}_{p^2}$. Actually, what we can do is take a random walk in the graph from a fixed starting vertex, and output a representative of the j-invariant at the end of the path. We will see that it is enough to take a walk of size linear in λ to end a vertex that is statistically indistinguishable from uniform. Thus, the assumptions with uniform sampling and with sampling as just described are equivalent, and we will formulate them using the uniform distribution for simplicity.*

Given two supersingular elliptic curves $E/\mathbb{F}_{p^2}, E'/\mathbb{F}_{p^2}$, the general isogeny problem is to find an isogeny $\varphi : E \to E'$, where by finding an isogeny we mean producing a polynomial-size representation of the isogeny.

Assumption 7 (isogeny)**.** *For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\left[\begin{matrix} p \leftarrow \mathcal{G}(1^\lambda), E, E' \leftarrow \mathbf{E}_{(p+1)^2}, \\ \varphi \leftarrow \mathcal{A}(p, E, E') \end{matrix} \quad : \quad \varphi \in \mathbf{Iso}(E, E'), \right]$$

is negligible in λ.

$$E \dashrightarrow^{\varphi} E'$$

Often, we specify to some extent the degree of the isogeny to be found. In particular, we will be using the following assumption in our constructions.

Assumption 8 (ℓ-power isogeny)**.** *Let ℓ. For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\left[\begin{matrix} p \leftarrow \mathcal{G}(1^\lambda), E, E' \leftarrow \mathbf{E}_{(p+1)^2}, \\ e, \varphi \leftarrow \mathcal{A}(p, E, E') \end{matrix} \quad : \quad \varphi \in \mathbf{Iso}(E, E'), \deg(\varphi) = \ell^e \right]$$

is negligible in λ.

In most constructions, the curve E is actually a fixed curve. This could be for functionality or efficiency reasons, but in any case it does not change the hardness of the problem, as one could solve the general problem by finding isogenies from the fixed curve to each of the challenge curves, computing the dual of one of the isogenies and composing them. The fastest classical algorithm known for these problems uses a meet-in-the-middle strategy, and has heuristic running time of $\tilde{O}(p^{1/2})$ bit operations [71, 103].

For isogenies of a fixed degree d, one could instead obtain a quantum algorithm with running time $\tilde{O}(\sqrt[3]{d})$, where time is quantified as the number of isogeny evaluation queries, using Tani's quantum claw-finding algorithm [153]. However, based on the recent proposition of Adj et al. [5] that the van Oorschot–Wiener algorithm [158] is a better classical solution, Jaques and Schanck [105] argued that, in fact, running the query-optimal version of Tani's algorithm to achieve $\tilde{O}(\sqrt[3]{d})$ time would require enough hardware that could be repurposed to run van Oorschot–Wiener algorithm in time $\tilde{O}(\sqrt[4]{d})$. Adopting a reasonable constraint on such hardware, they therefore estimate that both the best classical and quantum algorithms require $\tilde{O}(\sqrt{d})$ time to break the assumption.

The key-exchange protocol CSIDH [35], and other schemes derived from it, use a variant of this assumption in which the curves are defined over $\mathbb{F}_p$ instead of $\mathbb{F}_{p^2}$. In this case, a quantum subexponential algorithm is available, running in time $L_p(1/2)$ [19].

The following assumption has been used in [38] to prove collision-resistance of a proposed hash function. Intuitively, it means that even when the adversary chooses the second curve by taking an isogeny from the first, there is no way from them to produce a different isogeny connecting the two curves.

Assumption 9 (ℓ-power second isogeny)**.** *Let ℓ. For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\begin{bmatrix} p \leftarrow \mathcal{G}(1^\lambda), E \leftarrow \mathbf{E}_{(p+1)^2}, \\ E', e_1, e_2, \varphi_1, \varphi_2 \leftarrow \mathcal{A}(p, E) \end{bmatrix} : \begin{matrix} \varphi_1, \varphi_2 \in \mathbf{Iso}(E, E'), \\ \deg(\varphi_1) = \ell^{e_1}, \deg(\varphi_2) = \ell^{e_2} \end{matrix}$$

is negligible in λ.

The problem of determining the endomorphism ring of a supersingular curve is closely related to the problem of finding isogenies, as endomorphisms are isogenies from a curve to itself.[2]

[2] There are several possible meanings of 'determine the endomorphism ring', but we assume the output should be a $\mathbb{Z}$-basis in the quaternion algebra $\mathbf{B}_{p,\infty}$.

Assumption 10 (endomorphism ring)**.** *Let ℓ_1, ℓ_2 be different prime numbers. For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\left[\begin{matrix} p \leftarrow \mathcal{G}(1^\lambda), E \leftarrow \mathbf{E}_{(p+1)^2}, \\ (w_0, w_1, w_2, w_3) \leftarrow \mathcal{A}(p, E) \end{matrix} \quad : \quad \mathrm{End}(E) \cong \langle w_0, w_1, w_2, w_3 \rangle_{\mathbb{Z}} \subseteq \mathbf{B}_{p,\infty}\right]$$

is negligible in λ.

Essentially, assumption 10 tells us that it is hard to compute the forward direction of Deuring's correspondence. This problem was studied in [117], in which a classical algorithm to solve it was obtained, but with expected running time $\tilde{O}(p)$. It was later improved by Galbraith to $\tilde{O}(p^{\frac{1}{2}})$, under heuristic assumptions [71]. Interestingly, the best quantum algorithm for this problem, due to Biasse, Jao and Sankar [19], runs in time $\tilde{O}(p^{\frac{1}{4}})$, only providing a quadratic speedup over classical algorithms. This has largely motivated the use of supersingular isogeny problems in cryptography.

Note that it is essential that the curve is chosen randomly, as for a few special curves the endomorphism ring is easy to compute. In fact, these 'easy curves' are those often used as the starting curve in isogeny assumptions. One example of easy curve when $p = 3 \bmod 4$ is $E_0/\mathbb{F}_{p^2}$ defined by the equation $Y^2 = X^3 + X$, with j-invariant $j = 1728$, which has endomorphism ring

$$\mathcal{O} = \left\langle 1, \mathbf{i}, \frac{1+\mathbf{i}}{2}, \frac{\mathbf{ij}+\mathbf{j}}{2} \right\rangle.$$

Heuristically, assumption 10 implies assumptions 7, 8 [61, 140]. To compute an endomorphism of E, we take two random walks $\phi_1 : E \to E_1$ and $\phi_2 : E \to E_2$, and obtain an isogeny $\psi : E_1 \to E_2$. Then the composition $\hat{\phi}_2 \psi \phi_1$ is an endomorphism of E. Repeating the process, it is plausible to find four endomorphisms that are linearly independent, thus generating a subring of $\mathrm{End}(E)$. Repeating the process further, we expect to obtain a $\mathbb{Z}$-basis of the full endomorphism ring after having constructed at most $O(\log p + \log D)$ such endomorphisms, where D is a bound on the degree of the isogenies ψ. Indeed the subring index N is bounded by the product of the degrees of its generators which is $(pD)^{O(1)}$, any randomly chosen new element will be in that subring with a probability $1/N$, and every new element not in the subring will decrease the index by at least a factor of 2.

For the converse, suppose that we can compute the endomorphism rings of both E and E', represented as $\mathbb{Z}$-modules in $\mathbf{B}_{p,\infty}$. The strategy is to compute a lattice I in $\mathbf{B}_{p,\infty}$ of appropriate norm that is a left ideal of $\mathrm{End}(E)$ and a right

ideal of $\mathrm{End}(E')$, and to translate it back to the geometric setting to obtain an isogeny. This approach motivated the quaternion ℓ-isogeny algorithm of Kohel-Lauter-Petit-Tignol [61,118,140], which solves in polynomial time the following problem:

Problem 1 (quaternion path)**.** *Let p, ℓ be distinct prime numbers. Let $\mathcal{O}_0, \mathcal{O}_1$ be two maximal orders in $B_{p,\infty}$. Find $k \in \mathbb{N}$ and an ideal I of norm ℓ^k such that I is a left $\mathcal{O}_0$-ideal and its right order is isomorphic to $\mathcal{O}_1$.*

The algorithm can be adapted to produce ideals of B-powersmooth norm for $B \approx \frac{7}{2}\log p$ and using $O(\log p)$ different primes, instead of ideals of norm a power of ℓ. We will use that version in our second signature scheme.

For completeness, we mention that ordinary curve versions of assumptions 7 and 10 are not known to be equivalent. In fact, there is a subexponential algorithm for computing the endomorphism ring of ordinary curves [20], whereas the best classical algorithm known for computing isogenies is still exponential. There is, however, a subexponential quantum algorithm for computing an isogeny between ordinary curves [40], which is why the main interest in cryptography is the supersingular case.

Variants of the isogeny assumption 7, in which some extra information is provided, were first introduced in [54,103] to build an identification scheme, a key exchange protocol and a public-key encryption scheme.

In order to state them we need to introduce some additional notation. Let $\mathcal{G}_{\ell_1,\ell_2}$ be a prime generator outputting primes p of size λ and of the form $\ell_1^{e_1}\ell_2^{e_2} f \pm 1$, where typically ℓ_1 and ℓ_2 will be small prime numbers, and f is a small cofactor, so roughly $\ell_1^{e_1} \approx \ell_2^{e_2} \approx O(p^{1/2})$. Let $E \in \mathbf{E}_p$ with p of this form. We denote by $\mathbf{BE}_n$ the set of $\mathbb{Z}_p$-bases of $E[n]$, and by $\mathbf{Z}_{\ell_i}$ the distribution that outputs uniformly random pairs of elements in $\mathbb{Z}_{\ell_i^{e_i}}$, conditioned on at least one of them not being divisible by ℓ_i.

The *Computational Supersingular Isogeny (CSSI)* assumption is the same as the ℓ-power isogeny assumption (8), but the images of some points through the sought isogeny are given to the attacker.

Assumption 11 (CSSI)**.** *Let ℓ_1, ℓ_2 be different prime numbers. For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\left[\begin{array}{l} p \leftarrow \mathcal{G}_{\ell_1,\ell_2}(1^\lambda), E \leftarrow \mathbf{E}_{(p+1)^2}, \{R_1, S_1 \leftarrow \mathbf{BE}_{\ell_1^{e_1}}\}_{i=1,2}, \\ (m_1, n_1) \leftarrow \mathbf{Z}_{\ell_1}, \varphi : E \to E' = E/\langle m_1 R_1 + n_1 S_1\rangle, \\ e, \tilde{\varphi} \leftarrow \mathcal{A}(p, E, E', \{R_i, S_i\}_{i=1,2}, \varphi(R_2), \varphi(S_2)) \end{array} : \tilde{\varphi} = \varphi\right]$$

can be bounded by $\mathsf{negl}(\lambda)$.

$$\begin{array}{ccc} E & \overset{\varphi}{\dashrightarrow} & E' \\ \{R_i, S_i\}_{i=1,2} & & \varphi(R_2), \varphi(S_2) \end{array}$$

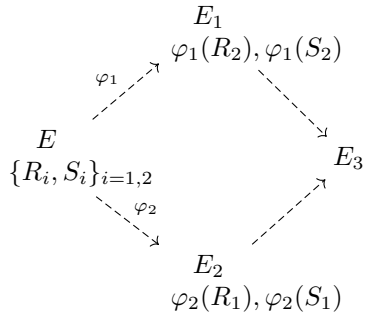
The fastest known algorithms for this problem use a meet-in-the-middle argument. The classical [71, 103] and quantum [66, 103] algorithms have heuristic running time respectively of $\tilde{O}(\ell_1^{e_1/2})$ and $\tilde{O}(\ell_1^{e_1/3})$ bit operations, which is respectively $\tilde{O}(p^{1/4})$ and $\tilde{O}(p^{1/6})$ in the context of De Feo-Jao-Plût [66].

The *Supersingular CDH (SSCDH)* problem shares its structure with the classical CDH problem. From a curve E, two random isogenies $\varphi_1 : E \to E_1, \varphi_2 : E \to E_2$ of degrees $\ell_1^{e_1}, \ell_2^{e_2}$, respectively, are taken. The goal is to find the curve $E_3 = E/H$, where H is spanned by the kernels of φ_1, φ_2.

Assumption 12 (SSCDH)**.** *Let ℓ_1, ℓ_2 be different prime numbers. For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\left[\begin{array}{l} p \leftarrow \mathcal{G}_{\ell_1,\ell_2}(1^\lambda), E \leftarrow \mathbf{E}_{(p+1)^2}, \{R_i, S_i \leftarrow \mathbf{BE}_{\ell_i^{e_i}}\}_{i=1,2} \\ (m_1, n_1) \leftarrow \mathbf{Z}_{\ell_1}, \varphi_1 : E \to E_1 = E/\langle m_1 R_1 + n_1 S_1 \rangle, \\ (m_2, n_2) \leftarrow \mathbf{Z}_{\ell_2}, \varphi_2 : E \to E_2 = E/\langle m_2 R_2 + n_2 S_2 \rangle, \\ E_3 \leftarrow \mathcal{A}\left(p, E, \{E_i, R_i, S_i, \varphi_{3-i}(R_i), \varphi_{3-i}(S_i)\}_{i=1,2}\right) \end{array} : \begin{array}{c} \\ \\ \\ E_3 \cong \\ E/\langle m_i R_i + n_i S_i \rangle_{i=1,2} \end{array}\right]$$

can be bounded by $\mathsf{negl}(\lambda)$.



Similarly, the *Supersingular DDH (SSDDH)* assumption can be defined. It works in the same way, except that instead of asking for E_3, the problem asks the attacker to distinguish the correct E_3 from random.

Finally, they introduced the *Decisional Supersingular Product (DSSP)* assumption, which is similar to the above, but asks the attacker to decide whether the pair E_2, E_3 is the correct one, or is random.

Assumption 13 (DSSP)**.** *Let ℓ_1, ℓ_2 be different prime numbers. For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\left[\begin{array}{l} p \leftarrow \mathcal{G}_{\ell_1,\ell_2}(1^\lambda), E \leftarrow \mathbf{E}_{(p+1)^2}, R_i, S_i \leftarrow \mathbf{BE}_{\ell_2^{e_i}} \text{ for } i = 1,2, \\ (m_1, n_1) \leftarrow \mathbf{Z}_{\ell_1}, \varphi_1 : E \to E_1 = E/\langle m_1 R_1 + n_1 S_1 \rangle, \\ (m_2, n_2) \leftarrow \mathbf{Z}_{\ell_2}, \varphi_2 : E \to E_2^0 = E/\langle m_2 R_2 + n_2 S_2 \rangle, \\ E_3^0 = E_1/\varphi_1(\langle m_2 R_2 + n_2 S_2 \rangle), \\ E_2^1 \leftarrow \mathbf{E}_p, (m', n') \leftarrow \mathbf{Z}_{\ell_1}, E_3^1 = E_2^1/\langle m' R_1 + n' S_1 \rangle, b \leftarrow \{0,1\}, \\ \tilde{b} \leftarrow \mathcal{A}(p, E, E_1, R_1, S_1, R_2, S_2, \varphi_1(R_2), \varphi_1(S_2), E_2^b, E_3^b) \end{array} : \tilde{b} = b\right]$$

can be bounded by $\mathsf{negl}(\lambda)$.

We stress that assumptions 11, 12 and 13 are potentially stronger than assumptions 7 and 8 because special primes are used and extra points are revealed. In [138], Petit studied how these additional isogeny images lead to attacks for certain parameter configurations. Furthermore, it is shown in Section 4 of [74] that if $\mathrm{End}(E)$ is known and one can find any isogeny from E to E' then one can compute the specific isogeny of degree $\ell_1^{e_1}$.

In [154, Definitions 2 and 3], the authors consider the problem of, given two curves E_1, E_2 such that there exists an isogeny φ between them, and a basis $\{P, Q\}$ of the N-torsion of E_1, computing $\varphi(P), \varphi(Q)$. We consider the decisional variant of this problem. However, we cannot expect indistinguishability between images of torsion points and random points of the N-torsion of E_2, as we always have that

$$e(\varphi(R), \varphi(S)) = e(R, S)^{\deg \varphi}.$$

We therefore impose this on the latter.

Assumption 14 (DIP (decisional isogeny-pairing))**.** *Let $d, N \in \mathbb{N}$ such that*

$\gcd(d, N) = 1$. *For any PPT adversary* $\mathcal{A}$*,*

$$\mathsf{Pr}\left[\begin{array}{l} p \leftarrow \mathcal{G}_{\ell_1,\ell_2}(1^\lambda), E \leftarrow \mathbf{E}_{(p+1)^2}, R, S \leftarrow \mathbf{BE}_N, \\ H \leftarrow \{\textit{subgroups of } E[d] \textit{ of size } d\}, \varphi : E \to E' = E/H, \\ \overline{R}^0, \overline{S}^0 \leftarrow E'[N] \textit{ conditioned on } e(\overline{R}^0, \overline{S}^0) = e(R,S)^d, \quad : \quad \tilde{b} = b \\ \overline{R}^1 = \varphi(R), \overline{S}^1 = \varphi(S), b \leftarrow \{0,1\}, \\ \tilde{b} \leftarrow \mathcal{A}(p, E, E', R, S, \overline{R}^b, \overline{S}^b) \end{array}\right]$$

can be bounded by $\mathsf{negl}(\lambda)$*.*

We prove that sampling elements of the second distribution is efficient. Indeed, let R, S be a basis of $E_2[N]$. We can identify torsion points $xR + yS$ with elements $(x, y) \in \mathbb{Z}_N \times \mathbb{Z}_N$. Then we are looking for pairs $(a, b), (c, d)$ that verify the pairing equation. We can sample them in the following way. We write $N = \prod_{i=1}^{k} \ell_i^{e_i}$, where the ℓ_i are the prime factors of N. We denote the order of x by $|x|$.

1. Choose $a, b \leftarrow \mathbb{Z}_N$ such that $|(a, b)| = N$.
2. If $\exists a^{-1} \in \mathbb{Z}_{\ell_i^{e_i}}$, choose $c_i \leftarrow \mathbb{Z}_{\ell_i^{e_i}}$, else if $\exists b^{-1} \in \mathbb{Z}_{\ell_i^{e_i}}$, choose $d_i \leftarrow \mathbb{Z}_{\ell_i^{e_i}}$.
3. Solve $ad_i - bc_i = t \bmod \ell_i^{e_i}$ for all $i = 1, \ldots, k$.
4. Recover $a, b, c, d \bmod N$ via Chinese remainder theorem.

We now show why this algorithm works and produces uniformly random pairs verifying the condition above. We first note that in Step 2, we will always have that either a or b has multiplicative inverse. We note that $|(a, b)| = N$ over $\mathbb{Z}_N$ implies $|(a, b)| = \ell_i^{e_i}$ over $\mathbb{Z}_{\ell_i^{e_i}}$, and since

$$|(a, b)| = \operatorname{lcm}(|a|, |b|),$$

this in turn implies that either a or b is of maximal order in $\mathbb{Z}_{\ell_i^{e_i}}$.

We have that

$$e(aR + bS, cR + dS) = e(R, S)^{ad-bc},$$

using that the pairing is bilinear and alternating. Then we want to impose condition (3),

$$e(R, S)^{ad-bc} = e(P, Q)^{\deg \varphi},$$

which is equivalent to

$$x(ad - bc) = \deg \varphi,$$

where x is the discrete logarithm of $e(R, S)$ with respect to $e(P, Q)$ (this can be efficiently computed as long as N is smooth). Therefore, the pairs satisfying condition (3) above are the solutions of the equation

$$ad - bc = t,$$

where $t = x^{-1} \deg \varphi$ (note that x is invertible because $\{R, S\}$ is a basis of $E_2[N]$). Finally, the equation modulo prime powers can be solved as

$$d_i = a^{-1}(t + bc_i) \mod \ell_i^{e_i}, \qquad \text{or} \qquad c_i = b^{-1}(ad_i - t) \mod \ell_i^{e_i},$$

depending on whether a or b is invertible.

Part I

Classical zero-knowledge proofs

Chapter 3

Proofs of same opening

This chapter is based on the paper 'Smaller QA-NIZK proofs of same opening for bilateral commitments' [146], which is a joint work with Carla Ràfols, and was published at Africacrypt 2020.

Bilinear groups have been used to design countless cryptographic protocols, some of them with no equivalent in other settings. In particular, such groups have been very useful to design non-interactive zero-knowledge (NIZK) proofs in the common reference string (CRS) model. The first works to realize that pairings allowed for the construction of efficient NIZK proofs were [27,87,95,96], culminating in the work of Groth–Sahai [97]. The latter presents a NIZK proof system for satisfiability of most types of linear and quadratic equation in bilinear groups, in the CRS model and under standard, constant size and weak assumptions. Groth–Sahai proofs are one of the fundamental building blocks in pairing-based cryptography, with well-known applications as anonymous credentials [68], e-Cash [10], ring signatures [37], shuffles [94], signatures of knowledge [14], and tight CCA encryption [99].

Groth–Sahai proofs follow the usual commit-and-prove paradigm: first, the prover commits to the solution of the equation, and then produces a "proof" formed of some group elements, which the verifier uses together with the commitments to get convinced of the satisfiability of the equation. The commit-and-prove framework is used implicitly in the original work of Groth–Sahai [97], and formalized explicitly in [62,68]. In this view, a NIZK proof proves some property of a committed value, and many different statements about a single committed

value can be proven.[1] This formalization is also a conceptually cleaner approach. It allows to differentiate clearly between the "commit" and the "proof" part among all the elements computed by the prover. In this work we also make the separation between commitment and proof, so when we discuss proof sizes we refer exclusively to the latter part.

For many equation types, the Groth–Sahai proof system is still the state of the art. Few improvements are known, like the general techniques to replace dual mode commitments by ElGamal ciphertexts [62], aggregation of many Groth–Sahai proofs [85, 109], which are of limited applicability, or some techniques to encode partial satisfiability [145].

A notable exception are quasi-adaptive NIZK (QA-NIZK) arguments of membership in linear spaces over a source group [109, 115, 124], introduced by Jutla–Roy [108], which allow to prove satisfiability of linear equations. More precisely, let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be an asymmetric bilinear group equipped with a pairing. Such QA-NIZK arguments allow to prove that a vector $[\boldsymbol{y}]_1 \in \mathbb{G}_1^n$ is of the form $\boldsymbol{y} = \mathbf{M}\boldsymbol{w}$, for some public matrix $[\mathbf{M}]_1 \in \mathbb{G}_1^{n\times t}$. These arguments are extremely efficient: under an assumption weaker than DDH, their size is only 1 group element, for most distributions of $[\mathbf{M}]_1$.[2] The same statement proven with Groth–Sahai proofs requires $O(t)$ elements for committing to $\boldsymbol{w}$ and $O(n)$ elements to prove that $\boldsymbol{y}$ is of this form.

Because of their efficiency, these arguments have many applications, for instance to different flavors of identity-based encryption [108] or group signatures [125]. These arguments also have a close relation to structure-preserving signatures [2, 4, 114]. Membership in linear spaces naturally encodes statements about ciphertexts and commitments: for example, two ElGamal ciphertexts (or more generally, any 'algebraic' commitment scheme, like Pedersen or Groth–Sahai commitments) encrypt the same message if their difference is in a certain linear space dependent of the public key. More generally, QA-NIZK arguments allow to aggregate proofs easily: proving that two vectors of ElGamal commitments open pairwise to the same value requires only one group element, using the constructions of Kiltz–Wee [115], and the security relies on the KerMDH assumption (5). On the other hand, with the Groth–Sahai proof system, this requires two elements of each group $\mathbb{G}_1, \mathbb{G}_2$ for each pair of ciphertexts.

In this chapter, we consider the problem of proving that two commitments, one in $\mathbb{G}_1$ and one in $\mathbb{G}_2$, open to the same value. This statement appears naturally when one wants to prove quadratic relations in asymmetric bilinear

[1] In contrast, if one thinks of Groth–Sahai proofs as NIZK proofs of satisfiability of quadratic equations, formally commitments cannot be reused across proofs.

[2] More precisely, $[\mathbf{M}]_1$ should be taken from a witness sampleable distribution.

groups. Indeed, suppose that we want to prove that a commitment opens to a bit, that is, that the opening of some commitments satisfies the quadratic equation $X(X-1)=0$. This often appears as part of a larger proof, for example in ring signatures [37,82,84], e-voting [36] or range proofs [31]. To prove that a commitment opens to a bit, Groth–Sahai proofs proceed as follows:

1. Rewrite the equation as $X(Y-1)=0$.
2. Commit to a solution: $[c]_1 = \mathsf{Com}(x;r)$ and $[d]_2 = \mathsf{Com}(y;s)$.
3. Prove satisfiability of the equation $X(Y-1)=0$ using the commitments c, d and providing some additional proof elements.
4. Prove that the commitments c, d open to the same value.

We note that step 4 is proving the linear equation $X = Y$. Informally, the idea is that step 3 is a quadratic check which requires commitments in different groups, and step 4 makes sure there is some consistency between these values. Formally, the need for it arises from the fact that Groth–Sahai proofs work for disjoint sets of variables in $\mathbb{G}_1$ and $\mathbb{G}_2$.

This is one of the main techniques for proving quadratic equations in $\mathbb{Z}_p$ in bilinear groups (in the CRS model and under standard assumptions), and any efficiency improvement in the same opening step (4) would have a direct impact on the overall efficiency. We note that there is another construction, introduced very recently in [48], that proves that a commitment over $\mathbb{G}_1$ opens to either 0 or 1. Their approach consists of using a pairing to compile interactive arguments into non-interactive ones, and they manage to prove that a commitment opens to a bit with 7 group elements. For comparison, the Groth–Sahai approach requires 10 group elements using our approach. Groth–Sahai proofs still seem better for proving that n commitments to a bit: in [48] the proof scales linearly, whereas if we use the aggregated version of our scheme, n proofs require $6n+3$ elements.

Our results. To the best of our knowledge, there are two ways of proving step 4. One is to use standard Groth–Sahai proofs, which requires 2 group elements in each of $\mathbb{G}_1$ and $\mathbb{G}_2$. The alternative is to use QA-NIZK arguments of membership in linear spaces. However, because the statement is split between $\mathbb{G}_1$ and $\mathbb{G}_2$, we need to resort to arguments of membership in bilateral spaces, which show, for two vectors $[\boldsymbol{x}]_1, [\boldsymbol{y}]_2$, and some matrices $[\mathbf{M}]_1, [\mathbf{M}]_2$ that there exists some $\boldsymbol{w}$ such that $\boldsymbol{x} = \mathbf{M}\boldsymbol{w}$ and $\boldsymbol{y} = \mathbf{N}\boldsymbol{w}$. These were constructed by

González et al. [85] under the SKerMDH assumption (6) However, this does not improve step (4) over the cost of Groth–Sahai proofs. The proof of González et al. only improves on the state of the art for the aggregated case, namely to show that n pairs of commitments open (pairwise) to the same value with a proof made of 2 elements in $\mathbb{G}_1$ and 2 elements in $\mathbb{G}_2$, independent of n. However, this is not an improvement for a single pair of commitments.

Noticing the gap between one element for one-sided proofs and four elements for bilateral proofs, a natural question is how much we can reduce the proof size in the bilateral case. We give a construction which reduces the proof size of [85] to three elements, while maintaining the same computational assumption in the soundness proof.

We note that this is the first concrete improvement for step (4) since the publication of the work of Groth–Sahai. Our result is a sophisticated combination of the techniques of Kiltz–Wee [115] and González et al. [85]. Additionally, we argue that our constructions are optimal, by showing that any two-element proof is vulnerable to a simple attack.

Our techniques. We briefly review the linear space membership proof of Kiltz–Wee [115]. Their core idea is a clever translation to the bilinear group setting of a hash proof system, which is essentially a NIZK proof in the symmetric key setting. Given a matrix $\mathbf{M} \in \mathbb{Z}_p^{m\times t}$, the starting point is a proof system for the language

$$\mathcal{L}_{\mathbf{M}} = \{[\boldsymbol{c}]_1 \leftarrow \mathbb{G}_1^m \mid \exists \boldsymbol{w} \text{ s. t. } \boldsymbol{c} = \mathbf{M}\boldsymbol{w}\}$$

which works as follows: prover and verifier share a key $\mathbf{K} \leftarrow \mathbb{Z}_p^{m\times(k+1)}$, where k will depend on the hardness assumption used to ensure soundness. The projection $[\mathbf{M}^\top\mathbf{K}]_1$ is published in the CRS. The prover sends $[\boldsymbol{\pi}]_1 = \boldsymbol{w}^\top[\mathbf{M}^\top\mathbf{K}]_1$, and the verifier checks that

$$[\boldsymbol{c}^\top]_1\mathbf{K} \stackrel{?}{=} [\boldsymbol{\pi}]_1.$$

Intuitively, the proof is sound because if $\boldsymbol{c}$ is not in $\mathrm{Im}(\mathbf{M})$ then $\boldsymbol{c}^\top\mathbf{K}$ is uniformly random given $\mathbf{M}^\top\mathbf{K}$, and thus there is no way for the prover to produce such a proof.

Kiltz–Wee take this idea and remove the need for a shared secret key by using a bilinear group. Now the CRS includes $[\mathbf{A}, \mathbf{K}\mathbf{A}]_2$, for a matrix $\mathbf{A} \in \mathbb{Z}_p^{(k+1)\times k}$. This partially fixes $\mathbf{K}$ without revealing it, the goal being that the verifier can use these elements to verify without needing to know $\mathbf{K}$ as before. The proof is still the same, but the verification is now

$$e([\boldsymbol{c}^\top]_1, [\mathbf{K}\mathbf{A}]_2) \stackrel{?}{=} e([\boldsymbol{\pi}]_1, [\mathbf{A}]_2).$$

By assuming the hardness of the KerMDH problem on $\mathbf{A}$, i.e., it is hard to find non-trivial cokernel elements of $\mathbf{A}$, we are essentially back to the argument of the hash proof system. For the right choice of distribution of $\mathbf{A}$, the assumption is believed to hold starting at $k = 1$, so in this case we have that the proof is formed of 2 group elements.

However, this can be taken one step further. Assuming that the distribution of $[\mathbf{M}]_1$ is witness sampleable, that is, that we can efficiently sample $\tilde{\mathbf{M}}$ such that $[\tilde{\mathbf{M}}]_1$ is distributed as $[\mathbf{M}]_1$, then it is enough to use the truncated matrix $\overline{\mathbf{A}} \in \mathbb{Z}_p^{k \times k}$ instead of $\mathbf{A}$, thus using $\mathbf{K} \in \mathbb{Z}_p^{m \times k}$, which yields proofs consisting of only one group element.

We now consider the natural generalization of this approach to bilateral proofs, as developed by González et al. [85].[3] Consider the following language:

$$\mathcal{L}_{\mathbf{M},\mathbf{N}} = \{([\boldsymbol{c}]_1, [\boldsymbol{d}]_2) \leftarrow \mathbb{G}_1^m \times \mathbb{G}_2^n \mid \exists \boldsymbol{w} \text{ s. t. } \boldsymbol{c} = \mathbf{M}\boldsymbol{w}, \boldsymbol{d} = \mathbf{N}\boldsymbol{w}\}.$$

To account for two-sided statements, we consider one key $\mathbf{K}$ for $\mathbb{G}_1$ and one key $\mathbf{L}$ for $\mathbb{G}_2$, and so we publish the following elements in the CRS:

$$[\mathbf{M}^\top \mathbf{K} + \mathbf{Z}, \mathbf{A}, \mathbf{L}\mathbf{A}]_1, [\mathbf{N}^\top \mathbf{L} - \mathbf{Z}, \mathbf{A}, \mathbf{K}\mathbf{A}]_2,$$

where $\mathbf{Z} \in \mathbb{Z}_p^{t \times k}$. The prover produces the proofs $[\boldsymbol{\pi}]_1 = \boldsymbol{w}^\top [\mathbf{M}^\top \mathbf{K} + \mathbf{Z}]_1$ and $[\boldsymbol{\theta}]_2 = \boldsymbol{w}^\top [\mathbf{N}^\top \mathbf{L} - \mathbf{Z}]_2$, and the verifier checks the equation

$$e([\boldsymbol{c}^\top]_1, [\mathbf{K}\mathbf{A}]_2) + e([\mathbf{L}\mathbf{A}]_1, [\boldsymbol{d}]_2) \stackrel{?}{=} e([\boldsymbol{\pi}]_1, [\mathbf{A}]_2) + e([\mathbf{A}]_1, [\boldsymbol{\theta}]_2). \tag{3.1}$$

Intuitively, the term $\mathbf{Z}$ in the CRS elements produces terms in the verification equation that will not cancel out unless $\boldsymbol{w}$ is the same in both sides. In a similar way as above, the soundness of this scheme reduces to the hardness of a SKerMDH problem. However, split kernel problems are easy for $k = 1$, and so we must take at least $k = 2$. This has a direct impact on the sizes of the keys $\mathbf{K}$ and $\mathbf{L}$, and so this approach yields proofs of two group elements in $\mathbb{G}_1$, and two in $\mathbb{G}_2$, and two verification equations.

Our strategy to reduce the proof size is to use only one element in $\mathbb{G}_2$, so instead of having $\boldsymbol{\theta} = (\theta, \hat{\theta})$ as above, we reuse the same θ. To make it work, we require the condition that the columns of $\mathbf{N}^\top \mathbf{L}$ are equal, so that $\boldsymbol{\theta} = (\theta, \theta)$, and it is enough to send it once. This introduces extra complexity in the CRS generation, and the simulation of the CRS for the adversary in the soundness

[3] The actual construction requires some masking terms to ensure zero-knowledge, but we omit these for simplicity of the presentation.

security reduction, particularly in the aggregated case. We present the proof directly for the most efficient case, $k = 2$.

To solve these new issues, we need to reformulate the problem slightly. Instead of considering the pair of commitments $([\boldsymbol{c}]_1, [\boldsymbol{d}]_2)$ as the statement, we consider just $[\boldsymbol{c}]_1$, and build a proof of F-knowledge of $F(\boldsymbol{w}) = [\boldsymbol{w}]_{1,2}$. Indeed, in applications the commitment $[\boldsymbol{d}]_2$ is an artifact of the proof, as when proving quadratic statements we need to split the commitments between $\mathbb{G}_1$ and $\mathbb{G}_2$ to exploit the pairing. Regarding zero-knowledge, this change implies that the simulator knows the opening of one of the commitments. We note that both openings are required for proving zero-knowledge in Groth–Sahai proofs.

We stress that our modified formalization is due to the intricacies of the soundness reduction, and has no actual impact in most applications. This is because, as we have seen in the proof of $X(X-1) = 0$ above, the commitment in $\mathbb{G}_2$ is a byproduct of the proof, and thus can be seen as part of it, while the 'meaningful' statement is about the commitment in $\mathbb{G}_1$.

Interestingly, our trick of reusing θ does not work for both sides, and in fact in Section 3.4 we show an attack for any two-element proof of this form. We argue that the general form of any proof of bilateral same opening consisting of only two elements must have a verification equations that looks essentially like equation (3.1) above, but with π, θ scalars instead of vectors; then we show a simple algebraic attack that exploits the two-sided nature of the proof.

Organization. In Section 3.1 we describe the commitments used and explain how the problem of same opening can be seen as a problem of membership in a linear space. Sections 3.2 and 3.3 contain the non-aggregated and aggregated versions of the scheme, respectively. We introduce the simpler version first for readability. In Section 3.4 we argue that our constructions have optimal proof size.

3.1 Linear relations in a bilinear group

3.1.1 Dual-mode algebraic commitment schemes

We present the type of commitments for which our QA-NIZK arguments can be used. These generalize many common schemes, like (multi-)Pedersen commitments and Groth–Sahai commitments. Our commitments are in the source groups, $\mathbb{G}_\gamma$ for $\gamma = 1, 2$, of a bilinear group. Let $\mathbf{F} \in \mathbb{Z}_p^{m \times n}$ and $\mathbf{U} \in \mathbb{Z}_p^{m \times \ell}$ be full-rank matrices. The commitment key is $ck = [\mathbf{F}, \mathbf{U}]_\gamma$, and the commitment

to a message $\boldsymbol{x} \in \mathbb{Z}_p^n$ with randomness $\boldsymbol{r} \in \mathbb{Z}_p^\ell$ is defined as

$$\mathsf{Com}_{ck}(\boldsymbol{x}; \boldsymbol{r}) = [\mathbf{F}\boldsymbol{x} + \mathbf{U}\boldsymbol{r}]_\gamma.$$

Choosing the appropriate distributions for $[\mathbf{F}, \mathbf{U}]_\gamma$, we can have two commitment keys, one that produces a perfectly binding commitment scheme and one that produces a perfectly hiding commitment scheme, and these two key distributions are computationally indistinguishable under a MDDH assumption (see [64] for details). In the description of our schemes and the soundness proofs we will use the perfectly binding key, switching to perfectly hiding to argue that our schemes are zero-knowledge.

The most well-known example is Groth–Sahai commitments to integers: given $x \in \mathbb{Z}_p$ and randomness $r \in \mathbb{Z}_p$, this is an instantiation of the commitment defined above, with the matrices

$$\mathbf{F} \leftarrow \mathbb{Z}_p^2, \qquad \mathbf{U} \leftarrow \mathbb{Z}_p^2,$$

when in perfectly binding mode, and

$$\mathbf{F} \leftarrow \mathbb{Z}_p^2, \qquad \mathbf{U} = \lambda\mathbf{F},$$

for $\lambda \leftarrow \mathbb{Z}_p$, when in perfectly hiding mode.

3.1.2 Linear equations in a bilinear group

A set of linear equations split between the two sides of a bilinear group can be written as

$$\begin{pmatrix} [\boldsymbol{c}]_1 \\ [\boldsymbol{d}]_2 \end{pmatrix} = \begin{pmatrix} [\mathbf{M}]_1 \\ [\mathbf{N}]_2 \end{pmatrix} \boldsymbol{X},$$

where $\boldsymbol{X}$ is the vector of unknowns, $[\boldsymbol{c}, \mathbf{M}]_1$ are the coefficients in $\mathbb{G}_1$ and $[\boldsymbol{d}, \mathbf{N}]_2$ are the coefficients in $\mathbb{G}_2$. Thus, proving satisfiability of this system is equivalent to proving that there exist some vector $\boldsymbol{w}$ such that

$$\boldsymbol{w} \in \mathbf{Im} \begin{pmatrix} \mathbf{M} \\ \mathbf{N} \end{pmatrix}.$$

Thus, these proofs are usually seen as proofs of membership in a linear subspace, in this case split between $\mathbb{G}_1$ and $\mathbb{G}_2$. The problem of same opening of two algebraic commitments,

$$[\boldsymbol{c}]_1 = \mathsf{Com}_{ck_1}(\boldsymbol{x}; \boldsymbol{r}) = [\mathbf{F}\boldsymbol{x} + \mathbf{U}\boldsymbol{r}]_1, \qquad [\boldsymbol{d}]_1 = \mathsf{Com}_{ck_2}(\boldsymbol{x}; \boldsymbol{s}) = [\mathbf{G}\boldsymbol{x} + \mathbf{V}\boldsymbol{s}]_2$$

can be seen in this framework of membership in linear spaces, where

$$\begin{pmatrix} [\boldsymbol{c}]_1 \\ [\boldsymbol{d}]_2 \end{pmatrix} = \left(\begin{array}{c|c|c} [\mathbf{F} & \mathbf{U} & \mathbf{0}]_1 \\ [\mathbf{G} & \mathbf{0} & \mathbf{V}]_2 \end{array}\right) \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{r} \\ \boldsymbol{s} \end{pmatrix}.$$

Since we are particularly interested in the case of same opening, we present our constructions directly for this application, although it would be easy to generalize to any matrices $[\mathbf{M}]_1, [\mathbf{N}]_2$, as long as they verify some conditions on their dimensions. As a warm-up, we develop first a non-aggregated version of the proof, as the main ideas are easier to visualize in this case.

3.2 Non-aggregated scheme

Given $x \in \mathbb{Z}_p$ and two commitments $[\boldsymbol{c}]_1, [\boldsymbol{d}]_2$ to x, we provide a proof of both commitments opening to the same element x. More precisely, given a group description gk and commitment keys $ck_1 = [\boldsymbol{f}, \boldsymbol{u}]_1 \in \mathbb{G}_1^{2\times 2}$ and $ck_2 = [\boldsymbol{g}, \boldsymbol{v}]_2 \in \mathbb{G}_2^{2\times 2}$, we want to prove F-knowledge in the language

$$\mathcal{L}_{gk,ck_1} = \{[\boldsymbol{c}]_1 \in \mathbb{G}_1^2 \mid \exists x, r \text{ s. t. } [\boldsymbol{c}]_1 = \mathsf{Com}_{ck_1}(x; r) = [x\boldsymbol{f} + r\boldsymbol{u}]_1\},$$

where $F(x, r) = [x]_{1,2}$.

- $gk := (p, \mathcal{P}_1, \mathcal{P}_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$.
- $\mathcal{K}_0(gk)$: set $ck_1 = [\boldsymbol{f}, \boldsymbol{u}]_1 \leftarrow \mathcal{D}_{\mathsf{par}}$, where $\mathcal{D}_{\mathsf{par}}$ is witness sampleable, that is, there exists an efficiently sampleable distribution $\tilde{\mathcal{D}}_{\mathsf{par}}$ outputting $(\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{u}})$ such that $[\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{u}}]_1$ is distributed as $[\boldsymbol{f}, \boldsymbol{u}]_1$.
- $\mathcal{K}_1(gk, ck_1)$: set $ck_2 = [\boldsymbol{g}, \boldsymbol{v}]_2$, where $\boldsymbol{g}, \boldsymbol{v} \leftarrow \mathbb{Z}_p^2$. Choose $a_1, a_2 \leftarrow \mathbb{Z}_p$ and also $\boldsymbol{k}_u, \hat{\boldsymbol{k}}_u, \boldsymbol{l}_v, \hat{\boldsymbol{l}}_v \leftarrow \mathbb{Z}_p^2$ conditioned on

$$\boldsymbol{l}_v^\top \boldsymbol{v} = \hat{\boldsymbol{l}}_v^\top \boldsymbol{v}, \tag{3.2}$$

  Finally, choose $z_2 \leftarrow \mathbb{Z}_p$ and set

$$w = \frac{\boldsymbol{k}_u^\top \boldsymbol{f}}{\boldsymbol{l}_v^\top \boldsymbol{g}}, \qquad z_1 = z_2 w,$$

$$\hat{w} = \frac{\hat{\boldsymbol{k}}_u^\top \boldsymbol{f}}{\hat{\boldsymbol{l}}_v^\top \boldsymbol{g}}, \qquad \hat{z}_1 = z_2 \hat{w}.$$

Algorithm $\mathcal{K}_1$ outputs the following CRS:

$$\begin{pmatrix} gk, ck_1, [\boldsymbol{k}_u^\top \boldsymbol{u}]_1, [\hat{\boldsymbol{k}}_u^\top \boldsymbol{u}]_1, [a_1 w]_1, [a_2 \hat{w}]_1, [a_1 w \boldsymbol{l}_v]_1, [a_2 \hat{w} \hat{\boldsymbol{l}}_v]_1, [z_1]_1, [\hat{z}_1]_1, \\ ck_2, [\boldsymbol{l}_v^\top \boldsymbol{v}]_2, [a_1]_2, [a_2]_2, [a_1 \boldsymbol{k}_u]_2, [a_2 \hat{\boldsymbol{k}}_u]_2, [z_2]_2 \end{pmatrix}.$$

- $\mathcal{P}(\mathsf{crs}, ([\boldsymbol{c}]_1, x, r) \in \mathcal{R})$: commit to x in $\mathbb{G}_2$ by choosing $s \leftarrow \mathbb{Z}_p$ and setting

$$[\boldsymbol{d}]_2 = \mathsf{Com}_{ck_2}(x, s) = [x\boldsymbol{g} + s\boldsymbol{v}]_2.$$

  Choose $\delta \leftarrow \mathbb{Z}_p$ and output $[\boldsymbol{d}]_2$ and

$$[\pi]_1 = [r\boldsymbol{k}_u^\top \boldsymbol{u} + \delta z_1]_1, \qquad [\theta]_2 = [s\boldsymbol{l}_v^\top \boldsymbol{v} + \delta z_2]_2,$$
$$[\hat{\pi}]_1 = [r\hat{\boldsymbol{k}}_u^\top \boldsymbol{u} + \delta \hat{z}_1]_1,$$

- $\mathcal{V}(\mathsf{crs}, [\boldsymbol{c}]_1, ([\boldsymbol{d}, \theta]_2, [\pi, \hat{\pi}]_1)$: The algorithm outputs 1 iff the following equations hold:

$$e\left([\boldsymbol{c}^\top]_1, [a_1 \boldsymbol{k}_u]_2\right) - e([a_1 w \boldsymbol{l}_v^\top]_1, [\boldsymbol{d}]_2) \stackrel{?}{=} e([\pi]_1, [a_1]_2) - e([a_1 w]_1, [\theta]_2),$$
$$e\left([\boldsymbol{c}^\top]_1, [a_2 \hat{\boldsymbol{k}}_u]_2\right) - e([a_2 \hat{w} \hat{\boldsymbol{l}}_v^\top]_1, [\boldsymbol{d}]_2) \stackrel{?}{=} e([\hat{\pi}]_1, [a_2]_2) - e([a_2 \hat{w}]_1, [\theta]_2).$$

Completeness. Both equations are analogous, and it is easy to see that for honest provers, using that $\boldsymbol{f}^\top \boldsymbol{k}_u = w(\boldsymbol{l}_v^\top \boldsymbol{g})$, we have that

$$\boldsymbol{c}^\top (a_1 \boldsymbol{k}_u) - (a_1 w \boldsymbol{l}_v^\top)\boldsymbol{d} = (x\boldsymbol{f}^\top + r\boldsymbol{u}^\top)(a_1 \boldsymbol{k}_u) - (a_1 w \boldsymbol{l}_v^\top)(x\boldsymbol{g} + s\boldsymbol{v}) =$$
$$= a_1 x \boldsymbol{f}^\top \boldsymbol{k}_u - a_1 x (w \boldsymbol{l}_v^\top \boldsymbol{g}) + (r\boldsymbol{u}^\top \boldsymbol{k}_u) a_1 - a_1 w (s\boldsymbol{v}^\top \boldsymbol{l}_v) = \pi a_1 - a_1 w \theta.$$

F-extractor. We now define the algorithm that, given the extraction key $xk = (\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{u}, \boldsymbol{v})$, outputs a function of the witness, in this case $F(x, r) = [x]_{1,2}$.

- $\mathsf{Ext}_{xk}([\boldsymbol{c}]_1, [\boldsymbol{d}]_2)$: knowing $\boldsymbol{f}, \boldsymbol{u}$, we can find a vector $\boldsymbol{u}^\perp$ such that $\boldsymbol{u}^\top \boldsymbol{u}^\perp = 0$ and $\boldsymbol{f}^\top \boldsymbol{u}^\perp = 1$, and compute $[\boldsymbol{c}^\top]_1 \boldsymbol{u}^\perp = [x]_1$. Similarly, we obtain $[x]_2$ from $[\boldsymbol{d}]_2$, using $\boldsymbol{g}, \boldsymbol{v}$.

Theorem 6. *The above scheme is computationally F-knowledge sound under the $\mathcal{RL}_2$-SKerMDH assumption (6). More precisely, there exists an adversary $\mathcal{B}$ against the $\mathcal{RL}_2$-SKerMDH problem such that for any PPT adversary $\mathcal{A}$, we have that*

$$\mathsf{Adv}_{F-\mathsf{KnowledgeSoundness}}\mathcal{A}(1^\lambda) \leq \mathsf{Adv}_{\mathcal{RL}_2\text{-SKerMDH}}\mathcal{B}(1^\lambda).$$

Proof. We assume the existence of an adversary $\mathcal{A}$ against the F-knowledge soundness of the scheme (that is, $\mathcal{A}$ is able to produce a statement and and an accepting proof such that

$$\mathsf{Ext}_{xk}([\boldsymbol{c}]_1, [\boldsymbol{d}]_2) = ([x]_1, [y]_2),$$

and $x \neq y$), and we use it to build an adversary $\mathcal{B}$ against the $\mathcal{RL}_2$-SKerMDH problem. $\mathcal{B}$ receives the challenge matrix

$$[\mathbf{A}]_{1,2} = [\boldsymbol{a}_1|\boldsymbol{a}_2]_{1,2} = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \\ r_1 & r_2 \end{bmatrix}_{1,2},$$

and builds the environment for $\mathcal{A}$ as follows. $\mathcal{B}$ samples $\boldsymbol{f}, \boldsymbol{u} \leftarrow \tilde{\mathcal{D}}_{\mathsf{par}}$ and $\boldsymbol{k}'_u, \hat{\boldsymbol{k}}'_u \leftarrow \mathbb{Z}_p^2$, and $\boldsymbol{u}^\perp \leftarrow \mathbb{Z}_p^2$ conditioned on $\boldsymbol{u}^\top \boldsymbol{u}^\perp = 0$. Implicitly, $\mathcal{B}$ defines

$$\boldsymbol{k}_u = \boldsymbol{k}'_u + a_1^{-1} r_1 \boldsymbol{u}^\perp, \qquad \hat{\boldsymbol{k}}_u = \hat{\boldsymbol{k}}'_u + a_2^{-1} r_2 \boldsymbol{u}^\perp.$$

Observe that this implies that

$$a_1 \boldsymbol{k}_u = a_1 \boldsymbol{k}'_u + r_1 \boldsymbol{u}^\perp, \qquad a_2 \hat{\boldsymbol{k}}_u = a_2 \hat{\boldsymbol{k}}'_u + r_2 \boldsymbol{u}^\perp, \tag{3.3}$$

which $\mathcal{B}$ can compute in $\mathbb{G}_2$. For the other side, $\mathcal{B}$ samples $\boldsymbol{g}, \boldsymbol{v} \leftarrow \mathbb{Z}_p^2$ and $\boldsymbol{l}'_v \leftarrow \mathbb{Z}_p^2$, and let $\boldsymbol{v}^\perp \in \mathbb{Z}_p^2$ be the unique vector such that $\boldsymbol{v}^\top \boldsymbol{v}^\perp = 0$ and

$$\boldsymbol{f}^\top \boldsymbol{u}^\perp = \boldsymbol{g}^\top \boldsymbol{v}^\perp. \tag{3.4}$$

$\mathcal{B}$ defines

$$w = \frac{\boldsymbol{k}'^\top_u \boldsymbol{f}}{\boldsymbol{l}'^\top_v \boldsymbol{g}}, \qquad \hat{w} = \frac{\hat{\boldsymbol{k}}'^\top_u \boldsymbol{f}}{\boldsymbol{l}'^\top_v \boldsymbol{g}}, \tag{3.5}$$

(note that $\boldsymbol{l}'_v$ is the same in both), and implicitly

$$\boldsymbol{l}_v = \boldsymbol{l}'_v + (a_1 w)^{-1} r_1 \boldsymbol{v}^\perp, \qquad \hat{\boldsymbol{l}}_v = \boldsymbol{l}'_v + (a_2 \hat{w})^{-1} r_2 \boldsymbol{v}^\perp,$$

which means that

$$a_1 w \boldsymbol{l}_v = a_1 w \boldsymbol{l}'_v + r_1 \boldsymbol{v}^\perp, \qquad a_2 \hat{w} \hat{\boldsymbol{l}}_v = a_2 \hat{w} \boldsymbol{l}'_v + r_2 \boldsymbol{v}^\perp, \tag{3.6}$$

and these can be computed in $\mathbb{G}_1$. Note that, by construction,

$$\frac{a_1 \boldsymbol{f}^\top \boldsymbol{k}_u}{a_1 w \boldsymbol{g}^\top \boldsymbol{l}_v} = \frac{a_1 \boldsymbol{f}^\top \boldsymbol{k}'_u + r_1 \boldsymbol{f}^\top \boldsymbol{u}^\perp}{a_1 w \boldsymbol{g}^\top \boldsymbol{l}'_v + r_1 \boldsymbol{g}^\top \boldsymbol{v}^\perp} = 1,$$

where we have used equalities (3.5) and (3.4), and therefore $w = \frac{\boldsymbol{f}^\top \boldsymbol{k}_u}{\boldsymbol{g}^\top \boldsymbol{l}_v}$. A similar argument shows that $\hat{w} = \frac{\boldsymbol{f}^\top \hat{\boldsymbol{k}}_u}{\boldsymbol{g}^\top \hat{\boldsymbol{l}}_v}$. $\mathcal{B}$ can also compute

$$[\boldsymbol{k}_u^\top \boldsymbol{u}]_1 = [\boldsymbol{k}_u'^\top \boldsymbol{u}]_1, \qquad [\hat{\boldsymbol{k}}_u^\top \boldsymbol{u}]_1 = [\hat{\boldsymbol{k}}_u'^\top \boldsymbol{u}]_1, \qquad [\boldsymbol{l}_v^\top \boldsymbol{v}]_2 = [\boldsymbol{l}_v'^\top \boldsymbol{v}]_2 = [\hat{\boldsymbol{l}}_v^\top \boldsymbol{v}]_2.$$

Finally, choose $z_2 \leftarrow \mathbb{Z}_p$ and set

$$z_1 = w z_2, \qquad \hat{z}_1 = \hat{w} z_2,$$

completing the CRS. The CRS is then sent to adversary $\mathcal{A}$, who outputs a statement $[\boldsymbol{c}]_1$ and a proof $[\boldsymbol{d}]_2, [\pi]_1, [\hat{\pi}]_1, [\theta]_2$ such that

$$\boldsymbol{c}^\top (a_1 \boldsymbol{k}_u) - (a_1 w \boldsymbol{l}_v^\top) \boldsymbol{d} = \pi a_1 - (a_1 w)\theta,$$
$$\boldsymbol{c}^\top (a_2 \hat{\boldsymbol{k}}_u) - (a_2 \hat{w} \hat{\boldsymbol{l}}_v^\top) \boldsymbol{d} = \hat{\pi} a_2 - (a_2 \hat{w})\theta.$$

Notice that, using the equalities (3.3) and (3.6), we can rewrite these expressions in terms of the columns of $\mathbf{A}$. Indeed, these are equivalent to

$$\boldsymbol{c}^\top (\boldsymbol{k}'_u | \hat{\boldsymbol{k}}'_u | \boldsymbol{u}^\perp) \boldsymbol{a}_1 - \boldsymbol{d}^\top (w \boldsymbol{l}'_v | \hat{w} \boldsymbol{l}'_v | \boldsymbol{v}^\perp) \boldsymbol{a}_1 = (\pi, \hat{\pi}, 0) \boldsymbol{a}_1 - (w\theta, \hat{w}\theta, 0) \boldsymbol{a}_1,$$
$$\boldsymbol{c}^\top (\boldsymbol{k}'_u | \hat{\boldsymbol{k}}'_u | \boldsymbol{u}^\perp) \boldsymbol{a}_2 - \boldsymbol{d}^\top (w \boldsymbol{l}'_v | \hat{w} \boldsymbol{l}'_v | \boldsymbol{v}^\perp) \boldsymbol{a}_2 = (\pi, \hat{\pi}, 0) \boldsymbol{a}_2 - (w\theta, \hat{w}\theta, 0) \boldsymbol{a}_2.$$

We rearrange this as a solution of the $\mathcal{RL}_2$-SKerMDH problem that the reduction $\mathcal{B}$ can compute:

$$e([(\boldsymbol{c}^\top \boldsymbol{k}'_u - \pi | \boldsymbol{c}^\top \hat{\boldsymbol{k}}'_u - \hat{\pi} | \boldsymbol{c}^\top \boldsymbol{u}^\perp)]_1, [\mathbf{A}]_2) = e([(w(\boldsymbol{d}^\top \boldsymbol{l}'_v - \theta) | \hat{w}(\boldsymbol{d}^\top \boldsymbol{l}'_v - \theta) | \boldsymbol{d}^\top \boldsymbol{v}^\perp)]_2, [\mathbf{A}]_1).$$

It remains to argue that this is not the trivial solution. To do so, we look at the third component. As $\{\boldsymbol{f}, \boldsymbol{u}\}$ and $\{\boldsymbol{g}, \boldsymbol{v}\}$ are bases of $\mathbb{Z}_p^2$, we can write

$$\boldsymbol{c} = x\boldsymbol{f} + r\boldsymbol{u}, \qquad \boldsymbol{d} = y\boldsymbol{g} + s\boldsymbol{v}$$

for some $x, y, r, s \in \mathbb{Z}_p$. Since the proof provided by the adversary is false, it must be that $x \neq y$. Then, in the first equation, the third component on the left is

$$\boldsymbol{c}^\top \boldsymbol{u}^\perp = x \boldsymbol{f}^\top \boldsymbol{u}^\perp,$$

while the corresponding component on the right is

$$\boldsymbol{d}^\top \boldsymbol{v}^\perp = y\boldsymbol{g}^\top \boldsymbol{v}^\perp.$$

Since $\boldsymbol{f}^\top \boldsymbol{u}^\perp = \boldsymbol{g}^\top \boldsymbol{v}^\perp$ and $x \neq y$, these values are different. We conclude that we have found a nontrivial solution of the $\mathcal{RL}_2$-$\mathsf{SKerMDH}$ problem. □

Theorem 7. *The above scheme is composable zero-knowledge, with simulation trapdoor* $\tau = (\boldsymbol{k}_u, \hat{\boldsymbol{k}}_u, \boldsymbol{l}_v)$.

Proof. We switch to a game in which the commitments in $\mathbb{G}_2$ are perfectly hiding instead of perfectly binding, and prove that in this case the scheme has perfect zero-knowledge. The CRS simulator generates the CRS as in the honest execution of the protocol, and also outputs $\tau = (\boldsymbol{k}_u, \hat{\boldsymbol{k}}_u, \boldsymbol{l}_v)$ as the simulation trapdoor. The proof simulator chooses $\delta \leftarrow \mathbb{Z}_p$ and uses τ to produce:

$$[\boldsymbol{d}_{\mathsf{sim}}]_2 = \mathsf{Com}_{ck_2}(0; s) = s[\boldsymbol{v}]_2$$

$$[\pi_{\mathsf{sim}}]_1 = [\boldsymbol{c}^\top]_1 \boldsymbol{k}_u + \delta[z_1] \qquad [\theta_{\mathsf{sim}}]_2 = [\boldsymbol{d}_{\mathsf{sim}}^\top] \boldsymbol{l}_v + \delta[z_2]$$
$$[\hat{\pi}_{\mathsf{sim}}]_1 = [\boldsymbol{c}^\top]_1 \hat{\boldsymbol{k}}_u + \delta[\hat{z}_1]$$

We have that $\boldsymbol{d}_{\mathsf{sim}}$ is distributed as $\boldsymbol{d}$, as the commitment is perfectly hiding, and $\pi_{\mathsf{sim}}, \hat{\pi}_{\mathsf{sim}}, \theta_{\mathsf{sim}}$ are uniformly random elements conditioned on satisfying the verification equations for any fixed $\boldsymbol{c}, \boldsymbol{d}$, which is the same distribution that $\pi, \hat{\pi}, \theta$ have in an honest execution. □

3.3 Aggregated scheme

Given $\boldsymbol{x} \in \mathbb{Z}_p^n$ and two commitments $[\boldsymbol{c}]_1, [\boldsymbol{d}]_2$ to $\boldsymbol{x}$, we provide a proof of both commitments opening to the same vector $\boldsymbol{x}$. More precisely, given a group description gk and commitment keys $ck_1 = [\mathbf{F}, \mathbf{U}]_1$, and $ck_2 = [\mathbf{G}, \mathbf{V}]_2$, where $\mathbf{F} \in \mathbb{Z}_p^{m_1 \times n}, \mathbf{G} \in \mathbb{Z}_p^{m_2 \times n}$ and $\mathbf{U} \in \mathbb{Z}_p^{m_1 \times \ell_1}, \mathbf{V} \in \mathbb{Z}_p^{m_2 \times \ell_2}$, we want to prove F-knowledge in the language

$$\mathcal{L}_{gk,ck_1} = \{[\boldsymbol{c}]_1 \in \mathbb{G}_1^{m_1} \mid \exists \boldsymbol{x}, \boldsymbol{r} \text{ s. t. } [\boldsymbol{c}]_1 = \mathsf{Com}_{ck_1}(\boldsymbol{x}; \boldsymbol{r})\},$$

where $F(\boldsymbol{x}, \boldsymbol{r}) = [\boldsymbol{x}]_{1,2}$.

- $gk := (p, \mathcal{P}_1, \mathcal{P}_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$.

- $\mathcal{K}_0(gk)$: set $ck_1 = [\mathbf{F}, \mathbf{U}]_1 \leftarrow \mathcal{D}_{\mathsf{par}}$, where $\mathcal{D}_{\mathsf{par}}$ is witness sampleable, that is, there exists an efficiently sampleable distribution $\tilde{\mathcal{D}}_{\mathsf{par}}$ outputting $(\tilde{\mathbf{F}}, \tilde{\mathbf{U}})$ such that $[\tilde{\mathbf{F}}, \tilde{\mathbf{U}}]_1$ is distributed as $[\mathbf{F}, \mathbf{U}]_1$.

- $\mathcal{K}_1(gk, ck_1)$: set $ck_2 = [\mathbf{G}, \mathbf{V}]_2$, where $\mathbf{G} \leftarrow \mathbb{Z}_p^{m_2 \times n}, \mathbf{V} \leftarrow \mathbb{Z}_p^{m_2 \times \ell_2}$. Also choose $a_1, a_2 \leftarrow \mathbb{Z}_p$ and $\boldsymbol{k}_u, \hat{\boldsymbol{k}}_u \leftarrow \mathbb{Z}_p^{m_1}$. Set $\boldsymbol{l}_v, \hat{\boldsymbol{l}}_v \leftarrow \mathbb{Z}_p^{m_2}$ conditioned on

$$\boldsymbol{l}_v^\top \mathbf{V} = \hat{\boldsymbol{l}}_v^\top \mathbf{V}, \qquad \boldsymbol{k}_u^\top \mathbf{F} = w(\boldsymbol{l}_v^\top \mathbf{G}), \qquad \hat{\boldsymbol{k}}_u^\top \mathbf{F} = \hat{w}(\hat{\boldsymbol{l}}_v^\top \mathbf{G}), \tag{3.7}$$

  for some $w, \hat{w} \leftarrow \mathbb{Z}_p$. Choose $z_2 \leftarrow \mathbb{Z}_p$ and set

$$z_1 = wz_2, \qquad \hat{z}_1 = \hat{w}z_2.$$

  Algorithm $\mathcal{K}_1$ outputs the following CRS:

$$\begin{pmatrix} gk, [\mathbf{U}^\top \boldsymbol{k}_u]_1, [\mathbf{U}^\top \hat{\boldsymbol{k}}_u]_1, [a_1 w]_1, [a_2 \hat{w}]_1, [a_1 w \boldsymbol{l}_v]_1, [a_2 \hat{w} \hat{\boldsymbol{l}}_v]_1, [z_1]_1, [\hat{z}_1]_1, \\ [\mathbf{V}^\top \boldsymbol{l}_v]_2, [a_1]_2, [a_2]_2, [a_1 \boldsymbol{k}_u]_2, [a_2 \hat{\boldsymbol{k}}_u]_2, [z_2]_2 \end{pmatrix}.$$

- $\mathcal{P}(\mathsf{crs}, ([\boldsymbol{c}]_1, (\boldsymbol{x}, \boldsymbol{r})) \in \mathcal{R})$: commit to $\boldsymbol{x}$ in $\mathbb{G}_2$ as $[\boldsymbol{d}]_2$. Choose $\delta \leftarrow \mathbb{Z}_p$ and output $[\boldsymbol{d}]_2$ and

$$\begin{aligned} [\pi]_1 &= [\boldsymbol{r}^\top \mathbf{U}^\top \boldsymbol{k}_u + \delta z_1]_1, & [\theta]_2 &= [\boldsymbol{s}^\top \mathbf{V}^\top \boldsymbol{l}_v + \delta z_2]_2, \\ [\hat{\pi}]_1 &= [\boldsymbol{r}^\top \hat{\mathbf{U}}^\top \boldsymbol{k}_u + \delta \hat{z}_1]_1, \end{aligned}$$

- $\mathcal{V}(\mathsf{crs}, [\boldsymbol{c}]_1, ([\boldsymbol{d}, \theta]_2, [\pi, , \hat{\pi}]_1))$: The algorithm outputs 1 iff the following equations hold:

$$\begin{aligned} e\left([\boldsymbol{c}^\top]_1, [a_1 \boldsymbol{k}_u]_2\right) - e([a_1 w \boldsymbol{l}_v^\top]_1, [\boldsymbol{d}]_2) &\stackrel{?}{=} e([\pi]_1, [a_1]_2) - e([a_1 w]_1, [\theta]_2), \\ e\left([\boldsymbol{c}^\top]_1, [a_2 \hat{\boldsymbol{k}}_u]_2\right) - e([a_2 \hat{w} \hat{\boldsymbol{l}}_v^\top]_1, [\boldsymbol{d}]_2) &\stackrel{?}{=} e([\hat{\pi}]_1, [a_2]_2) - e([a_2 \hat{w}]_1, [\theta]_2). \end{aligned}$$

Completeness. It is easy to check that, if the prover is honest,

$$\begin{aligned} &\boldsymbol{c}^\top (a_1 \boldsymbol{k}_u) - (a_1 w \boldsymbol{l}_v^\top) \boldsymbol{d} = (\boldsymbol{x}^\top \mathbf{F}^\top + \boldsymbol{r}^\top \mathbf{U}^\top)(a_1 \boldsymbol{k}_u) - (a_1 w \boldsymbol{l}_v^\top)(\mathbf{G}\boldsymbol{x} + \mathbf{V}\boldsymbol{s}) = \\ &= a_1 \boldsymbol{x}^\top \mathbf{F}^\top \boldsymbol{k}_u - a_1 (w \boldsymbol{l}_v^\top \mathbf{G}) \boldsymbol{x} + a_1 \boldsymbol{r}^\top \mathbf{U}^\top \boldsymbol{k}_u - a_1 w \boldsymbol{l}_v^\top \mathbf{V} \boldsymbol{s} = \pi a_1 - a_1 w \theta. \end{aligned}$$

We have used that $\boldsymbol{k}_u^\top \mathbf{F} = w(\boldsymbol{l}_v^\top \mathbf{G})$. The second equation is completely analogous.

Note on dimensions. For this scheme to work and be secure, we require some relations between the dimensions of the different elements involved.

(1) We want our commitments to be perfectly binding to be able to open the commitments in the source groups, so we require that $m_i \geq n + \ell_i$, for $i = 1, 2$.

(2) To be able to find $\boldsymbol{l}_v, \hat{\boldsymbol{l}}_v$ verifying the equations (3.7), we need to solve the linear system

$$\begin{pmatrix} \mathbf{G}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^\top \\ \mathbf{V}^\top & -\mathbf{V} \end{pmatrix} \begin{pmatrix} \boldsymbol{l}_v \\ \hat{\boldsymbol{l}}_v \end{pmatrix} = \begin{pmatrix} \mathbf{F}^\top \boldsymbol{k}_u \\ \mathbf{F}^\top \hat{\boldsymbol{k}}_u \\ \mathbf{0} \end{pmatrix}.$$

Since $\mathbf{F}$ is only known in $\mathbb{G}_1$, the system cannot be fully solved over $\mathbb{Z}_p$. However, we do not need the full solution over $\mathbb{Z}_p$, as only the projection $\mathbf{V}^\top \boldsymbol{l}_v$ needs to be given in $\mathbb{G}_2$, while the full $\boldsymbol{l}_v$ is necessary in $\mathbb{G}_1$. Thus we proceed as follows: we start by sampling $\boldsymbol{t} \leftarrow \mathbb{Z}_p^{\ell_2}$ and setting $\mathbf{V}^\top \boldsymbol{l}_v = \mathbf{V}^\top \hat{\boldsymbol{l}}_v = \boldsymbol{t}$. Then we consider the system

$$\begin{pmatrix} \mathbf{G}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^\top \\ \mathbf{V}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{pmatrix} \begin{pmatrix} \boldsymbol{l}_v \\ \hat{\boldsymbol{l}}_v \end{pmatrix} = \begin{pmatrix} \mathbf{F}^\top \boldsymbol{k}_u \\ \mathbf{F}^\top \hat{\boldsymbol{k}}_u \\ \boldsymbol{t} \\ \boldsymbol{t} \end{pmatrix}.$$

The matrix is known over $\mathbb{Z}_p$ and the right hand side is known over $\mathbb{G}_1$ (since $\mathbf{F}$ is known over $\mathbb{G}_1$ and the rest is known over $\mathbb{Z}_p$), so the system can be solved over $\mathbb{G}_1$ using Gaussian elimination. The system has solutions if $2m_2 \geq 2n + 2\ell_2$, which is implied by condition (1) above.

(3) In the proof of the zero-knowledge property, we want to be able to switch the commitment in $\mathbb{G}_2$ to perfectly hiding, so we need to ensure that it has enough randomness. Thus $\ell_2 \geq n$.

(4) Consider the matrices $(\mathbf{F}|\mathbf{U})$ and $(\mathbf{G}|\mathbf{V})$. These are of size $m_i \times (n + \ell_i)$, for $i = 1, 2$, respectively. In the soundness reduction we will be interested in finding nonzero vectors $\boldsymbol{u}^\perp, \boldsymbol{v}^\perp$ such that $\boldsymbol{w}^\top \boldsymbol{u}^\perp = 0$ for any vector $\boldsymbol{w}$ outside of the span of the columns of $\mathbf{F}$, and the same for $\boldsymbol{v}^\perp$ and $\mathbf{G}$. Additionally, we will require that

$$\mathbf{F}^\top \boldsymbol{u}^\perp = \mathbf{G}^\top \boldsymbol{v}^\perp.$$

As we have already established that $m_i \geq n + \ell_i$, we might need to add more columns to the matrices $(\mathbf{F}|\mathbf{U})$ and $(\mathbf{G}|\mathbf{V})$ so that they form bases of $\mathbb{Z}_p^{m_i}$, so let $\overline{\mathbf{U}}, \overline{\mathbf{V}} \in \mathbb{Z}_p^{m_i \times (m_i - n)}$ be the augmented matrices such that $(\mathbf{F}|\overline{\mathbf{U}})$ and $(\mathbf{G}|\overline{\mathbf{V}})$ are bases of $\mathbb{Z}_p^{m_i}$ for $i = 1, 2$, respectively. Then the vectors $\boldsymbol{u}^\perp, \boldsymbol{v}^\perp$ are given by the nontrivial solutions of the linear system

$$\begin{pmatrix} \overline{\mathbf{U}}^\top & \mathbf{0} \\ \mathbf{0} & \overline{\mathbf{V}}^\top \\ \mathbf{F}^\top & -\mathbf{G}^\top \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^\perp \\ \boldsymbol{v}^\perp \end{pmatrix} = \mathbf{0}.$$

This matrix is of size $(m_1 + m_2 - n) \times (m_1 + m_2)$, and therefore it has nontrivial solutions.

F-extractor. We now define the algorithm that, given the extraction key $xk = (\mathbf{F}, \mathbf{G}, \mathbf{U}, \mathbf{V})$, outputs a function of the witness, in this case $F(\boldsymbol{x}, \boldsymbol{r}) = [\boldsymbol{x}]_{1,2}$.

- $\mathsf{Ext}_{xk}([\boldsymbol{c}]_1, [\boldsymbol{d}]_2)$: as above, consider $\overline{\mathbf{U}}, \overline{\mathbf{V}}$ so that $(\mathbf{F}|\overline{\mathbf{U}})$ and $(\mathbf{G}|\overline{\mathbf{V}})$ are bases of $\mathbb{Z}_p^{m_i}$ for $i = 1, 2$, respectively. Knowing $\mathbf{F}, \overline{\mathbf{U}}$, we can find a matrix $\mathbf{U}^\perp \in \mathbb{Z}_p^{m_1 \times n}$ such that $\overline{\mathbf{U}}^\top \mathbf{U}^\perp = \mathbf{0}$ and $\mathbf{F}^\top \mathbf{U}^\perp = \mathbf{I}$, and compute $[\boldsymbol{c}^\top]_1 \mathbf{U}^\perp = [\boldsymbol{x}]_1$. Similarly, we obtain $[\boldsymbol{x}]_2$ from $[\boldsymbol{d}]_2$, using $\mathbf{G}, \overline{\mathbf{V}}$.

Theorem 8. *The above proof system is computationally F-knowledge sound under the $\mathcal{RL}_2$-SKerMDH assumption (6). More precisely, there exists an adversary $\mathcal{B}$ against the $\mathcal{RL}_2$-SKerMDH problem such that for any PPT adversary $\mathcal{A}$, we have that*

$$\mathsf{Adv}_{F-\mathsf{KnowledgeSoundness}}\mathcal{A}(1^\lambda) \leq \mathsf{Adv}_{\mathcal{RL}_2\text{-SKerMDH}}\mathcal{B}(1^\lambda).$$

Proof. Assume that there is an adversary $\mathcal{A}$ against the soundness of the scheme ($\mathcal{A}$ is able to produce a statement and and an accepting proof such that

$$\mathsf{Ext}_{xk}([\boldsymbol{c}]_1, [\boldsymbol{d}]_2) = ([\boldsymbol{x}]_1, [\boldsymbol{y}]_2),$$

and $\boldsymbol{x} \neq \boldsymbol{y}$). We use it to build an adversary $\mathcal{B}$ against the $\mathcal{RL}_2$-SKerMDH problem. $\mathcal{B}$ receives the challenge matrix

$$[\mathbf{A}]_{1,2} = [\boldsymbol{a}_1 | \boldsymbol{a}_2]_{1,2} = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \\ r_1 & r_2 \end{bmatrix}_{1,2},$$

and builds the environment for $\mathcal{A}$ as follows. We sample $\mathbf{G} \leftarrow \mathbb{Z}_p^{m_2 \times n}, \mathbf{V} \leftarrow \mathbb{Z}_p^{m_2 \times \ell_2}$, and let $\overline{\mathbf{V}}$ be as in (4) above. We choose $w, \hat{w} \leftarrow \mathbb{Z}_p$ and $\boldsymbol{l}'_v \leftarrow \mathbb{Z}_p^{m_2}$ Let $\boldsymbol{v}^\perp \in \mathbb{Z}_p^{m_2}$ such that $\overline{\mathbf{V}}^\top \boldsymbol{v}^\perp = 0$. Implicitly set

$$\boldsymbol{l}_v = \boldsymbol{l}'_v + (a_1 w)^{-1} r_1 \boldsymbol{v}^\perp, \qquad \hat{\boldsymbol{l}}_v = \boldsymbol{l}'_v + (a_2 \hat{w})^{-1} r_2 \boldsymbol{v}^\perp.$$

Observe that this implies that

$$a_1 w \boldsymbol{l}_v = a_1 w \boldsymbol{l}'_v + r_1 \boldsymbol{v}^\perp, \qquad a_2 \hat{w} \hat{\boldsymbol{l}}_v = a_2 \hat{w} \boldsymbol{l}'_v + r_2 \boldsymbol{v}^\perp, \tag{3.8}$$

which we can compute over $\mathbb{G}_1$. For the other side, we sample $(\mathbf{F}, \mathbf{U}) \leftarrow \tilde{\mathcal{D}}_{\mathsf{par}}$ and define $\overline{\mathbf{U}}$ as in (4) above. We also sample $\boldsymbol{k}'_u, \hat{\boldsymbol{k}}'_u \leftarrow \mathbb{Z}_p^{m_1}$ conditioned on

$$\boldsymbol{k}'^\top_u \mathbf{F} = w(\boldsymbol{l}'^\top_v \mathbf{G}), \qquad \hat{\boldsymbol{k}}'^\top_u \mathbf{F} = \hat{w}(\hat{\boldsymbol{l}}'^\top_v \mathbf{G}). \tag{3.9}$$

Let $\boldsymbol{u}^\perp \in \mathbb{Z}_p^{m_1}$ such that $\overline{\mathbf{U}}^\top \boldsymbol{u}^\perp = 0$ and

$$\mathbf{F}^\top \boldsymbol{u}^\perp = \mathbf{G}^\top \boldsymbol{v}^\perp. \tag{3.10}$$

We implicitly define

$$\boldsymbol{k}_u = \boldsymbol{k}'_u + a_1^{-1} r_1 \boldsymbol{u}^\perp, \qquad \hat{\boldsymbol{k}}_u = \hat{\boldsymbol{k}}'_u + a_2^{-1} r_2 \boldsymbol{u}^\perp.$$

which means that

$$a_1 \boldsymbol{k}_u = a_1 \boldsymbol{k}'_u + r_1 \boldsymbol{u}^\perp, \qquad a_2 \hat{\boldsymbol{k}}_u = a_2 \hat{\boldsymbol{k}}'_u + r_2 \boldsymbol{u}^\perp. \tag{3.11}$$

Note that, by construction,

$$a_1 w \mathbf{G}^\top \boldsymbol{l}_v = a_1 w \mathbf{G}^\top \boldsymbol{l}'_v + r_1 \mathbf{G}^\top \boldsymbol{v}^\perp = a_1 \mathbf{F}^\top \boldsymbol{k}'_u + r_1 \mathbf{F}^\top \boldsymbol{u}^\perp = a_1 \mathbf{F}^\top \boldsymbol{k}_u$$

where we have used equalities (3.9) and (3.10), and therefore $\mathbf{F}^\top \boldsymbol{k}_u = w(\mathbf{G}^\top \boldsymbol{l}_v)$ A similar argument shows that $\mathbf{F}^\top \hat{\boldsymbol{k}}_u = \hat{w}(\mathbf{G}^\top \hat{\boldsymbol{l}}_v)$. We can also compute

$$[\boldsymbol{k}_u^\top \mathbf{U}]_1 = [\boldsymbol{k}'^\top_u \mathbf{U}]_1, \qquad [\hat{\boldsymbol{k}}_u^\top \mathbf{U}]_1 = [\hat{\boldsymbol{k}}'^\top_u \mathbf{U}]_1, \qquad [\boldsymbol{l}_v^\top \mathbf{V}]_2 = [\boldsymbol{l}'^\top_v \mathbf{V}]_2 = [\hat{\boldsymbol{l}}_v^\top \mathbf{V}]_2.$$

Finally, choose $z_2 \leftarrow \mathbb{Z}_p$ and set

$$z_1 = w z_2, \qquad \hat{z}_1 = \hat{w} z_2,$$

completing the CRS. The CRS is then sent to adversary $\mathcal{A}$, who outputs a statement $[\boldsymbol{c}]_1, [\boldsymbol{d}]_2$ and a proof $[\pi]_1, [\hat{\pi}]_1, [\theta]_2$ such that

$$\boldsymbol{c}^\top(a_1\boldsymbol{k}_u) - (a_1 w\boldsymbol{l}_v^\top)\boldsymbol{d} = \pi a_1 - (a_1 w)\theta,$$
$$\boldsymbol{c}^\top(a_2\hat{\boldsymbol{k}}_u) - (a_2\hat{w}\hat{\boldsymbol{l}}_v^\top)\boldsymbol{d} = \hat{\pi} a_2 - (a_2\hat{w})\theta.$$

Notice that, using equalities (3.11) and (3.8), we can rewrite these expressions in terms of the columns of $\mathbf{A}$. Indeed, these are equivalent to

$$\boldsymbol{c}^\top(\boldsymbol{k}'_u|\hat{\boldsymbol{k}}'_u|\boldsymbol{u}^\perp)\boldsymbol{a}_1 - \boldsymbol{d}^\top(w\boldsymbol{l}'_v|\hat{w}\boldsymbol{l}'_v|\boldsymbol{v}^\perp)\boldsymbol{a}_1 = (\pi, \hat{\pi}, 0)\boldsymbol{a}_1 - (w\theta, \hat{w}\theta, 0)\boldsymbol{a}_1,$$
$$\boldsymbol{c}^\top(\boldsymbol{k}'_u|\hat{\boldsymbol{k}}'_u|\boldsymbol{u}^\perp)\boldsymbol{a}_2 - \boldsymbol{d}^\top(w\boldsymbol{l}'_v|\hat{w}\boldsymbol{l}'_v|\boldsymbol{v}^\perp)\boldsymbol{a}_2 = (\pi, \hat{\pi}, 0)\boldsymbol{a}_2 - (w\theta, \hat{w}\theta, 0)\boldsymbol{a}_2,$$

We rearrange this as a solution of the $\mathcal{RL}_2$-SKerMDH problem that the reduction can compute:

$$e([(\boldsymbol{c}^\top\boldsymbol{k}'_u - \pi|\boldsymbol{c}^\top\hat{\boldsymbol{k}}'_u - \hat{\pi}|\boldsymbol{c}^\top\boldsymbol{u}^\perp)]_1, [\mathbf{A}]_2) = e([(w(\boldsymbol{d}^\top\boldsymbol{l}'_v - \theta)|\hat{w}(\boldsymbol{d}^\top\boldsymbol{l}'_v - \theta)|\boldsymbol{d}^\top\boldsymbol{v}^\perp)]_2, [\mathbf{A}]_1).$$

It remains to argue that this is not the trivial solution. To do so, we look at the third component. As the columns of $(\mathbf{F}|\overline{\mathbf{U}})$ and $(\mathbf{G}|\overline{\mathbf{V}})$ are bases of $\mathbb{Z}_p^{m_i}$ for $i = 1, 2$, respectively, we can write

$$\boldsymbol{c} = \mathbf{F}\boldsymbol{x} + \overline{\mathbf{U}}\boldsymbol{r}, \qquad \boldsymbol{d} = \mathbf{G}\boldsymbol{y} + \overline{\mathbf{V}}\boldsymbol{s},$$

for some $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}_p^n, \boldsymbol{r}, \boldsymbol{s} \in \mathbb{Z}_p^\ell$. Since the proof provided by the adversary is false, it must be that $\boldsymbol{x} \neq \boldsymbol{y}$. Then, in the first equation, the third component on the left is

$$\boldsymbol{c}^\top\boldsymbol{u}^\perp = \boldsymbol{x}^\top\mathbf{F}^\top\boldsymbol{u}^\perp$$

, while the corresponding component on the right is

$$\boldsymbol{d}^\top\boldsymbol{v}^\perp = \boldsymbol{y}^\top\mathbf{G}^\top\boldsymbol{v}^\perp$$

. Since $\mathbf{F}^\top\boldsymbol{u}^\perp = \mathbf{G}^\top\boldsymbol{v}^\perp$ and $\boldsymbol{x} \neq \boldsymbol{y}$, these values are different. We conclude that we have found a nontrivial solution of the $\mathcal{RL}_2$-SKerMDH problem. □

Theorem 9. *The above proof system is composable zero-knowledge, with simulation trapdoor* $\tau = (\boldsymbol{k}_u, \hat{\boldsymbol{k}}_u, \boldsymbol{l}_v)$.

The proof is completely analogous to the proof of Theorem 7.

3.4 Optimality of our constructions

We argue that our constructions are optimal in terms of proof size, at least based on this general strategy of commit-and-prove schemes, and where the prover is limited to linear algebraic operations on the group elements, and verification is a pairing equation. To the best of our knowledge, this is the approach that is always taken in the literature. We prove optimality by arguing that any such proof formed of two elements (plus the commitments) is vulnerable to an attack.

We now consider any proof in which we have two commitments $[\boldsymbol{c}]_1$ and $[\boldsymbol{d}]_2$ to the values x and y, respectively, and we have a a two-element proof $[\pi]_1, [\theta]_2$ of same opening, that is, $x = y$. We consider a CRS formed of elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, and we assume that each side of the CRS is closed under linear combination. We can do this without loss of generality, since given the CRS it is easy to compute linear combinations of its elements.

Then the general verification equation of such a proof looks like this:

$$e([\boldsymbol{c}^\top]_1, [\boldsymbol{k}_1]_2) + e([\boldsymbol{k}_2^\top]_1, [\boldsymbol{d}]_2) + e([\pi]_1, [k_3]_2) + e([k_4]_1, [\theta]_2) = [0]_T, \quad (3.12)$$

where $[\boldsymbol{k}_1, k_3]_2, [\boldsymbol{k}_2, k_4]_1$ are elements (some of them vectors of elements) of the CRS. We note two omissions from this general equation: there is no affine term and there are no "quadratic" terms, i.e., terms in $\boldsymbol{c}^\top \boldsymbol{d}, \pi \boldsymbol{d}, \boldsymbol{c}\theta$ or $\pi\theta$. This is because the linear terms (those in equation (3.12)) force π and θ to be linear in the witness, and so the terms above are quadratic. The quadratic condition causes the appearance of terms with coefficient xy, which must cancelled out with other quadratic terms of the same coefficient. We note that, unlike in the linear part, this check does not make a distinction when $x = y$ or $x \neq y$, so we conclude that these quadratic terms do not contribute to achieving soundness. The intuition behind this is that we are proving membership in a linear space, and non-linear operations take us out of the space.

This leaves us with the equation (3.12) above. We now observe a very simple attack on any scheme with a verification equation like this. We set

$$\begin{aligned} [\boldsymbol{c}]_1 &= \boldsymbol{\alpha}[k_4]_1, & [\boldsymbol{d}]_2 &= \boldsymbol{\beta}[k_3]_2, \\ [\pi]_1 &= -\boldsymbol{\beta}^\top [\boldsymbol{k}_2]_1, & [\theta]_2 &= -\boldsymbol{\alpha}^\top [\boldsymbol{k}_1]_2, \end{aligned}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta} \leftarrow \mathbb{Z}_p^2$. It is trivial to verify that the first term in the equation cancels out with the fourth and the second with the third, and with overwhelming probability the openings of $[\boldsymbol{c}]_1$ and $[\boldsymbol{d}]_2$ do not match. Intuitively, this attack works because of the two-sided nature of the proof: the elements that are given in the CRS to ensure verifiability in one side are used to fool the other. Indeed,

in an honest execution the first term is expected to cancel out with the third, and the second with the fourth, while in this attack the pairs are jumbled.

One could also consider one-sided two-element proofs, i.e., of the form $[\pi_1, \pi_2]_1$ or $[\theta_1, \theta_2]_2$, but these can be handled in a very similar way. For example, in the first case, the general verification equation would be

$$e([\boldsymbol{c}^\top]_1, [\boldsymbol{k}_1]_2) + e([\boldsymbol{k}_2^\top]_1, [\boldsymbol{d}]_2) + e([\pi_1]_1, [k_3]_2) + e([\pi_2]_1, [k_4]_2) = [0]_T, \quad (3.13)$$

and the attack would consist of setting

$$\begin{aligned} [\boldsymbol{c}]_1 &= \alpha[\boldsymbol{k}_2]_1, & [\boldsymbol{d}]_2 &= \boldsymbol{\beta}(r[k_3]_2 + s[k_4]_2) - \alpha[\boldsymbol{k}_1]_2, \\ [\pi_1]_1 &= -r\boldsymbol{\beta}^\top[\boldsymbol{k}_2]_1, & [\pi_2]_1 &= -s\boldsymbol{\beta}^\top[\boldsymbol{k}_2]_1, \end{aligned}$$

for $\boldsymbol{\beta} \leftarrow \mathbb{Z}_p^2, \alpha, r, s \leftarrow \mathbb{Z}_p$. Thus we conclude that, with this approach, there is no possible proof of same opening of commitments in different groups which consists of less than three group elements, making our constructions optimal.

Chapter 4

Verifiable computation

This chapter is based on the paper 'Circuit satisfiability arguments from two-tiered commitments', which is a joint work with Carla Ràfols.

4.1 Introduction

In the last few years, we have seen a plethora of results advancing the state of the art in succinct non-interactive arguments (SNARGs) for proving circuit satisfiability or membership in other NP-complete languages. Succinct arguments can be used to construct proofs of correct computation with very efficient verification. Recent advances cover both the case of a deterministic, polynomial time computation in which the input of the circuit is public or the more demanding setting in which part or all of the input is secret.

There are a few different approaches to constructing these arguments with different security properties and tradeoffs, and based on different assumptions. Elliptic curves with a bilinear map, or pairing, are particularly amenable for designing SNARGs with efficient public verification. In particular, the most efficient arguments currently are pairing-based zero-knowledge succinct non-interactive arguments of knowledge (zkSNARKs) [49, 79, 90, 92, 126], in which the communication is a constant number of group elements (just three in the most efficient case [92]), regardless of the size of the circuit.

Among the most important downsides of zkSNARKs is the fact that they rely on knowledge of exponent assumptions or the generic or algebraic group mod-

els, which are very strong assumptions. In particular, these are non-falsifiable assumptions [133], which means that one cannot efficiently check if an adversary is breaking the assumption. Additionally, SNARKs require a structured reference string (SRS) that is also circuit-dependent, which must be generated by a trusted third party or by means of multi-party computation. To avoid this, recent research has focused on SNARKs with updateable SRS [39, 70, 93, 128], which means that the SRS can be efficiently non-interactively updated by any party, and the resulting argument is sound as long as a single update is honest. Besides being updateable, these works also construct a universal SRS, i.e. a SRS of size linear in some integer N that works for all circuits of size at most N.

Ideally, from a security point of view, one would like to avoid the reliance of non-falsifiable assumptions to construct efficient non-interactive arguments. Achieving these goals combined with practical efficiency is a difficult task. Most straightforward solutions (e.g. [97]) are of size linear in the circuit. One exception (in the pairing-based setting, which is the focus of this work) are [111, 112], that build compact NIZKs from standard assumptions that are linear in the witness for NC1 circuits (and linear in the size of the circuit for general circuits), and have only an additive overhead in the security parameter.[1] Another exception is the scheme for delegation of computation of Kalai et al. [110], that is proven secure under some non-standard (but falsifiable) assumptions in bilinear maps, and that has a proof size of $O(d \log W)$ group elements, where d is the depth and W is the width of the circuit. Both of these constructions are efficient asymptotically but have rather large constants.

On the other hand, González and Ràfols [86] presented a proof, based on falsifiable assumptions and using a circuit-dependent SRS, in which the proof size depends not on the size of the circuit, but the multiplicative depth. More precisely, they obtained a proof of size $O(d)$ group elements (with small constants), with linear sized SRS under a non-standard W-assumption, although with a circuit dependent SRS. Their proof exploits the fact that the input of the circuit is known. It is possible to turn it into a NIZK argument of circuit satisfiability of size $O(n + d)$ group elements, where n is the size of the secret input, if the circuit is boolean. This uses the fact that ElGamal encryptions to the input bits are trapdoor-extractable.

Another line of research, derived from [24], and followed by Bulletproofs [28], builds interactive arguments that rely on very weak assumptions like the discrete

[1] In all the schemes described before, the overhead is multiplicative in the security parameter. This is hidden in the fact that we usually count the proof size in number of group elements, and each group element is linear in the security parameter.

logarithm (DLog) assumption, and that can also be instantiated in DLog groups without pairings. These proofs require sending $O(\log M)$ elements, where M is the number of multiplicative gates in the circuit. Moreover, these proofs have a transparent setup, which means that there is no need to trust a third party to generate the CRS. They require $O(\log M)$ rounds of interaction, and they can be made non-interactive with the Fiat–Shamir transformation. A recent paper [50] made use of a pairing to speed up the verification step, bringing it down from $O(M)$ to $O(\log M)$ at the cost of introducing a structured (but updateable) setup but without modifying the assumptions.

This work avoids the use of non-falsifiable assumptions by taking an hybrid approach that exploits the advances in constructing non-interactive arguments under standard assumptions with the power of interactive proofs and the recursive inner product argument of [24, 28], with the end goal of obtaining proofs with less rounds of interactions and reduced communication complexity with respect to the state of the art.

Our results. Let C be a circuit with public input and M multiplication gates. We slice the circuit into multiplicative levels $1, \ldots, d$. The contributions of this chapter are the following:

- We construct an argument with communication complexity $O(\log d)$ group elements and $O(\log d)$ rounds of interaction under falsifiable assumptions. In particular, for NC1 circuits, this improves the state of the art under falsifiable assumptions (communication and rounds $O(\log M)$ [28, 50]) by an order of magnitude. Our starting point is the construction of González and Ràfols [86]. We add a layer of interactivity, by using the proofs for inner pairing products of Bünz et al. [29], which use two-tier commitment schemes as in [91]. For the soundness proof, we inherit the reliance on some (falsifiable) W-assumptions in bilinear groups from [86] and we require an additional assumption, which is still falsifiable and circuit-independent.[2]

- The construction of [86] requires a SRS (generated by a trusted third party) that depends on the circuit being used. We present an alternative proof that uses a universal and updateable SRS. This construction is in the usual preprocessing model as in similar works [39, 70, 78, 128], where the

[2] An important point, discussed in [86] is that it is is not difficult to construct succinct arguments for correct circuit evaluation based on non-falsifiable assumptions. As soundness itself is falsifiable (it is possible to decide if the adversary has broken soundness because the input is public), the tautological assumption that "the scheme is sound" is falsifiable. The challenge is to rely on falsifiable *and* circuit-independent assumptions.

SRS is universal but some deterministic circuit-dependent preprocesssing of the SRS is allowed (performed by some party called encoder, or indexer, but that can be done by any party from the SRS and the description of the circuit). Our construction works in two steps: first we give a 3-round Sigma protocol with communication complexity $O(d)$, and we analyze its security under falsifiable and circuit dependent assumptions. Then, we use the inner pairing product argument again to achieve $O(\log d)$ communication complexity.

- The verifier complexity for the construction based on [86] is $O(d)$. On the other hand, a naive approach for the last construction would result in a verifier linear in M. The key issue is is how to sample a random vector in a large space in a verifiable way. We resort to the same techniques as in recent updateable and universal zkSNARKs [39,128]. These works reduce the verification complexity to constant.[3] In our case, these techniques allow to reduce the verification complexity to $O(d)$, with some limitations (e.g. if we use the techniques of [39] the matrix describing the linear constraints has to be sparse).

- The interactive constructions can be transformed into NIZK proofs for circuit satisfiability with an overhead of $O(\log n)$ group elements, where n is the size of the secret part of the input. This works for both arithmetic and boolean circuits, unlike in [86], where the overhead is $O(n)$ and only works for boolean circuits.

In Table 4.1, we compare our constructions with known alternatives based on falsifiable assumptions.

Our techniques. Our starting point is the work of [86]. On a high level, their construction of correct circuit evaluation, with a communication cost of $O(d)$, works as follows. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a pairing. Let $\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i$ be a correct assignment to the left, right and output wires, respectively, at multiplicative level i of the circuit, for $i = 1, \ldots, d$. For each level of the circuit, they need to ensure that:

(a) The output is consistent with the input wires, that is,

$$\boldsymbol{a}_i \circ \boldsymbol{b}_i = \boldsymbol{c}_i,$$

[3] We note that this is only possible in the preprocessing model, where the verifier does not need to read the circuit description but only a short encoding of it.

Scheme	Prover	Proof size	Verifier	RS	Univ.	Rounds	RS Size
[24], [28]	$O(M)$	$O(\log M)$	$O(s+M)$	URS	✓	$O(\log M)$	$O(M)$
[91]	$O(N)$	$O(N^{1/3})$	$O(s+N)$	URS	✓	7	$O(M)$
[50]	$O(M)$	$O(\log M)$	$O(s+\log M)$	SRS	✓	$O(\log M)$	$O(M)$
[86]	$O(M \log M)$	$O(d)$	$O(s+d)$,	SRS	✗	1	O(M)
Sec. 4.4	$O(M \log M)$	$O(\log d)$	$O(s+d)$	SRS	✗	$O(\log d)$	$O(M)$
Sec. 4.5	$O(M \log M)$	$O(d)$	$O(s+d^2)$	SRS	✓	3	$O(W)$
Sec. 4.5	$O(M \log M)$	$O(\log d)$	$O(s+d^2)$	SRS	✓	$O(\log d)$	$O(W)$

Table 4.1: Efficient arguments for proving correct circuit evaluation. M is the number of multiplication gates, s is the size of the public input, d is the multiplicative depth of the circuit and W its width (i.e. the maximum number of gates in a single multiplicative depth). URS and SRS stand for unstructured and structured reference strings, respectively. The 'Univ.' column reflects whether the RS is universal or not.

(b) The left and right wires are consistent with the outputs of previous levels, that is,

$$\boldsymbol{a}_i = \sum_{j<i} \boldsymbol{m}_{ij} \cdot \boldsymbol{c}_j, \qquad \boldsymbol{b}_i = \sum_{j<i} \tilde{\boldsymbol{m}}_{ij} \cdot \boldsymbol{c}_j,$$

for some coefficients $\boldsymbol{m}_{ij}, \tilde{\boldsymbol{m}}_{ij}$ are part of the description of the circuit and specify the correct wiring and correct addition gate evaluation.

The approach is commit to $\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i$ with shrinking commitments $[L]_1, [R]_2, [O]_1$, respectively, and succinctly prove some relations between these to prove (a) and (b) in a level-wise aggregated manner, by verifying some pairing product equations at each level. The way to prove (a) and (b) is not new, as it draws, respectively, on techniques from zkSNARKs [79] and QA-NIZK arguments of membership in linear spaces [108, 115]. The contribution is the reduction to a falsifiable assumption.

By proving each level this way, [86] proves a correct and consistent assignment for the whole circuit, in a process that they call 'knowledge transfer'. The idea is that the knowledge of the input is transferred to lower levels of the circuit,

Scheme	Assumption
[24], [28]	DLog
[91]	DLog, KerMDDH
[50]	DLog
[86]	W-RSDH (16), $\mathcal{LG}_{\mathcal{R},2}$-SMDDH (4)
Sect. 4.4	W-DLog, W-RSDH (16) , $\mathcal{LG}_{\mathcal{R},2}$-SMDDH (4)
Sect. 4.5	W-DLog, W-RSDH (16), Assumption 17

Table 4.2: W is the width of the circuit, and $\mathcal{R}$ is an interpolation set with $\#\mathcal{R} = W$.

although at lower levels the commitments are never opened or extracted.[4] Obviously, this requires sending $O(d)$ group elements. To avoid this, we add a layer of interactivity on top of their protocol, further compressing the $[L]_1, [R]_2, [O]_1$, with a second shrinking commitment scheme, into $[CL, CR, CO]_T$, respectively. This is inspired by the idea of two-tiered commitments in [91]. The challenge now becomes ensuring that proving relations in the now twice-committed values guarantees the satisfiability of the original relations.

This additional layer is based on the interactive proof techniques of [24] for circuit satisfiability and the extension to inner product of pairing equations [29]. In [24], the authors provide a Sigma protocol for proving knowledge of a solution of an arithmetic circuit with M multiplication gates (i.e. a system of $O(M)$ linear and quadratic equations) with communication $O(\log M)$. The only assumption required is the binding property of the commitment scheme used. Our setting is not exactly the same as theirs, as we want to prove quadratic and linear relations on $[L]_1, [R]_2, [O]_1$, which are elements of $\mathbb{G}_1, \mathbb{G}_2$, as opposed to elements of $\mathbb{Z}_p$. Nevertheless, this is exactly the setting described in [29]. It is simple to translate their constructions to the new setting, using the commitments from [3]. Since this new layer is extractable, in the security proof we can recover the $[L]_1, [R]_2, [O]_1$, and rely on the security proofs of [86].

When proving (b), [86] makes use of a QA-NIZK proof that takes care of the linear relations very efficiently, but has the downside of requiring a circuit-

[4] Although QA-NIZK arguments for membership in a linear space generated by some matrix $\mathbf{M}$, as presented in [115], are proven sound under standard assumptions, this does not cover the case where $\mathbf{M}$ is a full rank matrix, which is the case needed to prove (b). In this case, we only know how to prove soundness under non-falsifiable assumptions, e.g. [33, 86] prove a relaxed notion of soundness that guarantees the knowledge transfer property under a decisional assumption.

dependent SRS. Thus, to achieve a universal SRS, it is not enough to add a new layer on top of [86], but we need to replace their whole proof for linear equations with a completely different one.

For this, we get loosely inspired by recent updateable and universal preprocessing SNARKs. On a high level, our strategy is as follows. Suppose that we want to prove all the linear relations that the inputs of level i have with outputs of previous levels. We can think of the solution of the equations as a vector that is orthogonal to the subspace $\mathbf{Y}$ spanned by the vectors of coefficients of the equations. More precisely, if $\boldsymbol{a}_i, \boldsymbol{b}_i$ is a valid assignment to input wires at level i wires as a function of the output wires $\boldsymbol{c}_{i-1}$ up to level $i-1$, then $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}_{i-1})$ is orthogonal to the matrix

$$\mathbf{Y}_i = \begin{pmatrix} -\mathbf{I} & \mathbf{0} & \mathbf{M}_i \\ \mathbf{0} & -\mathbf{I} & \tilde{\mathbf{M}}_i \end{pmatrix}.$$

We use a three-round Sigma protocol in which the prover commits to the solution, and then is challenged to prove that this solution is orthogonal to a vector $\boldsymbol{y}$ in the row space of $\mathbf{Y}$ sampled by the verifier. For this, we characterize the property that 'two commitments with respect to commitment key ck open to orthogonal vectors' as a problem of divisibility of polynomials. In our case ck is a vector of Lagrangian basis polynomials associated to an interpolation set $\mathcal{R}$ evaluated in some secret point x (secret in the sense that it is only known in the source groups). For this, we can resort to the univariate sumcheck of [13], that works when $\mathcal{R}$ is a multiplicative subgroup of a finite field. In fact, we use a simpler characterization of this property that does not use that $\mathcal{R}$ is a multiplicative subgroup. Although, for efficiency, $\mathcal{R}$ should be chosen to be a multiplicative group, we think it is simpler that the completeness of the argument does not use this fact.

The problem with this approach is that sampling and sending a commitment to a uniformly random vector in the row space $\mathbf{Y}$ is linear in its dimension, that is, the number of equations defining the circuit, which is, in this case, linear in the number of gates. To avoid the linear verifier, we recall the sampling techniques of [39, 128].

So far, all the results we have described are for the case where the circuit has a public input. We can achieve zero-knowledge by reducing to this case with an extractable commitment to the input (which incurs an overhead of $O(\log n)$, where n is the input size) and with standard techniques to hide the intermediate values.

Organization. We recall some chapter-specific definitions and assumptions in section 4.2. In section 4.3, we recall and adapt as necessary the schemes that we will use as building blocks for our constructions. In section 4.4, we present our main construction, showing how to modify the construction of [86] to achieve $O(\log d)$ communication. In section 4.5, we describe the alternative scheme for proving linear equations, which does not require a circuit-dependent CRS. Finally, we discuss how to add the zero-knowledge property to our constructions in section 4.6.

4.2 Preliminaries

4.2.1 Promise problems and knowledge transfer arguments.

The idea of [86] of aggregating all the proofs in a single multiplicative level of the circuit seems to allow naturally for some kind of modular security proof as, essentially, the same security proof needs to be repeated d times, once per level. However, proving soundness in a modular way, and under standard assumptions (where shrinking commitments to each of the level wires cannot be extracted), seems out of reach.

Fortunately, [86] observed that a relaxed notion of soundness is sufficient. Namely, it is enough to prove that some knowledge transfer property is satisfied: at level i it is sufficient to prove that (a) *if* the prover knows openings to all the commitments outputs of multiplication at depth at most $i-1$, then, it also knows an opening of the right and left wires at level i, and (b) *if* the prover knows an opening to the input wires at level i, then it knows an opening at level $i+1$. For each level, this is formalized as an argument for a promise problem [65]: completeness is defined in the usual way for a language $\mathcal{L}$, but for soundness we only prove that the adversary cannot cheat for statements not in $\mathcal{L}$ for which the promise holds (that is, the language for which we say the adversary is successful in breaking soundness is a proper subset of the complement of $\mathcal{L}$).

4.2.2 Assumptions

We now describe the assumptions that we will use through the chapter, in addition to those of section 2.3. Let $\mathcal{G}$ be a bilinear group generator. On input a security parameter 1^λ, it produces $gk = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are groups of prime order $p = \Theta(2^\lambda)$, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear map.

We will consider the following distribution for MDDH-type assumptions.

Let $\mathcal{R} = \{r_1, \ldots, r_q\} \subseteq \mathbb{Z}_p$ be an interpolation set, and let $\lambda_i(X)$ is the ith Lagrangian basis polynomial associated to $\mathcal{R}$.

$$\mathcal{LG}_{\mathcal{R},2} : \mathbf{A} = \begin{pmatrix} \lambda_1(s_1) & \lambda_1(s_2) \\ \lambda_2(s_1) & \lambda_2(s_2) \\ \vdots & \vdots \\ \lambda_q(s_1) & \lambda_q(s_2) \end{pmatrix},$$

where $s_i \leftarrow \mathbb{Z}_p$.

Assumption 15 (q-Discrete Logarithm Assumption [23])**.** *For all non-uniform PPT adversaries $\mathcal{A}$, and relative to $gk \leftarrow \mathcal{G}(1^\lambda)$ and the coin tosses of adversary $\mathcal{A}$,*

$$\mathsf{Pr}\left[x' \leftarrow \mathcal{A}\left(gk, \{[x^i]_{1,2}\}_{i=0}^{q}\right) : x = x'\right] \leq \mathsf{negl}(\lambda).$$

The next assumption was introduced in [86]. Given $\mathcal{R}$ as above, we consider its vanishing polynomial $t(X) = \prod_{r\in\mathcal{R}}(X - r)$.

Assumption 16 ($\mathcal{R}$-Rational Strong Diffie–Hellman Assumption in $\mathbb{G}_1$ [86])**.** *For all non-uniform PPT adversaries $\mathcal{A}$,*

$$\mathsf{Pr}\left[[z, w]_1 \leftarrow \mathcal{A}\left(gk, \mathcal{R}, \{[x^i]_{1,2}\}_{i=0}^{q-1}, [x^q]_2\right) : z \neq 0 \wedge z = wt(x)\right] \leq \mathsf{negl}(\lambda),$$

where the probability is taken over $gk \leftarrow \mathcal{G}(1^\lambda)$ and the coin tosses of $\mathcal{A}$.

This assumption is a generalization of the q-SFrac Assumption considered in [81], where with the same input, the adversary has to output $[r_1(s)/r_2(s)]_1$ for some fixed polynomials $r_1(X), r_2(X)$. This assumption is a flexible version [132] for $r_1(X) = 1, r_2(X) = t(X)$, where the adversary chooses the generator.

In section 4.5, we make use of the following new assumption. Let $\mathcal{R}$, $t(X)$ and $\{\lambda_i(X)\}_{i=1}^{q}$ be as above, and let $v \in \mathbb{Z}_p \setminus \mathcal{R}$.

Assumption 17. *For all non-uniform PPT adversaries $\mathcal{A}$,*

$$\mathsf{Pr}\left[\begin{array}{l} [a, \{b_i, c_i, d_i\}_{i=1}^{n}]_1 \leftarrow \mathcal{A}\left(gk, \mathcal{R}, v, \{[x^i]_{1,2}\}_{i=0}^{q-1}, [x^q]_2\right) : \\ a\lambda_i(x) + b_i(x - v) + c_i t(x) = 0 \wedge b_i(x - v) + d_i = 0 \end{array}\right] \leq \mathsf{negl}(\lambda),$$

where the probability is taken over $gk \leftarrow \mathcal{G}(1^\lambda)$ and the coin tosses of $\mathcal{A}$.

This assumption can be thought of as a sort of KerMDH assumption in $\mathbb{G}_2$ for the matrix distribution

$$\Lambda_{n,v} : \mathbf{A} = \left(\begin{array}{cccc|cccc} \lambda_1(x) & \dots & \lambda_n(x) & & & & & \\ x-v & & & & x-v & & & \\ & \ddots & & & & \ddots & & \\ & & x-v & & & & x-v & \\ t(x) & & & & & & & \\ & \ddots & & & & & & \\ & & t(x) & & & & & \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \end{array}\right),$$

in which the additional information $\{[x^i]_1\}_{i=1}^{q-1}$ is provided to the adversary. Note that knowing the elements in the matrix in $\mathbb{G}_2$ is equivalent to knowing $\{[x^i]_2\}_{i=1}^{q}$. The generic hardness of this assumption is argued below.

Proposition 10. *Assumption 17 holds against generic adversaries.*

Proof. A generic adversary against the problem receives $\{[x^i]_{1,2}\}_{i=1}^{n-1}, [x^n]_2$, and is asked to find a nontrivial vector

$$[a, b_1, \dots, b_n, c_1, \dots, c_n, d_1, \dots, c_n]_1$$

such that

$$\begin{aligned} a\lambda_i(x) + b_i(x-v) + c_i t(x) &= 0, \\ b_i(x-v) + d_i &= 0, \end{aligned}$$

for all $i = 1 \dots, n$. We show that a generic adversary $\mathcal{A}$ cannot find such solution. For simplicity, we focus on the case $v = 0$, although the proof is essentially the same for any $v \notin \mathcal{R}$.

Essentially, all that the adversary can do is choose a, b_i, c_i, d_i as polynomials of x of degree up to $n-1$, since they are not provided with higher powers of x in $\mathbb{G}_1$. The equation $b_i x + d_i = 0$ tells us that necessarily $\deg b_i \leq n-2$. We distinguish two cases:

- If $b_i = 0$, the first equation becomes $a\lambda_i(x) + c_i t(x) = 0$. Observe that $\lambda_i(x)$ and $t(x)$ have $n-1$ common roots, and $t(x)$ has an additional root $(x - r_i)$ which is not a root of $\lambda_i(x)$. Therefore, for the two terms to cancel out, the only possibility is that a contains the factor $(x - r_i)$.

- If $b_i \neq 0$, then the term $b_i x$ must cancel out with the other two terms. This term has degree $n-1$. Since $\deg t(x) = n$, there is no way that these two terms cancel out. As for the first term, $a\lambda_i(x)$ has degree at least $n-1$. But for the terms $a\lambda_i(x)$ and $b_i x$ to cancel out, a would need to contain the root x, since $\lambda_i(x)$ does not, and so its degree would be n and again they would not cancel out.

We have seen that the second case cannot happen, and the first implies that a contains the root $(x - r_i)$ for all i. But notice that a is the same for every equation, and so we would have $\deg a \geq n$, which is something that a generic adversary cannot compute in $\mathbb{G}_1$. Therefore, the only possibility is that $a = b_i = c_i = d_i = 0$ for all $i = 1, \ldots, n$, leading to the trivial solution.

□

4.2.3 Commitment schemes

We describe the commitment schemes used in this paper.

Commitments to group elements.

We describe a commitment scheme for committing to group elements $[m] \in \mathbb{G}$. We use the constructions of [97]. Let $ck = [\boldsymbol{u}_1, \boldsymbol{u}_2] \in \mathbb{G}^{2\times 2}$ be a commitment key. The commitment to $[m]$ with randomness $(t_1, t_2) \leftarrow \mathbb{Z}_p^2$ is defined as

$$\mathsf{Com}_{ck}([m]; t_1, t_2) = \begin{bmatrix} 0 \\ m \end{bmatrix} + t_1[\boldsymbol{u}_1] + t_2[\boldsymbol{u}_2].$$

When $\boldsymbol{u}_1, \boldsymbol{u}_2$ are sampled as linearly independent vectors, the commitment scheme is perfectly hiding and computationally binding under the $\mathcal{L}_1$-MDDH (just usual DDH) assumption (4). It is easy to see that the commitment scheme is homomorphic, i.e.,

$$\mathsf{Com}_{ck}([m + m']; t_1 + t_1', t_2 + t_2') = \mathsf{Com}_{ck}([m]; t_1, t_2) + \mathsf{Com}_{ck}([m']; t_1', t_2').$$

Shrinking commitments to group elements in a pairing group.

This construction was introduced in [3]. Consider the setting in which we have a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ available. Let $ck_1 = [x_0, \boldsymbol{x}]_2 \in \mathbb{G}_2^{1+m}$ be the commitment keys for committing to vectors in $\mathbb{G}_1^m$, and $ck_2 = [y_0, \boldsymbol{y}]_1 \in \mathbb{G}_1^{1+m}$

for committing to vectors in $\mathbb{G}_2^m$. We define

$$\mathsf{Com}_{ck_1}([\boldsymbol{m}]_1;[t]_1) = [t,\boldsymbol{m}]_1 \cdot [x_0,\boldsymbol{x}]_2 = [t]_1[x_0]_2 + \sum_i [m_i]_1[x_i]_2,$$

$$\mathsf{Com}_{ck_2}([\boldsymbol{m}]_2;[t]_2) = [y_0,\boldsymbol{y}]_1 \cdot [t,\boldsymbol{m}]_2 = [y_0]_1[t]_2 + \sum_i [y_i]_1[m_i]_2.$$

The first commitment is perfectly hiding, and computationally binding under the $\mathcal{U}_1$-KerMDH assumption (5) in $\mathbb{G}_1$, and the analogous is true for the second commitment. Both schemes are homomorphic, that is,

$$\mathsf{Com}_{ck}([\boldsymbol{m}+\boldsymbol{m}'];[t+t']) = \mathsf{Com}_{ck}([\boldsymbol{m}];[t]) + \mathsf{Com}_{ck}([\boldsymbol{m}'];[t']).$$

Note that, similarly to Pedersen commitments, they have the property that the message can "absorb" the commitment key: for any vector $(s_0,\boldsymbol{s}) \in \mathbb{Z}_p^{1+m}$,

$$\mathsf{Com}_{ck\circ(s_0,\boldsymbol{s})}([\boldsymbol{m}];[t]) = [t,\boldsymbol{m}]\cdot[ck\circ(s_0,\boldsymbol{s})] = [s_0 t,\boldsymbol{m}\circ\boldsymbol{s}]\cdot[ck] = \mathsf{Com}_{ck}([\boldsymbol{m}\circ\boldsymbol{s}];[s_0 t]).$$

Lagrangian Pedersen commitments.

We introduce a type of commitment often used in SNARKs, and formalized in [127]. Let $\mathcal{R} = \{r_1,\ldots,r_n\} \subseteq \mathbb{Z}_p$. Then the Lagrange basis polynomials with respect to $\mathcal{R}$ are

$$\lambda_i(X) = \prod_{j\neq i} \frac{(X-r_j)}{(r_i-r_j)}.$$

We define $\boldsymbol{\lambda}(X) = (\lambda_1(X),\ldots,\lambda_n(X))$. Let $ck = (\mathcal{R},\{[s^i]\}_{i=0}^{m-1})$ for some $s \leftarrow \mathbb{Z}_p$. Given $\boldsymbol{m} \in \mathbb{Z}_p^n$, we define the Lagrangian Pedersen commitment of $\boldsymbol{m}$ as

$$\mathsf{Com}_{ck}(\boldsymbol{m}) = \boldsymbol{m}\cdot[\boldsymbol{\lambda}(s)]_1 = \sum_{i=1}^{n} m_i[\lambda_i(s)].$$

This commitment scheme computationally binding under the n-DLog assumption (15).

4.3 Building blocks

In this section, we lay out some schemes that we will use as building blocks in our construction. In Section 4.3.1, we briefly recall the proofs of membership in linear spaces of [115] and [85]. In Sections 4.3.2, 4.3.3 and 4.3.4, we adapt the schemes from [24] to the bilinear setting. The translations are mostly straightforward, with some minor changes due to the new setting.

4.3.1 Membership in linear spaces

Given a pairing group, the QA-NIZK argument of [115] proves membership in the language

$$\mathcal{L}_{[\mathbf{M}]_1} = \{[\boldsymbol{u}]_1 \in \mathbb{G}_1^d \mid \exists \boldsymbol{w} \in \mathbb{Z}_p^N \text{ such that } \boldsymbol{u} = \mathbf{M}\boldsymbol{w}\}.$$

The CRS contains $[\mathbf{M}^\top \mathbf{K}]_1, [\mathbf{A}, \mathbf{K}\mathbf{A}]_2$, for a key $\mathbf{K} \leftarrow \mathbb{Z}_p^{d\times(k+1)}$ and a matrix $\mathbf{A} \leftarrow \mathcal{D}_k$, where $\mathcal{D}_k$ is a distribution for which the MDDH problem is hard. The proof consists of

$$[\boldsymbol{\pi}]_1 = \boldsymbol{w}^\top [\mathbf{M}^\top \mathbf{K}]_1,$$

and verification checks the equation

$$e([\boldsymbol{u}^\top]_1, [\mathbf{K}\mathbf{A}]_2) = e([\boldsymbol{\pi}]_1, [\mathbf{A}]_2).$$

The scheme is perfect zero-knowledge and computationally sound under the $\mathcal{L}_k$-KerMDH assumption (5). In its most efficient instantiation, and assuming that the distribution of $\mathbf{M}$ is witness-sampleable, we can take $k = 1$, and replace $\mathbf{A}$ by the truncated square matrix $\overline{\mathbf{A}}$. Hence, the right hand side of the verification equation is just one element in $\mathbb{G}_T$.

The bilateral counterpart of this scheme, introduced by [85], proves membership in the language

$$\mathcal{L}_{[\mathbf{M}]_1,[\mathbf{N}]_2} = \left\{([\boldsymbol{u}]_1, [\boldsymbol{v}]_2) \in \mathbb{G}_1^{d_1} \times \mathbb{G}_2^{d_2} \mid \exists \boldsymbol{w} \in \mathbb{Z}_p^N \text{ such that } \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} = \begin{pmatrix} \mathbf{M} \\ \mathbf{N} \end{pmatrix} \boldsymbol{w}\right\}.$$

The argument is very similar to the one above, but it uses two keys $\mathbf{K}$ and $\mathbf{L}$, one for each side of the bilinear group, and a masking term $\mathbf{Z}$. The proof consists of

$$\boldsymbol{\pi} = \boldsymbol{w}^\top [\mathbf{M}^\top \mathbf{K} + \mathbf{Z}]_1, \qquad \boldsymbol{\theta} = \boldsymbol{w}^\top [\mathbf{N}^\top \mathbf{L} - \mathbf{Z}]_2,$$

and the verifier checks the equation

$$e([\boldsymbol{u}^\top]_1, [\mathbf{K}\overline{\mathbf{A}}]_2) + e([\mathbf{L}\overline{\mathbf{A}}]_1, [\boldsymbol{v}]_2) = e([\boldsymbol{\pi}]_1, [\overline{\mathbf{A}}]_2) + e([\overline{\mathbf{A}}]_1, [\boldsymbol{\theta}]_2).$$

Again, the scheme is perfect zero-knowledge, and computationally sound under the $\mathcal{L}_k$-SKerMDH assumption (6). The only difference is that this assumption is not secure for $k = 1$, so we must take at least $k = 2$, but the proof size is still a constant number (4) of group elements.

4.3.2 Polynomial evaluation argument

We describe a scheme for committing to a Laurent polynomial $[t(Z)] = \sum_{i=-d_1}^{d_2} [t_i] Z^i$ with $t_0 = 0$, and later revealing verifiable images of points, on demand. For the CRS we just need the commitment keys ck. Note that we can rewrite the polynomial as

$$[t(Z)] = [t_{-d_1}, \ldots, t_{-1}, t_1, \ldots, t_{d_2}] \cdot (Z^{-d_1}, \ldots, Z^{-1}, Z, \ldots, Z^{d_2}).$$

- $\mathcal{P} \to \mathcal{V}$: for $i = -d_1, \ldots, -1, 1, \ldots, d_2$, pick $\boldsymbol{\rho}_i \leftarrow \mathbb{Z}_p^2$ and commit to the coefficients of the polynomial using the commitments to group elements from section 4.2.3:
$$[C_i] = \mathsf{Com}_{ck}([t_i]; \boldsymbol{\rho}_i).$$
Send $\{C_i\}_{i=-d_1}^{-1}, \{C_i\}_{i=1}^{d_2}$ to the verifier.
- $\mathcal{V} \to \mathcal{P}$: pick $z \leftarrow \mathbb{Z}_p$ and send it to the prover.
- $\mathcal{P} \to \mathcal{V}$: compute $[t] = [t(z)]$ and
$$\boldsymbol{\rho} = (\boldsymbol{\rho}_{-d_1}, \ldots, \boldsymbol{\rho}_{-1}, \boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_{d_2}) \cdot (z^{-d_1}, \ldots, z^{-1}, z, \ldots, z^{d_2}).$$
Send $[t], \boldsymbol{\rho}$ to the verifier.
- $\mathcal{V}$: check that
$$[C_{-d_1}, \ldots, C_{-1}, C_1, \ldots, C_{d_2}] \cdot (z^{-d_1}, \ldots, z^{-1}, z, \ldots, z^{d_2}) = \mathsf{Com}_{ck}([t]; \boldsymbol{\rho}).$$

Theorem 11. *The scheme above is complete, perfect zero-knowledge, and computationally knowledge sound under the binding property of the underlying commitment scheme.*

Proof. It is straightforward to check that the scheme is complete. For soundness, fix C_i for $i \in \{1, \ldots, m\}$, where $m = d_1 + d_2 - 1$. Rewind the protocol m times to receive answers for challenges $z_{(1)}, \ldots, z_{(m)}$:

$$(C_{-d_2}, \ldots, C_{d_1}) \cdot (z_{(k)}^{-d_2}, \ldots, z_{(k)}^{d_1}) \text{ opens to } (t_{(k)}; \boldsymbol{\rho}_{(k)}).$$

Then

$$\begin{pmatrix} z_{(1)}^{-d_2} & \cdots & z_{(1)}^{d_1} \\ \vdots & & \vdots \\ z_{(m)}^{-d_2} & \cdots & z_{(m)}^{d_1} \end{pmatrix} \begin{pmatrix} C_1 \\ \vdots \\ C_m \end{pmatrix} \text{ opens to } \left(\begin{array}{c|c} t_{(1)} & \boldsymbol{\rho}_{(1)} \\ \vdots & \vdots \\ t_{(m)} & \boldsymbol{\rho}_{(m)} \end{array} \right),$$

and so

$$\begin{pmatrix} C_1 \\ \vdots \\ C_m \end{pmatrix} \text{ opens to } \underbrace{\begin{pmatrix} z_{(1)}^{-d_2} & \cdots & z_{(1)}^{d_1} \\ \vdots & & \vdots \\ z_{(m)}^{-d_2} & \cdots & z_{(n)}^{d_1} \end{pmatrix}^{-1}}_{=:\Lambda} \left(\begin{array}{c|c} t_{(1)} & \boldsymbol{\rho}_{(1)} \\ \vdots & \vdots \\ t_{(m)} & \boldsymbol{\rho}_{(m)} \end{array}\right).$$

Thus we can either recover the coefficients t_i of the polynomial, or break the binding property of the commitment scheme. The matrix Λ is invertible with overwhelming probability, as it is a shifted Vandermonde matrix and the $z_{(k)}$ are independent.

To prove the zero-knowledge property, we describe a transcript simulator. Assume that we are given an evaluation point z and an evaluation $[t]$. We sample $C_i \leftarrow \mathbb{Z}_p^2$ for $i = -d_1, \ldots, -1, 1, \ldots, d_2 - 1$, and $\boldsymbol{\rho} \leftarrow \mathbb{Z}_p^2$. Then we set

$$[C_{d_2}] = \frac{1}{z^{d_2}} \left(\mathsf{Com}_{ck}([t]; \boldsymbol{\rho}) - \sum_{i=-d_1}^{-1} C_i z^i - \sum_{i=1}^{d_2-1} C_i z^i \right).$$

We observe that the C_i are uniformly random, conditioned on satisfying the verification equation. This is the same distribution as in an honest execution of the protocol, as the commitments are perfectly hiding.

□

4.3.3 Lifted inner product argument

We will also make use of the inner pairing product argument of [29], which is an adaptation of the inner product argument of [24] to the pairing setting. This scheme allows us to prove knowledge of vectors $[\boldsymbol{a}]_1, [\boldsymbol{b}]_2$ such that $[\boldsymbol{a}]_1 \cdot [\boldsymbol{b}]_2 = [z]_T$, given $[z]_T$ and (non-hiding) commitments to $\boldsymbol{a}, \boldsymbol{b}$.

More precisely, let $[\boldsymbol{x}]_1 \leftarrow \mathbb{G}_1^n, [\boldsymbol{y}]_2 \leftarrow \mathbb{G}_y^n$ be the commitment keys. Then, the scheme is a proof of knowledge for the relation

$$\{(([A, B]_T), ([\boldsymbol{a}]_1, [\boldsymbol{b}]_2)) \mid A = \boldsymbol{a} \cdot \boldsymbol{x} \wedge B = \boldsymbol{y} \cdot \boldsymbol{b} \wedge z = \boldsymbol{a} \cdot \boldsymbol{b}\}.$$

The core of the protocol is a 2-move argument that reduces the statement to another of the same form but with vectors half the length of the previous ones. Thus, after a logarithmic number of rounds, the witness has constant size and can easily be revealed to the verifier.

The resulting scheme is knowledge sound under the binding property of the underlying commitment scheme, and can be made non-interactive in the random oracle model. Regarding efficiency, the communication cost is $O(\log n)$, while the prover and verifier computation costs are $O(n)$. The details of the scheme are as follows.

We commit to $[\boldsymbol{a}]_1, [\boldsymbol{b}]_2$ as described in section 4.2.3, but without randomness:

$$[A]_T = \mathsf{Com}_{ck_1}([\boldsymbol{a}]_1) = \sum_{i=1}^{n} [a_i]_1 [x_i]_2 = [\boldsymbol{a}]_1 \cdot [\boldsymbol{x}]_2,$$
$$[B]_T = \mathsf{Com}_{ck_2}([\boldsymbol{b}]_2) = \sum_{i=1}^{n} [y_i]_1 [b_i]_2 = [\boldsymbol{y}]_1 \cdot [\boldsymbol{b}]_2.$$

The core of the protocol is a 2-move argument that reduces the statement to another of the same form but with shorter vectors.

- $\mathcal{P} \to \mathcal{V}$: for $m \mid n$, parse the witness as

  $$[\boldsymbol{a}]_1 = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m]_1, \qquad [\boldsymbol{b}]_2 = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m]_2,$$

  and the commitment keys as

  $$[\boldsymbol{x}]_2 = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m]_2, \qquad [\boldsymbol{y}]_1 = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m]_1.$$

  Then the prover computes "commitments" to $\boldsymbol{a}, \boldsymbol{b}, z$ with shifted commitment keys,

  $$[A_k]_T = \sum_{i=\max\{1,1-k\}}^{\min\{m,m-k\}} [\boldsymbol{a}_{i+k}]_1 \cdot [\boldsymbol{x}_i]_2,$$
  $$[B_k]_T = \sum_{i=\max\{1,1-k\}}^{\min\{m,m-k\}} [\boldsymbol{y}_i]_1 \cdot [\boldsymbol{b}_{i+k}]_2,$$
  $$[z_k]_T = \sum_{i=\max\{1,1-k\}}^{\min\{m,m-k\}} [\boldsymbol{a}_i]_1 \cdot [\boldsymbol{b}_{i+k}]_2,$$

  for $k = 1 - m, \ldots, m - 1$, and sends these to the verifier.

- $\mathcal{V} \to \mathcal{P}$: pick $s \leftarrow \mathbb{Z}_p$ and send it to the prover.

At this point, the prover can compute

$$[\boldsymbol{a}']_1 = \sum_{i=1}^{m} s^i [\boldsymbol{a}_i]_1, \qquad [\boldsymbol{b}']_2 = \sum_{i=1}^{m} s^{-i} [\boldsymbol{b}_i]_1.$$

as a new and shorter witness, and prover and verifier can compute the corresponding statement, analogous to the original, as

$$ck'_1 = \sum_{i=1}^{m} s^{-i} [\boldsymbol{x}_i]_2, \qquad ck'_2 = \sum_{i=1}^{m} s^{i} [\boldsymbol{y}_i]_1,$$

$$[A']_T = \mathsf{Com}_{ck'_1}([\boldsymbol{a}']_1) = \sum_{k=1-m}^{m-1} s^k [A_k]_T$$

$$[B']_T = \mathsf{Com}_{ck'_2}([\boldsymbol{b}']_2) = \sum_{k=1-m}^{m-1} s^{-k} [B_k]_T$$

$$[z']_T = \sum_{k=1-m}^{m-1} s^{-k} [z_k]_T$$

The prover and verifier can recursively shorten the problem, until the witness is very short and can be revealed directly.

Theorem 12. *The scheme above is complete, and knowledge sound under the binding property of the underlying commitment scheme.*

Proof. Completeness follows by inspection. For knowledge soundness, our goal is to build an extractor that, given (rewindable) access to the prover, either breaks the binding property of the commitment scheme (that is, it finds a nontrivial linear dependence relation between the components of $[\boldsymbol{x}]_2$ or $[\boldsymbol{y}]_1$), or succeeds in extracting a valid witness.

We observe that finding $[\boldsymbol{\lambda}']_1$ such that $[\boldsymbol{\lambda}']_1 \cdot [\boldsymbol{x}']_2 = 0$ allows us to recover $[\boldsymbol{\lambda}]$ such that $[\boldsymbol{\lambda}]_1 \cdot [\boldsymbol{x}]_2 = 0$, since

$$[\boldsymbol{\lambda}']_1 \cdot \left((s^{-1}, \ldots, s^{-m}) \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_m \end{bmatrix}_2 \right) = [\boldsymbol{\lambda}']_1 \cdot [\boldsymbol{x}']_2 = 0,$$

and this is nontrivial as long as $\boldsymbol{\lambda}'$ is nontrivial and $s \neq 0$. Thus, a break of the problem in any recursive step can be moved upstream all the way to the original statement. The same is true for $[\boldsymbol{y}]_1$. With this in mind, we just need to find such a relation in any of the recursive steps.

Assume that at some level we are given witnesses $[\boldsymbol{a}']_1, [\boldsymbol{b}']_2$ of shortened statements. From them we will extract the witness $[\boldsymbol{a}]_1, [\boldsymbol{b}]_2$ of the previous level. Given $2m-1$ challenges $s_{(1-m)}, \ldots, s_{(m-1)}$ for a fixed initial message from the prover, and the corresponding valid witnesses, we have that

$$[A'_{(j)}]_T = \mathsf{Com}_{ck'_1}([\boldsymbol{a}']_1) = [\boldsymbol{a}'_{(j)}]_1 \cdot [\boldsymbol{x}'_{(j)}]_2 = [\boldsymbol{a}'_{(j)}]_1 \cdot \sum_{i=1}^{m} s_{(j)}^{-i} [\boldsymbol{x}_i]_2, \tag{4.1}$$

and on the other hand

$$[A'_{(j)}]_T = \sum_{k=1-m}^{m-1} s_{(j)}^{k} [A_k]_T. \tag{4.2}$$

We rewrite the equality of the right hand sides as a system of linear equations:

$$\underbrace{\begin{pmatrix} s_{(1-m)}^{1-m} & \cdots & s_{(1-m)}^{m-1} \\ \vdots & & \vdots \\ s_{(m-1)}^{1-m} & \cdots & s_{(m-1)}^{m-1} \end{pmatrix}}_{=:\mathbf{S}} \begin{bmatrix} A_{1-m} \\ \vdots \\ A_{m-1} \end{bmatrix}_T =$$

$$= \begin{bmatrix} \boldsymbol{a}'_{(1-m)} & & \\ & \ddots & \\ & & \boldsymbol{a}'_{(m-1)} \end{bmatrix}_1 \begin{pmatrix} s_{(1)}^{1-m} & \cdots & s_{(1-m)}^{m-1} \\ \vdots & & \vdots \\ s_{(m)}^{1-m} & \cdots & s_{(m-1)}^{m-1} \end{pmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_m \end{bmatrix}_2,$$

Since the matrix $\mathbf{S}$ is a Vandermonde matrix, with the row j shifted by a factor $s_{(j)}^{1-m}$, and the $s_{(j)}$ are sampled uniformly at random, we have that this matrix is invertible with overwhelming probability. Thus we find coefficients $[\boldsymbol{a}_{k,i}]_1$ such that

$$[A_k]_T = \sum_{i=1}^{m} [\boldsymbol{a}_{k,i}]_1 [\boldsymbol{x}_i]_2. \tag{4.3}$$

Then, combining equations (4.1), (4.2) and (4.3), we have that either

$$s^{-i} [\boldsymbol{a}'_{(j)}]_1 = \sum_{k=1-m}^{m-1} s^{k} [\boldsymbol{a}_{k,i}]_1,$$

or we have found a non-trivial zero linear combination of the $[\boldsymbol{x}_i]_2$. In the former case,

$$[\boldsymbol{a}'_{(j)}]_1 = \sum_{k=1-m}^{m-1} s^{k+i} [\boldsymbol{a}_{k,i}]_1,$$

for all $i = 1, \ldots, m$. Observe that the left hand side does not depend on i. In particular, for $i = 1$ we have that $2 - m \leq k + i \leq m$, and for $i = k$ we have that $1 \leq k + i \leq 2m - 1$. Since coefficients of the same degree must match for all i, this implies that $\boldsymbol{a}_{k,i} = 0$ for $k + i < 1$ or $k + i > m$. A change of variable $\ell = k + i$ shows that

$$[\boldsymbol{a}'_{(j)}]_1 = \sum_{\ell=1}^{m} s^{\ell} [\boldsymbol{a}_{k,\ell-k}]_1.$$

Therefore, we have extracted the witness

$$[\boldsymbol{a}]_1 = [\boldsymbol{a}_{1,m-1}, \ldots, \boldsymbol{a}_{m,0}]_1.$$

A completely analogous procedure yields $[\boldsymbol{b}]_2$. It remains to show that $z = \boldsymbol{a} \cdot \boldsymbol{b}$. We have that

$$\sum_{k=1-m}^{m-1} s^{-k} [z_k]_T = [z'_{(j)}]_T = [\boldsymbol{a}'_{(j)}]_1 \cdot [\boldsymbol{b}'_{(j)}]_2 = \left(\sum_{i=1}^{m} s^{i} [\boldsymbol{a}_i]_1 \right) \cdot \left(\sum_{i=1}^{m} s^{-i} [\boldsymbol{b}_i]_2 \right)$$

for $j = 1 - m, \ldots, m - 1$ and, as above, we can extract

$$z_0 = z = \sum_{i=1}^{m} [\boldsymbol{a}_i]_1 \cdot [\boldsymbol{b}_i]_2 = [\boldsymbol{a}]_1 \cdot [\boldsymbol{b}]_2.$$

□

4.3.4 Pairing equations argument

Using the schemes presented so far as building blocks, we can build a proof of both quadratic and linear equations in a bilinear group. The goal is to prove knowledge of a solution of $O(d)$ equations of each type, using only $O(\log d)$ communication.

More precisely, we want to prove knowledge of $[L_i, O_i]_1, [R_i]_2$ such that

$$[L_i]_1 [R_i]_2 - [O_i]_1 [1]_2 = 0,$$

for all $i = 1, \ldots, d$, and also a number $O(d)$ of linear relations between these. The proof is essentially an adaptation of [24] to the bilinear setting, and works by embedding each equation into a different degree of a polynomial, which is then tested in a random point chosen by the verifier. This check can be rewritten as an inner product, and so the inner product argument of the previous section is used to reduce communication to $O(\log d)$.

For simplicity, we first describe the construction for proving these quadratic equations, and later we will discuss how to add linear equations in the same variables. We note that it is also straightforward to modify the proof to accommodate quadratic equations with linear terms or more variables.

Proof for quadratic equations.

For the CRS we just need the commitment keys

$$ck_1 = [x_1, \ldots, x_n]_2, \qquad ck_2 = [y_1, \ldots, y_n]_1.$$

Additionally, we assume that $[L_i, R_i, O_i]$ are known in both $\mathbb{G}_1$ and $\mathbb{G}_2$ by the prover.[5]

- $\mathcal{P} \to \mathcal{V}$: for $d = mn$ and $E \in \{L, R, O\}$ and $\gamma \in \{1, 2\}$, arrange the solutions in matrices and commit to the rows:

$$[\mathbf{E}]_\gamma = \begin{bmatrix} \boldsymbol{E}_1 \\ \vdots \\ \boldsymbol{E}_m \end{bmatrix}_\gamma = \begin{bmatrix} E_{11} & \ldots & E_{1n} \\ \vdots & & \vdots \\ E_{m1} & \ldots & E_{mn} \end{bmatrix}_\gamma \begin{array}{l} \to \quad CE_1^\gamma = \mathsf{Com}_{ck_\gamma}([\boldsymbol{E}_1]_\gamma; \rho_{E,1}^\gamma), \\ \qquad \vdots \\ \to \quad CE_m^\gamma = \mathsf{Com}_{ck_\gamma}([\boldsymbol{E}_m]_\gamma; \rho_{E,m}^\gamma), \end{array}$$

  and send $\{CL_i^\gamma, CR_i^\gamma, CO_i^\gamma\}_{i,\gamma=1}^{m,2}$ to the verifier.

- $\mathcal{V} \to \mathcal{P}$: pick $s \leftarrow \mathbb{Z}_p$ and send it to the prover.

- $\mathcal{P} \to \mathcal{V}$: let $\boldsymbol{s} = (s^m, s^{2m}, \ldots, s^{nm})^\top$, and define the polynomials

$$\begin{aligned} [\boldsymbol{p}(Z)] &= \sum_{i=1}^{m} [\boldsymbol{L}_i] s^i Z^i + \sum_{i=1}^{m} [\boldsymbol{R}_i] Z^{-i} + \sum_{i=1}^{m} [\boldsymbol{O}_i] Z^{m+i}, \\ \boldsymbol{q}(Z) &= \sum_{i=1}^{m} -s^i \boldsymbol{s} Z^{-m-i} \\ [\boldsymbol{p}'(Z)] &= [\boldsymbol{p}(Z)] \circ \boldsymbol{s} - 2[\boldsymbol{q}(Z)], \end{aligned}$$

  both in $\mathbb{G}_1$ and $\mathbb{G}_2$, and

$$[u(Z)]_T = [\boldsymbol{p}(Z)]_1 \cdot [\boldsymbol{p}'(Z)]_2.$$

[5] This is not a problem for our intended application, although it makes parts of the proof twice as large.

Then the degree 0 term of $u(Z)$ is

$$2\left(\sum_{i=1}^{m} \boldsymbol{L}_i \cdot (\boldsymbol{R}_i \circ \boldsymbol{s})s^i - \sum_{i=1} s^i \boldsymbol{s} \cdot \boldsymbol{O}_i\right),$$

and note that this encodes the original set of equations by embedding each within a different power of s. The prover sends the coefficients of $[u(Z)]_T$ to the verifier.

- $\mathcal{V} \to \mathcal{P}$: pick $z \leftarrow \mathbb{Z}_p$ and send it to the prover.

($\mathbf{O}(\sqrt{\mathbf{d}})$ argument – constant number of rounds)

- $\mathcal{P} \to \mathcal{V}$: compute

$$[\rho^\gamma]_\gamma = \sum_{i=1}^{m} [\rho^\gamma_{L,i}]_\gamma s^i z^i + \sum_{i=1}^{m} [\rho^\gamma_{R,i}]_\gamma z^{-i} + \sum_{i=1}^{m} [\rho^\gamma_{O,i}]_\gamma z^{m+i}.$$

and send $([\boldsymbol{p}(z)]_{1,2}, [\rho^1]_1, [\rho^2]_2)$ to the verifier.

- $\mathcal{V}$: compute $[\boldsymbol{p}'(z)]_{1,2}$ and check that

$$\sum_{i=1}^{m} CL_i^\gamma s^i z^i + \sum_{i=1}^{m} CR_i^\gamma z^{-i} + \sum_{i=1}^{m} CO_i^\gamma z^{m+i} = \mathsf{Com}_{ck_\gamma}([\boldsymbol{p}(z)]_\gamma; [\rho^\gamma]_\gamma),$$

for $\gamma = 1, 2$, and that

$$[u(z)]_T = [\boldsymbol{p}(z)]_1 \cdot [\boldsymbol{p}'(z)]_2,$$

where $u(Z)$ has independent term 0. Accept if all the conditions hold.

($\mathbf{O}(\log \mathbf{d})$ argument – $\mathbf{O}(\log \mathbf{d})$ number of rounds)

- $\mathcal{P} \longleftrightarrow \mathcal{V}$: prover and verifier engage in an inner product argument for $u(z) = \boldsymbol{p}(z) \cdot \boldsymbol{p}'(z)$. To do so, the verifier needs zero-randomness commitments to $\boldsymbol{p}(z)$ and $\boldsymbol{p}'(z)$. The prover computes and sends

$$[\rho^\gamma]_\gamma = \sum_{i=1}^{m} [\rho^\gamma_{L,i}]_\gamma s^i z^i + \sum_{i=1}^{m} [\rho^\gamma_{R,i}]_\gamma z^{-i} + \sum_{i=1}^{m} [\rho^\gamma_{O,i}]_\gamma z^{m+i},$$

for $\gamma = 1, 2$, to the verifier. Then the commitment to $[\boldsymbol{p}(z)]_\gamma$ can be computed as

$$\sum_{i=1}^{m} CL^\gamma_i s^i z^i + \sum_{i=1}^{m} CR^\gamma_i z^{-i} + \sum_{i=1}^{m} CO^\gamma_i z^{m+i} - \mathsf{Com}_{ck_\gamma}(0, [\rho^\gamma]_\gamma).$$

For the commitment to $\boldsymbol{p}'(z)$, we make the following observation. Let

$$ck'_2 = ck_2 \circ (s^{-m}, s^{-2m}, \ldots, s^{-nm})$$

be a new commitment key for $\mathbb{G}_2$. Note that this is equivalent to $ck_2 = ck'_2 \circ \boldsymbol{s}$. Then the verifier can compute the zero-randomness commitment to $\boldsymbol{p}'(z)$ as

$$\begin{aligned}
&\mathsf{Com}_{ck_2}([\boldsymbol{p}(z)]_2; 0) - \mathsf{Com}_{ck'_2}([2\boldsymbol{q}(z)]_2; 0) = \\
&= \mathsf{Com}_{ck'_2 \circ \boldsymbol{s}}([\boldsymbol{p}(z)]_2; 0) - \mathsf{Com}_{ck'_2}([2\boldsymbol{q}(z)]_2; 0) = \\
&= \mathsf{Com}_{ck'_2}([\boldsymbol{p}(z) \circ \boldsymbol{s} - 2\boldsymbol{q}(z)]_2; 0) = \mathsf{Com}_{ck'_2}([\boldsymbol{p}'(z)]_2; 0).
\end{aligned}$$

Finally, prover and verifier can initiate an inner product argument for

$$[u(z)]_T = [\boldsymbol{p}(z)]_1 \cdot [\boldsymbol{p}'(z)]_2,$$

with commitment keys ck_1 and ck'_2, and commitments as described above.

Theorem 13. *The scheme above is complete, and knowledge sound under the binding property of the underlying commitment scheme.*

Proof. Completeness follows by inspection. For soundness, the techniques of this proof are essentially the same that we have used in the previous building blocks. We distinguish between the $O(\sqrt{d})$ and the $O(\log d)$ cases. In the first, assume that we fix the first three messages, and then get $4m + 1$ responses to $4m + 1$ challenges $z_{(j)}$. By solving the linear system that arises from the

homomorphic property of the commitments, we extract $[L_i, R_i, O_i, H_i]_{1,2}$ from the equations

$$[\boldsymbol{p}(z_{(j)})] = \sum_{i=1}^{m} [L_i] s^i z_{(j)}^i + \sum_{i=1}^{m} [R_i] z_{(j)}^{-i} + \sum_{i=1}^{m} [O_i] z_{(j)}^{m+i}.$$

Note that we have as many equations as unknowns, and the challenges $z_{(j)}$ are independent, so the system is solvable. We have that

$$u(z_{(j)}) = \boldsymbol{p}(z_{(j)}) \cdot \boldsymbol{p}'(z_{(j)}).$$

for $4m + 1$ independent points $z_{(j)}$, so these are equal as polynomials. Then, since the degree-0 term of $u(Z)$ is 0 and the degree-0 term of $\boldsymbol{p}(Z) \cdot \boldsymbol{p}'(Z)$ is

$$2 \left(\sum_{i=1}^{m} \boldsymbol{L}_i \cdot (\boldsymbol{R}_i \circ \boldsymbol{s}) s^i - \sum_{i=1} s^i \boldsymbol{s} \cdot \boldsymbol{O}_i \right),$$

we conclude that this expression is 0.

Now assume that we have $n(m+1)+1$ challenges $s_{(j)}$ in the second round, with their corresponding answers, for the same fixed first round. Then the expression above, as a polynomial in s, is a polynomial of degree $n(m+1)$, and we have found $n(m+1)+1$ roots, so the polynomial is identically zero. Thus, the original equations hold.

For the $O(\log d)$ case, extract a witness for the inner product argument, using its knowledge soundness, and then we are back in the same situation as in the $O(\sqrt{d})$ case, discussed above.

□

Efficiency. Without the inner product argument, the most expensive parts of the communication are the first message, which is $O(m)$ group elements, the third, which is $O(\deg u)$, where $\deg u = 2m$, and the fifth, which is $O(n)$. This is optimized by taking $m \approx n \approx \sqrt{d}$. With the inner product argument, the dependence on n is dropped after $\log d$ iterations (each of them with constant communication), so the optimal case is $m = 2, n = d/2$, which yields communication complexity $O(\log d)$.

Zero-knowledge. Adding the zero-knowledge property to the construction is quite straightforward. We start by reviewing which elements of the transcript might leak information about the witness. In the first message, the prover sends

commitments, which are perfectly hiding and thus do not leak information. In the third, we send the coefficients of $u(Z)$. Note that this depends on $\boldsymbol{p}(Z)$, whose coefficients depend on the witness. Thus we need to mask $\boldsymbol{p}(Z)$ to avoid leaking information, so we replace our previous definition by

$$[\boldsymbol{p}(Z)] = \sum_{i=1}^{m} [\boldsymbol{L}_i] s^i Z^i + \sum_{i=1}^{m} [\boldsymbol{R}_i] Z^{-i} + \sum_{i=1}^{m} [\boldsymbol{O}_i] Z^{m+i} + [\boldsymbol{D}] Z^{2m+1},$$

where $\boldsymbol{D} \leftarrow \mathbb{Z}_p^n$ is a blinding term that the prover commits to in the first round, as $CD = \mathsf{Com}(\boldsymbol{D}, \delta)$. Later messages, in both versions, only depend on $\boldsymbol{p}(z)$. Note that the degree-0 term remains unchanged.

This modification allows to hide the evaluation of $p(Z)$ at one point, but in the protocol we are giving the coefficients of $u(Z)$, which potentially allow for many evaluations. Thus, we replace sending the coefficients of $u(Z)$ in the clear by committing to them, using the construction of Section 4.3.2, and later revealing the evaluation at z.

Theorem 14. *The scheme above, with the modification described, is zero-knowledge, under the hiding property of the underlying commitment scheme.*

Proof. Choose CL_i, CR_i, CO_i, CH_i randomly. This is distributed correctly because the commitments are perfectly hiding. Choose $\boldsymbol{p} \leftarrow \mathbb{Z}_p^n$ and set CD, ρ such that

$$\mathsf{Com}_{ck}(\boldsymbol{p}; \rho) = \sum_{i=1}^{m} [CL_i] s^i z^i + \sum_{i=1}^{m} [CR_i] z^{-i} + \sum_{i=1}^{m} [CO_i] z^{m+i} + [CD] z^{2m+1}.$$

We now need to simulate $u(Z)$ such that $u(0) = 0$ and $u(z) = \boldsymbol{p} \cdot \boldsymbol{p}'$. Given $z, u(z)$, we fake the commitments as in the proof of the zero-knowledge property of the polynomial commitment scheme.

□

Adding the linear equations.

We now turn to proving that the values $[L_i, O_i]_1, [R_i]_2$ also satisfy a set of Q linear equations, more precisely

$$\sum_{i=1}^{m} [\boldsymbol{L}_i]_1 \cdot [\boldsymbol{w}_{q,\ell,i}]_2 + \sum_{i=1}^{m} [\boldsymbol{w}_{q,r,i}]_1 \cdot [\boldsymbol{R}_i]_2 + \sum_{i=1}^{m} [\boldsymbol{O}_i]_1 \cdot [\boldsymbol{w}_{q,o,i}]_2 = 0,$$

for $q = 1, \ldots, Q$.

The key of the previous protocol is building a polynomial $u(Z)$ that can be written as an inner product of two vectors of polynomials $\boldsymbol{p}, \boldsymbol{p}'$, and whose independent term is

$$2\left(\sum_{i=1}^{m} \boldsymbol{L}_i \cdot (\boldsymbol{R}_i \circ \boldsymbol{s}) s^i - \sum_{i=1} s^i \boldsymbol{s} \cdot \boldsymbol{O}_i\right),$$

which, when seen as a polynomial in s, encodes each starting equation in a different degree of the polynomial. Thus, to add the linear equations to the protocol, we just need to modify the polynomials $\boldsymbol{p}, \boldsymbol{p}'$ accordingly, so that the degree 0 term also encodes the linear equations in another set of degrees.

Thus, we set

$$\boldsymbol{w}_{\ell,i}(s) = \sum_{q=1}^{Q} \boldsymbol{w}_{q,\ell,i} s^{d+q}, \qquad \boldsymbol{w}_{r,i}(s) = \sum_{q=1}^{Q} \boldsymbol{w}_{q,r,i} s^{d+q}$$
$$\boldsymbol{w}_{o,i}(s) = -s^i \boldsymbol{s} + \sum_{q=1}^{Q} \boldsymbol{w}_{q,o,i} s^{d+q},$$

and

$$[\boldsymbol{p}(Z)] = \sum_{i=1}^{m} [\boldsymbol{L}_i] s^i Z^i + \sum_{i=1}^{m} [\boldsymbol{R}_i] Z^{-i} + \sum_{i=1}^{m} [\boldsymbol{O}_i] Z^{m+i},$$
$$\boldsymbol{q}(Z) = \sum_{i=1}^{m} \boldsymbol{w}_{\ell,i}(s) s^{-i} Z^{-i} + \sum_{i=1}^{m} \boldsymbol{w}_{r,i}(s) Z^i + Z^{-m} \sum_{i=1}^{m} \boldsymbol{w}_{o,i}(s) Z^{-i}$$
$$[\boldsymbol{p}'(Z)] = [\boldsymbol{p}(Z)] \circ \boldsymbol{s} + 2[\boldsymbol{q}(Z)],$$

It is straightforward to check that $u(Z) = \boldsymbol{p}(Z) \cdot \boldsymbol{p}'(Z)$ now has degree 0 term

$$2\left(\sum_{i=1}^{m} \boldsymbol{L}_i \cdot (\boldsymbol{R}_i \circ \boldsymbol{s}) s^i + \sum_{i=1}^{m} \boldsymbol{L}_i \cdot \boldsymbol{w}_{\ell,i}(s) + \sum_{i=1}^{m} \boldsymbol{w}_{r,i}(s) \cdot \boldsymbol{R}_i + \sum_{i=1}^{m} \boldsymbol{O}_i \cdot \boldsymbol{w}_{o,i}(s)\right),$$

which encodes the quadratic relations in the degrees 1 to d on s, and the linear relations in the degrees $d+1, \ldots, d+Q$. The adaptation of the complete description of the protocol and the security proofs is completely straightforward.

4.4 Circuit satisfiability with logarithmic communication

4.4.1 Overview

In [86], González and Ràfols introduced a novel technique for building arguments for circuit satisfiability that are secure under falsifiable assumptions, mainly the RSDH assumption (16) and , and have a communication cost of $O(d)$ group elements, where d is the depth of the circuit. We will build on top of the arguments of [86], adding an interactive layer on top of them.

Let $\boldsymbol{x}, \boldsymbol{y}$ be the input and output, respectively, of a circuit with N multiplication gates. Let $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ be a correct assignment to the left, right and output wires of multiplication gates. The strategy of [85] consists of two main arguments, each of them dealing with a type of gate in the circuit. They slice the circuit in levels of multiplicative gates, and commit to the left, right and output wires $\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i$ of each level $i = 1, \ldots, d$ of the circuit using shrinking commitments. We denote these commitments by $[L_i]_1, [R_i]_2$ and $[O_i]_1$, respectively.

The problems of proving quadratic and linear equations for each level of the circuit are formalized as promise problems. This is just to be able to present the proofs in a modular way, but when putting the proof for the whole circuit together they obtain the usual soundness guarantees.

- Quadratic constraints: let $\mathcal{R} = \{r_1, \ldots, r_m\}$ be an interpolation set, and let $t(X) = \prod_{i=1}^{m}(X - r_i)$. For each i, they provide a constant-size proof of the quadratic equations of each level by sending and verifying the equations

$$L_i R_i - O_i = H_i t(s),$$

  as described in detail in Figure 4.1. In Section 4.4.2, we will avoid communication linear in d by sending instead linear combinations of these.

- Linear constraints: these ensure that the input of a certain level are consistent with the output of the previous levels, and are of the form

$$\boldsymbol{a}_i = \sum_{j<i} \boldsymbol{m}_{ij} \cdot \boldsymbol{c}_j, \qquad \boldsymbol{b}_i = \sum_{j<i} \tilde{\boldsymbol{m}}_{ij} \cdot \boldsymbol{c}_j.$$

$\underline{\mathsf{Setup}(gk, \mathcal{R})}$:
Sample $\overline{s} \leftarrow \mathbb{Z}_p^*$;
Output $\mathsf{crs} =$
$\Big(gk, \{[\lambda_1(\overline{s})]_\gamma, \ldots, [\lambda_m(\overline{s})]_\gamma\}_{\gamma\in\{1,2\}}$,
$\{[\overline{s}^i]_1\}_{i\in\{1,\ldots,m-2\}}, [t(\overline{s})]_2\Big)$.

$\underline{\mathcal{P}(\mathsf{crs}, \boldsymbol{a}, \boldsymbol{b})}$:
$\ell(X) = \sum_{i=1}^m a_i\lambda_i(X)$;
$r(X) = \sum_{i=1}^m b_i\lambda_i(X)$;
$o(X) = \sum_{i=1}^m c_i\lambda_i(X)$;
$h(X) = (\ell(X)r(X) - o(X))/t(X)$;
$[L]_1 = [\ell(\overline{s})]_1$; $[R]_2 = [r(\overline{s})]_2$;
$[O]_1 = [o(\overline{s})]_1$; $[H]_1 = [h(\overline{s})]_1$;
Output $[H]_1$.

$\underline{\mathcal{V}(\mathsf{crs}, \boldsymbol{a}, \boldsymbol{b}, [L]_1, [R]_2, [O]_1, [H]_1)}$:
Check if:
$e([L]_1, [R]_2) - e([O]_1, [1]_2) = e([H]_1, [t(\overline{s})]_2)$;
output 1 in this case and 0 otherwise.

Figure 4.1: proofs for quadratic equations.

Observe that the equations can be rewritten as an orthogonality problem:

$$\left(\begin{array}{c|c|c} -\mathbf{I}_N & \mathbf{0} & \mathbf{M} \\ \mathbf{0} & -\mathbf{I}_N & \tilde{\mathbf{M}} \end{array}\right) \left(\begin{array}{c} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{c} \end{array}\right) = \mathbf{0} \tag{4.4}$$

They prove all linear constraints in an aggregated manner through a QA-NIZK linear subspace proof of

$$\left(\begin{array}{c} \boldsymbol{o} \\ \boldsymbol{\ell} \\ \boldsymbol{r} \end{array}\right) = \underbrace{\left(\begin{array}{c} \mathbf{T}_O \\ \mathbf{T}_L \\ \mathbf{T}_R \end{array}\right)}_{:=\mathbf{T}} \boldsymbol{c},$$

where $\boldsymbol{o}, \boldsymbol{\ell}, \boldsymbol{r}$ are the vectors formed by the commitments O_i, L_i, R_i for $i = 1, \ldots, d$, respectively, and $\boldsymbol{T}$ is the matrix that encodes the linear constraints, aggregated by level. To do so, they use a modification of the linear space proofs of Section 4.3.1. We will reformulate the linear space membership problem as an inner product problem, and use the argument presented in the previous section.

The key idea of their construction is that these two arguments serve a function of "knowledge transfer". This means that if a prover knows the witness up

$\underline{\mathsf{Setup}(gk)}$:
Generate crs_i as in Figure 4.1, for each level $i = 1, \ldots, d$.
Generate $\mathsf{crs}_{\mathsf{lin}}$ as the CRS for the bilateral linear spaces proof.

$\underline{\mathcal{P}(\{\mathsf{crs}_i\}_{i=1}^d, \mathsf{crs}_{\mathsf{lin}}, (\boldsymbol{x}, \boldsymbol{y}), (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}))}$:
For $i = 1, \ldots, d$, compute Lagrangian Pedersen commitments $[L_i]_1, [R_i]_2, [O_i]_1$ to $\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i$, respectively.
(Quadratic constraints) For $i = 1, \ldots, d$, produce a proof $\Pi_{\mathsf{quad},i}$ that $\boldsymbol{a}_i \circ \boldsymbol{b}_i$ is an opening of $[O]_i$.
(Linear constraints) Produce a proof Π_{lin} that $[L_i]_1, [R_i]_2$ verify the linear constraints that relate them to the outputs of previous levels.
Output $\left(\{[L_i]_1, [R_i]_2, [O_i]_1, \Pi_{\mathsf{quad},i}\}_{i=1}^d, \Pi_{\mathsf{lin}}\right)$.

$\underline{\mathcal{V}\left(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{y}, \left(\{[L_i]_1, [R_i]_2, [O_i]_1, \Pi_{\mathsf{quad},i}\}_{i=1}^d, \Pi_{\mathsf{lin}}\right)\right)}$:
Verify $\{\Pi_{\mathsf{quad},i}\}_{i=1}^d, \Pi_{\mathsf{lin}}$, and check if:
$[O_d]_1 = \sum y_i [\lambda_i(s)]_1$;
output 1 in this case and 0 otherwise.

Figure 4.2: proof of circuit satisfiability.

to a certain level and they are able to produce an accepting proof, then they must also know the witness of the next level. Thus knowledge of the input propagates through the whole circuit.

The two arguments are put together to produce the full argument for circuit satisfiability, which is summarized in Figure 4.2.

Note. The actual protocol requires two evaluation points $\overline{s}, \overline{s}'$, and so everything in the proof is doubled, that is, $8d$ elements are sent instead of $4d$. This is true also for our construction, although for simplicity of the exposition we have decided to present it for just one point.

Theorem 15. *The protocol of Figure 4.2 is complete, and sound under the $\mathcal{R}$-RSDH assumption (16) and the $\mathcal{LG}_{\mathcal{R},2}$-SMDDH assumption (4).*

In the following sections, we show how to build on top of this proof to improve the communication complexity.

4.4.2 Argument for quadratic equations

As we have seen, the proof for quadratic equations in [86] amounts to sending commitments $[L, O, H]_1, [R]_2$ and proving that they satisfy the equations

$$[L]_1[R]_2 - [O]_1[1]_2 - [H]_1[t]_2 = [0]_T,$$

for a known constant $[t]_2$. This requires sending $O(d)$ group elements, as there is one such equation for each multiplicative level of the circuit. To achieve sublinear complexity, we use the construction of Section 4.3.4 to prove knowledge of a solution of the equations, instead of sending such solution directly. This brings the communication complexity down to $O(\log d)$ group elements.

4.4.3 Argument for linear equations

Let $\boldsymbol{w} = (c_1, \ldots, c_N)$ be the vector of multiplication gate outputs of the circuit, where N is the number of multiplication gates, and let $\boldsymbol{u} = (\boldsymbol{o}, \boldsymbol{\ell}, \boldsymbol{r})$ be the level-wise shrinking commitments to outputs, left inputs and right inputs, respectively. We encode the linear relations between them as $\boldsymbol{u} = \mathbf{M}\boldsymbol{w}$, for a matrix $\mathbf{M}$, of size $3d \times N$, that depends on the circuit and the commitment keys used in the shrinking commitments $\boldsymbol{u}$.

Observe that the vector $\boldsymbol{u}$ is partly known in $\mathbb{G}_1$ and partly in $\mathbb{G}_2$. However, for simplicity of the exposition, let us assume for now that $\boldsymbol{u} = (L_1, \ldots, L_d)$, and so the matrix $\mathbf{M}$ is of size $d \times N$ and is known over $\mathbb{G}_1$. Recall that the QA-NIZK argument of [115] proves membership in the language

$$\mathcal{L}_{[\mathbf{M}]_1} = \{\boldsymbol{u} \in \mathbb{G}_1^d \mid \exists \boldsymbol{w} \in \mathbb{Z}_p^N \text{ such that } \boldsymbol{u} = \mathbf{M}\boldsymbol{w}\},$$

using a proof $[\boldsymbol{\pi}]_1$ that in the most efficient instantiation is just one group element, and the verification is

$$e([\boldsymbol{u}^\top]_1, [\mathbf{K}\mathbf{A}]_2) = e([\boldsymbol{\pi}]_1, [\mathbf{A}]_2).$$

We observe that the left hand side can be seen as an inner product

$$[\boldsymbol{u}]_1 \cdot [\mathbf{K}\overline{\mathbf{A}}]_2 = [\boldsymbol{\pi}\overline{\mathbf{A}}]_T.$$

Therefore, instead of sending directly $[\boldsymbol{u}]_1$ and a proof $[\boldsymbol{\pi}]_1$ that $\boldsymbol{u} = \mathbf{M}\boldsymbol{w}$, we send only $[\boldsymbol{\pi}]_1$, and engage in an inner product argument, as described in Section 4.3.3. As the vectors involved in the inner product argument are of size d, this allows us to prove the linear relations with communication $O(\log d)$.

Note that shrinking commitments to $\boldsymbol{u}, \mathbf{K}\overline{\mathbf{A}}$ are required for the inner product argument. For $\boldsymbol{u}$, we already have a commitment, given at the beginning of the argument for quadratic equations (Section 4.4.2), which we also use here.

Since the full vector $\boldsymbol{u}$ is split between $\mathbb{G}_1$ and $\mathbb{G}_2$, we replace the argument of [115] by the bilateral version of [85], which proves membership in the language

$$\mathcal{L}_{[\mathbf{M}]_1,[\mathbf{N}]_2} = \left\{ (\boldsymbol{u},\boldsymbol{v}) \in \mathbb{G}_1^{d_1} \times \mathbb{G}_2^{d_2} \mid \exists \boldsymbol{w} \in \mathbb{Z}_p^N \text{ such that } \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} = \begin{pmatrix} \mathbf{M} \\ \mathbf{N} \end{pmatrix} \boldsymbol{w} \right\}.$$

Recall that the proof $[\boldsymbol{\pi}]_1, [\boldsymbol{\theta}]_2$ consists of 4 group elements in the most efficient case, and the verification equation is

$$e([\boldsymbol{u}^\top]_1, [\mathbf{K}\overline{\mathbf{A}}]_2) + e([\mathbf{L}\overline{\mathbf{A}}]_1, [\boldsymbol{v}]_2) = e([\boldsymbol{\pi}]_1, [\overline{\mathbf{A}}]_2) + e([\overline{\mathbf{A}}]_1, [\boldsymbol{\theta}]_2).$$

Again, it is easy to reformulate this equation as an inner product

$$[\boldsymbol{u}, \mathbf{L}\overline{\mathbf{A}}]_1 \cdot [\mathbf{K}\overline{\mathbf{A}}, \boldsymbol{v}] = [\boldsymbol{\pi}\overline{\mathbf{A}} + \overline{\mathbf{A}}\boldsymbol{\theta}]_T.$$

We note that actually [86] does not exactly use the QA-NIZK arguments of [85, 115], but a modified version with a more involved key generation step. Nevertheless, the verification equation remains essentially the same, so it can be written as an inner product equation, and thus we still achieve complexity $O(\log d)$.

4.4.4 Circuit satisfiability proof

By putting together the previous sections, we obtain a scheme for proving circuit satisfiability. We provide a high-level description, and refer to the specific sections for more detailed information.

$\underline{\mathsf{Setup}(gk)}$:
Generate the crs from sections 4.4.1, 4.4.2, and 4.4.3.

$\underline{\mathcal{P}(\mathsf{crs}, (\boldsymbol{x},\boldsymbol{y}), (\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}))}$:
For $i = 1, \ldots, d$, compute commitments $[L_i]_1, [R_i]_2, [O_i]_1$ to $\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i$, respectively.
(Quadratic constraints) Use section 4.4.2.
(Linear constraints) Use section 4.4.3.
Output both proofs $\Pi_{\mathsf{quad}}, \Pi_{\mathsf{lin}}$.

$\underline{\mathcal{V}(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{y}, (\Pi_{\mathsf{quad}}, \Pi_{\mathsf{lin}}))}$:
Verify both proofs.

Figure 4.3: new proof of circuit satisfiability.

Theorem 16. *The protocol is complete, and sound under the binding property of the underlying commitment scheme, the $\mathcal{R}$-RSDH assumption (16), and the $\mathcal{LG}_{\mathcal{R},2}$-SMDDH assumption (4).*

Efficiency. There are d commitments L_i, R_i, O_i. For the quadratic equations, we prove the d equations

$$L_i R_i - O_i - H_i t = 0,$$

which requires communication $O(\log d)$. The linear equations are handled by proving the inner product equation

$$[\boldsymbol{u}, \mathbf{L}\overline{\mathbf{A}}]_1 \cdot [\mathbf{K}\overline{\mathbf{A}}, \boldsymbol{v}] = [\boldsymbol{\pi}\overline{\mathbf{A}} + \overline{\mathbf{A}}\boldsymbol{\theta}]_T,$$

where the vectors involved are of size $O(d)$, and thus again this takes communication $O(\log d)$.

4.5 Achieving universal SRS

The construction in the previous section is very efficient, but suffers from the downside of requiring a circuit-dependent SRS. This is due to the use of quasi-adaptive proofs to deal with the linear equations. In this section, we take a different approach for proving linear equations, inspired by recent work on universal and updateable pairing-based zkSNARKs.

The high-level idea is as follows. Satisfiability of a homogeneous linear system can be seen as an orthogonality condition: a solution must be orthogonal to the rows of matrix $\mathbf{Y}$, consisting of the vectors of coefficients of the equations. Our construction is a Sigma protocol, described in Section 4.5.1 in which the prover first commits to the solution $\boldsymbol{c}$. Next, it is given a challenge $\boldsymbol{d}$ from the row space of $\mathbf{Y}$, and finally proves that $\boldsymbol{c} \cdot \boldsymbol{d} = 0$. With overwhelming probability, $\boldsymbol{c}$ will only be orthogonal to $\boldsymbol{d}$ if it is orthogonal to all the rows of d.

A naive verifier that samples $\boldsymbol{d}$ uniformly from the row space of $\mathbf{Y}$ requires linear time. For the sake of clarity, we first present the Sigma protocol with challenge sampled uniformly from the row space of $\mathbf{Y}$, and in the next section we will discuss the modifications required to avoid linear complexity.

4.5.1 An interactive Argument for linear equations

We design a two-round argument for linear equations that will be used to prove, for each level i separately, that there is an opening $\boldsymbol{a}_i$ of a commitment $[L_i]_1$ that

is correctly defined as a function of the input and the outputs of multiplicative gates of previous levels, which is a vector $\boldsymbol{c} = (\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{i-1})$ committed as

$$[\boldsymbol{O}]_1 = [O_0, \ldots, O_{i-1}]_1.$$

What we would like to prove is that, *if* $[\boldsymbol{O}]_1$ opens to $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{i-1})$, then $[L]_1$ opens to $\boldsymbol{a}_i$, which is in the correct linear relation to $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{i-1})$. Thus, in this section we construct an argument for proving satisfiabiliy of equations of the form

$$\boldsymbol{a} = \begin{pmatrix}\mathbf{M}_1| & \ldots & |\mathbf{M}_k\end{pmatrix} \begin{pmatrix}\boldsymbol{c}_0 \\ \vdots \\ \boldsymbol{c}_k\end{pmatrix}, \qquad k \in \mathbb{N}.$$

Note that $\boldsymbol{c}_0$ corresponds to the input of the circuit, which is public.

We observe that a witness $(\boldsymbol{a}, \boldsymbol{c}) \in \mathbb{Z}_p^n \times (\mathbb{Z}_p)^{n(k+1)}$ satisfies all the linear equations if and only if $(\boldsymbol{a}, \boldsymbol{c})$ is orthogonal to the row space defined by the matrix

$$\mathbf{Y} = \begin{pmatrix}-\mathbf{I}_n| & \mathbf{M}\end{pmatrix} = \begin{pmatrix}-\mathbf{I}_n| & \mathbf{M}_0| & \ldots & |\mathbf{M}_k\end{pmatrix}.$$

We specify the languages

$$\mathcal{L}_{\mathsf{yes}} = \{(\boldsymbol{c}, [\boldsymbol{O}, L]_1) \mid O_j = \boldsymbol{\lambda}(x) \cdot \boldsymbol{c}_j \wedge L = \boldsymbol{\lambda}(x) \cdot (\mathbf{M}\boldsymbol{c})\},$$
$$\mathcal{L}_{\mathsf{no}} = \{(\boldsymbol{c}, [\boldsymbol{O}, L]_1) \mid O_j = \boldsymbol{\lambda}(x) \cdot \boldsymbol{c}_j \wedge L \neq \boldsymbol{\lambda}(x) \cdot (\mathbf{M}\boldsymbol{c})\}.$$

We need to prove that our argument is complete for $\mathcal{L}_{\mathsf{yes}}$ and sound for $\mathcal{L}_{\mathsf{no}}$. Note that we do not claim or prove anything for tuples which are not in $\mathcal{L}_{\mathsf{yes}}$ or $\mathcal{L}_{\mathsf{no}}$. González and Ràfols use a QA-NIZK argument (with a different security analysis) to construct a NI argument for these same languages with the same security guarantees, so our new argument can just replace theirs, resulting in a universal and updateable argument.

The SRS generation is split in two algorithms: KeyGenU, which produces the universal SRS, with powers of x and KeyGenD, which produces the circuit-dependent SRS from public information and the output of the previous algorithm. Updateability follows directly from the structure of the SRS ([93]).

The commitments are part of the statement. When using this argument to prove circuit satisfiability, an additional first message from prover to verifier in which the commitments are sent (or committed to, when communication is sublinear in d), is necessary.

- KeyGenU($gk, \mathcal{R} \subseteq \mathbb{Z}_p, v \in \mathbb{Z}_p$): sample $x \leftarrow \mathbb{Z}_p$ and output

$$\mathsf{crs}_U = \left\{\mathcal{R}, v, \{[x^i]_{1,2}\}_{i=0}^{m-1}, [x^m]_2\right\}.$$

- KeyGenD($\mathbf{M}, \mathsf{crs}_U$): let $\{\lambda_i(X)\}_{i=1}^n$ be the Lagrange interpolation polynomials with interpolation set $\mathcal{R}$. Define the vectors of polynomials

$$\boldsymbol{\lambda}(X) = (\lambda_1(X), \ldots, \lambda_n(X)), \qquad \tilde{\boldsymbol{\lambda}}(X) = \boldsymbol{\lambda}(X) \circ \left(\frac{1}{\lambda_1(v)}, \ldots, \frac{1}{\lambda_n(v)}\right).$$

  We then define

$$\boldsymbol{D}_L(X) = -\mathbf{I}_n \tilde{\boldsymbol{\lambda}}(X), \qquad \boldsymbol{D}_{O,j}(X) = \mathbf{M}_j \tilde{\boldsymbol{\lambda}}(X),$$

  that is,

$$(\boldsymbol{D}_L(X), \boldsymbol{D}_{O,0}(X), \ldots, \boldsymbol{D}_{O,k}(X)) = \mathbf{Y}(\tilde{\boldsymbol{\lambda}}(X), \ldots, \tilde{\boldsymbol{\lambda}}(X)).$$

  Compute $[D_{O,j,i}, D_{L,i}] = [D_{O,j,i}(x), D_{L,i}(x)]$, and output

$$\mathsf{crs}_D = \left\{ \left\{ [D_{L,i}]_2, \{[D_{O,j,i}]_2\}_{j=0}^k \right\}_{i=1}^n \right\}.$$

The argument proceeds as follows:

- $\mathcal{V} \to \mathcal{P}$: choose $\boldsymbol{\alpha} \leftarrow \mathbb{Z}_p^n$ and send to the prover.
- $\mathcal{P} \to \mathcal{V}$: compute $\boldsymbol{d} = \boldsymbol{\alpha}^\top \mathbf{Y}$, parse it as $\boldsymbol{d} = (-\boldsymbol{\alpha}, \boldsymbol{d}_0, \ldots, \boldsymbol{d}_k)$, and define the polynomials

$$D_L(X) = \boldsymbol{\alpha}^\top \cdot \boldsymbol{D}_L(X), \qquad D_{O,j}(X) = \boldsymbol{\alpha}^\top \cdot \boldsymbol{D}_{O,j}(X),$$

$$P(X) = \left(-\boldsymbol{a} \circ \boldsymbol{\alpha} + \sum_{j=0}^k \boldsymbol{c}_j \circ \boldsymbol{d}_j \right) \cdot \tilde{\boldsymbol{\lambda}}(X), \qquad Q(X) = \frac{P(X)}{X - v},$$

  and $H(X)$ such that

$$L(X)D_L(X) + \sum_{j=0}^k O_j(X) D_{O,j}(X) = P(X) + H(X)t(X).$$

  Compute $[Q]_1 = [Q(x)]_1, [P]_1 = [P(x)]_1$ and $[H]_1 = [H(x)]_1$ and send these to the verifier.

- $\mathcal{V}$: Define

$$D_L(x) = \boldsymbol{\alpha}^\top \cdot \boldsymbol{D}_L(x), \qquad D_{O,j}(x) = \boldsymbol{\alpha}^\top \cdot \boldsymbol{D}_{O,j}(x),$$

and accept iff the following equations hold:

$$[L]_1[D_L]_2 + \sum_{j=0}^{k} O_j(X) D_{O,j}(X) = [Q]_1[x-v]_2 + [H]_1[t(x)]_2,$$
$$[P]_1[1]_2 = [Q]_1[x-v]_2.$$

Theorem 17. *The scheme above is complete, and promise-sound under the hardness of Assumption 17.*

Proof. The only non-trivial steps in completeness are ensuring that the polynomial $P(X)$ is divisible by $X-v$, and that $L(X)D_L(X)+\sum_{j=0}^{k} O_j(X)D_{O,j}(X)-P(X)$ is divisible by $t(X)$, assuming that the prover is honest. Observe that $\tilde{\boldsymbol{\lambda}}(v) = (1, \ldots, 1)$, and so

$$P(v) = \left(-\boldsymbol{a} \circ \boldsymbol{\alpha} + \sum_{i=0}^{k} \boldsymbol{c}_i \circ \boldsymbol{d}_i\right) \cdot \tilde{\boldsymbol{\lambda}}(v) =$$
$$= \sum_{j=1}^{n}(-a_j\alpha_j + \sum_{i=1}^{k} c_{i,j}d_{i,j}) = \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{c} \end{pmatrix} \cdot \begin{pmatrix} -\boldsymbol{\alpha} \\ \boldsymbol{d}_0 \\ \vdots \\ \boldsymbol{d}_k \end{pmatrix} = 0,$$

where the last equality follows because $\boldsymbol{d}$ is a linear combination of the rows of $\mathbf{Y}$, so any valid witness must be orthogonal to all the rows of $\mathbf{Y}$. Thus, $P(v) = 0$ and $(X - v) \mid P(X)$.

For the second condition, note that

$$L(r_i) = a_i, \quad O_j(r_i) = c_{j,i}, \quad D_L(r_i) = -\frac{\alpha_i}{\lambda_i(v)}, \quad , D_{O,j}(r_i) = \frac{1}{\lambda_i(v)}\boldsymbol{\alpha}^\top \boldsymbol{m}_{j,i},$$

where $\boldsymbol{m}_i$ is the ith column of $\mathbf{M}_j$. Then

$$L(r_i)D_L(r_i) + \sum_{j=0}^{k} O_j(r_i)D_O(r_i) = \frac{1}{\lambda_i(v)}\left(-a_i\alpha_i + \sum_{j=0}^{k} c_{j,i}\boldsymbol{\alpha}^\top \boldsymbol{m}_{j,i}\right),$$

$$P(r_i) = \left(-\boldsymbol{a}\circ\boldsymbol{\alpha} + \sum_{j=0}^{k}\boldsymbol{c}_j\circ\boldsymbol{d}_j\right)\cdot\tilde{\boldsymbol{\lambda}}(r_i) = \qquad .$$

$$= \frac{1}{\lambda_i(v)}\left(-a_i\alpha_i + \sum_{j=0}^{k} c_{j,i}d_{j,i}\right) = \frac{1}{\lambda_i(v)}\left(-a_i\alpha_i + \sum_{j=0}^{k} c_{j,i}\alpha^\top \boldsymbol{m}_{j,i}.\right)$$

Therefore, $(X-r_i) \mid \left(L(X)D_L(X) + \sum_{j=0}^{k} O_j(X)D_{O,j}(X) - P(X)\right)$ for any $i = 1, \ldots, n$.

For soundness, assume that an adversary is able to produce an accepting proof for a statement in $\mathcal{L}_{\mathsf{no}}$. That is, $[\boldsymbol{O}]_1$ is the legitimate value associated to $\boldsymbol{c}$, but the adversary lies in $[L]_1$. Since the proof is accepting, we have

$$LD_L + \sum_{j=0}^{k} O_j D_{O,j} = Q(x-v) + Ht(x),$$
$$P = Q(x-v).$$

With $\boldsymbol{c}$ (which is part of the statement)[6], we compute the legitimate values $[\tilde{\boldsymbol{O}}, \tilde{L}, \tilde{P}, \tilde{Q}, \tilde{H}]_1$, where $\boldsymbol{O} = \tilde{\boldsymbol{O}}$ but $L \neq \tilde{L}$, and

$$\tilde{L}D_L + \sum_{j=0}^{k} O_j D_{O,j} = \tilde{Q}(x-v) + \tilde{H}t(x),$$
$$\tilde{P} = \tilde{Q}(x-v),$$

and by subtracting the corresponding verification equations we have that

$$(L-\tilde{L})D_L = (Q-\tilde{Q})(x-v) + (H-\tilde{H})t(x),$$
$$P - \tilde{P} = (Q-\tilde{Q})(x-v),$$

[6]When using this argument for proving correct circuit evaluation, $\boldsymbol{c}$ is derived from the knowledge of the input and the knowledge transfer property at previous levels.

We now recall that

$$D_L = -\boldsymbol{\alpha} \cdot \tilde{\boldsymbol{\lambda}}(x).$$

By rewinding the proof n times, we obtain answers for challenge rows $\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(n)}$, and we have the system of linear equations

$$\begin{pmatrix} D_L^{(1)} \\ \vdots \\ D_L^{(n)} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{\alpha}^{(1)} \\ \vdots \\ -\boldsymbol{\alpha}^{(n)} \end{pmatrix} \tilde{\boldsymbol{\lambda}}(x) = \underbrace{\begin{pmatrix} -\boldsymbol{\alpha}^{(1)} \\ \vdots \\ -\boldsymbol{\alpha}^{(n)} \end{pmatrix} \begin{pmatrix} \frac{1}{\lambda_1(v)} & & \\ & \ddots & \\ & & \frac{1}{\lambda_n(v)} \end{pmatrix}}_{=:\mathbf{D}} \boldsymbol{\lambda}(x).$$

We observe that the matrix $\mathbf{D}$ is invertible, since the first factor is composed of uniformly random vectors, and the second is a diagonal matrix with nonzero elements in the diagonal. Moreover, both are known in $\mathbb{Z}_p$, as they depend only on the challenges and public information. By inverting $\mathbf{D}$, we obtain answers $Q^{(i)}, H^{(i)}, P^(i)$ such that

$$\begin{aligned} (L - \tilde{L})\lambda_i(x) &= (Q^{(i)} - \tilde{Q}^{(i)})(x - v) + (H^{(i)} - \tilde{H}^{(i)})t(x), \\ (P^{(i)} - \tilde{P}^{(i)}) &= (Q^{(i)} - \tilde{Q}^{(i)})(x - v), \end{aligned}$$

where $\tilde{Q}^{(i)}, \tilde{H}^{(i)}, \tilde{P}^{(i)}$ are the honest answers to the challenges $\boldsymbol{d}^{(i)} = \boldsymbol{e}_i^\top \mathbf{Y}$, where $\boldsymbol{e}_i$ is the ith vector of the canonical basis. This breaks Assumption 17. □

The scheme, as described above, covers the equations that relate each left wire to the previous output wires. We can do exactly the same for the right wires. However, notice that for the scheme to be secure, we need to commit to the right wires in $\mathbb{G}_1$:

$$[R]_1 = \boldsymbol{b} \cdot [\boldsymbol{\lambda}(x)]_1.$$

This clashes with the global protocol, since for quadratic equations we commit to the right wires in the same way, but in $\mathbb{G}_2$. However, if zero-knowledge is not required, and thus the commitments are not hiding, then the verifier can easily check that both are commitments to the same value by verifying the equation

$$e([R]_1, [1]_2) = e([1]_1, [R]_2).$$

4.5.2 Sublinear verification complexity through efficient challenge sampling

The bottleneck of the previous protocol, both from the point of view of verifier and communication complexity, is the fact that $\boldsymbol{d}$ is sampled uniformly from $\mathbf{Y}$, which requires $O(n)$ work from the verifier. When used in the argument for correct circuit evaluation, at each level this means that the proof at level i requires the verifier to do work that is linear in the number of wires of at level i. In total, this would result in a verifier that is linear in the number of multiplicative gates. For our argument, it is necessary for the verifier to compute:

$$[D_L]_2 = \boldsymbol{\alpha}^\top(-\mathbf{I}_n) \cdot [\tilde{\boldsymbol{\lambda}}(x)]_2, \qquad [DO, j]_2 = \boldsymbol{\alpha}^\top \mathbf{M}_j \cdot [\tilde{\boldsymbol{\lambda}}(x)]_2. \tag{4.5}$$

for some vector $\boldsymbol{\alpha} \in \mathbb{Z}_p^n$ that is 'sufficiently' random. The typical strategy [24] to reduce communication complexity is to choose $\boldsymbol{\alpha} = (1, \alpha, \ldots, \alpha^n)$. This reduces communication but not verifier work. For this reason, we resort to the literature of universal and updateable zkSNARKS, where this problem has also been dealt with, to propose various alternatives.

- **Sparse Matrices.** When the matrix $\mathbf{M}_j$ is sparse, we can use the techniques described in Marlin [39], and refined in Lunar [32], to sample such a vector for $\boldsymbol{\alpha} = (\lambda_1(y), \ldots, \lambda_n(y))$. The verifier sends $y \leftarrow \mathbb{Z}_p$, and the prover sends for each level i, $[D_L]_2$ and $[D_{O,j}]_2$, $j = 0, \ldots, i-1$ for this value of α, together with a proof of correct evaluation of $[D_L]_2$ and $[D_{O,j}]_2$. The proof uses the fact that the matrices $\mathbf{M}_{i,j}$ are sparse and is constant size for each i, j (thus the verifier is $O(s + d^2)$, where s is the size of the public input).
- **Amortized setting.** It is also possible to use techniques from Sonic [128] to get better amortized complexity. The idea is that we can define for each level i a bivariate polynomials $\boldsymbol{P}_i(X, Y)$ such that $(D_L, D_{O,0}, \ldots, D_{O,i-1}) = \boldsymbol{P}_i(x, y)$, that is, these commitments are the evaluation at the secret point x in the SRS and the point $\boldsymbol{y}$ chosen by the verifier. Many values $\boldsymbol{P}_i(x, y_k)$ corresponding to different verifier challenges $y_1, \ldots, y_\ell$ can be verified with a cost dominated by the cost of evaluating $\boldsymbol{P}_i(X, Y)$ at a single point z_1, z_2.

4.5.3 Aggregation of multiplicative levels

So far, we have focused on an argument for proving that linear relations hold between the outputs of a level and the inputs of the next, given commitments

$[\boldsymbol{O}, L]_1$ to these. However, our goal is to end up with a protocol whose communication complexity is sublinear in the multiplicative depth d of the circuit, and thus we cannot use this level-wise approach directly.

The trick to achieve sublinear complexity is the same that we already used in Section 4.4.3. Consider first the naive version of the protocol in which we prove all levels individually, in parallel. After compiling it into a non-interactive protocol, using the Fiat–Shamir transformation, we clearly have a protocol whose complexity depends linearly on d. We observe that the verification step of this protocol consists of checking that

$$[L_i]_1[D_{L,i}]_2 + [\boldsymbol{O}_i]_1[\boldsymbol{D}_{\boldsymbol{O},i}]_2 = [Q_i]_1[x - v]_2 + [H_i]_1[t(x)]_2,$$
$$[P_i]_1[1]_2 = [Q_i]_1[x - v]_2,$$

for each level $i = 1, \ldots, d$. Then, we can reformulate the problem as proving knowledge of $[L_1, \ldots, L_d, O_1, \ldots, O_d]_1$, such that they verify the equations above.

To solve this, we can again make use of Section 4.3.4, which allows us to prove knowledge of a solution to these equations with communication complexity $O(\log d)$.

4.5.4 Circuit satisfiability proof with universal CRS

Again, we put together the proofs for quadratic and linear equations, this time using the new proof of satisfiability of linear equations described above.

$\underline{\mathsf{Setup}(gk)}$:
Generate the crs from sections 4.4.1, 4.4.2, and 4.5.3.

$\underline{\mathcal{P}(\mathsf{crs}, (\boldsymbol{x}, \boldsymbol{y}), (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}))}$:
For $i = 1, \ldots, d$, compute commitments $[L_i]_1, [R_i]_2, [O_i]_1$ to $\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i$, respectively.
(Quadratic constraints) Use section 4.4.2.
(Linear constraints) Use section 4.5.3.
Output both proofs $\Pi_{\mathsf{quad}}, \Pi_{\mathsf{lin}}$.

$\underline{\mathcal{V}(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{y}, (\Pi_{\mathsf{quad}}, \Pi_{\mathsf{lin}}))}$:
Verify both proofs.

Figure 4.4: new proof of circuit satisfiability (with universal SRS).

Theorem 18. *The protocol is complete, and sound under the binding property of the underlying commitment scheme, the $\mathcal{R}$-RSDH assumption (16), and assumption 17.*

4.6 Zero-knowledge

In this section, we describe how to add the zero-knowledge property to the proofs of correct circuit evaluation presented above. For clarity, we first present the most elementary approach, and afterwards we discuss how to modify it for improved efficiency.

Let C be a circuit with multiplicative depth d and secret input $\boldsymbol{x} \in \mathbb{Z}_p^n$, and let $[\boldsymbol{ck}] \leftarrow \mathbb{G}^2$ be a commitment key for the Lagrangian Pedersen commitment scheme. In parallel with the rest of the protocol, we also perform the following Sigma protocol:

- $\mathcal{P} \to \mathcal{V}$: choose a blinding term $x_0 \leftarrow \mathbb{Z}_p$. For $i = 0, \ldots, n$, choose randomness $r_i \leftarrow \mathbb{Z}_p^n$ and commit to the input x_i:

$$[\boldsymbol{u}_i] = \mathsf{Com}_{\boldsymbol{ck}}(x_i, r_i) = (x_i, r_i) \cdot [\boldsymbol{ck}],$$

  and send $[u] = [\boldsymbol{u}_0, \ldots, \boldsymbol{u}_n] \in \mathbb{G}^{2n}$ to the verifier.

- $\mathcal{V} \to \mathcal{P}$: choose $z \leftarrow \mathbb{Z}_p$ and send it to the prover.

- $\mathcal{P} \to \mathcal{V}$: let $\boldsymbol{z} = (1, z, z^2, \ldots, z^n)$, $\boldsymbol{x} = (x_0, \ldots, x_n)$, and $\boldsymbol{r} = (r_0, , \ldots, r_n)$. Compute

$$x_z = \boldsymbol{x} \cdot \boldsymbol{z}, \qquad r_z = \boldsymbol{r} \cdot \boldsymbol{z}, \tag{4.6}$$

  and send them to the verifier.

- $\mathcal{V}$: the verifier accepts iff

$$\sum_{i=0}^{n} z^i [\boldsymbol{u}_i] = \mathsf{Com}_{\boldsymbol{ck}}(x_z, r_z).$$

Theorem 19. *The scheme above is complete, zero-knowledge, and knowledge sound under the binding property of the commitment scheme.*

Proof. Completeness follows from the homomorphic property of the commitment scheme. To argue zero-knowledge, we show how to indistinguishably simulate proofs: choose $z, x_z, r_z \leftarrow \mathbb{Z}_p$, choose $[\boldsymbol{u}_1], \ldots [\boldsymbol{u}_n] \leftarrow \mathbb{G}^2$, and set

$$[\boldsymbol{u}_0] = \mathsf{Com}_{\boldsymbol{ck}}(x_z, r_z) - \sum_{i=1}^{n} z^i [\boldsymbol{u}_i].$$

In a real execution of the protocol, the element x_z are uniformly random, due to the blinding term x_0, and r_z, z are clearly uniformly random. Due to the perfectly hiding property of the commitment scheme, the $[\boldsymbol{u}_i]$ are uniformly random, conditioned by the verification equation. Thus, the algorithm described perfectly simulates honest proofs.

Finally, we argue that the scheme is knowledge sound by describing a witness extractor. By rewinding the proof $n+1$ times for the same commitments but different challenges $z^{(i)}$, we obtain answers $x_z^{(i)}, r_z^{(i)}$ such that

$$\begin{pmatrix} x_z^{(0)} \\ \vdots \\ x_z^{(n)} \end{pmatrix} = \begin{pmatrix} 1 & z^{(0)} & \ldots & {z^{(0)}}^n \\ \vdots & \vdots & & \vdots \\ 1 & z^{(n)} & \ldots & {z^{(n)}}^n \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_n \end{pmatrix}$$

Since this matrix is a Vandermonde matrix spanned by independent $z^{(i)}$, with overwhelming probability it is invertible, and thus we can extract $\boldsymbol{x}$.

□

Equipped with this Sigma protocol, it is very easy to add the zero-knowledge property to our constructions, simply by running this in parallel with the main scheme. The commitments to the secret inputs are perfectly hiding, so they do not leak any information about them. On the other hand, the knowledge soundness ensures that during security proofs we can extract the whole input and proceed as if it was public.

With the addition of zero-knowledge as above, the communication complexity incurs an overhead of $O(n)$, where n is the size of the secret input. This is because we commit individually to each element of the secret input, but actually we can do much better than this. Reproducing the techniques of [88] or [91], we can have shrinking commitments to the secret input that can be opened through rewinding in the security proof. These yield a cost of $O(n^{\frac{1}{2}})$ and a 5-round protocol, or $O(n^{\frac{1}{3}})$ and a 7-round protocol, respectively. Furthermore, we can also use the inner product arguments of knowledge of [24,28] for equation

(4.6): we send a constant-size commitment to $\boldsymbol{x}$, and use these arguments to prove knowledge of $\boldsymbol{x}$ satisfying (4.6). This reduces the overhead to $O(\log n)$, using a $O(\log n)$-round protocol.

Part II

Isogeny-based cryptosystems

Chapter 5

Isogeny signatures

This chapter is based on the paper 'Identification protocols and signature schemes based on supersingular isogeny problems' [75], which is a joint work with Steven Galbraith and Christophe Petit. It was first published at Asiacrypt 2017, where it received the best paper award. An extended version was published in Journal of Cryptology [76].

A recent research area is cryptosystems whose security is based on the difficulty of finding a path in the isogeny graph of supersingular elliptic curves [38, 47, 66, 103, 104]. As seen in section 2.4.5, the only known quantum algorithm for these problems, due to Biasse, Jao and Sankar [19], has exponential complexity. Hence, additional motivation for the study of these cryptosystems is that they are possibly suitable for post-quantum cryptography. The latter has received greater focus due to the recent standardization process initiated by NIST.[1]

Some of the first constructions in supersingular isogeny cryptography include the collision-resistant hash function of Charles, Goren and Lauter [38], the key exchange protocol of Jao and De Feo [103], and the public key encryption scheme and Sigma protocol of De Feo, Jao and Plût [66]. Focusing on signatures, Jao and Soukharev [104] presented an undeniable signature, and Xi, Tian and Wang [166] presented a designated verifier signature.

[1] U.S. Department of Commerce, National Institute of Standards and Technology, Post-Quantum Cryptography project, 2016. Available at https://csrc.nist.gov/projects/post-quantum-cryptography, last retrieved September 13th, 2019.

In this chapter, we present two public key signature schemes whose security relies on computational problems related to finding a path in the isogeny graph of supersingular elliptic curves.

Our techniques. The first scheme is obtained relatively simply from the De Feo–Jao–Plût [66] Sigma protocol by using the Fiat–Shamir transformation to turn it into a non-interactive signature scheme. We also use a variant of the Fiat–Shamir transformation due to Unruh to obtain a post-quantum signature scheme. Essentially the same signature scheme was independently published by Yoo, Azarderakhsh, Jalali, Jao and Soukharev [167], but our version has improved signature size. This scheme has the advantage of being simple to describe, at least to a reader who is familiar with the previous work in the subject, and easy to implement. On the other hand, it inherits the disadvantages of [66], in particular it relies on a non-standard isogeny problem using small isogeny degrees, reveals auxiliary points, and uses special primes.

The fastest classical attack on the first scheme has heuristic running time of $\tilde{O}(p^{1/4})$ bit operations, and the fastest quantum attack (see Section 5.1 of [66]) has running time of $\tilde{O}(p^{1/6})$. Galbraith, Petit, Shani and Ti [74] and Petit [139] showed that revealing auxiliary points may be dangerous in certain contexts. It is therefore highly advisable to build cryptographic schemes based on the most general, standard and potentially hardest isogeny problems.

Our second scheme uses completely different ideas and relies on the difficulty of a more standard computational problem, namely the endomorphism ring problem (assumption 10, or equivalently assumptions 7 or 8). This computational problem has heuristic classical complexity of $\tilde{O}(p^{1/2})$ bit operations, and quantum complexity $\tilde{O}(p^{1/4})$. In particular, the second scheme does not involve sending auxiliary points and so avoids the attacks of [74, 139]. The Sigma protocol that is very similar to the proof of graph isomorphism. One obtains a signature scheme by applying the Fiat–Shamir or Unruh transformations.

We briefly sketch the main ideas behind our second scheme. The public key is a pair of elliptic curves (E_0, E_1) and the private key is an isogeny $\varphi : E_0 \to E_1$. To interactively prove knowledge of φ, one chooses a random isogeny $\psi : E_1 \to E_2$ and sends E_2 to the verifier. The verifier sends a bit b. If $b = 0$ the prover reveals ψ. If $b = 1$ the prover reveals an isogeny $\eta : E_0 \to E_2$. In either case, the verifier checks that the response is correct. The interaction is repeated a number of times until the verifier is convinced that the prover knows an isogeny from E_0 to E_1.

However, the subtlety is that we cannot just set $\eta = \psi \circ \varphi$, as then E_1

would appear on the path in the graph from E_0 to E_2 and so we would have leaked the private key. The crucial idea is to use the algorithm of Kohel–Lauter–Petit–Tignol [118] to produce a "pseudo-canonical" isogeny $\eta : E_0 \to E_2$ that is independent of φ. The algorithm of Kohel–Lauter–Petit–Tignol is based on the theory of quaternion algebras.

Organization. In section 5.1 we give a result on mixing in the isogeny graph, and discuss how to efficiently represent isogeny data. Sections 5.2 and 5.3 describe our two signature schemes, respectively, and section 5.4 concludes the chapter. In a first reading to get the intuition of our schemes without all implementation details, one can safely skip some parts, namely sections 5.1, 5.3.3 and 5.3.4.

5.1 Algorithmic considerations

5.1.1 Random walks in isogeny graphs

Let p, ℓ be two different primes, and let j be a supersingular invariant in characteristic p. We define a *random step* of degree ℓ from j as the process of randomly and uniformly choosing a neighbour of j in $\mathsf{G}_\ell(\mathbb{F}_{p^2})$, and returning that vertex. For a composite degree $n = \prod_i \ell_i$, we define a *random walk* of degree n from j_0 as a sequence of j-invariants j_i such that j_i is a random step of degree ℓ_i from j_{i-1}. We do not require the primes ℓ_i to be distinct.

The output of random walks in expander graphs converges quickly to a uniform distribution. In our signature scheme we will be using random walks of *B-powersmooth* degree n, namely $n = \prod_i \ell_i^{e_i}$, with all prime powers $\ell_i^{e_i} \leq B$, with B as small as possible. To analyse the output distribution of these walks we will use a generalization[2] of classical random walk theorems [100].

We first make a few observations about the relation between random walks and adjacency matrices. Let $\mathbf{A}$ be the adjacency matrix of an undirected k-regular graph, with eigenvalues $\lambda_1 = k \geq \lambda_2 \geq \cdots \geq \lambda_r$. We define the *normalized adjacency matrix* as $\overline{\mathbf{A}} = \frac{1}{k}\mathbf{A}$. Let J be a random variable over the set of vertices. Let $\boldsymbol{p} \in [0,1]^r$ be a vector that, on entry i, contains the probability of J being the ith vertex (so the sum of all the entries is 1).

– $\overline{\mathbf{A}}$ has eigenvalues $\overline{\lambda_i} = \frac{\lambda_i}{k}$, and the same eigenvectors as $\mathbf{A}$.

[2] Random walk theorems are usually stated for a single graph whereas our walks will switch from one graph to another, all with the same vertex set but different edges.

- The matrices $\mathbf{A}$ and $\overline{\mathbf{A}}$ have orthogonal diagonalization, since they are symmetric due to the graph being undirected.
- Given the vector of probabilities $\boldsymbol{p}$, the probabilities after a random step in the graph are given by the vector $\overline{\mathbf{A}}\boldsymbol{p}$.
- Given the uniform vector $\boldsymbol{u} = (\frac{1}{r}, \ldots, \frac{1}{r})$, we have $\overline{\mathbf{A}}\boldsymbol{u} = \boldsymbol{u}$. Therefore, $\boldsymbol{u}$ is an eigenvector for the eigenvalue 1.

Theorem 20 (Random walk theorem)**.** *Let p be a prime number, and let j_0 be a supersingular invariant in characteristic p. Let J be a random variable giving the final j-invariant reached by a random walk in $\mathsf{G}_\ell(\overline{\mathbb{F}}_{p^2})$ of degree $n = \prod_i \ell_i^{e_i}$, starting from j_0. Let $N = \#V$. Then for every j-invariant $\tilde{j}$ we have*

$$\left| \mathsf{Pr}[\mathsf{J} = \tilde{j}] - \frac{1}{N} \right| \leq \prod_i \left(\frac{2\sqrt{\ell_i}}{\ell_i + 1} \right)^{e_i}.$$

Proof. Let $\boldsymbol{p}_t$ be the probability vector of J after the first t steps in the random walk. In particular, since we are starting from a fixed point, we have that $\boldsymbol{p}_0$ has 1 in the entry corresponding to j_0 and 0 everywhere else. We want to prove that $\boldsymbol{p}_t$ quickly converges to $\boldsymbol{u}$, so we are interested in bounding the 2-norm of $\boldsymbol{p}_t - \boldsymbol{u}$, which we denote by $|\boldsymbol{p}_t - \boldsymbol{u}|$.

For any probability vector $\boldsymbol{p}$, it is easy to see that $(\boldsymbol{p} - \boldsymbol{u}) \perp \boldsymbol{u}$. Since the graph is Ramanujan, in particular $\lambda_1 > \lambda_2$, and thus the first eigenspace is spanned by $\boldsymbol{u}$ only. Then, for $t \geq 1$,

$$\boldsymbol{p}_{t-1} - \boldsymbol{u} = \sum_{i=2}^{N} \mu_i \boldsymbol{v}_i,$$

where $\boldsymbol{v}_i$ is an eigenvalue corresponding to λ_i, for some coefficients μ_i. Thus

$$|\boldsymbol{p}_t - \boldsymbol{u}| = \left|\overline{\mathbf{A}}\boldsymbol{p}_{t-1} - \boldsymbol{u}\right| = \left|\overline{\mathbf{A}}(\boldsymbol{p}_{t-1} - \boldsymbol{u})\right| = \left|\sum_{i=2}^{N} \mu_i \overline{\mathbf{A}}\boldsymbol{v}_i\right| = \left|\sum_{i=2}^{N} \mu_i \overline{\lambda_i}\boldsymbol{v}_i\right| =$$

$$= \left(\sum_{i=2}^{N} \mu_i \overline{\lambda_i} |\boldsymbol{v}_i|^2\right)^{\frac{1}{2}} \leq \left(\sum_{i=2}^{N} \mu_i \overline{\lambda_2} |\boldsymbol{v}_i|^2\right)^{\frac{1}{2}} = \overline{\lambda_2} \left(\sum_{i=2}^{N} \mu_i |\boldsymbol{v}_i|^2\right)^{\frac{1}{2}} = \overline{\lambda_2} |\boldsymbol{p}_t - \boldsymbol{u}|,$$

where we have used that $\overline{\mathbf{A}}\boldsymbol{u} = \boldsymbol{u}$ and that $\lambda_2 \geq \lambda_i$ for all $i = 2, \ldots, N$. Iterating on this process, we obtain

$$|\boldsymbol{p}_t - \boldsymbol{u}| \leq \prod_i \overline{\lambda_{2,i}}^{e_i} |\boldsymbol{p}_0 - \boldsymbol{u}| \leq \prod_i \overline{\lambda_{2,i}}^{e_i},$$

since

$$|\boldsymbol{p}_0 - \boldsymbol{u}|^2 = \left(1 - \frac{1}{N}\right)^2 + (N-1)\left(\frac{1}{N}\right)^2 \leq 1 - \frac{2}{N} + \frac{1}{N^2} + \frac{1}{N} \leq 1.$$

Therefore,

$$\left|\mathsf{Pr}[\mathsf{J} = j] - \frac{1}{N}\right| \leq |\boldsymbol{p}_t - \boldsymbol{u}|_\infty \leq |\boldsymbol{p}_t - \boldsymbol{u}| \leq \prod_i \left(\frac{\lambda_{2,i}}{\ell_i + 1}\right)^{e_i} \leq \prod_i \left(\frac{2\sqrt{\ell_i}}{\ell_i + 1}\right)^{e_i},$$

using the bound on $\lambda_{2,i}$ given by the Ramanujan property.

□

Armed with this theorem, it is now easy to prove that random walks in our graph achieve very fast mixing, i.e. that the distribution of the output of relatively short walks is statistically indistinguishable from random. More precisely, we want the right-hand term to be smaller than $1/(p^{1+\varepsilon})$ for an arbitrary positive constant ε, and at the same time we will want the powersmooth bound B to be as small as possible. The following lemma shows that taking $B \approx 2(1+\epsilon)\log p$ suffices asymptotically.

Lemma 21. *Let $\epsilon > 0$. There is a function $c_p = c(p)$ such that $\lim_{p\to\infty} c_p = 2(1+\epsilon)$, and, for each p,*

$$\prod_{\substack{\ell_i \text{ prime} \\ e_i := \max\{e \in \mathbb{N} \mid \ell_i^e < c_p \log p\}}} \left(\frac{\ell_i + 1}{2\sqrt{\ell_i}}\right)^{e_i} > p^{1+\epsilon}.$$

Proof. Let B be an integer. We have

$$\prod_{\substack{\ell_i \text{ prime} \\ e_i := \max\{e \in \mathbb{N} \mid \ell_i^e < B\}}} \left(\frac{\ell_i + 1}{2\sqrt{\ell_i}}\right)^{e_i} > \prod_{\substack{\ell_i < B \\ \ell_i \text{ prime}}} \left(\frac{\ell_i + 1}{2\sqrt{\ell_i}}\right) > \prod_{\substack{\ell_i < B \\ \ell_i \text{ prime}}} \left(\frac{\sqrt{\ell_i}}{2}\right).$$

Taking logarithms, using the prime number theorem and replacing the sum by an integral we have

$$\log\left(\prod_{\substack{\ell_i < B \\ \ell_i \text{ prime}}} \left(\frac{\sqrt{\ell_i}}{2}\right)\right) = \sum_{\substack{\ell_i < B \\ \ell_i \text{ prime}}} \frac{1}{2}\log \ell_i - \sum_{\substack{\ell_i < B \\ \ell_i \text{ prime}}} \log 2 \approx$$

$$\approx \frac{1}{2}\int_1^B \log x \frac{1}{\log x} dx - \log 2 \frac{B}{\log B} = \frac{1}{2}B - \log 2 \frac{B}{\log B} \approx \frac{1}{2}B$$

if B is large enough. Taking $B = c\log(p)$ where $c = 2(1+\epsilon)$ gives $\frac{1}{2}B = (1+\epsilon)\log p = \log(p^{1+\epsilon})$, which proves the lemma. □

5.1.2 Efficient representation of isogeny data

Our schemes require representing/transmitting elliptic curves and isogenies. In this section we first explain how to represent certain mathematical objects appearing in our protocol as bitstrings in a canonical way so that minimal data needs to be sent and stored. Next, we discuss different representations of isogeny paths and their impact on the efficiency of our signature schemes. As these paths will be sent from one party to another, the second party needs an efficient way to verify that the bitstring received corresponds to an isogeny path between the right curves.

Extension fields. Let p be a prime number. Every supersingular j-invariant is defined over $\mathbb{F}_{p^2}$. A canonical representation of $\mathbb{F}_{p^2}$-elements is obtained via a canonical choice of degree 2 irreducible polynomial over $\mathbb{F}_p$. Canonical representations in any other extension fields are defined in a similar way. Although there are only about $p/12$ supersingular j-invariants in characteristic p, we are not aware of an efficient method to encode these invariants into $\log p$ bits, so we represent supersingular j-invariants with the $2\log p$ bits it takes to represent an arbitrary $\mathbb{F}_{p^2}$-element.

Ellitic curves. Elliptic curves are defined by their j-invariant up to isomorphism. Hence, rather than sending the coefficients of the elliptic curve equation, it suffices to send the j-invariant. For any invariant j there is a canonical elliptic curve equation $E_j : y^2 = x^3 + \frac{3j}{1728-j}x + \frac{2j}{1728-j}$ when $j \neq 0, 1728$, $y^2 = x^3 + 1$ when $j = 0$, and $y^2 = x^3 + x$ when $j = 1728$. If one needs a particular group order then one might need to take a twist.

Isogenies. We are interested in representing chains $E_0, E_1, \ldots, E_n$ of isogenies $\phi_i : E_{i-1} \to E_i$, each of prime degree ℓ_i, where $i = 1, \ldots, n$. Here ℓ_i are always very small primes. A useful feature of our protocols is that isogeny chains can always be chosen such that the isogeny degrees are increasing $\ell_i \geq \ell_{i-1}$. First we need to discuss how to represent the sequence of isogeny degrees. If all degrees are equal to a constant ℓ (e.g., $\ell = 2$) then it is only necessary to state the length. If the degrees are different then the most compact representation seems

to be

$$N = \prod_{i=1}^{n} \ell_i$$

which might be a global system parameter, or may be sent as part of the protocol. The receiver can recover the sequence of isogeny degrees from N by factoring using trial division and ordering the primes by size. This representation is possible due to our convention the isogeny degrees are increasing and since the degrees are all small.

Now we discuss how to represent the curves themselves in the chain of isogenies. We give several methods.

1. There are two naive representations. One is to send all the j-invariants $j_i = j(E_i)$ for $0 \leq i \leq n$. This requires $2(n+1)\log_2(p)$ bits. Note that the verifier is able to check the correctness of the isogeny chain by checking that $\Phi_{\ell_i}(j_{i-1}, j_i) = 0$ for all $1 \leq i \leq n$, where Φ_{ℓ_i} is the ℓ_i-th modular polynomial. The advantage of this method is that verification is relatively quick (just evaluating a polynomial that can be precomputed and stored).

   The other naive method is to send the x-coordinate of a kernel point $P_i \in E_{j_i}$ on the canonical curve. Given j_{i-1} and the kernel point P_{i-1} one computes the isogeny ϕ_i on $E_{j_{i-1}}$ whose image is isomorphic to E_{j_i} using the Vélu formula and hence deduces j_i. Note that the kernel point is not unique and is typically defined over an extension of the field. Both these methods require sending a lot of data.

   A refinement of the second method is used in our first signature scheme, where ℓ is fixed and one can publish a point that defines the kernel of the entire isogeny chain. More precisely, a curve E and points $R, S \in E[\ell^n]$ are fixed. Each integer $0 \leq \alpha < \ell^n$ defines a subgroup $\langle R + [\alpha]S \rangle$ and hence an ℓ^n isogeny. It suffices to send α, which requires $\log_2(\ell^n)$ bits. In the case $\ell = 2$ this is just n bits, which is smaller than all the other suggestions in this section.

2. One can improve upon the naive method in several simple ways. One method is to send every second j-invariant. The verifier accepts this as a valid path if, for all odd integers i, the greatest common divisor over $\mathbb{F}_{p^2}[y]$

   $$\gcd(\Phi_{\ell_i}(j_{i-1}, y), \Phi_{\ell_{i+1}}(y, j_{i+1}))$$

   is a non-constant polynomial, which will almost always be $(y - j_i)$.

Another method is to send only some least significant bits (more than $\log_2(\ell_i + 1)$ of them) of the j_i instead of the entire value. The verifier can reconstruct the isogeny path by factoring $\Phi_{\ell_i}(j_{i-1}, y)$ over $\mathbb{F}_{p^2}$ (it will always split completely in the supersingular case) and then selecting j_i to be the root that has the correct least significant bits (depending on how many bits are used there may occasionally be a non-unique choice of root but, considering the complete path, the compressed representation should lead to a unique sequence of j-invariants).

3. An optimal compression method seems to be to define a well-ordering on $\mathbb{F}_{p^2}$ (e.g., lexicographic order on the binary representation of the element). Instead of j_i one sends the index k such that when the $\ell_i + 1$ roots of $\Phi_{\ell_i}(j_{i-1}, y)$ are written in order, j_i is the k-th root. It is clear that the verifier can reconstruct the value j_i and hence can reconstruct the whole chain from this information. The sequence of integers k can be encoded as a single integer in terms of a "base $\prod_{j=1}^{i}(\ell_i + 1)$" representation.

   If the walk is non-backtracking and the primes ℓ_i are repeated then one can remove the factor $(y - j_{i-2})$ that corresponds to the dual isogeny of the previous step, this can save some bandwidth.

   We say that this method is 'optimal', since it is hard to imagine doing better than $\log_2(\ell_i + 1)$ bits for each step in general,[3] though we have no proof that one cannot do better. However, note that the verifier now needs to perform polynomial factorisation, which may cause some overhead in a protocol. Note that in the case where all $\ell_i = 2$ and the walk is non-backtracking then this method also requires n bits, which matches the method we use in our first signature scheme (mentioned in item 1 above).

4. A variant of the optimal method is to use an ordering on points/subgroups rather than j-invariants. At each step one sends an index k such that the isogeny $\phi : E_{i-1} \to E_i$ is defined by the k-th cyclic subgroup of $E_{j_{i-1}}[\ell_i]$. Again the verifier can reconstruct the path, but this requires factoring ℓ_i-division polynomials.

   More precisely, given a canonical ordering on the field of definition of $E[\ell]$, one can define a canonical ordering of the cyclic kernels, hence represent them by a single integer in $\{0, \ldots, \ell\}$. One can extend this canonical

[3] In the most general case, when all primes ℓ_i are distinct, then there are $\prod_i(\ell_i + 1)$ possible isogeny paths and thus one cannot expect to represent an arbitrary path using fewer than $\log_2(\prod_i \ell_i)$ bits.

ordering to kernels of composite degrees in various simple ways (see also [8, Section 3.2]). If two curves are connected by two distinct isogenies of the same degree then either one can be chosen (it makes no difference in our protocols), so the ambiguity in exceptional cases is never a problem for us.

In practice, since these points may be defined over an extension of $\mathbb{F}_{p^2}$, we believe that ordering the roots of $\Phi_{\ell_i}(j_{i-1}, y)$ is significantly more efficient than ordering kernel subgroups.

Computation. Finally we give a brief analysis of the complexity of the basic operations required for our schemes, assuming fast (quasi-linear) modular and polynomial arithmetic.

As discussed above, an isogeny step of prime degree ℓ can be described by a single integer in $\{0, \ldots, \ell\}$. Similarly, by combining integers in a product, an isogeny of degree $\prod_i \ell_i^{e_i}$ can be described by a single positive integer smaller than $\prod_i (\ell_i + 1)^{e_i}$. This integer can define either a list of subgroups (specified in terms of some ordering), or a list of supersingular j-invariants (specified in terms of an ordering on the roots of the modular polynomial). In the first case, at each step the verifier, given a j-invariant, will need to compute the curve equation, then its full ℓ_i torsion (which may be over a large field extension), then to sort with respect to some canonical ordering the cyclic subgroups of order ℓ_i to identify the correct one, and finally to compute the next j-invariant with Vélu's formulae [163]. In the second case, at each step the verifier, given a j-invariant, will need to specialize one variable of the ℓ_i-th modular polynomial, then to compute all roots of the resulting univariate polynomial and finally to sort the roots to identify the correct one. The second method is more efficient as it does not require running Vélu's formulae over some large field extension, and the root-finding and sorting routines are applied on smaller inputs. We assume that the modular polynomials are precomputed.

In our second signature scheme we will have $\ell_i^{e_i} = O(\log p)$. The cost of computing an isogeny increases with the size of ℓ_i. Hence it suffices to analyse the larger case, for which $e_i = 1$ and $\ell_i = O(\log p)$. Assuming precomputation of the modular polynomials and using [162] for polynomial factorization, the most expensive part of an isogeny step is evaluating the modular polynomials $\Phi_{\ell_i}(x, y)$ at $x = j_{i-1}$. As these polynomials are bivariate with degree ℓ_i in each variable, they have $O(\ell_i^2)$ monomials, and so this requires $O(\log^2 p)$ field operations for a total cost of $\tilde{O}(\log^3 p)$ bit operations, since j-invariants are defined over $\mathbb{F}_{p^2}$. In our first signature scheme based on the De Feo-Jao-Plût

protocol we have $\ell_i = O(1)$ so each isogeny step costs $\tilde{O}(\log p)$ bit operations.

Alternatively, isogeny paths can be given as a sequence of j-invariants. To verify the path is correct one must compute $\Phi_{\ell_i}(j_{i-1}, j_i)$, which still requires $\tilde{O}(\log^3 p)$ bit operations. However, in practice it would be much quicker to not require root-finding algorithms. Also, all the steps can be checked in parallel, and all the steps of a same degree are checked using the same polynomial, so we expect many implementation optimizations to be possible.

5.1.3 Heuristic assumptions used in this chapter

This chapter makes use of several heuristic assumptions. All these assumptions say that some forms of the following approximations are valid.

Approximation 1. *Let $\mathcal{N}_1$ be a set and let $\mathcal{N}_2 \subseteq \mathcal{N}_1$. Let χ be a probability distribution on $\mathcal{N}_1$. We approximate $\mathsf{Pr}[x \in \mathcal{N}_2 \mid x \leftarrow \chi]$ by $|\mathcal{N}_2|/|\mathcal{N}_1|$.*

In several cases, $\mathcal{N}_1$ will be the set of positive integers up to some bound, and $\mathcal{N}_2$ will be a subset of integers with some factorization pattern. In this case, we will approximate $|\mathcal{N}_2|/|\mathcal{N}_1|$ by the value naturally expected from the density of primes.

Approximation 2. *Let B be a positive integer and let $\mathcal{N}_1 := \{1, 2, \ldots, B\}$. Let $\mathcal{N}_2 \subseteq \mathcal{N}_1$ be the subset of integers in $\mathcal{N}_1$ satisfying some factorization pattern. We approximate $\mathsf{Pr}[x \in \mathcal{N}_2 \mid x \leftarrow \chi]$ by the expected value of $|\mathcal{N}_2|/|\mathcal{N}_1|$ following the density of primes.*

More precisely:

- In Section 5.3.3, Step 2c, the existence of β_2 is guaranteed if some linear system is invertible over $\mathbb{Z}_N$. Here N is an integer of cryptographic size, and the system is randomized through the selection of α and β_1 in Steps 2a and 2b. We assume that the probability of having a non invertible system is negligible.
- In Lemma 30, we generate candidates for the ideals I_i according to some distribution on the set of solutions of a quadratic form. Here there are $O(\log p)$ candidate ideals, and we assume that only $O(\log p)$ trials are needed to find the correct one.
- In Section 5.3.3, Step 1, we construct a random element in an ideal I according to a specific distribution, and assume the reduced norm of this element will be a prime with a probability as given by the prime number theorem.

- In Section 5.3.3, Steps 2b and 2d, we generate integer elements according to a specific distribution, and we assume that the probability that these numbers are "Cornacchia-nice" (in the sense that Cornacchia's algorithm will run efficiently on them, which translates into some factorization pattern) only depends on their size, and is as expected for numbers of these sizes.

All assumptions except for the second one come from our use of (the powersmooth variant of) the quaternion isogeny algorithm in [118]. We expect that the first two assumptions above can be removed with a finer analysis, maybe together with some minor algorithmic changes and a moderate efficiency loss. In the case of the second assumption, trying all possible solutions to the quadratic form will maintain a polynomial complexity, though of a slightly bigger degree. One might then reduce that degree by exploiting the structure of all solutions leading to the same ideals.

On the other hand, a rigorous proof for the remaining assumptions seem to be beyond the reach of existing analytic number theory techniques. We stress that these sorts of assumptions are generally believed to be true by analytic number theory experts "unless there is a good reason for them to be false", such as some congruence condition. In the later case, we expect that simple tweaks to our algorithms will restore their correctness and improve their complexity.

5.2 First signature scheme

This section presents a signature scheme obtained from the Sigma protocol of De Feo-Jao-Plût [66]. First we describe their scheme. The independent work [167] presents a signature scheme which is obtained in the same way, by applying the Fiat–Shamir or Unruh transformation to the De Feo-Jao-Plût Sigma protocol. Nevertheless, here we obtain a smaller signature size.

5.2.1 De Feo-Jao-Plût Sigma protocol

Let p be a large prime of the form $\ell_1^{e_1}\ell_2^{e_2}f \pm 1$, where ℓ_1, ℓ_2 are small primes (typically $\ell_1 = 2$ and $\ell_2 = 3$). We start with a supersingular elliptic curve E_0 defined over $\mathbb{F}_{p^2}$ with

$$\#E_0(\mathbb{F}_{p^2}) = (\ell_1^{e_1}\ell_2^{e_2}f)^2,$$

and we fix a basis $\{R_1, S_1\}$ of $E_0[\ell_1^{e_1}]$. We choose a primitive $\ell_1^{e_1}$-torsion point as

$$P_1 = aR_1 + bS_1,$$

for some $a, b \in \mathbb{Z}_{\ell_1^{e_1}}$. We define $E_1 = E_0/\langle P_1 \rangle$, and we denote the corresponding $\ell_1^{e_1}$-isogeny by $\varphi : E_0 \to E_1$.

Let $\{R_2, S_2\}$ be a basis of $E_0[\ell_2^{e_2}]$. The public key is

$$(E_0, E_1, R_1, S_1, R_2, S_2, \varphi(R_2), \varphi(S_2)).$$

The private key is the point P_1. The interaction goes as follows. We describe one of the t parallel repetitions.

- $\mathcal{P} \to \mathcal{V}$: the prover chooses a random primitive $\ell_2^{e_2}$-torsion point P_2 as $P_2 = \alpha R_2 + \beta S_2$ for some $\alpha, \beta \in \mathbb{Z}_{\ell_2^{e_2}}$. Note that

  $$\varphi(P_2) = \alpha\varphi(R_2) + \beta\varphi(S_2).$$

  The prover defines the curves $E_2 = E_0/\langle P_2 \rangle$ and $E_3 = E_1/\langle \varphi(P_2) \rangle = E_0/\langle P_1, P_2 \rangle$, and uses Vélu's formulae to compute the following diagram.

  $$\begin{array}{ccc} E_0 & \xrightarrow{\varphi} & E_1 \\ \downarrow{\scriptstyle \psi} & & \downarrow{\scriptstyle \psi'} \\ E_2 & \xrightarrow{\varphi'} & E_3 \end{array}$$

  The prover sends E_2 and E_3 to the verifier.

- $\mathcal{V} \to \mathcal{P}$: the verifier challenges the prover with a random bit $b \leftarrow \{0, 1\}$.

- $\mathcal{P} \to \mathcal{V}$: we distinguish two cases. If $b = 0$, the prover reveals P_2 and $\varphi(P_2)$ (for example, by sending the integers α, β). If $b = 1$, the prover reveals $\psi(P_1)$.

- $\mathcal{V}$: in both cases, the verifier accepts the proof if the points revealed have the right order and are the kernels of isogenies between the right curves. We iterate this process to reduce the cheating probability.

We can repeat the above scheme t times in parallel, with t large enough to make the scheme non-trivial. Note that the response to challenge 0 is two points

while the response to challenge 1 is one point. In other words, at first sight, the responses have different lengths. Compression techniques can be used in this case to ensure that responses all have the same length (see Section 4.2 of [167]).

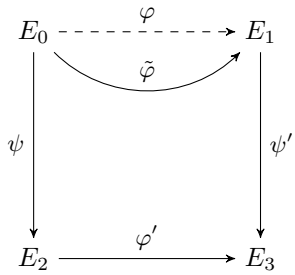
The following theorem is the main security result for this section. The basic ideas of the proof are by De Feo-Jao-Plût [66], but we give a slightly different formalisation.

Theorem 22. *The scheme described above is a complete, 2-special sound and honest verifier zero-knowledge Sigma protocol, under the hardness of assumption 13.*

Proof. It is straightforward to check that the scheme is complete. We now show that parallel executions of the Sigma protocol are sound and honest verifier zero knowledge.

For soundness, suppose that $\mathcal{A}$ is an adversary that takes as input the public key and succeeds in the Sigma protocol with probability ε. Given a challenge instance $(E_0, E_1, R_1, S_1, R_2, S_2, \varphi(R_2), \varphi(S_2))$ for Problem 11, we run $\mathcal{A}$ on this tuple as the public key. In the first round, $\mathcal{A}$ outputs commitments $(E_{i,2}, E_{i,3})$ for $1 \leq i \leq t$. We then send a challenge in $\{0,1\}^t$ to $\mathcal{A}$ and, with probability ϵ outputs a response that satisfies the verification algorithm. Now, we use the standard replay technique: Rewind $\mathcal{A}$ to the point where it had output its commitments and then respond with a different challenge. With probability ϵ, $\mathcal{A}$ outputs a valid response.

Now, choose some index i such that the challenges differ. We now restrict our focus to the ith repetition of the protocol. Then, we have that $\mathcal{A}$ sent E_2, E_3 and can answer both challenges $c = 0$ and $c = 1$ successfully. Hence we have an explicit description of the isogenies ψ, ψ' and φ' in the following diagram.



From this, one has an explicit description of an isogeny $\tilde{\varphi} = \hat{\psi}' \circ \varphi' \circ \psi$ from E_0 to E_1. The degree of $\tilde{\varphi}$ is $\ell_1^{e_1}\ell_2^{2e_2}$. One can determine $\ker(\tilde{\varphi}) \cap E_0[\ell_1^{e_1}]$ by

iteratively testing points in $E_0[\ell_1^j]$ for $j = 1, \dots e_1$. Hence, one determines the kernel of φ, as desired. This proves soundness.

Now we show the honest verifier zero-knowledge property, i.e. we show that one can simulate transcripts of the protocol without knowing the secret key. When $b = 0$ we simulate correctly by choosing $u, v \in \mathbb{Z}_{\ell_2^{e_2}}$ and setting $E_2 = E_0/\langle uR_2 + vS_2\rangle$ and $E_3 = E_1/\langle u\varphi(R_2) + v\varphi(S_2)\rangle$. When $b = 1$ we choose a random curve E_2 and a random point $R \in E_2[\ell_1^{e_1}]$ and we publish $E_2, E_3 = E_2/\langle R\rangle$ and answer with the point R (hence defining the isogeny). Although (E_2, E_3) are a priori not distributed as in an honest execution, assumption 13 implies it is computationally hard to distinguish the simulation from the real game. Hence the scheme has computational zero knowledge.

□

5.2.2 Classical signatures from the De Feo-Jao-Plût Sigma protocol

One can apply the Fiat–Shamir transformation (figure 2.1) to the De Feo-Jao-Plût scheme to obtain a signature scheme. We also observe that the scheme is recoverable and so one can apply the optimization described above. In this section we fully specify the signature scheme resulting from the transformation of figure 2.1, together with some optimisations.

Our main focus is to minimize signature size. Hence, we use the most space-efficient variant of the Fiat–Shamir transform. Next we need to consider how to minimize the amount of data that needs to be sent to specify the isogenies. Several approaches were considered in section 5.1.2. For the pair of vertical isogenies, it seems to be most compact to represent them using a representation of the kernel (this is more efficient than specifying two paths in the isogeny graph). However this requires additional points in the public key. For the horizontal isogeny there are several possible approaches, but we think that the most compact is to use the representation in terms of specifying roots of the modular polynomial. One can easily find other implementations that allow different tradeoffs of public key size versus signature size.

Key Generation Algorithm: On input a security parameter λ generate a prime p with at least 4λ bits, such that $p = \ell_1^{e_1}\ell_2^{e_2} f \pm 1$, with ℓ_1, ℓ_2, f small

(ideally $f = 1$, $\ell_1 = 2$, $\ell_2 = 3$) and $\ell_1^{e_1} \approx \ell_2^{e_2}$. Choose[4] a supersingular elliptic curve E_0 with $\#E_0(\mathbb{F}_{p^2}) = (\ell_1^{e_1}\ell_2^{e_2}f)^2$ and j-invariant j_0. Fix points $R_2, S_2 \in E_0(\mathbb{F}_{p^2})[\ell_2^{e_2}]$ and a random primitive $\ell_1^{e_1}$-torsion point $P_1 \in E_0[\ell_1^{e_1}]$. Compute the isogeny $\varphi : E_0 \to E_1$ with kernel generated by P_1, and let j_1 be the j-invariant of the image curve. Set $R_2' = \varphi(R_2)$, $S_2' = \varphi(S_2)$. Choose a hash function H with $t = 2\lambda$ bits of output. The secret key is P_1, and the public key is $(p, j_0, j_1, R_2, S_2, R_2', S_2', H)$. One can reduce the size of the public key by using different representations of isogeny paths, but for simplicity we use this variant.

Signature Algorithm: For $i = 1, \ldots, t$, choose random integers $0 \leq \alpha_i < \ell_2^{e_2}$. Compute the isogeny $\psi_i : E_0 \to E_{2,i}$ with kernel generated by $R_2 + [\alpha_i]S_2$ and let $j_{2,i} = j(E_{2,i})$. Compute the isogeny $\psi_i' : E_1 \to E_{3,i}$ with kernel generated by $R_2' + [\alpha_i]S_2'$ and let $j_{3,i} = j(E_{3,i})$. Compute $h = H(m, j_{2,1}, \ldots, j_{2,t}, j_{3,1}, \ldots, j_{3,t})$ and parse the output as t challenge bits b_i. For $i = 1, \ldots, t$, if $b_i = 0$ then set $z_i = \alpha_i$. If $b_i = 1$ then compute $\psi_i(P_1)$ and compute a representation z_i of the j-invariant $j_{2,i} \in \mathbb{F}_{p^2}$ and the isogeny with kernel generated by $\psi_i(P_1)$ (for example, as a sequence of integers representing which roots of the ℓ_1-division polynomial to choose at each step of a non-backtracking walk, or using a compact representation of $\psi_i(P_1)$ in reference to a canonical basis of $E_{2,i}[\ell_1^{e_1}]$). Return the signature $\sigma = (h, z_1, \ldots, z_t)$.

Verification Algorithm: On input a message m, a signature σ and a public key PK, recover the parameters p, E_0, E_1. For each $1 \leq i \leq t$, using the information provided by z_i, one recomputes the j-invariants $j_{2,i}, j_{3,i}$. In the case $b_i = 0$ this is done using $z_i = \alpha_i$ by computing the isogeny from E_0 with kernel generated by $R_2 + [\alpha_i]S_2$ and the isogeny from E_1 with generated by $R_2' + [\alpha_i]S_2'$. When $b_i = 1$ then the value $j_{2,i}$ is provided as part of z_i, together with a description of the isogeny from $E_{2,i}$ to $E_{3,i}$.

One then computes

$$h' = H(m, j_{2,1}, \ldots, j_{2,t}, j_{3,1}, \ldots, j_{3,t})$$

and checks that the value equals h from the signature. The signature is accepted if this is true and is rejected otherwise.

[4] Costello-Longa-Naehrig [47] choose a special j-invariant in $\mathbb{F}_p$ for efficiency reasons in their implementation of the supersingular key exchange protocol. One could also choose a random j-invariant by performing a random isogeny walk from any fixed j-invariant.

Theorem 23. *The signature scheme described above is (clasically) EU-CMA secure under the hardness of assumptions 11 and 13, in the random oracle model.*

Proof. Assumption 11 implies that the previous scheme is a Sigma protocol for a hard relation. Then, the result follows immediately from theorems 2 and 22. □

Efficiency. As isogenies are of degree roughly $\sqrt{p}$, the scheme requires to use primes p of size 4λ to defeat meet-in-the-middle attacks. Assuming H is some fixed hash function and therefore not sent, the secret key is simply $x(P_1) \in \mathbb{F}_{p^2}$. A trivial representation requires $2\log p = 8\lambda$ bits; however, with a canonical ordering of the cyclic subgroups, this can be reduced to $\frac{1}{2}\log p = 2\lambda$ bits.

The public key is p and then $j_0, j_1, x(R_2), x(S_2), x(R_2'), x(S_2') \in \mathbb{F}_{p^2}$ which requires $13\log_2(p) \approx 52\lambda$ bits. The values of j_0, $x(R_2)$ and $x(S_2)$ can be canonically fixed by the protocol, in which case the public key is only $7\log p \approx 28\lambda$ bits. The values of $x(R_2')$ and $x(S_2')$ can also be avoided but at the expense of larger signature sizes. The signature size is analysed in Lemma 24.

De Feo et al [66] showed how to compute an ℓ^e-isogeny in around $e\log(e)$ exponentiations/Vélu computations using what they call an "optimal strategy". Assuming quasi-linear cost $\tilde{O}(\log(p^2)) = \tilde{O}(\lambda)$ for the field operations, the total computational complexity of the signing and verifying algorithms is $\tilde{O}(\lambda^3)$ bit operations.

Lemma 24. *The average signature bit-size of this scheme is*

$$t \;+\; \tfrac{t}{2}\lceil\log_2(\ell_2^{e_2})\rceil + \tfrac{t}{2}\left(2\lceil\log_2(p)\rceil + \lceil\log_2(\ell_1^{e_1})\rceil\right) \approx 6\lambda^2.$$

Proof. On average half the bits b_i of the hash value are zero and half are one. When $b_i = 0$ we send an integer α_i such that $0 \le \alpha_i < \ell_2^{e_2}$, which requires $\lceil\log_2(\ell_2^{e_2})\rceil \approx 2\lambda$ bits. When $b_i = 1$ we need to send $j_{2,i} \in \mathbb{F}_{p^2}$, which requires $2\lceil\log_2(p)\rceil$ bits, followed by a representation of the isogeny. One can represent a generator of the kernel of the isogeny with respect to some canonical generators P_1', Q_1' of $E_{2,i}[\ell_1^{e_1}]$ as β_i such that $0 \le \beta_i < \ell_1^{e_1}$, thus requiring $\lceil\log_2(\ell_1^{e_1})\rceil$ bits. Alternatively one can represent the non-backtracking sequence of j-invariants in terms of an ordering on the roots of the ℓ_1-th modular polynomial. This also can be done in $\lceil\log_2(\ell_1^{e_1})\rceil$ bits. For security level λ one can take $t = \lambda$, giving $\ell_1^{e_1} \approx \ell_2^{e_2} \approx 2^{2\lambda}$, $p \approx 2^{4\lambda}$ and so signatures are around $6\lambda^2$ bits. The more conservative choice $t = 2\lambda$ gives signatures of around $12\lambda^2$ bits. □

5.2.3 Post-quantum signatures from the De Feo-Jao-Plût Sigma protocol

Next, we describe the signature scheme resulting from applying Unruh's transform to the Sigma protocol of De Feo-Jao-Plût, and we discuss its efficiency.

Remark 2. *In Unruh [156] the set Γ is of a fixed size and all responses have the same length. The quantum random oracle G is used to commit to all responses at the same time, and its domain and image sets have the same size to ensure that G is binding in an unconditional or at least statistical sense (i.e. a computationally binding commitment would not suffice). In our protocols however, the challenges are just one bit, and the responses to challenges 0 and 1 have different lengths. We therefore use two quantum random oracles G_0 and G_1 to hide responses to challenges 0 and 1 respectively.*

Key Generation Algorithm: On input a security parameter λ generate a prime p with at least 6λ bits, such that $p = \ell_1^{e_1}\ell_2^{e_2}f \pm 1$, with ℓ_1, ℓ_2, f small (ideally $f = 1$, $\ell_1 = 2$, $\ell_2 = 3$) and $\ell_1^{e_1} \approx \ell_2^{e_2} > 2^{3\lambda}$. Choose a supersingular elliptic curve E_0 with $\#E_0(\mathbb{F}_{p^2}) = (\ell_1^{e_1}\ell_2^{e_2}f)^2$ and j-invariant j_0. Fix a canonical basis $\{R_2, S_2\}$ for $E_0(\mathbb{F}_{p^2})[\ell_2^{e_2}]$ and a random primitive $\ell_1^{e_1}$-torsion point $P_1 \in E_0[\ell_1^{e_1}]$. Compute the isogeny $\varphi : E_0 \to E_1$ with kernel generated by P_1, and let j_1 be the j-invariant of the image curve. Set $R_2' = \varphi(R_2)$, $S_2' = \varphi(S_2)$. Choose a hash function $H : \{0,1\}^* \to \{0,1\}^t$ with $t = 3\lambda$ bits of output, and two hash functions $G_i : \{0,1\}^{N_i} \to \{0,1\}^{N_i}$ for $i = 0, 1$, such that every element has polynomially many preimages. Here N_i is an upper bound on the bitlength of the responses in the protocol when the challenge bit is i. The secret key is P_1, and the public key is $(p, j_0, j_1, R_2, S_2, R_2', S_2', H, G)$. One can reduce the size of the public key by using different representations of isogeny paths, but for simplicity we use this variant.

Signing Algorithm: For $i = 1, \ldots, t$, choose random integers $0 \leq \alpha_i < \ell_2^{e_2}$. Compute the isogeny $\psi_i : E_0 \to E_{2,i}$ with kernel generated by $R_2 + [\alpha_i]S_2$ and let $j_{2,i} = j(E_{2,i})$. Compute the isogeny $\psi_i' : E_1 \to E_{3,i}$ with kernel generated by $R_2' + [\alpha_i]S_2'$ and let $j_{3,i} = j(E_{3,i})$. For $i = 1, \ldots, t$, set $z_{i,0} = \alpha_i$ and $z_{i,1}$ as a representation of the j-invariant $j_{2,i} \in \mathbb{F}_{p^2}$ and the isogeny with kernel generated by $\psi_i(P_1)$ (for example, as a sequence of integers representing which roots of the ℓ_1-modular polynomial to choose at each step of a non-backtracking walk, or using a compact representation of $\psi_i(P_1)$ in reference to a canonical basis of $E_{2,i}[\ell_1^{e_1}]$).

Compute $g_{i,0} = G_0(z_{i,0})$ and $g_{i,1} = G_1(z_{i,1})$ for $1 \leq i \leq t$. Compute

$$h = H(m, j_{2,1}, \ldots, j_{2,t}, j_{3,1}, \ldots, j_{3,t}, g_{1,0}, g_{1,1}, \ldots, g_{t,0}, g_{t,1})$$

and parse the output as t challenge bits h_i. Return the signature

$$\sigma = (h, z_{1,h_1}, \ldots, z_{t,h_t}, g_{1,1-h_1}, \ldots, g_{t,1-h_t}).$$

Verification Algorithm: On input a message m, a signature σ and a public key PK, recover the parameters p, E_0, E_1. For each $1 \leq i \leq t$, using the information provided by RSP_i, one recomputes the j-invariants $j_{2,i}, j_{3,i}$. In the case $h_i = 0$ this is done using $\text{RSP}_i = \alpha_i$ by computing the isogeny from E_0 with kernel generated by $R_2 + [\alpha_i]S_2$ and the isogeny from E_1 with generated by $R'_2 + [\alpha_i]S'_2$. When $h_i = 1$ then the value $j_{2,i}$ is provided as part of RSP_i, together with a description of the isogeny from $E_{2,i}$ to $E_{3,i}$.

The verifier computes $g_{i,h_i} = G_{h_i}(z_{i,h_i})$ for $1 \leq i \leq t$ (padding to N bits using zeros) and checks that the hash value

$$h' = H(m, j_1, j_{2,1}, \ldots, j_{2,t}, g_{1,0}, g_{1,1}, \ldots, g_{t,0}, g_{t,1}).$$

is the same as h from the signature. In this case the verifier accepts the proof, otherwise it is rejected.

We now show that this scheme is a secure signature.

Theorem 25. *The signature scheme described above is (quantumly) existentially unforgeable against chosen-message attacks under the hardness of assumptions 11 and 13, in the quantum random oracle model.*

Proof. This follows immediately from theorems 3 and 22. □

Efficiency. There are four reasons why the post-quantum variant of the signature is less efficient than the variant in Section 5.2.2. First, the prime p is larger in the post-quantum case due to the quantum attack on the isogeny problem due to Biasse, Jao and Sankar [19]. Second, one must compute responses to both values of the challenge bit, which essentially doubles the computation compared with the non-post-quantum case. Thirdly, one needs to send the values $g_{i,j}$ as part of the signature, which increases signature size. Note that we have introduced an optimisation that only sends half the values $g_{i,j}$, since the missing values can be recomputed by the verifier. And fourth, the chosen value of t will be larger when aiming for quantum security.

We now compute the average signature size. When $h_i = 0$, responses are of the form α_i for a random integer $0 \leq \alpha_i < \ell_2^{e_2}$, and thus requiring $N_0 \approx \log {\ell_2}^{e_2} \approx \frac{1}{2} \log p$ bits each. When $h_i = 1$, responses encode the j-invariant $j_{2,i}$, which takes $\lceil 2 \log p \rceil$ bits to represent, and the isogeny with kernel generated by $\psi_i(P_1)$, which has degree $\ell_1^{e_1}$, and thus requires $\lceil \log \ell_1^{e_1} \rceil$ bits, for a total of $N_1 \approx \frac{5}{2} \log p$. Finally, we note that the average response length $\frac{3}{2} \log p$ is doubled as in Unruh transform a commitment $g_{i,1-h_i} = G_{1-h_i}(z_{i,1-h_i})$ to the other challenge value is simultaneously transmitted. The average size of signatures is therefore $t + t \cdot 3 \log p$. For λ bits of security, we choose $\log p = 6\lambda$ and $t = 3\lambda$, obtaining an average signature size of $54\lambda^2$.

5.3 Second signature scheme

We now present our main result. The main advantage of this scheme compared with the one in the previous section is that its security is based on the general problem of computing an isogeny between two supersingular curves, or equivalently on computing the endomorphism ring of a supersingular elliptic curve. Unlike the scheme in the previous section, the prime has no special property and no auxiliary points are revealed.

5.3.1 Sigma protocol based on the endomorphism ring computation

The concept is similar to the graph isomorphism Sigma protocol, in which we reveal one of two graph isomorphisms, but never enough information to deduce the secret isomorphism.

As recalled in Section 5.1.2, although it is believed that computing endomorphism rings of supersingular elliptic curves is a hard computational problem in general, there are some particular curves for which it is easy.

The following construction is explained in Lemma 2 of [118]. We choose

$$E_0 : y^2 = x^3 + x$$

over a field $\mathbb{F}_{p^2}$, where $p = 3 \bmod 4$ and $\#E_0(\mathbb{F}_{p^2}) = (p+1)^2$. Unlike the scheme in Section 5.2, no constraint on the prime p or group order is necessary. We have $j(E_0) = 1728$. When $p = 3 \bmod 4$, the quaternion algebra $\mathbf{B}_{p,\infty}$ ramified at p and ∞ can be canonically represented as

$$\mathbf{B}_{p,\infty} = \mathbb{Q}\langle \mathbf{i}, \mathbf{j} \rangle = \mathbb{Q} + \mathbb{Q}\mathbf{i} + \mathbb{Q}\mathbf{j} + \mathbb{Q}\mathbf{k},$$

where $\mathbf{i}^2 = -1$, $\mathbf{j}^2 = -p$ and $\mathbf{k} := \mathbf{ij} = -\mathbf{ji}$. The endomorphism ring of E_0 is isomorphic to the maximal order $\mathcal{O}_0$ with $\mathbb{Z}$-basis

$$\left\{1, \mathbf{i}, \frac{1+\mathbf{k}}{2}, \frac{\mathbf{i}+\mathbf{j}}{2}\right\}.$$

Indeed, there is an isomorphism of quaternion algebras $\theta : B_{p,\infty} \to \mathrm{End}(E_0) \otimes \mathbb{Q}$ sending $(1, \mathbf{i}, \mathbf{j}, \mathbf{k})$ to $(1, \phi, \pi, \pi\phi)$ where $\pi(x, y) = (x^p, y^p)$ is the Frobenius endomorphism, and $\phi(x, y) = (-x, \iota y)$ with $\iota^2 = -1$.

Let L be the product of prime powers ℓ^e up to $B = 2\log(p)$ (this choice is based on Lemma 21). In other words, let $\ell_1, \ldots, \ell_r$ be the list of all primes up to B and let $L = \prod_{i=1}^{r} \ell_i^{e_i}$ where $\ell_i^{e_i} \leq B < \ell_i^{e_i+1}$.

To generate the public and private keys, we take a random isogeny (walk in the graph) $\varphi : E_0 \to E_1$ of powersmooth degree L and, using this knowledge, compute $\mathrm{End}(E_1)$. The public information is E_1. The secret is $\mathrm{End}(E_1)$ or, equivalently, a path from E_0 to E_1. Under the assumption that computing the endomorphism ring is hard, the secret key cannot be computed from the public key only.

To prove knowledge of φ the prover will choose a random isogeny $\psi : E_1 \to E_2$ and give E_2 to the verifier. The verifier challenges the prover to give either the isogeny $\psi : E_1 \to E_2$ or an isogeny $\eta : E_0 \to E_2$. The fundamental problem is to find an isogeny η that does not leak any information about φ (in particular, the isogeny path corresponding to $\psi \circ \varphi$ would not be a secure response). Our scheme uses the following three algorithms, that are explained in detail in later sections, that allow a 'pseudo-canonical' isogeny η to be computed.

Translate isogeny path to ideal: Given $E_0, \mathcal{O}_0 = \mathrm{End}(E_0)$ and a chain of isogenies from E_0 to E_1, to compute $\mathcal{O}_1 = \mathrm{End}(E_1)$ and a left $\mathcal{O}_0$-ideal I whose right order is $\mathcal{O}_1$.

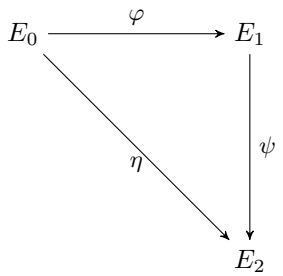
Find new path: Given a left $\mathcal{O}_0$-ideal I corresponding to an isogeny $E_0 \to E_2$, to produce a new left $\mathcal{O}_0$-ideal J corresponding to an "independent" isogeny $E_0 \to E_2$ of powersmooth degree.

Translate ideal to isogeny path: Given $E_0, \mathcal{O}_0, E_2, I$ such that I is a left $\mathcal{O}_0$-ideal whose right order is isomorphic to $\mathrm{End}(E_2)$, to compute a sequence of prime degree isogenies giving the path from E_0 to E_2.

The public key is a pair (E_0, E_1) and the private key is an isogeny $\varphi : E_0 \to E_1$ of powersmooth degree L. Below we describe the interaction between the prover and the verifier.

- $\mathcal{P} \to \mathcal{V}$: the prover performs a random walk starting from E_1 of powersmooth degree L in the graph, obtaining a curve E_2 and an isogeny $\psi : E_1 \to E_2$, and sends E_2 to the verifier.
- $\mathcal{V} \to \mathcal{P}$: the verifier challenges the prover with a random bit $b \leftarrow \{0, 1\}$.
- $\mathcal{P} \to \mathcal{V}$: if $b = 0$, the prover sends ψ to the verifier. If $b = 1$, the prover does the following:
  - Compute $\mathrm{End}(E_2)$ and translate the isogeny path between E_0 and E_2 into a corresponding ideal I giving the path in the quaternion algebra.
  - Use the **Find new path** algorithm to compute a "pseudo-canonical" path between $\mathrm{End}(E_0)$ and $\mathrm{End}(E_2)$ in the quaternion algebra, represented by an ideal J.
  - Translate the ideal J to an isogeny path η from E_0 to E_2.
  - Send η to the verifier.
- $\mathcal{V}$: the verifier accepts the proof if the answer to the challenge is indeed an isogeny between E_1 and E_2 or between E_0 and E_2, respectively.

The isogenies involved in this protocol are summarized in the following diagram:



It is easy to see that this a canonical, recoverable Sigma protocol, but it is not non-trivial as the challenge is only one bit. Thus, we repeat the protocol to reduce the cheating probability.

The two translation algorithms mentioned above in the $b = 1$ case will be described in Section 5.3.4. They rely on the fact that $\mathrm{End}(E_0)$ is known. The algorithms are efficient when the degree of the random walk is powersmooth,

and for this reason all isogenies in our protocols will be of powersmooth degree. The powersmooth version of the quaternion isogeny algorithm of Kohel-Lauter-Petit-Tignol will be described and analysed in Section 5.3.3. The random walks are taken of sufficiently large degree such that their output has close to uniform distribution, by Theorem 20 and Lemma 21.

In the next subsection we will prove the following result.

Theorem 26. *The scheme described above is a complete, 2-special sound and honest verifier zero-knowledge Sigma protocol.*

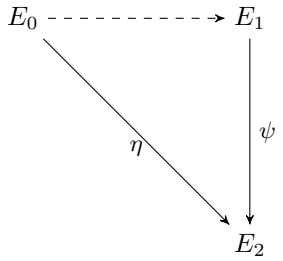
The advantage of this construction over De Feo-Jao-Plût's scheme is that it relies on a more standard and potentially harder computational problem. In the rest of this section, we first give a proof of Theorem 26, then we provide details of the algorithms involved in our scheme.

5.3.2 Proof of theorem 26

We shall prove that the Sigma protocol described in the previous section is complete, 2-special sound and honest verifier zero-knowledge. It follows that t parallel executions of the protocol make it non-trivial.

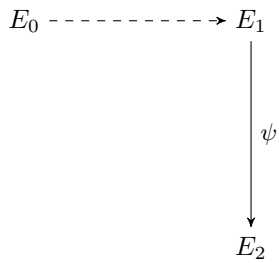
Completeness. Let φ be an isogeny between E_0 and E_1 of B-powersmooth degree, for $B = O(\log p)$. If the challenge received is $b = 0$, it is clear that the prover knows a valid isogeny $\psi : E_1 \to E_2$, so the verifier accepts the proof. If $b = 1$, the prover follows the procedure described above and the verifier accepts. In the next subsections we will show that this procedure is polynomial time.

2-special soundness. Let (E_0, E_1) be challenge instance of assumption 7 and set it to be the public key for the scheme. Suppose we are given transcripts (a, c_1, c_2, z_1, z_2) for the single-bit scheme such that $\mathcal{V}(\mathsf{pk}, a, c_i, z_i) = 1$ for $i = 1, 2$. Let $E_2 = a$. Since $c_1 \neq c_2$ the responses z_1 and z_2 therefore give two isogenies $\psi : E_1 \to E_2$, $\eta : E_0 \to E_2$. Given these two valid answers, an extraction algorithm can compute an isogeny $\phi : E_0 \to E_1$ as $\phi = \hat{\psi} \circ \eta$, where $\hat{\psi}$ is the dual isogeny of ψ. The extractor outputs ϕ, which is a valid witness. Moreover using the algorithms of Sections 5.3.3 and 5.3.4 This is summarized in the following diagram.



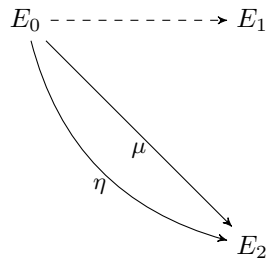
Honest-verifier zero-knowledge. We shall prove that there exists a probabilistic polynomial time simulator $\mathcal{S}$ that outputs transcripts indistinguishable from transcripts of interactions with an honest verifier, in the sense that the two distributions are statistically close. Note that $\mathcal{O}_0 = \mathrm{End}(E_0)$ is public information so is known to the simulator. The simulator starts by taking a random bit $b \leftarrow \{0, 1\}$.

– If $b = 0$, take a random walk from E_1 of powersmooth degree L, as in the real protocol, obtaining a curve E_2 and an isogeny $\psi : E_1 \to E_2$. The simulator outputs the transcript $(E_2, 0, \psi)$.



In this case, it is clear that the distributions of every element in the transcript are the same as in the real interaction, as they are generated in the same way. This is possible because, when $b = 0$, the secret is not required for the prover to answer the challenge.

– If $b = 1$, take a random walk from E_0 of length L to obtain a curve E_2 and an isogeny $\mu : E_0 \to E_2$, then proceed as in an honest execution of the protocol, running the **Find new path** algorithm, to produce another isogeny $\eta : E_0 \to E_2$. The simulator outputs the transcript $(E_2, 1, \eta)$.



The reason to output η instead of μ is to ensure that the transcript distributions are indistinguishable from the distributions in the real scheme. This is proven below, in Lemma 27.

We first study the distribution of E_2 up to isomorphism. Let J_r be the output of the random walk from E_1 to produce $j(E_2)$ in the real interaction, and let J_s be the output of the random walk from E_0 to produce $j(E_2)$ in the simulation.

Let G be the set of all supersingular j-invariants, namely the vertex set of the isogeny graph. Note that $\#\mathsf{G} = N \approx p/12$. By theorem 20 and lemma 21, and since the isogeny walks have degree L, we have, for any $j \in \mathsf{G}$

$$\left|\mathsf{Pr}(\mathsf{J}_r = j) - \frac{1}{N_p}\right| \leq \frac{1}{p^{1+\epsilon}}, \qquad \left|\mathsf{Pr}(\mathsf{J}_s = j) - \frac{1}{N_p}\right| \leq \frac{1}{p^{1+\epsilon}}.$$

Therefore

$$\sum_{j\in\mathsf{G}} |\mathsf{Pr}(\mathsf{J}_r = j) - \mathsf{Pr}(\mathsf{J}_s = j)| \leq N \cdot \max_{j\in\mathsf{G}} |\mathsf{Pr}(\mathsf{J}_r = j) - \mathsf{Pr}(\mathsf{J}_s = j)| \leq$$
$$\leq N \cdot \left(\frac{1}{p^{1+\epsilon}} + \frac{1}{p^{1+\epsilon}}\right) \approx \frac{1}{6p^{\epsilon}}$$

which is a negligible function of λ for any constant $\epsilon > 0$. In other words, the statistical distance, between the distribution of $j(E_2)$ in the real signing algorithm and the simulation, is negligible. Now, since η is produced in the same way from E_0 and E_2 in the simulation and in the real protocol execution, we have that the statistical distance between the distributions of η is also negligible. This follows from Lemma 27 in Section 5.3.3, which states that the output of the quaternion path algorithm does not depend on the input ideal, only on its ideal class.

5.3.3 Quaternion isogeny path algorithm

In this section we sketch the quaternion isogeny algorithm from Kohel–Lauter–Petit–Tignol [118] and we evaluate its complexity when $p = 3 \bmod 4$. The original paper does not give a precise complexity analysis; it only claims that the algorithm runs in heuristic probabilistic polynomial time. This is the algorithm used for the **Find new path** procedure in the Sigma protocol.

The algorithm takes as input two maximal orders $\mathcal{O}, \mathcal{O}'$ in the quaternion algebra $\mathbf{B}_{p,\infty}$, and it returns a sequence of left $\mathcal{O}$-ideals $I_0 = \mathcal{O} \supset I_1 \supset \ldots \supset I_e$ such that the right order of I_e is in the same equivalence class as $\mathcal{O}'$. In addition, the output is such that the index of I_{i+1} in I_i is a small prime for all i. The paper [118] focuses on the case where the norm of I_e is ℓ^e for some integer e, but it mentions that the algorithm can be extended to the case of powersmooth norms. We will only describe and use the powersmooth version. In our application there are some efficiency advantages from using isogenies whose degree is a product of small powers of distinct primes, rather than a large power of a small prime.

Note that the ideals returned by the quaternion isogeny path algorithm (or equivalently the right orders of these ideals) correspond to vertices of the path in the quaternion algebra graph, and to a sequence of j-invariants by Deuring's correspondence. In the next subsection we will describe how to make this correspondence explicit; here we focus on the quaternion algorithm itself.

An important feature of the algorithm is that paths between two arbitrary maximal orders $\mathcal{O}$ and $\mathcal{O}'$ are always constructed as a concatenation of two paths from each maximal order to a special maximal order. In our scheme and the discussion below we fix $\mathcal{O}_0 = \langle 1, \mathbf{i}, \frac{1+\mathbf{k}}{2}, \frac{\mathbf{i}+\mathbf{j}}{2} \rangle$ where $\mathbf{i}^2 = -1$ and $\mathbf{j}^2 = -p$.

We focus on the case where $\mathcal{O} = \mathcal{O}_0$, and assume that instead of a second maximal order $\mathcal{O}'$ we are given the corresponding left $\mathcal{O}_0$-ideal I as input (the two variants of the problem are equivalent). This will be sufficient for our use of the algorithm. We assume that I is given by a $\mathbb{Z}$-basis of elements in $\mathcal{O}_0$. Denote by $n(\alpha)$ and $n(I)$ the reduced norm of an element or ideal respectively. Note that in our context $n(I)$ is known, as it is the degree of the known isogeny. The equivalence class of maximal orders defines an equivalence class of $\mathcal{O}_0$-ideals, where two ideals I and J are in the same class if and only if $I = Jq$ with $q \in \mathbf{B}^*_{p,\infty}$. Therefore, our goal is, given a left $\mathcal{O}_0$-ideal I, to compute another left $\mathcal{O}_0$-ideal J with powersmooth norm in the same ideal class. Further, in order to be able to later apply Algorithm 2, we require the norm of I to be odd. Without loss of generality we assume there is no integer $s > 1$ such that $I \subseteq s\mathcal{O}_0$, and that $I \neq \mathcal{O}_0$. The algorithm proceeds as follows:

1. Compute an element $\delta \in I$ and an ideal $I' = I\overline{\delta}/n(I)$ of prime norm N.
2. Find $\beta \in I'$ with norm NS where S is powersmooth and odd.
3. Output $J = I'\overline{\beta}/N$.

Steps 1 and 3 of this algorithm rely on the following simple result [118, Lemma 5]: if I is a left $\mathcal{O}$-ideal of reduced norm N and α is an element of I, then $I\overline{\alpha}/N$ is a left $\mathcal{O}$-ideal of norm $n(\alpha)/N$. Clearly, I and J are in the same equivalence class.

To compute δ in step 1, first a Minkowski-reduced basis $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ of I is computed [135]. To obtain lemma 27 below we make sure that the Minkowski basis is uniformly randomly chosen among all such bases[5]. Then random elements $\delta = \sum_i x_i\alpha_i$ are generated with integers x_i in an interval $[-m, m]$, where m is determined later, until the norm of δ is equal to $n(I)$ times a prime. A probable prime suffices in this context (actually step 1 is not strictly needed, but aims to simplify step 2), so we can use the Miller–Rabin test to discard composite numbers with high probability.

Step 2 is the core of the algorithm and actually consists of the following substeps:

2a. Find α such that $I' = \mathcal{O}_0 N + \mathcal{O}_0\alpha$.

2b. Find $\beta_1 \in \mathcal{O}_0$ with odd norm NS_1, where S_1 is powersmooth.

2c. Find $\beta_2 \in \mathbb{Z}\mathbf{j} + \mathbb{Z}\mathbf{k}$ such that $\alpha = \beta_1\beta_2 \bmod N\mathcal{O}_0$.

2d. Find $\beta_2' \in \mathcal{O}_0$ with odd powersmooth norm S_2 and $\lambda \in \mathbb{Z}_N^*$ such that $\beta_2' = \lambda\beta_2 \bmod N\mathcal{O}_0$.

2e. Set $\beta = \beta_1\beta_2'$.

In step 2a, proposition 5 ensures that such α exits, and we can take random linear combinations of the Minkowski basis until the condition is met.

In step 2b, the algorithm actually searches for $\beta_1 = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$. A large enough powersmooth number S_1 is fixed a priori, then the algorithm generates small random values of c, d until the norm equation

$$a^2 + b^2 = S_1 - p(c^2 + d^2)$$

[5] One can enumerate all Minkowski bases efficiently. In [118], an arbitrary Minkowski basis was chosen.

can be solved efficiently using Cornacchia's algorithm (for example, until the right hand side is a prime equal to 1 modulo 4).

Step 2c is just linear algebra modulo N. As argued in [118] it has a negligible chance of failure, in which case one can just go back to Step 2b.

In step 2d the algorithm a priori fixes S_2 large enough, then searches for integers a, b, c, d, λ with $\lambda \notin N\mathbb{Z}$, and such that

$$N^2(a^2 + b^2) + p\left((\lambda C + cN)^2 + (\lambda D + dN)^2\right) = S_2,$$

where we have $\beta_2 = C\mathbf{j} + D\mathbf{k}$. If necessary S_2 is multiplied by a small prime such that $p(C^2+D^2)S_2$ is a square modulo N, after which the equation is solved modulo N, leading to two solutions for λ. An arbitrary solution is chosen, and then looking at the equation modulo N^2 leads to a linear space of solutions for $(c, d) \in \mathbb{Z}_N$. The algorithm chooses random solutions until the equation

$$a^2 + b^2 = \left(S_2 - p^2\left((\lambda C + cN)^2 + (\lambda D + dN)^2\right)\right)/N^2$$

can be efficiently solved with Cornacchia's algorithm.

The overall algorithm is summarized in Algorithm 1.

We now prove two lemmas on this algorithm. The first lemma shows that the output of this algorithm only depends on the ideal class of I but not on I itself. This is important in our Sigma protocol, as otherwise part of the secret isogeny φ could potentially be recovered from η. The second lemma gives a precise complexity analysis of the algorithm, where [118] only showed probabilistic polynomial time complexity. Both lemmas are of independent interest.

Lemma 27. *The output distribution of the quaternion isogeny path algorithm only depends on the equivalence class of its input. (In particular, the output distribution does not depend on the particular ideal class representative chosen for this input.)*

Proof. Let I_1 and I_2 be two left $\mathcal{O}_0$-ideals in the same equivalence class, that is, there exists $q \in \mathbf{B}^*_{p,\infty}$ such that $I_2 = I_1 q$. We show that the distribution of the ideal I' computed in Step 1 of the algorithm is identical for I_1 and I_2. As the inputs are not used anymore in the remainder of the algorithm, this will prove the lemma.

In the first step the algorithm computes a Minkowski basis of its input, uniformly chosen among all possible Minkowski bases. Let $B_1 = \{\alpha_{11}, \alpha_{12}, \alpha_{13}, \alpha_{14}\}$ be a Minkowski basis of I_1. Then, since the norm is multiplicative, we have that $B_2 = \{\alpha_{11}q,\ \alpha_{12}q, \alpha_{13}q, \alpha_{14}q\}$ is a Minkowski basis of I_2. The algorithm then

Algorithm 1 Find new path algorithm

Input: $\mathcal{O}_0 = \langle 1, \mathbf{i}, \frac{1+\mathbf{k}}{2}, \frac{\mathbf{i}+\mathbf{j}}{2} \rangle$, I a left $\mathcal{O}_0$-ideal, $n(I)$.
Output: J a left $\mathcal{O}_0$-ideal of powersmooth norm such that $I = Jq$ for some $q \in B_{p,\infty}$.

1: $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ Minkowski-reduced basis of I.
2: $\alpha_i \leftarrow \{\pm\alpha_i\}$ for $i = 1, 2, 3, 4$.
3: **loop**
4: $\quad \{x_1, x_2, x_3, x_4\} \leftarrow [-m, m]^4$. Start with $m = \lceil \log p \rceil$ and do exhaustive search in the box, increasing m if necessary.
5: $\quad \delta := \sum_{i=1}^{4} x_i \alpha_i$
6: $\quad$ **if** $N := n(\delta)/n(I)$ is prime **then return** $N, I' := I\overline{\delta}/n(I)$
7: Set an a priori powersmooth bound $s = \frac{7}{2} \log p$, and odd integers S_1, S_2 with $S_1 > p \log p$, $S_2 > p^3 \log p$ and s-powersmooth product $S_1 S_2$.
8: Choose $\alpha \in I'$ such that $\gcd(n(\alpha), N^2) = N$, so that $I' = \mathcal{O}_0 N + \mathcal{O}_0 \alpha$.
9: **while** a, b are not found **do**
10: $\quad c, d \leftarrow [-m, m]^2$, for $m = \lfloor \sqrt{NS_1/2p} \rfloor$. Increase S_1 and s if necessary.
11: $\quad a, b \leftarrow$ Solution of $a^2 + b^2 = NS_1 - p(c^2 + d^2)$ (solve using Cornacchia's algorithm).
12: $\beta_1 = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$
13: Set β_2 as a solution of $\alpha = \beta_1 \beta_2 \bmod N\mathcal{O}_0$ with $\beta_2 \in \mathbb{Z}\mathbf{j} + \mathbb{Z}\mathbf{k}$.
14: Write $\beta_2 = C\mathbf{j} + D\mathbf{k}$. Try small odd primes r in increasing order until we find one such that $\left(\frac{(C^2+D^2)S_2 r}{N}\right) = 1$, and set $S_2 = S_2 r$. Update s accordingly.
15: $\lambda \leftarrow$ Solution of $p\lambda^2(C^2 + D^2) = S_2 \bmod N$.
16: **while** a, b are not found **do**
17: $\quad c, d \leftarrow$ Solution of $p\lambda^2(C^2 + D^2) + 2p\lambda N(Cc + Dd) = S_2 \bmod N^2$.
18: $\quad a, b \leftarrow$ Solution of $a^2 + b^2 = \left(S_2 - p^2\left((\lambda C + cN)^2 + (\lambda D + dN)^2\right)\right)/N^2$ (solve using Cornacchia's algorithm). Increase S_2 and s if necessary.
19: $\beta_2' = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$
20: $J = I'\overline{\beta_1 \beta_2'}/N$

computes random elements $\delta = \sum_i x_i \alpha_i$ for integers x_i in an interval $[-m, m]$. Clearly, for any element δ_1 computed when the input is I_1, there is a corresponding element $\delta_2 = \delta_1 q$ computed when the input is I_2. This is repeated until the norm of δ is a prime times $n(I)$. As $n(I_2) = n(I_1)n(q)$, the stopping condition is equivalent for both. Finally, an ideal I of prime norm is computed as $I\overline{\delta}/n(I)$. Clearly, when $\delta_2 = \delta_1 q$, we have that

$$\frac{I_2\overline{\delta}_2}{n(I_2)} = \frac{I_1 q\overline{q}\overline{\delta}_1}{n(q)n(I_1)} = \frac{I_1\overline{\delta}_1}{n(I_1)}.$$

This shows that the prime norm ideal computed in Step 1 only depends on the equivalence class of the input. □

The expected running time given in the following lemma relies on several heuristics related to the factorization of numbers generated following certain distributions (see Section 5.1.3). Intuitively all these heuristics say that asymptotically those numbers behave in the same way as random numbers of the same size.

Lemma 28. *Let* $X := \max |c_{ij}|$ *where* $c_{ij} \in \mathbb{Z}$ *are integers such that*

$$c_{i1} + c_{i2}\mathbf{i} + c_{i3}\frac{1+\mathbf{k}}{2} + c_{i4}\frac{\mathbf{i}+\mathbf{j}}{2},$$

for $i = 1, 2, 3, 4$, *form a* $\mathbb{Z}$*-basis for* I. *If* $\log X = O(\log p)$ *then Algorithm 1 heuristically runs in time* $\tilde{O}(\log^3 p)$, *and produces an output of norm* S *with* $\log(S) \approx \frac{7}{2}\log(p)$ *which is* $\left(\frac{7}{2} + o(1)\right)\log p$*-powersmooth.*

Proof. The Minkowski basis can be computed in $O(\log^2 X)$, for example using the algorithm of [135].

For generic ideals the reduced norms of all Minkowski basis elements[6] are in $O(\sqrt{p})$ (see [118, Section 3.1]). In the first loop we initially set $m = \lceil \log p \rceil$. Assuming heuristically that the numbers N generated behave like random numbers we expect the box to produce some prime number. The resulting N will be in $\tilde{O}(\sqrt{p})$. For some non generic ideals the Minkowski basis may contain a pair of elements with norms significantly smaller than $O(\sqrt{p})$; in that case we can expect to finish the loop for smaller values of m by setting $x_3 = x_4 = 0$, and to obtain some N of a smaller size.

[6] The reduced norm of an ideal element is the norm of this element divided by the norm of the ideal.

Rabin's pseudo-primality test performs a single modular exponentiation (modulo a number of size $\tilde{O}(\sqrt{p})$), and it is passed by composite numbers with a probability at most $1/4$. The test can be repeated r times to decrease this probability to $1/4^r$. Assuming heuristically that the numbers tested behave like random numbers, the test will only be repeated a significant amount of times on actual prime numbers, so in total it will be repeated $O(\log p)$ times. This leads to a total complexity of $\tilde{O}(\log^3 p)$ bit operations for the first loop, using fast (quasi-linear) modular multiplication.

The other two loops involve solving equations of the form $x^2 + y^2 = M$. For such an equation to have solutions it is sufficient that M is a prime and $M = 1 \bmod 4$, a condition that is heuristically satisfied after $2 \log M$ random trials. Choosing S_1 and S_2 as in the algorithm ensures that the right-hand term of the equation is positive, and (assuming this term behaves like a random number of the same size) is of the desired form for some choices (c, d), at least heuristically. Cornacchia's algorithm runs in time $\tilde{O}(\log^2 M)$, which is also $\tilde{O}(\log^2 p)$ in the algorithm. The pseudo-primality tests will require $\tilde{O}(\log^3 p)$ operations in total, and their cost will dominate both loops.

Computing β_2 is just linear algebra modulo $N \approx \tilde{O}(\sqrt{p})$ and this cost can be neglected. The last two steps can similarly be neglected. As a result, we get an overall cost of $\tilde{O}(\log^3 p)$ bit operations for the whole algorithm.

Let $s = \frac{7}{2} \log p$. We have

$$n(J) = n(I')n(\beta_1)n(\beta_2')/N^2,$$

and thus, neglecting $\log \log p$ factors, we have that

$$\log n(J) \approx \frac{1}{2} \log p + \log p + 3 \log p - \log p = \frac{7}{2} \log p.$$

We make the heuristic assumption that $\log n(J) = \left(\frac{7}{2} + o(1)\right) \log p$. Moreover, heuristically

$$\prod_{p_i^{e_i} < s} p_i^{e_i} \approx (s)^{s/\log s} \approx p^{7/2 + o(1)},$$

so we can expect to find $S_1 S_2$ that is s-powersmooth and of the correct size. □

Note. A subtle issue is to understand in what sense the output of Algorithm 1 is a 'random' isogeny. The algorithm appears to make many random choices: first a 'random ideal' I' is chosen, then a 'random' element β_1 is constructed, then an 'arbitrary' β_2 is constructed, and finally the ideal J is output. However,

a crucial observation is Lemma 27: since J is equivalent to I, the output does not actually depend heavily on these choices (intuitively, all the choices 'cancel each other out'). There is only a small set of actual isogenies η that will be output by this algorithm (once the parameter L and other smoothness bounds are fixed). For this reason, we can view the output as 'independent' of I (and hence of φ) and the isogeny η as a 'pseudo-canonical' choice of isogeny from E_0 to E_2.

5.3.4 Step-by-step Deuring correspondence

We now discuss algorithms to convert isogeny paths into paths in the quaternion algebra, and vice versa. This will be necessary in our protocols, as we are sending curves and isogenies, whereas the process uses the quaternion path algorithm.

All the isogeny paths that we will need to translate in our signature scheme will start from the special j-invariant $j_0 = 1728$. We recall (see beginning of Section 5.3.1) that this corresponds to the curve E_0 with equation $y^2 = x^3 + x$ and endomorphism ring

$$\mathcal{O}_0 = \mathrm{End}(E_0) = \left\langle 1, \phi, \frac{1+\pi\phi}{2}, \frac{\pi+\phi}{2} \right\rangle.$$

Moreover, there is an isomorphism of quaternion algebras sending $(1, \mathbf{i}, \mathbf{j}, \mathbf{k})$ to $(1, \phi, \pi, \pi\phi)$.

For any isogeny $\varphi : E_0 \to E_1$ of degree n, we can associate a left $\mathrm{End}(E_0)$-ideal $I = \mathrm{Hom}(E_1, E_0)\varphi$ of norm n, corresponding to a left $\mathcal{O}_0$-ideal with the same norm in the quaternion algebra $\mathbf{B}_{p,\infty}$. Conversely every left $\mathcal{O}_0$-ideal arises in this way [117, Section 5.3]. In our protocol, we will need to make this correspondence explicit, namely we will need to pair up each isogeny from E_0 with the correct $\mathcal{O}_0$-ideal. Moreover, we need to do this for 'large' degree isogenies to ensure a good distribution via our random walk theorem.

Translating an ideal to an isogeny path.

Let E_0 and $\mathcal{O}_0 = \mathrm{End}(E_0)$ be given, together with a left $\mathcal{O}_0$-ideal I corresponding to an isogeny of degree n. We assume I is given as a $\mathbb{Z}$-basis $\{\alpha_1, \ldots, \alpha_4\}$. We explicitly find the isogeny by determining its kernel [165].

Assume for the moment that n is a small prime. One can compute generators for all cyclic subgroups of $E_0[n]$, each one uniquely defining a degree n isogeny which can be computed with Vélu's formulae. A generator P then corresponds

to the basis $\{\alpha_1, \ldots, \alpha_4\}$ if and only if $\alpha_j(P) = 0$ for all $1 \leq j \leq 4$. To evaluate $\alpha(P)$ with $\alpha \in I$ and $P \in E_0[n]$, we first write

$$\alpha = \frac{u + v\mathbf{i} + w\mathbf{j} + x\mathbf{k}}{2},$$

then we compute P' such that $[2]P' = P$, and finally we evaluate

$$[u]P' + [v]\phi(P') + [w]\pi(P') + [x]\pi(\phi(P')).$$

To show that any such P' works, write $\beta = u + v\mathbf{i} + w\mathbf{j} + x\mathbf{k}$. Since $\beta = \alpha \circ [2]$ it follows that $E_0[2] \subseteq \ker(\beta)$. If $\beta(P') = 0$ then

$$\alpha(P) = \alpha([2]P') = (\alpha \circ [2])(P') = \beta(P') = 0.$$

Since any other choice of P' is $P' + T$ for some $T \in E_0[2]$ the choice of P' does not matter.

An alternative to trying all subgroups is to choose a pair $\{P_1, P_2\}$ of generators for $E_0[n]$ and, for some $\alpha \in I$, solve the discrete logarithm instance (if possible) $\alpha(P_2) = [x]\alpha(P_1)$. It follows that $\alpha(P_2 - [x]P_1) = 0$ and so we have determined a candidate point in the kernel of the isogeny. Both solutions are too expensive for large n.

When $n = \ell^e$ the degree n isogeny can be decomposed into a composition of e degree ℓ isogenies. If P is a generator for the kernel of the degree ℓ^e isogeny, then $\ell^{e-i+1}P$ is the kernel of the degree ℓ^i isogeny corresponding to the first i steps. One can therefore match ideals with kernels step-by-step, with successive approximations of I or P respectively. This algorithm is more efficient than the previous one, but it still requires to compute ℓ^e-torsion points, which in general may be defined over a degree ℓ^e extension of $\mathbb{F}_{p^2}$. To ensure that the ℓ^e torsion is defined over $\mathbb{F}_{p^2}$ one can choose p such that $\ell^e \mid (p \pm 1)$ as in the De Feo–Jao–Plût protocols. However, for general p, this translation algorithm would still be too expensive.

We solve this efficiency issue by using powersmooth degree isogenies in our protocols. When $n = \prod_i \ell_i^{e_i}$ with distinct primes ℓ_i, one reduces to the prime power case as follows. For simplicity we assume that 2 does not divide n. The isogeny of degree n can be decomposed into a sequence of prime degree isogenies. For simplicity we assume the isogeny steps are always performed in increasing degree order. However, rather than working with points on a sequence of elliptic curves, we work entirely on E_0. Using a Chinese Remainder Theorem-like representation, points in $E_0[n]$ can be represented as a sequence

of points in $E_0[\ell_i^{e_i}]$. When one wishes to compute the corresponding sequence of isogenies $\varphi_i : E_{i-1} \to E_i$, each of degree $\ell_j^{e_j}$, it is necessary to transport the appropriate kernel points across to E_{i-1} along the isogenies already computed.

Given a left $\mathcal{O}_0$-ideal I, Algorithm 2 progressively identifies the corresponding isogeny sequence. When determining points in $\ker(\alpha) \cap E_0[\ell_i^{e_i}]$ the algorithm uses a natural optimisation of reducing the coefficients of α modulo $\ell_i^{e_i}$.

Algorithm 2 Translating ideal to isogeny path

Input: $\mathcal{O}_0 = \mathrm{End}(E_0) = \langle 1, \phi, \frac{1+\pi\phi}{2}, \frac{\pi+\phi}{2} \rangle$, $I = \langle \alpha_1, \alpha_2, \alpha_3, \alpha_4 \rangle, n = \prod_{i=1}^{r} \ell_i^{e_i}$ with $2 \nmid n$.

Output: the isogeny corresponding to I through Deuring's correspondence.

for $i = 1, \ldots, r$ **do**
  Compute a basis $\{P_{i1}, P_{i2}\}$ for the $\ell_i^{e_i}$ torsion on E_0
  for $j = 1, 2$ **do**
    Compute P'_{ij} such that $P_{ij} = [2]P'_{ij}$
$\varphi_0 = [1]_{E_0}$
for $i = 1, \ldots, r$ **do**
  for $k = 1, 2, 3, 4$ **do**
    $\alpha_{ik} = \alpha_k$ with its coefficients reduced modulo $\ell_i^{e_i}$.
    Write $\alpha_{ik} = (u_{ik} + v_{ik}\mathbf{i} + w_{ik}\mathbf{j} + x_{ik}\mathbf{k})/2$.
    for $j = 1, 2$ **do**
      $P_{ijk} = [u_{ik}]P'_{ij} + [v_{ik}]\phi(P'_{ij}) + [w_{ik}]\pi(P'_{ij}) + [x_{ik}]\pi(\phi(P'_{ij}))$
  Solve ECDLP to compute Q_i of order $\ell_i^{e_i}$ such that $\alpha_{ik}(Q_i) = 0$ for all k
  Compute ϕ_i = isogeny with kernel $\langle \varphi_{i-1}(Q_i) \rangle$ (compute with Vélu's formulae).
  Set $\varphi_i = \phi_i \varphi_{i-1}$
Output $\varphi_0, \phi_1, \ldots, \phi_r$.

In our protocols we will have $\ell_i^{e_i} = O(\log n) = O(\log p)$. Moreover, we will be using $O(\log p)$ different primes. The complexity of Algorithm 2 under these assumptions is given by the following lemma. Note that almost all primes ℓ_i are such that $\sqrt{B} < \ell_i \leq B$ and so $e_i = 1$, hence we ignore the obvious ℓ-adic speedups that can be obtained in the rare cases when ℓ_i is small.

Lemma 29. *Let $n = \prod \ell_i^{e_i}$ with $\log n = O(\log p)$ and $\ell_i^{e_i} = O(\log p)$. Then Algorithm 2 can be implemented to run in time $\tilde{O}(\log^6 p)$ bit operations for the first loop, and $\tilde{O}(\log^5 p)$ for the rest of the algorithm.*

Proof. Without any assumption on p, the $\ell_i^{e_i}$ torsion points will generally be defined over degree $\ell_i^{e_i}$ extension fields, hence they will be of size $O(\log^2 p)$. However, the isogenies themselves will be rational, i.e. defined over $\mathbb{F}_{p^2}$. This means their kernel is defined by a polynomial over $\mathbb{F}_{p^2}$. Isogenies over $\mathbb{F}_{p^2}$ of degree d can be evaluated at any point in $\mathbb{F}_{p^2}$ using $O(d)$ field operations in $\mathbb{F}_{p^2}$.

Let $d = \ell_i^{e_i}$. To compute a basis of the d-torsion, we first factor the division polynomial over $\mathbb{F}_{p^2}$. This polynomial has degree $O(d^2) = O(\log^2 p)$. Using the algorithm in [113], this can be done in $\tilde{O}(\log^4 p)$ bit operations. Since the isogenies are defined over $\mathbb{F}_{p^2}$, this will give factors of degree at most $(d-1)/2$, each one corresponding to a cyclic subgroup. We then randomly choose some factor with a probability proportional to its degree, and we factor it over its splitting field, until we have found a basis of the d-torsion. After $O(1)$ random choices we will have a basis of the d-torsion. Each factorization costs $\tilde{O}(\log^5 p)$ using the algorithm in [162], and verifying that two points generate the d-torsion can be done with $O(d)$ field operations. It then takes $O(d)$ field operations to compute generators for all kernels. As $r = O(\log p)$ we deduce that the first loop requires $\tilde{O}(\log^6 p)$ bit operations.

Computing P_{ijk} involves Frobenius operations and multiplications by scalars bounded by d (and so $O(\log\log p)$ bits). This requires $O(\log\log p)$ field operations, that is a total of $\tilde{O}(\log^3 p)$ bit operations. Any cyclic subgroup of order $\ell_i^{e_i}$ is generated by a point $Q_i = aP_{i1} + bP_{i2}$, and the image of this point by α_{ik} is $aP_{i1k} + bP_{i2k}$. One can determine the integers a, b by an ECDLP computation or by testing random choices. There are roughly $\ell_i^{e_i} = O(\log p)$ subgroups, and testing each of them requires at most $O(\log\log p)$ field operations, so finding Q_i requires $\tilde{O}(\log p)$ field operations. Evaluating $\varphi_{i-1}(Q_i)$ requires $O(\log^2 p)$ field operations. Computing the isogeny ϕ_i can be done in $O(\log p)$ field operations using Vélu's formulae. As $r = O(\log p)$ we deduce that the second loop requires $\tilde{O}(\log^5 p)$ bit operations. □

We stress that in our signature algorithm, Algorithm 2 will be run $O(\log p)$ times. However the torsion points are independent of both the messages and the keys, so they can be precomputed. Hence the "online" running time of Algorithm 2 is $\tilde{O}(\log^5 p)$ bit operations per execution.

Translating an isogeny path to an ideal.

Let $E_0, E_1, \ldots, E_r$ be an isogeny path and suppose $\phi_i : E_{i-1} \to E_i$ is of degree $\ell_i^{e_i}$. We define $I_0 = \mathcal{O}_0$. Then, for $i = 1, \ldots, r$, we compute an element $\alpha_i \in I_{i-1}$

and an ideal

$$I_i = I_{i-1}\ell_i^{e_i} + \mathcal{O}_0\alpha_i$$

that corresponds to the isogeny $\phi_i \circ \cdots \circ \phi_1$. This is analogous in the powersmooth case to the notation I_i as used in Section 5.3.3, in particular $I_0 = \mathcal{O}_0 \supset I_1 \supset \ldots \supset I_r$. The idea is to determine suitable endomorphisms $\alpha_i \in I_{i-1}$ with the desired norm and that kill the required kernel point.

Algorithm 3 Translating isogeny path to ideal

Input: $E_0, E_1, \ldots, E_r$ isogeny path, $\phi_i : E_{i-1} \to E_i$ of degree $\ell_i^{e_i}$.
Output: the ideal path $I_0, \ldots, I_r$ corresponding to the isogeny path.

Let $I_0 = \mathcal{O}_0$
for $i = 1, \ldots, r$ **do**
    Find Q_i of order $\ell_i^{e_i}$ that generates the kernel of ϕ_i
    Compute $[\beta](Q_i)$ for all $\beta \in \{1, \mathbf{i}, \frac{\mathbf{i}+\mathbf{j}}{2}, \frac{1+\mathbf{k}}{2}\}$
    Let $\{\beta_1, \beta_2, \beta_3, \beta_4\}$ a basis of I_{i-1}
    Let $f_i(w, x, y, z) = n(w\beta_1 + x\beta_2 + y\beta_3 + z\beta_4)$
    repeat
        Pick a random solution to $f_i(w, x, y, z) = 0 \bmod \ell_i^{e_i}$
        Set $\alpha_i = w\beta_1 + x\beta_2 + y\beta_3 + z\beta_4$
    until $[\alpha_i](Q_i) = 0$
    Set $I_i = I_{i-1}\ell_i^{e_i} + \mathcal{O}_0\alpha_i$
    Perform basis reduction on I_i

In our protocols we will have $\ell_i^{e_i} = O(\log n) = O(\log p)$; moreover we will be using $O(\log p)$ different primes. The complexity of Algorithm 3 for these parameters is given by the following lemma.

Lemma 30. *Let $n = \prod_{i=1}^{r} \ell_i^{e_i}$ with $\log n = O(\log p)$ and $\ell_i^{e_i} = O(\log p)$. Assuming natural heuristics, Algorithm 3 can be implemented to run in expected time $\tilde{O}(\log^4 p)$ and the output is a $\mathbb{Z}$-basis with integers bounded by X such that $\log X = O(\log p)$.*

Proof. The input consists of a sequence of isogenies. Recall that the representation of an isogeny is usually done by explicitly specifying a kernel point (or else equivalent information, such as a polynomial whose roots are the kernel points). The $\ell_i^{e_i}$ torsion points will generally be defined over degree $\ell_i^{e_i}$ extension fields, hence they will be of size $O(\log^2 p)$. Isogenies of degree d can be evaluated at any point using $O(d)$ field operations.

When the degree is odd, the isogeny ϕ_i is naturally given by a polynomial ψ_i such that the roots of ψ_i correspond to the x-coordinates of affine points in $\ker \varphi_i$. To identify a generator Q_i, we first factor ψ_i over $\mathbb{F}_{p^2}$. Using the algorithm in [162] this can be done with $\tilde{O}(\log^3 p)$ bit operations. We choose a random irreducible factor with a probability proportional to its degree, we use this polynomial to define a field extension of $\mathbb{F}_{p^2}$, and we check whether the corresponding point is of order $\ell_i^{e_i}$. If not, we choose another irreducible factor and repeat. We expect to repeat this $O(1)$ times, and each step requires $\tilde{O}(\log p)$ bit operations. Therefore, the total cost for line 3 is $\tilde{O}(\log^3 p)$.

Step 4 requires $O(\log \log p)$ field operations to compute a point Q_i' such that $[2]Q_i' = Q_i$. After that, it mostly requires $O(\log p)$ field operations to compute the Frobenius map. The total cost of this step is therefore $\tilde{O}(\log^3 p)$.

Basis elements for all the ideals I_i appearing in the algorithm can be reduced modulo $\mathcal{O}_0 n$, hence their coefficients are of size $\log n = O(\log p)$.

To compute a random solution to f_i modulo $\ell_i^{e_i}$, we choose uniformly random values for w, x, y and, when the resulting quadratic equation in z has solutions modulo $\ell_i^{e_i}$, we choose a random one. As $\ell_i^{e_i} = O(\log p)$ the cost of this step can be neglected. Computing $[\alpha_i](Q_i)$ requires $O(\log \log p)$ operations over a field of size $O(\log^2 p)$. On average we expect to repeat the loop $O(\ell_i^{e_i}) = O(\log p)$ times, resulting in a total cost of $\tilde{O}(\log^3 p)$. Computing each f_i costs $\tilde{O}(\log p)$ bit operations.

As $r = O(\log p)$ the total cost of the algorithm is $\tilde{O}(\log^4 p)$. One can check that all integers in the algorithm are bounded in terms of n, and so coefficients are of size X where $\log X = O(\log n) = O(\log p)$. □

Recall that the condition $\log X = O(\log p)$ is needed in Lemma 28.

5.3.5 Classical signatures from the endomorphism ring computation

In this section, we give the details of our second signature scheme based on our new Sigma protocol, with security relying on computing the endomorphism ring of a supersingular elliptic curve.

Key Generation Algorithm: On input a security parameter λ generate a prime p with 2λ bits, which is congruent to 3 modulo 4. Let $E_0 : y^2 = x^3 + Ax$ over $\mathbb{F}_p$ be supersingular, and let $\mathcal{O}_0 = \mathrm{End}(E_0)$. Fix B, S_1, S_2 as

small as possible[7] such that $S_k := \prod_i \ell_{k,i}^{e_{k,i}}$, $\ell_{k,i}^{e_{k,i}} < B$, $\gcd(S_1, S_2) = 1$, and $\prod \left(\frac{2\sqrt{\ell_{k,i}}}{\ell_{k,i}+1}\right)^{e_{k,i}} < (p^{1+\epsilon})^{-1}$. Perform a random isogeny walk of degree S_1 from the curve E_0 with j-invariant $j_0 = 1728$ to a curve E_1 with j-invariant j_1. Compute $\mathcal{O}_1 = \mathrm{End}(E_1)$ and the ideal I corresponding to this isogeny. Choose a hash function H with t bits of output (e.g., $t = \lambda$ or, more conservatively, $t = 2\lambda$). The public key is $\mathsf{pk} = (p, j_1, H)$ and the secret key is $\mathsf{sk} = \mathcal{O}_1$, or equivalently I.

Signing Algorithm: On input a message m and keys $(\mathsf{pk}, \mathsf{sk})$, recover the parameters p and j_1. For $i = 1, \ldots, t$, generate a random isogeny walk w_i of degree S_2, ending at a j-invariant $j_{2,i}$. Compute $h := H(m, j_{2,1}, \ldots, j_{2,t})$ and parse the output as t challenge bits b_i. For $i = 1, \ldots, t$, if $b_i = 1$ use w_i and Algorithm 3 of Section 5.3.4 to compute the corresponding ideal I_i and hence its right order $\mathcal{O}_{2,i} = \mathrm{End}(E_{2,i})$, then use the algorithm of Section 5.3.3 on input II_i to compute a "fresh" path between $\mathcal{O}_0$ and $\mathcal{O}_{2,i}$, and finally use Algorithm 2 to compute an isogeny path w_i' from j_0 to $j_{2,i}$. If $b_i = 0$ set $z_i := w_i$, otherwise set $z_i := w_i'$. Return the signature $\sigma = (h, z_1, \ldots, z_t)$.

Verification Algorithm: On input a message m, a signature σ and a public key pk, recover the parameters p and j_1. For each $1 \leq i \leq t$ one uses z_i to compute the image curve $E_{2,i}$ of the isogeny. Hence the verifier recovers the j-invariants $j_{2,i}$ for $1 \leq i \leq t$. The verifier then recomputes the hash $H(m, j_{2,1}, \ldots, j_{2,t})$ and checks that the value is equal to h, accepting the signature if this is the case and rejecting otherwise.

We now show that this scheme is a secure signature.

Theorem 31. *The signature scheme described above is (classically) EU-CMA secure under the hardness of assumption 7, in the random oracle model.*

Proof. Assumption 7 implies that the previous scheme is a Sigma protocol for hard relation. Then, the result follows immediately from theorems 2 and 26. □

Efficiency: As the best classical algorithm for computing the endomorphism ring of a supersingular elliptic curve runs in time $\tilde{O}(\sqrt{p})$ one can take $\log p = 2\lambda$. By Theorem 20 and Lemma 21, taking $B \approx 2(1+\epsilon)\log p$ ensures that the outputs

[7] The exact procedure is irrelevant here.

of random walks are distributed uniformly enough. Random walks then require $2(1+\epsilon)\log p$ bits to represent, so signatures are

$$t + \frac{t}{2}\left(2(1+\epsilon)\lceil\log p\rceil + \frac{7}{2}\lceil\log p\rceil\right)$$

bits on average, depending on the challenge bits. For λ bits of security, we choose $t = \lambda$, so the average signature length is approximately $\lambda + (\frac{\lambda}{2})(4(1+\epsilon)\lambda + 7\lambda) \approx \frac{1}{2}(11+4\epsilon)\lambda^2 \approx \frac{11}{2}\lambda^2$. The conservative choice $t = 2\lambda$ gives signatures of size approximately $11\lambda^2$ bits.

Private keys are $2(1+\epsilon)\log p \approx 4\lambda$ bits if a canonical representation of the kernel of the isogeny between E_0 and E_1 is stored. This can be reduced to 2λ bits for generic E_1: if I is the ideal corresponding to this isogeny, it is sufficient to store another ideal J in the same class, and for generic E_1 there exists one ideal of norm $n \approx \sqrt{p}$. To represent this ideal in the most efficient way, it is sufficient to give n and a second integer defining the localization of I at every prime factor ℓ of n, for canonical embeddings of $B_{p,\infty}$ into $M_2(\mathbb{Q}_\ell)$, where $M_2(\mathbb{Q}_\ell)$ is the group of 2×2 matrices over the ℓ-adics. This reduces storage costs to roughly 2λ bits. Public keys are $3\log p = 6\lambda$ bits. A signature mostly requires t calls to the Algorithms of Sections 5.3.3 and 5.3.4, for a total cost of $\tilde{O}(\lambda^6)$. Verification requires to check $O(\lambda)$ isogeny walks, each one comprising $O(\lambda)$ steps with a cost $O(\lambda^2)$ field operations each when modular polynomials are precomputed, hence a total cost of $\tilde{O}(\lambda^6)$ bit operations (under the same heuristic assumptions as in Lemma 28).

Optimization with Non Backtracking Walks: In our description of the signature scheme we have allowed isogeny paths to "backtrack". We made this choice to simplify the convergence analysis of random walks and because it does not affect the asymptotic complexity of our schemes significantly. However in practice at any concrete security parameter, it will be better to use non-backtracking random walks as they will converge more quickly to a uniform distribution [6].

5.3.6 Post-quantum signatures from the endomorphism ring computation

We briefly describe the signature scheme arising from applying Unruh's transform to the Sigma protocol of Section 5.3.

Key Generation Algorithm: On input a security parameter λ generate a prime p with 4λ bits, which is congruent to 3 modulo 4. Let $E_0 : y^2 = x^3 + Ax$ over $\mathbb{F}_p$ be supersingular, and let $\mathcal{O}_0 = \mathrm{End}(E_0)$. Set $t = 3\lambda$. Fix B, S_1, S_2 as in the key generation algorithm of Section 5.3.5. Perform a random isogeny walk of degree S_1 from the curve E_0 with j-invariant $j_0 = 1728$ to a curve E_1 with j-invariant j_1. Compute $\mathcal{O}_1 = \mathrm{End}(E_1)$ and the ideal I corresponding to this isogeny.

Choose a hash function $H : \{0,1\}^* \to \{0,1\}^t$. Let $N_0 \approx 2\log p$ and $N_1 \approx \frac{7}{2}\log p$ be upper bounds for the bitlengths of the representations of isogeny paths in the algorithm, respectively in responses to challenges 0 and 1. For $i = 0, 1$ let $G_i : \{0,1\}^{N_i} \to \{0,1\}^{N_i}$ be a hash function such that every element has polynomially many preimages. The public key is $\mathsf{pk} = (p, j_1, H, G_0, G_1)$ and the secret key is $\mathsf{sk} = \mathcal{O}_1$, or equivalently I.

Signing Algorithm: On input a message m and keys $(\mathsf{pk}, \mathsf{sk})$, recover the parameters p and j_1. For $i = 1, \ldots, t$ generate a random isogeny walk w_i of degree S_2, ending at a j-invariant $j_{2,i}$.

For $i = 1, \ldots, t$ apply Algorithm 3 of Section 5.3.4 to compute the ideal I_i corresponding to the isogeny path w_i, then use the algorithm of Section 5.3.3 on input II_i to compute a "fresh" ideal corresponding to a path between $\mathcal{O}_0$ and $\mathcal{O}_{2,i}$, and finally use Algorithm 2 to compute an isogeny path w_i' from j_0 to $j_{2,i}$.

Compute $g_{i,0} = G_0(w_i)$ and $g_{i,1} = G_1(w_i')$ for $1 \le i \le t$, where the bitstrings w_i and w_i' are padded with zeroes to become binary strings of length N. Compute $h := H(m, j_1, j_{2,1}, \ldots, j_{2,t}, g_{1,0}, g_{1,1}, \ldots, g_{t,0}, g_{t,1})$ and parse the output as t challenge bits h_i. For $i = 1, \ldots, t$, if $h_i = 0$ then set $\mathrm{RSP}_i = w_i$ and if $h_i = 1$ then set $\mathrm{RSP}_i = w_i'$. Return the signature $\sigma = (h, \mathrm{RSP}_1, \ldots, \mathrm{RSP}_t, g_{1,1-h_1}, \ldots, g_{t,1-h_t})$.

Verification Algorithm: On input a message m, a signature σ and a public key pk, recover the parameters p and j_1.

For each $1 \le i \le t$ one uses RSP_i to compute the image curve $E_{2,i}$ of the isogeny (if $h_i = 0$ then RSP_i is a path from E_1 and if $h_i = 1$ then it is a path from E_0). Hence the verifier recovers the j-invariants $j_{2,i}$ for $1 \le i \le t$.

The verifier then computes $g_{i,h_i} = G_{h_i}(\mathrm{RSP}_i)$ for $1 \le i \le t$ (again padding to N bits using zeros). Finally the verifier computes the hash value

$$h' = H(m, j_1, j_{2,1}, \ldots, j_{2,t}, g_{1,0}, g_{1,1}, \ldots, g_{t,0}, g_{t,1}).$$

If $h' = h$ then the verifier accepts the signature and otherwise rejects.

We now show that this scheme is a secure signature.

Theorem 32. *The signature scheme described above is (quantumly) EU-CMA secure under the hardness of assumption 7, in the quantum random oracle model.*

Proof. Assumption 7 implies that the previous scheme is a Sigma protocol for hard relation. Then, the result follows immediately from theorems 3 and 26. □

Efficiency: For the same reasons as in the application of the Unruh transform to the De Feo-Jao-Plût scheme, this signature scheme is less efficient than its classical counterpart. Again, we only send half the values $g_{i,j}$, since the missing values can be recomputed by the verifier.

The average signature size is $t + t((2\log p + \frac{7}{2}\log p))$, on the basis that half the challenge bits are 0 and half of them are 1. For λ bits of security, we choose $\log p = 4\lambda$ and $t = 3\lambda$. Then the average signature size is approximately $66\lambda^2$.

5.3.7 Comparison

Tables 5.1 and 5.2 summarize the main efficiency features of the four signature schemes, based either on De Feo–Jao–Plût or on our new Sigma protocol, and using the Fiat–Shamir or Unruh transformations. The numbers provided were obtained by optimizing signature sizes first, then signing and verification time and finally key sizes; other trade-offs are of course possible. The scheme based on the De Feo–Jao–Plût Sigma protocol and Unruh transform was discovered independently in [167]; the version we give incorporates optimizations that reduce the signature sizes for the same security guarantees.[8] Signatures based on De Feo–Jao–Plût Sigma protocol are simpler and somewhat more efficient than signatures based on our new Sigma protocol. However, the latter have the advantage to rely on more standard and potentially harder computational problems. Schemes based on the Fiat–Shamir transformation are more efficient than schemes based on Unruh's transformation, but the latter provide security guarantees against quantum adversaries.

Table 5.1 and a quick comparison with RSA signatures suggest that isogeny-based signatures schemes may be efficient enough for practical use. Indeed, for RSA signatures, key sizes are cubic in the security parameter, and signing and verification times are respectively quasi-quadratic and quasi-linear in the key

[8] Both signature sizes depend linearly on a parameter t which we fixed in a more conservative manner than Yoo et al. With $t = 2\lambda$ their signatures are $69\lambda^2$ bits and ours are $48\lambda^2$ bits, and with $t = 3\lambda$ their signatures are $\lceil 103.5\lambda^2 \rceil$ bits and ours are $72\lambda^2$ bits.

	SK size	PK size	Signature	Signing	Verification
DFJP + FS	2λ	28λ	$6\lambda^2$	$\tilde{O}(\lambda^3)$	$\tilde{O}(\lambda^3)$
Sec 5.3 + FS	2λ	6λ	$\frac{11}{2}\lambda^2$	$\tilde{O}(\lambda^6)$	$\tilde{O}(\lambda^6)$
DFJP + U	3λ	42λ	$54\lambda^2$	$\tilde{O}(\lambda^3)$	$\tilde{O}(\lambda^3)$
Sec 5.3 + U	4λ	12λ	$66\lambda^2$	$\tilde{O}(\lambda^6)$	$\tilde{O}(\lambda^6)$

Table 5.1: Asymptotic efficiency of four signature schemes using De Feo–Jao–Plût and our Sigma protocol, and the Fiat–Shamir and Unruh transformations, as a function of the security parameter λ. All sizes are in bits and computation costs are in bit operations.

	128 bit			256 bit		
	SK	PK	Signature	SK	PK	Signature
DFJP + FS	256	3584	98304	512	7168	393216
Sec 5.3 + FS	256	768	90112	512	1536	360448
DFJP + U	384	5376	884736	768	10752	3538944
Sec 5.3 + U	512	1536	1081344	1024	3072	4325376

Table 5.2: Concrete efficiency of our signature schemes at security levels of 128 and 256 bits. Security level provided are against classical or quantum adversaries for schemes based on the Fiat–Shamir or Unruh transforms respectively. All sizes are in bits.

sizes (the latter assuming a small public key exponent is used), amounting to $\tilde{O}(\lambda^3)$ and $\tilde{O}(\lambda^6)$. As for concrete parameters, key sizes are much smaller for isogeny-based signatures than for RSA signatures and comparable to ECDSA signatures. Further work in this area should aim at decreasing signature sizes.

5.4 Conclusion

We have presented two signature schemes based on supersingular isogeny problems. Both schemes are built from a parallel execution of a Sigma protocol with bounded soundness, using the Fiat–Shamir transformation. The first scheme is built directly from the De Feo–Jao–Plût Sigma protocol with some optimization. A similar scheme was given by Yoo, Azarderakhsh, Jalali, Jao and Soukharev [167]. The second scheme is more involved, and introduces a new masking method for isogeny paths. A crucial ingredient for our second protocol is the quaternion isogeny algorithm of Kohel-Lauter-Petit-Tignol [118] in the powersmooth case, for which we provide a more complete description and analysis. The first scheme is significantly more efficient, but the second one is based on an arguably more standard and potentially harder computational problem.

Our schemes rely on problems that can potentially resist quantum algorithms. However, this family of problems is also rather new in cryptography. Among all of them, we believe that the problem of computing the endomorphism ring of a supersingular elliptic curve (on which our second signature scheme relies) is the most natural one to consider from an algorithmic theory point of view, and it was the subject of Kohel's PhD thesis in 1996 [117, Chapter 7]. The assumption is also potentially weaker than assumptions 11 and 13 considered in previous works (and used in our first signature scheme). Yet, even that problem is far from having received the same scrutiny as more established cryptography problems like discrete logarithms or integer factoring. We hope that this work will encourage the community to study its complexity.

Chapter 6

Isogeny encryption

This chapter is based on the paper 'SÉTA: supersingular encryption from torsion attacks' [57], which is a joint work with Cyprien Delpech de Saint Guilhem, Péter Kutas and Christophe Petit.

In 2011, Jao–De Feo [102] introduced the first isogeny-based public-key encryption scheme, based on a key agreement protocol. Their work was inspired by a construction of Stolbunov [152] for ordinary elliptic curves, with a switch to supersingular curves to thwart sub-exponential quantum algorithms that exist in the ordinary case [40].

The key agreement protocol follows a 'Diffie–Hellman-like' structure: Alice and Bob start from a public curve E_0 and choose random secret isogenies φ_1, φ_2 to reach curves E_1, E_2. Then they send the curves to each other and finally use their respective secrets to arrive at a common curve E_3, as shown in Figure 6.1.

In the supersingular case, the commutativity of this diagram is not immediately preserved as, say, Bob cannot evaluate his isogeny φ_2 on Alice's curve E_1 without some extra help. To solve this, Jao–De Feo proposed to send additional information in the protocol in the form of images of torsion points under the secret isogenies. With the help of these points, they ensured that each party could evaluate their secret isogeny on the other's curve.

However, as it happened in the previous chapter with the De Feo–Jao–Plût Sigma protocol, the isogeny problem upon which the security of the scheme is based on now differs from the original problem in several ways. First, it is a decisional problem, consisting on distinguishing E_3 from random, given

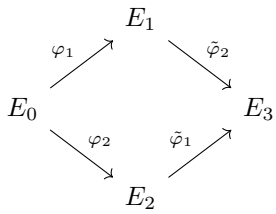


Figure 6.1: Sketch of the SIDH key agreement protocol.

E_0, E_1, E_2. This is analogous to the relation between the discrete logarithm and decisional Diffie–Hellman problems. Second, the adversary now has access to the images of some torsion points under the secret isogenies, which in principle could make the recovery of these isogenies easier. In addition, Jao–De Feo proposed to use special primes and rather small degree isogenies in their protocols to accelerate computations.

The introduction of this new hardness assumption fostered the proposal of new related assumptions that were used to prove the security of isogeny-based schemes. Many of these assumptions shared the need to reveal extra points. In particular, this family of assumptions and parameter choices are used in the SIKE submission[1] to the NIST process [7].

In 2017, Petit [139] studied the impact of the extra points in the hardness of these problems. He showed that for some choices of parameters, the problem could in fact be solved in polynomial time with classical algorithms. More precisely, Petit's algorithm solves the following problem: let E_0 be a special curve, for which the endomorphism ring is known, and let $\varphi : E_0 \to E$ be an isogeny of degree D. Let P, Q be a basis of the N-torsion of E_0. Then, given $E_0, E, \varphi(P), \varphi(Q)$, the problem is to compute φ. The algorithm's running time depends on the choices of D and N.

So far, Petit's techniques cannot be applied to the parameters proposed by Jao–De Feo, hence the proposed schemes [7,54,102] remain secure. Nevertheless and in anticipation of potential further cryptanalysis progress, it is desirable to design alternative cryptographic schemes that only rely on standard isogeny problems. This has so far only been achieved for signature schemes, as discussed in the previous chapter [53,75,76,151], and hash functions [38]. A special case

[1] In particular, SIKE is proven secure under the hardness of the "computational DH-like" isogeny problem (assumption 12), in the random oracle model.

is CSIDH [35], a key agreement protocol that relies on the original isogeny problem, but is restricted to supersingular elliptic curves over $\mathbb{F}_p$, and can be solved in quantum subexponential time.

More generally, any relaxation of the assumptions used in building isogeny-based PKE schemes and KEMs is of interest from a theoretical point of view, and could become crucial if further cryptanalysis progress occurs.

Our contributions. We provide new PKE schemes and KEMs based on isogeny problems. Key recovery security for our schemes only relies on the original isogeny problem for supersingular curves, and the standard OW-CPA and IND-CCA security rely on different problems than those used in SIDH and SIKE. We argue that, depending on future cryptanalysis progress, our schemes can provide an interesting alternative to the SIKE family.

We now briefly sketch the core idea of our constructions. Petit's algorithm crucially uses the fact that the endomorphism ring of E_0 is known in SIDH/SIKE. We exploit this fact to turn the attack into a decryption mechanism.

1. Let E_0 be a special curve as above. Alice takes a random isogeny $\varphi_s : E_0 \to E_s$ and publishes E_s as her public key, keeping φ_s as her secret key. A canonical method to compute a basis P, Q of the N-torsion of any E_s is also fixed as part of the scheme.

2. When Bob wants to send a message m to Alice, he encodes it into an isogeny $\varphi_m : E_s \to E_m$, creating the following diagram.

$$E_0 \xrightarrow{\varphi_s} E_s \xrightarrow{\varphi_m} E_m$$

   He sends $(j(E_m), \varphi_m(P), \varphi_m(Q))$ as the ciphertext.

3. To decrypt a message, Alice uses her secret isogeny φ_s and knowledge of the endomorphism ring of E_0 to compute endomorphisms of E_s. She can then recover the secret φ_m by running the attack on the ciphertext.

The endomorphism ring of E_s remains hidden to the adversary, so, even though the parameters are chosen to enable Petit's attack, it cannot be run unless $\mathrm{End}(E_s)$ is recovered. The task of recovering the endomorphism ring of a randomly sampled supersingular curve is also a hard problem, for which only exponential-time algorithms exist. As a consequence, an alternative secret key cannot be derived when given only E_0 and E_s.

Since we rely on Petit's attack for decryption, we send torsion point images that are larger than the ones used in SIDH, which suggests an easier underlying problem. However, a key difference is that there is no Diffie–Hellman-like structure in our case: we rely directly on the discrete logarithm-like problem, so in this sense our problem is harder.

We also deal with the algorithmic aspects of the construction, in particular addressing a potential timing dependency that arises from an uncommon case in which Petit's algorithm takes longer to recover the isogeny. We identify when this happens and tune our parameters to avoid this case completely.

We first build an OW-CPA secure PKE scheme and then we use a generic OAEP-style transformation to achieve IND-CCA security in the QROM. For KEMs, we present two alternative routes: one uses the transformations of [98], which work out of the box but have a non-tight security reduction; the other uses the work of [147], which has tighter reductions but requires the starting scheme to verify an additional property called sparse pseudorandomness.

Organization. We first briefly recall the SIDH/SIKE constructions, and recall the relevant generic transformations for encryption schemes, in section 6.1. We then present a generalization of the Charles-Goren-Lauter hash function and describe our construction as a trapdoor OWF, together with its inversion mechanism and the relevant algorithmic considerations, in section 6.2. In sections 6.3 and 6.4, we present the PKE schemes and KEMs, respectively. These three sections contain the core technical details of this work. In section 6.5, we discuss parameter selection and analyze the asymptotic complexity of our scheme. We finally compare our scheme with SIDH/SIKE in section 6.6, and conclude in section 6.7.

6.1 Preliminaries

6.1.1 SIDH and SIKE protocols

We give a high level description of SIDH and SIKE. We start with the original SIDH protocol of Jao–De Feo [102]. In the setup one chooses two small primes ℓ_1, ℓ_2 and a prime p of the form $p = \ell_1^{e_1}\ell_2^{e_2}f - 1$, where f is a small cofactor and e_1 and e_2 are large (in SIKE [7] they use $\ell_1^{e_1} = 2^{216}$, $\ell_2^{e_2} = 3^{137}$ and $f = 1$). Let E be the elliptic curve with j-invariant 1728.[2] Let R_1, S_1 be a basis of $E[\ell_1^{e_1}]$

[2]There is a less efficient variant in which a random curve E' is obtained through a random walk from E, and E' is used as the starting curve.

and let R_2, S_2 be a basis of $E[\ell_2^{e_2}]$. The protocol is as follows:

1. Alice chooses a random cyclic subgroup of $E[\ell_1^{e_1}]$ generated by $P_1 = m_1 R_1 + n_1 S_1$ and Bob chooses a random cyclic subgroup of $E[\ell_2^{e_2}]$ generated by $P_2 = m_2 R_2 + n_2 S_2$.

2. Alice computes the isogeny $\varphi_1 : E \to E/\langle P_1 \rangle$ and Bob computes the isogeny $\varphi_2 : E \to E/\langle P_2 \rangle$.

3. Alice sends the curve $E/\langle P_1 \rangle$ and the points $\varphi_1(R_2)$ and $\varphi_1(S_2)$ to Bob, and Bob similarly sends $(E/\langle P_2 \rangle, \varphi_2(R_1), \varphi_2(S_1))$ to Alice.

4. Alice and Bob both use the images of the torsion points to compute the shared secret which is the curve $E/\langle P_1, P_2 \rangle$ (e.g. Alice can compute $\varphi_2(P_1) = m_1 \varphi_2(R_1) + n_1 \varphi_2(S_1)$ and $E/\langle P_1, P_2 \rangle = E_2/\langle \varphi_2(P_1) \rangle$).

This key exchange protocol also leads to a PKE scheme in the same way as the Diffie–Hellman key exchange leads to ElGamal encryption. Let Alice's private key be the isogeny $\varphi_1 : E \to E/\langle P_1 \rangle$ and her public key be the curve $E/\langle P_1 \rangle$ together with the images of the torsion points $\varphi_1(R_2)$ and $\varphi_1(S_2)$. Encryption and decryption work as follows:

1. To encrypt a bitstring m, Bob chooses a random subgroup generated by $P_2 = m_2 R_2 + n_2 S_2$ and computes the corresponding isogeny $\varphi_2 : E \to E/\langle P_2 \rangle$. He computes the shared secret $E \to E/\langle P_1, P_2 \rangle$ and hashes the j-invariant of $E/\langle P_1, P_2 \rangle$ to a binary string s. The ciphertext corresponding to m is the tuple $(E/\langle P_2 \rangle, \varphi_2(R_1), \varphi_2(S_1), c := m \oplus s)$

2. In order to decrypt Bob's message, Alice computes $E/\langle P_1, P_2 \rangle$ and from this information computes s. Then she retrieves the message by computing $c \oplus s$.

This PKE scheme is IND-CPA secure [7,54,102]. In the SIKE submission [7], it is transformed using the constructions in [98, Section 3] to produce an IND-CCA secure KEM in the ROM.

6.1.2 Generic transformations for encryption

One common technique for boosting the security of encryption schemes is the use of well-known generic transformations, which can take weakly secure schemes and can upgrade them all the way to IND-CCA secure. The cost of this is

relatively small in terms of efficiency, as these transformations usually require to add a small number of hashes to the ciphertext. The downside is that these transformations often have highly non-tight security reductions. In this section, we reproduce those relevant to this work. All of these are based in the transformation by Fujisaki–Okamoto [69].

The next two transformations take a OW-CPA encryption scheme and produce an IND-CCA secure KEM in the quantum setting.

$\mathsf{QFO}_m^{\not\perp}$ transformation. We describe the $\mathsf{QFO}_m^{\not\perp}$ transformation from [98], which takes a OW-CPA secure PKE scheme and produces an IND-CCA secure key encapsulation mechanism. This is based on a previous transformation by Targhi-Unruh [155], which in turn is essentially a QROM secure version of the Fujisaki-Okamoto transformation [69]. Following the recommendation in [17, Section 16], we choose the variant with implicit rejection, that is, when the ciphertext is invalid, the decapsulation algorithm outputs a wrong key instead of $\perp$.

Let $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme, with message space $\mathsf{M} = \{0,1\}^\lambda$ and randomness space R. Also, let

$$G : \{0,1\}^\lambda \to \mathsf{R}, \qquad H : \{0,1\}^* \to \{0,1\}^\lambda, \qquad H' : \{0,1\}^\lambda \to \{0,1\}^\lambda$$

be three hash functions, modelled as random oracles. The $\mathsf{QFO}_m^{\not\perp}$ transformation outputs KEM presented in Figure 6.2.

$\overline{\mathsf{Setup}}(1^\lambda)$:	$\overline{\mathsf{Enc}}(pk)$:	$\overline{\mathsf{Dec}}(dk, c, d)$:
$(pk, sk) \leftarrow \mathsf{Setup}(1^\lambda)$	$m \leftarrow \mathsf{M}$	$m = \mathsf{Dec}_{sk}(\mathsf{c})$
$s \leftarrow \mathsf{M}$	$\mathsf{c} = \mathsf{Enc}_{pk}(m; G(m))$	**if** $\mathsf{c} \neq \mathsf{Enc}_{pk}(m; G(m))$ **or** $H'(m) \neq d$
return (pk, dk)	$d = H'(m)$	**return** $K = H(s, \mathsf{c}, d)$
$dk = (pk, sk, s)$	$K = H(m)$	**else return** $K = H(m)$.
	return (K, c, d)	

Figure 6.2: The $\mathsf{QFO}_m^{\not\perp}$ transform

Theorem 33 (Theorems 4.4 and 4.6 from [98])**.** *Let* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme with perfect correctness that is OW-CPA secure. Then the* $\mathsf{QFO}_m^{\not\perp}$ *transformation above produces a KEM that is IND-CCA secure in the quantum random oracle model. More precisely, for any quantum PPT adversary* $\mathcal{A}$ *there*

exists an adversary $\mathcal{B}$ such that

$$\mathsf{Adv}^{\mathsf{IND-CCA}}_{\mathsf{KEM},\mathcal{A}}(\lambda) \leq 8q^{3/2}\left(\mathsf{Adv}^{\mathsf{OW-CPA}}_{\mathsf{PKE},\mathcal{B}}(\lambda)\right)^{1/4},$$

where q is the number of queries made to any of the random oracles.

Note that the security reduction is highly non-tight.

SXY transformation. We now describe the SXY transformation, introduced in [147]. In this case, the security reduction is tight, but unlike other proposals, it requires an additional property from the original PKE scheme: *sparse pseudorandomness*. Informally, this means that the ciphertexts of a random message are computationally indistinguishable from uniformly random elements of the ciphertext space (pseudorandomness), and that at the same time the probability of a random element of the ciphertext space being a valid ciphertext is negligible (sparseness).

Definition 21 (Definition 3.2 from [147])**.** *A deterministic public-key encryption scheme $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$, with plaintext space M and ciphertext space C, is* sparse pseudorandom *if the following two properties are satisfied.*

- *Sparseness:*

$$\mathsf{Sparse}_{\mathsf{PKE}}(\lambda) := \max_{(pk,sk)\in\mathsf{Setup}(1^\lambda)} \frac{\#\mathsf{Enc}_{pk}(\mathsf{M})}{\#\mathsf{C}}$$

  is negligible in λ.

- *Pseudorandomness: for any PPT adversary $\mathcal{A}$,*

$$\mathsf{Adv}^{\mathsf{PR}}_{\mathsf{PKE},\mathcal{A}}(\lambda) := \mathsf{Pr}\begin{bmatrix}(pk,sk)\leftarrow\mathsf{Setup}(1^\lambda), m^*\leftarrow\mathsf{M},\\ \mathsf{c}_0^* = \mathsf{Enc}_{pk}(m^*), \mathsf{c}_1^*\leftarrow\mathsf{C}, & :\tilde{b}=b\\ b\leftarrow\{0,1\}, \tilde{b}\leftarrow\mathcal{A}(pk,\mathsf{c}_b^*)\end{bmatrix}$$

  is negligible in λ.

Let $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ be a sparse pseudorandom deterministic public-key encryption scheme with message space $\mathsf{M} = \{0,1\}^\lambda$ and ciphertext space C. Also, let

$$H : \{0,1\}^\lambda \to \{0,1\}^\lambda, \qquad H' : \{0,1\}^\ell \times \mathsf{C} \to \{0,1\}^\lambda$$

be two hash functions, modelled as random oracles. The SXY transformation outputs the following KEM:

$\overline{\mathsf{Setup}}(1^\lambda)$:	$\overline{\mathsf{Enc}}(pk)$:	$\overline{\mathsf{Dec}}(dk, c)$:
$(pk, sk) \leftarrow \mathsf{Setup}(1^\lambda)$	$m \leftarrow \mathsf{M}$	$m = \mathsf{Dec}_{sk}(\mathsf{c})$
$s \leftarrow \{0,1\}^\ell$	$\mathsf{c} = \mathsf{Enc}_{pk}(m)$	**if** $m = \perp$ **return** $K = H'(s, \mathsf{c})$
$dk = (pk, sk, s)$	$K = H(m)$	**if** $\mathsf{c} \neq \mathsf{Enc}_{pk}(m)$ **return** $K = H'(s, \mathsf{c})$
return (pk, dk)	**return** (K, c)	**else return** $K = H(m)$.

Figure 6.3: The SXY transform

Theorem 34 (Theorem 4.2 and Lemma 3.1 from [147])**.** *Let* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *be a deterministic PKE scheme with perfect correctness that is sparse pseudorandom. Assume that the ciphertext space* $\mathcal{C}$ *is efficiently sampleable. Then the SXY transformation above produces a KEM that is IND-CCA secure in the quantum random oracle model. More precisely, for any quantum PPT adversary* $\mathcal{A}$ *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{IND-CCA}}_{\mathsf{KEM},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda) + \mathsf{Sparse}_{\mathsf{PKE}}(\lambda) + 2^{\frac{-\ell+1}{2}} q_{H'},$$

where $q_{H'}$ *is the number of queries made to* H'*.*

6.2 Injective trapdoor OWFs from supersingular isogenies

We first present a generalisation of the CGL hash function [38] and then introduce a new family of trapdoor OWFs. We show that, for certain parameters, we can efficiently sample a statistically uniform function from the family and that any such function is injective and one-way. Finally, we show that sampling a function at random yields a trapdoor, i.e. a secret isogeny, which we can use to efficiently invert the function.

6.2.1 Charles-Goren-Lauter hash function

We now present the CGL hash function family as introduced in [38]. Essentially, the hash function works by taking a walk in the supersingular isogeny graph, with each choice determined by the input.

To select a hash function from the family, one selects a j-invariant $j \in \mathcal{J}_p$ which fixes a canonical curve $E/\mathbb{F}_{p^2}$ with $j(E) = j$. There are $\ell+1$ isogenies of degree ℓ connecting E to other vertices, so a canonical one of these is ignored and the other ℓ are numbered arbitrarily. Then, given a message $m = b_1 b_2 \dots b_n$, with $b_i \in \{1, \dots, \ell\}$, hashing starts by choosing a degree-ℓ isogeny from E according to symbol b_1 to arrive at a first curve E_1. Not allowing backtracking, there are then only ℓ isogenies out of E_1 and one is chosen according to b_2 to arrive at a second curve E_2. Continuing in the same way, m determines a unique walk of length n.

The output of the CGL hash function h_j is then the j-invariant of the final curve in the path, i.e. $h_j(m) \coloneqq j(E_n)$ where the walks starts at vertex j and is defined as above. We see that starting at a different vertex j' results in a different hash function $h_{j'}$.

We modify this hash function family in three ways.

- We consider a generalisation where we do not ignore one of the $\ell + 1$ isogenies from the starting curve E. That is, we take inputs $m = b_1 b_2 \dots b_n$ where $b_1 \in \{1, \dots, \ell + 1\}$ and $b_i \in \{1, \dots \ell\}$ for $i \geq 1$; this introduces a one-to-one correspondence between inputs and cyclic isogenies of degree ℓ^n originating from E.

- As in the previous chapter, we consider a generalisation where the walk takes place over different isogeny graph, for different small primes ℓ_i. Given an integer $D_m = \prod_{i=1}^{n} \ell_i^{e_i}$ where the ℓ_i are its prime factors, we introduce the notation

$$\mu(D_m) = \prod_{i=1}^{n} (\ell_i + 1) \cdot \ell_i^{e_i - 1}.$$

  We then take the message m to be an element of $\{1, \dots, \mu(D_m)\}$ represented as a tuple $(m_1, \dots, m_n)$, and each m_i is hashed along the graph of ℓ_i-isogenies. To ensure continuity, the j-invariants are chained along the hash functions, that is, we write $j_i = h_{j_{i-1}}(m_i)$, where j_{i-1} is the hash of m_{i-1}. Thus, only j_0 parametrises the overall hash function, which we denote by j. As before, this generalization returns the final j-invariant $j_n = h_{j_{n-1}}(m_n)$ as the hash of m.

- We also modify the CGL hash function to return the images of two given points under the D_m-isogeny φ_m from E_j to E_{j_n}.

For the rest of this work, as we will only make use of this family of generalised functions, we therefore refer by $\mathcal{H}^{p,D_m}$ to the hash function family

$$\mathcal{H}^{p,D_m} = \left\{ h_j^{D_m} : m, R, S \mapsto j(E_n), \varphi_m(R), \varphi_m(S) \right\}.$$

6.2.2 A new one-way function family

Given p, D_m and N, we define a family of functions

$$\mathcal{F}^{p,D_m,N} : \mathcal{J}_p \times \{1, \ldots, \mu(D_m)\} \to \mathcal{J}_p \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2,$$

which uses the generalised CGL hash function family $\mathcal{H}^{p,D_m}$. We define the function $f_j(m)$ to first compute the canonical curve E_j and compute a canonical basis (R_j, S_j) of the N-torsion group $E_j[N]$ (which is efficient if N is powersmooth). Next, the function computes $(j_c, R_c, S_c) = h_j^{D_m}(m, R_j, S_j)$. Succinctly, we have

$$f_j : m \mapsto \left(h_j^{D_m}(m, R_c, S_c) \right)$$

Statistically random sampling from the family. The starting curve E_0 with j-invariant $j(E_0) = 1728$ is fixed as part of the global parameters of the family $\mathcal{F}^{p,D_m,N}$. To select a random f_j from $\mathcal{F}$, a random isogeny of degree D_s, with cyclic kernel K_s, is chosen. This fixes $E_s \approx E_0/K_s$, and the corresponding j-invariant $j_s = j(E_s)$, thus fixing

$$f_{j_s} : [\mu(D_m))] \to \mathcal{J}_p \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2.$$

For well-chosen parameters, the statistically random sampling is guaranteed by theorem 20 and lemma 21.

Injectivity. We observe that, for the right choice of parameters, the functions are injective.

Lemma 35. *Let $N^2 > 4D_m$, then any function $f_j \in \mathcal{F}^{p,D_m,N}$ is injective.*

Proof. Suppose that a function f_j is not injective, i.e. that there are two distinct isogenies φ and φ' of degree D_m from E_j to E_c, corresponding to two distinct messages, with the same action on $E_j[N]$, implied by the colliding images of P_j and Q_j. Then, following [129, Section 4], their difference is also an isogeny between the same curves whose kernel contains the entire N-torsion. This, together with [150, Lemma V.1.2], implies that $4D_m \geq \deg(\varphi - \varphi') \geq N^2$. Taking $N^2 > 4D_m$ ensures that in fact $\varphi = \varphi'$ and therefore that f_j is injective. □

One-wayness. We now prove that the functions from this family are one-way under the hardness of an isogeny problem.

Lemma 36. *Let D_s be such that the distribution of j_s is statistically close to uniform. A function $f_j \in \mathcal{F}^{p,D_m,N}$ sampled at random as explained above, is quantum one-way under assumption 11, with isogeny degree $d = D_m$ and torsion degree N.*

Proof. Suppose that there is a PPT quantum adversary $\mathcal{A}$ that can break the one-wayness of f_j; that is, given j and $(j_c, R_c, S_c) = f_j(m^*)$ for $m^* \leftarrow [\mu(D_m)]$, $\mathcal{A}$ can recover m^* with non-negligible probability. We build a reduction $\mathcal{B}$ which receives a challenge $(E_1, E_2, \{R_i, S_i\}_{i=1,2}, \varphi(R_2), \varphi(S_2))$ for assumption 11, and returns φ.

The reduction passes $j(E_1)$ and $(j(E_2), \varphi(R_2), \varphi(S_2))$ to $\mathcal{A}$, who will return a corresponding input m with high probability. By reproducing the hashing of m, the reduction $\mathcal{B}$ can then recompute an isogeny $\tilde{\varphi}$ which is equivalent to φ. Note here that if m is a correct pre-image under the function f_j, then we are certain that it is the only one as, by Lemma 35, f_j is injective. With its knowledge of E_1, R_1 and S_1, $\mathcal{B}$ can then compute φ and return it.

□

We analyze the asymptotic cost of computing the one-way function in lemma 45, in section 6.5.3.

6.2.3 Computing inverses

In this section, we show how to use the algorithm of [139] to invert a given function $f_j \in \mathcal{F}^{p,D_m,N}$. We are given (j_c, R_c, S_c) as the output of $f_j(m)$ for some unknown m, and also the random isogeny $\phi_s : E_0 \to E_j$ of degree D_s used to select E_j at random. This gives us the composed isogeny

$$\phi = \phi_m \circ \phi_s : E_0 \to E_m$$

of degree $D = D_m D_s$, where ϕ_m is the walk determined by the message m, used in the computation of $f_j(m)$.

Computing ϕ_m given a suitable endomorphism of E_0.

In this section we assume that we know $\theta \in \mathrm{End}(E_0)$ and $d \in \mathbb{Z}$ such that $\mathrm{Tr}(\theta) = 0$ and $\deg(\phi \circ \theta \circ \hat{\phi} + [d]) = N$. Furthermore, we assume that D is odd, that $\gcd(D, N) = 1$ and that $-4\deg(\theta)$ is not a square modulo any prime divisor

of D. We will explain how to find such θ as part of the global parameters for our schemes in section 6.2.4; here we describe how to invert the function given such a θ.

Let $\psi = \phi \circ \theta \circ \hat{\phi} + [d] \in \mathrm{End}(E_m)$. We can compute ψ by the following method described in [139]. The endomorphism ψ has degree N and we know its action on $E_m[N]$, thus we can compute its kernel (since it is contained in $E_m[N]$). Since we are able to compute ψ, we can compute $\ker(\phi \circ \theta \circ \hat{\phi}) \cap E_m[D]$ efficiently. Now let $G = \ker(\phi \circ \theta \circ \hat{\phi}) \cap E_m[D]$. Lemma 37 below shows that in fact $G = \ker(\hat{\phi})$; from this we can recover first $\ker(\phi)$ and then $\ker(\phi_m)$, separating out ϕ_s. This then allows us to recover $m \in \{1, \ldots, \mu(D_m)\}$ which corresponds to $\ker(\phi_m)$. Algorithm 4 summarizes these steps in pseudocode.

Algorithm 4 Computing inverses

Require: $\mathsf{c}, \phi_s, \theta \in \mathrm{End}(E_0), d \in \mathbb{Z}$.
Ensure: $m \in \{1, \ldots, \mu(D_m)\}$ such that $f_{j_s}(m) = \mathsf{c}$.
1: Parse c as $(j_c, R_c, S_c) \in \mathbb{F}_{p^2} \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2$.
2: Compute the canonical curve $E_m = E_j$.
3: Let $\phi = \phi_m \circ \phi_s : E_0 \to E_m$.
4: Let $\psi = \phi \circ \theta \circ \hat{\phi} + [d] \in \mathrm{End}(E_m)$. ▷ Choices of θ and d ensure $\deg \psi = N$.
5: Compute $K_1 = \ker \psi \subseteq E_m[N]$ using d, θ, ϕ_s and $R_c, S_c \in E_m[N]$.
6: Compute $K_2 = \ker(\phi \circ \theta \circ \hat{\phi}) \cap E_m[D] = \ker(\psi - [d]) \cap E_m[D] = \ker(\hat{\phi})$.
7: Compute $\ker(\phi_m)$ using $\ker(\hat{\phi})$.
8: **return** m that corresponds to $\ker(\phi_m)$.

Lemma 37. *Let θ be such that $-\deg(\theta)$ is a quadratic nonresidue modulo every prime dividing D. Then G is cyclic and furthermore $G = \ker(\hat{\phi})$.*

Proof. It is clear that $\ker(\hat{\phi}) \subseteq G$ since it is contained in $\ker(\phi \circ \theta \circ \hat{\phi})$ and in $E_m[D]$ as well. We now show that G is cyclic. Let M be the largest divisor of D such that $E_m[M] \subseteq G$. Then ϕ can be decomposed as $\phi_{D/M} \circ \phi_M$. Then by [139, Lemma 5] the kernel of ϕ_M is fixed by θ. In the proof of [139, Lemma 6] it is shown that a subgroup of $E_0[M]$ can only be fixed by an endomorphism θ if $\mathrm{Tr}(\theta)^2 - 4\deg(\theta)$ is a square modulo M. Choosing θ as above therefore ensures that $M = 1$ which implies that G is cyclic. The order of G is a divisor of D since G is cyclic and every element of G has order dividing D. However, G contains $\ker(\hat{\phi})$ which is a group of order D. This implies that $G = \ker(\hat{\phi})$. □

We note that algorithm 4 runs in polynomial time, although we delay a

detailed complexity analysis until lemma 46 in section 6.5.3, after we have established the relations between the different parameters involved.

Avoiding a timing dependency.

The condition that $-\deg(\theta)$ is a quadratic nonresidue modulo every prime dividing D may seem strange at first, since in [139] the case when G is not cyclic is also considered. Without this condition, M will not always be equal to 1 and in that case the most time-consuming part of the algorithm is guessing a θ-invariant subgroup of $E_0[M]$—this is exponential in the number of prime factors of M and it can be expensive since D is powersmooth. In [139] it is shown that the expected running time of the attack remains polynomial time. This is however not sufficient for our purposes, as inversion could take a very long time on some inputs, and the variable inversion time creates a dependency between the input and the inversion time. By evoking this extra condition on θ and increasing the parameters slightly, we avoid a timing dependency entirely.

Detection of invalid inputs.

When provided with a valid ciphertext c, Algorithm 4 will always return the corresponding plaintext. To detect invalid inputs we proceed as follows. If any of the steps fails we return $\perp$ to indicate that the ciphertext is invalid. If the algorithm returns an output $\tilde{m}$ then we recompute the image $\tilde{\mathsf{c}}$ from it; if that matches the original c, then we return $\tilde{m}$ as a valid message; otherwise we return $\perp$.

6.2.4 Computation of the endomorphism

We now provide an algorithm for finding $\theta \in \mathrm{End}(E_0)$ which does not depend on ϕ_s or ϕ_m, only on their degrees, and can therefore be run as part of global parameter generation. This is essentially just a small modification of [139, Algorithm 2] but it is technical and may be skipped at a first reading.

The ring $\mathrm{End}(E_0)$ has an integral basis $\{1, i, \frac{ij+j}{2}, \frac{1+i}{2}\}$, with $i^2 = -p$ and $j^2 = -1$. The endomorphism ring contains the $\mathbb{Z}$-linear combinations of i, j, ij. We will be looking for θ in the form $ai + bj + cij$ with $a, b, c \in \mathbb{Z}$. This means that we are looking for a solution of the following Diophantine equation:

$$D^2(pa^2 + pb^2 + c^2) + d^2 = N \tag{6.1}$$

Furthermore, we need that $-4\deg(\theta)$ is a quadratic nonresidue modulo every prime divisor of D.

We make certain parameter restrictions which are partly necessary and partly for convenience. First we choose D to be odd since $-4\deg(\theta)$ is obviously a square modulo 2. We choose N to be a square modulo D^2, so the equation will be solvable modulo D^2 and we choose $N > D^5$. Let $D = \prod_{i=1}^{k} \ell_i^{e_i}$ be the prime decomposition of D, and let us denote by $T := \prod_{i=1}^{k} \ell_i$ the product of all distinct prime factors of D. We will also add the restriction that $D > T^3$. Let $A := pa^2 + pb^2 + c^2$. Algorithm 5 below computes a solution to Equation 6.1 such that $-A$ is a quadratic nonresidue modulo every prime number dividing D.

Algorithm 5 Computing θ

Require: D, N, p as above. Let T be the product of primes dividing D.
Ensure: solution to equation 6.1 such that $-A$ is a quadratic nonresidue modulo every prime dividing D.

1: Find u such that $u^2 \equiv N \pmod{D^2}$.
2: **for** every prime ℓ_i dividing D **do**
3: Let s_{ℓ_i} be a quadratic nonresidue modulo ℓ_i.
4: $r_i \leftarrow (s_{\ell_i} - \frac{-N+u^2}{D^2})(2u)^{-1} \pmod{\ell_i}$.
5: Compute a residue r modulo T with the property that $r \equiv r_i \pmod{\ell_i}$.
6: $\ell \leftarrow 0$.
7: $d \leftarrow D^2(T\ell + r) + u$.
8: $A \leftarrow \frac{N-d^2}{D^2}$.
9: **if** A is not a square modulo p **then**
10: $\ell \leftarrow \ell + 1$.
11: **go to** Step 7.
12: **else**
13: Find c such that $c^2 \equiv A \pmod{p}$.
14: **if** $\frac{A-c^2}{p}$ is a prime congruent to 1 modulo 4 **then**
15: Solve the equation $a^2 + b^2 = \frac{A-c^2}{p}$.
16: **else**
17: $\ell \leftarrow \ell + 1$.
18: **go to** Step 7.
19: **return** (a, b, c, d)

The following lemmas address the correctness and efficiency of algorithm 5.

Lemma 38. *Let A be the output of algorithm 5. Then $-A$ is a quadratic nonresidue modulo all ℓ_i.*

Proof. Let r_i, s_{ℓ_i} and u be as in Algorithm 5. Let r be an integer such that $r \equiv r_i \pmod{\ell_i}$. Then we show that for every i, the integer $\frac{-N+(D^2r+u)^2}{D^2}$ is not a quadratic residue modulo ℓ_i which implies that $-A$ is not a quadratic residue modulo every ℓ_i since $T\ell + r \equiv r_i \pmod{\ell_i}$ for every integer ℓ.

We have that

$$\frac{-N+(D^2r+u)^2}{D^2} = \frac{-N+u^2}{D^2} + D^2r^2 + 2ur.$$

By our choice of r we have that

$$\frac{-N+u^2}{D^2} + D^2r^2 + 2ur \equiv \frac{-N+u^2}{D^2} + 2ur_i \equiv s_{\ell_i} \pmod{\ell_i},$$

which is a quadratic nonresidue by the choice of s_{ℓ_i}.

□

Lemma 39. *Under plausible heuristic assumptions, algorithm 5 finds a solution to equation 6.1 with the required properties in polynomial time.*

Proof. Lemma 38 implies that $-(pa^2+pb^2+c^2)$ is a quadratic nonresidue modulo every ℓ_i. Observe that if $\ell < \frac{T}{2}$ we have that

$$N - (D^2(T\ell + r) + u)^2 > 0,$$

because of the conditions $N > D^5$ and $D > T^3$. This implies that whenever $\ell < \frac{T}{2}$ we have that $\frac{A-c^2}{p}$ in Step 13 is a positive number. Moreover, we can estimate the size of $\frac{A-c^2}{p}$ since

$$A = \frac{N - (D^2(T\ell + r) + u^2)^2}{D^2} \approx D^3,$$

which implies that $\frac{A-c^2}{p} \approx D^2$. By the prime number theorem and the Chebotarev density theorem we have that the number of primes smaller than D^2 and congruent to 1 modulo 4 is $O\left(\frac{D^2}{\log(D^2)}\right)$. Thus, after $O(\log p)$ iterations (which is much smaller than $\frac{T}{2}$) we will get that $\frac{A-c^2}{p}$ is a sum of two squares.

Finally, representing a prime number (congruent to 1 modulo 4) as a sum of two squares can be accomplished in polynomial time using Cornacchia's algorithm. All the other steps clearly run in polynomial time. □

Remark 3. *The proof implies that instead of having the two conditions $N > D^5$ and $D > T^3$ we could have had the condition $N > D^4T^3$.*

6.3 Public-key encryption scheme

We now build a PKE scheme using the family of trapdoor OWFs of Section 6.2 and show that it is OW-CPA secure; then we modify it to achieve IND-CCA security.

We define the $\mathsf{S\acute{E}TA}_{\mathsf{OW\text{-}CPA}}$ PKE scheme as the tuple $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ of PPT algorithms described below.

Parameters. Let λ denote the security parameter. Let E_0 be a fixed supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ with j-invariant $j(E_0) = 1728$. Let D_s, D_m and N be integers chosen according to the requirements of Section 6.2. Let $\theta \in \mathrm{End}(E_0)$ be computed as in Section 6.2.4. We let $\mathsf{params} = (\lambda, p, j_0, D_s, D_m, N, \theta)$.

Key generation. The $\mathsf{KeyGen}(\mathsf{params})$ algorithm proceeds as follows:

1. Sample a random cyclic subgroup $K_s \subseteq E_0(\overline{\mathbb{F}_{p^2}})$ of size D_s.
2. Compute the isogeny $\phi_s : E_0 \to E_s \coloneqq E_0/\langle K_s \rangle$.
3. Compute the j-invariant $j_s = j(E_s)$ and its canonical curve E_{j_s}.
4. Set $\mathsf{pk} \coloneqq j_s$ and $\mathsf{sk} \coloneqq K_s$.
5. Return $(\mathsf{pk}, \mathsf{sk})$.

Encryption. The $\mathsf{Enc}(\mathsf{params}, \mathsf{pk}, m)$ algorithm proceeds as follows. For a given $m \in \{0,1\}^{n_m}$, where $n_m = \lfloor \log_2 \mu(D_m) \rfloor$, first cast m as an integer in the set $\{1, \ldots, \mu(D_m)\}$ and then:

1. Parse $\mathsf{pk} = j_s \in \mathcal{J}_p$.
2. Compute $(j_c, R_c, S_c) \leftarrow f_{j_s}(m)$, where $f_{j_s} \in \mathcal{F}^{p, D_m, N}$.
3. Embed (j_c, R_c, S_c) as a binary string $\mathsf{c} \in \{0,1\}^{n_c}$ where n_c is sufficiently large to represent one j-invariant in $\mathcal{J}_p$ and two points in $E_{j_c}[N]$.
4. Return c.

Decryption. The $\mathsf{Dec}(\mathsf{params}, \mathsf{pk}, \mathsf{sk}, \mathsf{c})$ algorithm proceeds as follows:

1. Given $\mathsf{params}, \mathsf{sk}$ and $\mathsf{c} \in \{0,1\}^{n_c}$, parse c as $(j_c, R_c, S_c) \in \mathbb{F}_{p^2} \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2$; if that fails, return $\perp$.
2. Follow Algorithm 4 to recover $\tilde{m} \in \{1, \ldots, \mu(D_m)\}$; if this fails, set $\tilde{m} = \perp$.
3. If $\tilde{m} \neq \perp$; verify that $f_{j_s}(\tilde{m}) \stackrel{?}{=} \mathsf{c}$. If not, set $\tilde{m} = \perp$.
4. If $\perp$ was recovered, return $\perp$.
5. Otherwise, from $\tilde{m} \in \{1, \ldots, \mu(D_m)\}$, recover $m \in \{0,1\}^{n_m}$ and return it.

Theorem 40. *Let D_s be such that the distribution of j_s is statistically close to uniform. The PKE scheme described above is quantumly OW-CPA secure, under assumption 11.*

Proof. This follows directly from lemma 36. □

6.4 Key encapsulation mechanisms

We select two generic transformations to apply to our encryption scheme, obtaining an IND-CCA secure KEM in the QROM. The first works for any OW-CPA encryption scheme, but has the drawback a large tightness factor in the security reduction. The second has a tighter reduction, but requires the OW-CPA scheme to be *sparse pseudorandom*. We first provide a proof that our scheme satisfies this property and then use the two transformations to achieve IND-CCA security in the QROM. We refer to [17,98] for transformations in the classical ROM.

6.4.1 Sparse pseudorandomness

Recall that the SXY transformation [147] takes a weakly secure PKE and produces a CCA-secure KEM, with a tight security reduction. The downside is that it requires an additional property from the original PKE scheme: sparse pseudorandomness (definition 21).

We prove that our encryption scheme is sparse pseudorandom, under assumption 14. Recall that our encryption function is defined as

$$\mathsf{Enc}_{pk}(m) = (j(E_m), \varphi_m(R), \varphi_m(S)),$$

where $pk = E_s$ is a supersingular elliptic curve, $\{R, S\}$ is a basis of the N-torsion of E_s, and $\varphi_m : E_s \to E_m$ is the isogeny corresponding to the CGL hash function

with input m. The message space is $\mathsf{M} = \{0,1\}^n$. To guarantee that the two conditions above are satisfied, we must carefully choose the ciphertext space $\mathsf{C} \subseteq \mathcal{V} \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2$, where $\mathcal{V} = \mathcal{J}_p$ is the set of vertices of the supersingular isogeny graph. In particular, to have pseudorandomness we must ensure that there is no way to distinguish random elements of C from valid ciphertexts. We impose the following conditions on C:

- An element $(j(\overline{E}), \overline{R}, \overline{S}) \in \mathsf{C}$ must satisfy that $\overline{E}$ is isogenous to E and $\overline{R}, \overline{S} \in \overline{E}[N]$.
- The elements $\overline{R}, \overline{S}$ must be of order N and linearly independent.
- Note that $e(\varphi_m(R), \varphi_m(S)) = e(R,S)^{D_m}$, where e is the Weil pairing. Therefore $(j(\overline{E}), \overline{R}, \overline{S}) \in \mathsf{C}$ must satisfy that $e(\overline{R}, \overline{S}) = e(R,S)^{D_m}$.

Note that the third condition implies the second when N and D_m are coprime, which is the case for our constructions.

We now prove that our scheme is sparse pseudorandom.

Lemma 41. *Let $\epsilon > 0$. Assume that $p^{1-\varepsilon}N^3 > \mu(D_m)$ and D_m is large enough to ensure that the output of a random walk of degree D_m is close to uniform. Then the encryption scheme defined above is sparse in C.*

Proof. Our aim is to prove that $\#\mathsf{Enc}_{pk}(\mathsf{M})/\#\mathsf{C}$ is negligible. Since the encryption function is injective, we have that $\#\mathsf{Enc}_{pk}(\mathsf{M}) = \#\mathsf{M} = 2^{\lfloor \log \mu(D_m) \rfloor}$. On the other hand, $\#\mathsf{C}$ can be factored in the number of valid j-invariants times the number of valid pairs of points for each curve.

We observe that, if D_m is large enough, the mixing property of expander graphs ensures that the probability of ending a random walk of degree D_m at any j-invariant on the graph is bounded away from 0. Therefore the number of valid j-invariants is the size of the graph, which is $\lfloor p/12 \rfloor + k$ where $k \in \{0,1,2\}$.

For the number of valid pairs, we fix a supersingular j-invariant $j(\overline{E}) \in \mathcal{V}$. We observe that $\overline{E}[N] = \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$, and we are interested in finding how many choices of $(\overline{R}, \overline{S}) \in \overline{E}[N] \times \overline{E}[N]$ correspond to a valid ciphertext, that is, that they verify the pairing condition. There are roughly N^3 such pairs, as we have N^4 pairs in the torsion and we impose one equation on them.

Therefore

$$\frac{\#\mathsf{Enc}_{pk}(\mathsf{M})}{\#\mathsf{C}} \approx \frac{\mu(D_m)}{\frac{p}{12}N^3} < \frac{12}{p^{\varepsilon}},$$

which is negligible in the security parameter.

□

Proving pseudorandomness information-theoretically does not seem possible, given the result above, so we rely on a hardness assumption.

Lemma 42. *The encryption scheme defined above is pseudorandom under assumption 14.*

Proof. The pseudorandomness game is exactly distinguishing between the two distributions in assumption 14.

□

6.4.2 Applying the generic transformations

We are now in the conditions to apply both generic transformations to our encryption scheme of section 6.3, obtaining the following results in a straightforward way.

Corollary 43. *The scheme described in section 6.3, combined with the* $\mathsf{QFO}_m^{\not\perp}$ *transformation, is a quantumly IND-CCA secure KEM, under assumption 11, in the quantum random oracle model.*

Proof. Direct application of theorems 40 and 33. □

Corollary 44. *The scheme described in section 6.3, combined with the SXY transformation, is a quantumly IND-CCA secure KEM, under assumption 14, in the quantum random oracle model.*

Proof. Direct application of lemmas 41 and 42, and theorem 34. □

6.5 Parameter selection and efficiency

We first summarise the conditions on parameters for the scheme of section 6.3, ensuring security and efficient decryption, and then analyse the asymptotic costs and suggest concrete parameters.

6.5.1 Parameter requirements

Recall that λ is the security parameter, p is the characteristic of the field, D_s, D_m are the degrees of the secret key and message isogenies, respectively, N is the order of torsion points whose image is revealed, and T is the product of all distinct prime factors of $D = D_s D_m$.

Requirement	Condition
Efficiency of computations	$\log p = O(\lambda)$
Representation of N-torsion points	N powersmooth
Efficiency of key generation	D_s powersmooth
Efficiency of encryption	D_m powersmooth
Existence of θ	$D \equiv 1 \bmod 2$
Injectivity of functions	$N^2 > 4D_m$
Solvability of Diophantine equation	$D > p$ and $N > D^4$
Inversion is constant time	$N > D^5$, $D > T^3$ and $N \bmod D$ is square

Table 6.1: List of parameter conditions for efficiency.

Algorithmic requirements. We choose $\log p = O(\lambda)$ for efficient arithmetic. We require that N is powersmooth, with a powersmooth bound as small as possible, to efficiently represent N-torsion points. Key generation and encryption depend on performing a random walk in the isogeny graph. This can be done efficiently for isogenies of powersmooth degree. The conditions for efficient decryption and avoiding a timing dependency are discussed in section 6.2. In table 6.1, we list the conditions required for the efficiency of our algorithms.

Security requirements. Next, we focus on the conditions required for security. We first review the hardness of the computational problems involved, presented in section 2.4.5. Recall that assumption 7 can be broken in classical $\tilde{O}(\sqrt{p})$ time [58, 71]. By specifying the degree d of the isogeny to find, one can instead apply a claw-finding algorithm by computing all isogenies of degree $\sqrt{d}$ starting from E_1 and then looking for a collision with isogenies of degree $\sqrt{d}$ starting from E_2. Adapting the algorithms from [58, 71] results in $\tilde{O}(\sqrt{d})$ classical running time.

Assumption 10 can be broken in classical $\tilde{O}(\sqrt{p})$ time [71] and in quantum $\tilde{O}(\sqrt[4]{p})$ time [19]. We note that when $d > p$, it may actually be more efficient to break assumption 7 by first solving a related instance of assumption 10, independent of d and then computing the isogeny using the endomorphism ring instead of the claw-finding strategy. This may be the case in our setting, but since we are already considering explicitly assumption 10, we ensure that the choice of p is appropriate for security. For the ciphertext space to be sampleable, we also require N to be powersmooth.

Remark 4. *To achieve statistical uniformity of the j-invariants obtained through random walks, we must ensure that the walks are long enough, as discussed in*

Requirements	Condition
Assumption 10	$\log p \geq 4\lambda$
Assumption 11 for OW-CPA	$\log D_m \geq 2\lambda$
Sampleable $\mathcal{C}$	N powersmooth
Statistical uniformity of j_s, j_m	See Remark 4
Ciphertexts do not leak information	$\gcd(D, N) = 1$
Ciphertexts are sparse	$p \cdot N^3 > \mu(D_m)$

Table 6.2: List of parameter conditions necessary for security.

Section 6.2.2. This amounts to choosing D_s, D_m as the product $\prod \ell_i^{e_i}$, where $\ell_i^{e_i}$ are all the highest prime powers smaller than $2 \log p$, for all primes ℓ_i. However, recall that we also need N to be powersmooth, and at the same time $\gcd(D, N) = 1$, so we must distribute small primes between D and N. The simplest way is to alternate assigning a prime to D and one to N, in each case going up to the necessary bound imposed by the rest of the conditions. Alternative distributions of the primes could be considered to optimise computations.

Table 6.2 summarises the conditions required for security.

6.5.2 Concrete parameters

After reviewing parameter restrictions for efficiency and security we suggest concrete parameters.

The parameters that we need to specify is D_m, D_s, p, N and the endomorphism θ. To avoid specializing the problems in any way we choose a random large prime (450 bits) as opposed to a prime of a special form. First we give an example for the integer parameters. The numbers D_m, D_s and N are given by their prime decomposition to highlight their powersmoothness.

(a) $D_m = (17^8) \cdot (23^5) \cdot (31^5) \cdot (37^5) \cdot (53^3) \cdot (71^3) \cdot (73^4) \cdot (89^3) \cdot (97^3) \cdot (107^3)$

(b) $D_s = (101^2) \cdot (113^2) \cdot (811^3) \cdot (1229^2) \cdot (1291^2) \cdot (2153^2) \cdot 2999 \cdot 3313 \cdot 3323 \cdot 3517 \cdot 4007 \cdot 4889 \cdot 5209 \cdot 5557 \cdot 5623$

(c) $N = (21^8) \cdot (29^8) \cdot (41^8) \cdot (43^8) \cdot (59^8) \cdot (61^8) \cdot (67^8) \cdot (83^8) \cdot (103^8) \cdot (139^4) \cdot (149^4) \cdot (233^4) \cdot (283^4) \cdot (311^4) \cdot (443^4) \cdot (491^4) \cdot (599^4) \cdot (619^4) \cdot (631^4) \cdot (761^2) \cdot (1321^2) \cdot (1327^2) \cdot (1373^2) \cdot (1433^2) \cdot (1571^4) \cdot (1579^4) \cdot (1733^4) \cdot (1741^4) \cdot (1753^2) \cdot (1787^2) \cdot (1931^4) \cdot (2083^2) \cdot (2843^2) \cdot (2857^2) \cdot (2579^4) \cdot (2591^4) \cdot (2621^4) \cdot (2971^4) \cdot (3001^4) \cdot (3011^2) \cdot (3217^4) \cdot (3221^4) \cdot (3541^4) \cdot (3617^2) \cdot (3967^2) \cdot (4021^2) \cdot (4691^2) \cdot (5413^2) \cdot (6791^2) \cdot (7057^2) \cdot (7307^2) \cdot (7487^2) \cdot (7523^2) \cdot (7883^2) \cdot (6151^2) \cdot (6173^2) \cdot (6197^2) \cdot (7127^2) \cdot (8713^2) \cdot (8867^2) \cdot (9431^2) \cdot (9209^2) \cdot (8951^2) \cdot (9397^2) \cdot (9463^2) \cdot (9547^2) \cdot (9643^2) \cdot (9931^2) \cdot (10957^2) \cdot (11443^2) \cdot (11447^2)$

(d) $p = 230176781360103462133325777520657068921143060073775685635959971282821886726488206093893612689141113454628680660455129369525654117385259 1$

Now we turn our attention to θ. We implemented algorithm 5 in MAGMA [25] to compute a suitable solution of equation 6.1. We describe θ as a linear combination $ai + bj + cij$ as described in section 6.2.3. To make verification easier we also disclose the value d in the solution of equation 6.1:

1. $a = 4700046804358509319819862428243413283089600278375902907438377421082196898538929595378818129254297377088456585243627941929029192418234866548 7$
2. $b = 3098519396547805461012636243729083354843511120506702327385144248674792964230417880936080279712111562524815125415610483084803741597403096780 8$
3. $c = 3067668759255653909672530661908326434136489871369991357662318645291546831673839677853088182832098785291916003831085150626387002702688 19$
4. $d = 7166194938731789784593922401516621878685989320215035147302628432684449117257520669288979589497036094977019772975131377270923771558593024783878753050234241777558122190631021305595744469656083026107381185177047617078746203145803384316463965668566108399311752016825531224628633496234647956882453339473372623136494929818982771232391604517046351 5$

Running algorithm 5 took less than 2 minutes on a standard laptop, which makes the generation of θ efficient, as this only has to be computed once at the parameter generation phase.

6.5.3 Efficiency analysis

In this subsection we give an asymptotic analysis of the proposed one-way function and the schemes derived from it. We analyze the cost of computing and inverting the function.

Lemma 45. *With the choices of parameters of section 6.5.1, computing the one-way function of section 6.2.2 has a cost of* $\tilde{O}(\log^{4.5} p)$ *bit operations.*

Proof. The main cost of evaluating the one-way function is evaluating the isogeny ϕ_m at the torsion points P and Q. Since N is powersmooth, P and Q can be represented as sum of points of order $O(\log p)$. Furthermore, every prime divisor of N is also of size $O(\log p)$. Thus, first we give an estimate of computing the

image of a point R of order $O(\log p)$ under an isogeny of degree ℓ, where ℓ is a prime divisor of N.

The isogeny ϕ_m is defined over $\mathbb{F}_{p^2}$ and R is defined over an extension field $\mathbb{F}_{p^r}$ where $r = O(\log p)$. Evaluating a degree ℓ isogeny on R takes $\tilde{O}(\sqrt{\ell})$ field operations in $\mathbb{F}_{p^r}$, using the techniques of [15]. This translates to $\tilde{O}(r\sqrt{\ell}\log p)$ bit operations. Since N has $O(\log p)$ prime factors, this amounts to a total complexity of $\tilde{O}(\log^{3.5} p)$ bit operations for evaluating ϕ_m on a point R. Therefore, evaluating ϕ_m on P and Q requires $\tilde{O}(\log^{4.5} p)$ bit operations, using fast finite field arithmetic. □

Lemma 46. *With the choices of parameters of section 6.5.1, inverting the one-way function of section 6.2.2 (algorithm 4) has a cost of $\tilde{O}(\log^{4.5} p)$ bit operations.*

Proof. The most costly part of inverting the one-way function is the computation of the intersection of the kernel of $\psi - [d]$ and $E_m[D]$. This involves two major steps:

- Evaluating $\psi - [d]$ on a basis of the D-torsion, which allows for representing $\psi - [d]$ as 2×2 matrix M with entries from $\mathbb{Z}/D\mathbb{Z}$. This can be accomplished with $O(\log^{4.5} p)$ bit operations as it is a similar isogeny evaluation problem as discussed in lemma 45.

- Finding M explicitly and computing its kernel. This amounts to solving a discrete logarithm problem in a smooth rank 2 group, which can be done with a variant of the Pohlig–Hellman algorithm. For each ℓ prime divisor of D, naively this has a cost of $O(\log^2 p)$ operations in $\mathbb{F}_{p^r}$. However, we observe that the group orders are very smooth, which makes the exponentiation computation the most costly part of the Pohlig–Hellman algorithm. By reusing computations here, we can solve each discrete logarithm with $\tilde{O}(\log p)$ operations in $\mathbb{F}_{p^r}$. Recall that $r = O(\log p)$, so this yields a total cost of $\tilde{O}(\log^4 p)$ bit operations.

□

Note that the two lemmas above essentially give the cost of encryption / encapsulation and decryption / decapsulation, respectively, of the schemes of sections 6.3 and 6.4. This is due to the fact that all the schemes mostly consist of running and inverting the one-way function, plus a small number of hash function evaluations, depending on the case.

Communication costs.

The output of the one-way function is composed of a j-invariant $j_c \in \mathbb{F}_{p^2}$, which can be represented with $2\log p$ bits, and two torsion points $P_c, Q_c \in E_{j_c}[N]$, each of which can be represented with $2\log N$ bits by identifying each N-torsion point with a pair of elements in $\mathbb{Z}_N$. Therefore, the bit size of a ciphertext is

$$2\log p + 4\log N.$$

Further compression is possible, representing both torsion points with $3\log N$ bits, using the techniques in [46, Section 6.1].

The communication overhead of each of the schemes of Sections 6.3 and 6.4 is just a small number of hashes.

6.5.4 Road-map to greater efficiency

The estimates of lemmas 45 and 46 hold for conservative parameter choices and generic primes. We describe how using special primes can improve the efficiency of evaluation and inversion. If one manages to find a prime p where both N and D are defined over small extension fields (e.g., $\mathbb{F}_{p^4}$), then isogeny evaluation becomes a lot cheaper. Indeed, evaluating an isogeny of degree ℓ would take $O(\sqrt{\ell}\log p)$ bit operations and so evaluating and inverting the one-way function has an asymptotic complexity similar to SIKE. One has to note that having N defined over a small extension speeds up evaluation and having D defined over a small extension speeds up inversion. Applying the methods used for parameter selection in [45] and in [55] could potentially apply here as well. We leave the task of finding practical parameters and an efficient implementation for further work. Recent results [26, 120] improve on the attack from [139]. These results might reduce the unbalancedness between N, D in in this work.

6.6 Comparison with SIDH/SIKE

Prior to this work, the main method to obtain a PKE scheme from supersingular isogenies was to adapt the original key agreement protocol of [102] in an ElGamal fashion, as described in [54, Section 3.3]; we will refer to this as the SIDH encryption scheme. The SIKE KEM is derived from it through generic transformations. In the SIDH encryption scheme, key generation resembles a partial key agreement where one party generates their secret isogeny and publishes the target curve, together with the images of a torsion basis, as its long-term static key. In this section, we compare SÉTA with SIDH encryption and SIKE.

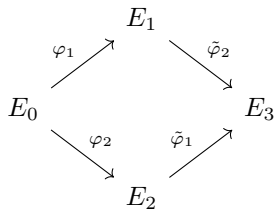


Figure 6.4: Sketch of the SIDH key agreement protocol.

6.6.1 Security

The security of SÉTA relies on the hardness of isogeny problems different from those of SIDH encryption or SIKE; future cryptanalysis progress could affect SIKE without affecting our schemes.

Encryption schemes. The IND-CPA security of the original encryption schemes of [54, 102] and their version in the SIKE specifications document [7] rely on the supersingular isogeny DDH and CDH problems, respectively; that is, given E_0, E_1, E_2 as in Figure 6.4 and the corresponding images of torsion points, respectively distinguish E_3 from random or compute E_3. Our work approaches the original "discrete logarithm"-like assumption (given two curves, compute an isogeny between them) as we reduce OW-CPA security to the hardness of this problem with additional images of torsion points. While OW-CPA is a weaker notion than IND-CPA, the generic transformations in the QROM will provide us with IND-CCA security anyway. We note that SIKE also uses the QROM, even for IND-CPA security. In both cases, the reductions to the respective hard problems are tight.

Importantly, in SIDH-based schemes the starting curve E_0 is fixed for efficiency reasons, and the schemes do not benefit from the additional hardness of isogeny problems that comes from a random starting curve. Furthermore, the curves E_0 and E_1 are somewhat close in the underlying isogeny graph because of the chosen degrees. In contrast, the security of our scheme of section 6.3 benefits from the full hardness of the problem and the use of longer isogenies.

However, since our schemes use images of larger torsion groups, our assumption is not formally weaker that the isogeny DDH and CDH assumptions that SIDH relies on. Nevertheless, our scheme could prove to be more valu-

able, depending on the direction in which cryptanalysis progresses. We consider different scenarios:

- Petit's attacks are improved to work with SIDH parameters. This will force SIDH to move to the less efficient setting of starting with a random curve.
- The requirement to know non-scalar endomorphisms of the starting curve is removed from Petit's attack. This would render both SIDH and SÉTA insecure but it would be a significant new attack.
- A new attack exploits the Diffie–Hellman structure of SIDH (Petit's attack does not). If knowledge of non-scalar endomorphisms was required, SIDH would need to use random starting curves. If not, SIDH would not remain secure, whereas SÉTA would. To the best of our knowledge, no attack of this kind is known at the moment.

Considering key recovery (definition 3) we see that, for [54], it is directly related to the CSSI assumption [54, Problem 5.2], as recovering the secret isogeny enables any attacker to complete the key agreement and decrypt the message. Not only does this problem include the torsion point images, which means that it can be weak against Petit's attacks [139], but the static nature of the key also opens the scheme to active attacks [74].

In contrast, our scheme of section 6.3 does not suffer from this; the torsion point images that we reveal depend only on the plaintext. Indeed, the key recovery problem for our scheme consists of recovering an equivalent isogeny between the curves E_0 and E_s *without additional torsion information* (and thus is protected by assumption 7), or, equivalently directly computing the endomorphism ring of E_s (assumption 10)—either of these options would allow to directly evaluate the endomorphism $\phi_s \circ \theta \circ \hat{\phi}_s \in \mathrm{End}(E_s)$ in the inversion algorithm of section 6.2.3. This guarantees stronger key-recovery security to our schemes, in contrast to SIDH and its variants. We formalise this in the following result.

Theorem 47. *The scheme of section 6.3 is quantumly secure against key recovery attacks under assumption 7 for the curves E_0 and E_s.*

Proof. Let $\mathcal{A}$ be an adversary against key recovery. Given E_1 and E_2 as in the statement above, $\mathcal{B}$ computes $j_s = j(E_2)$ and submits j_s to $\mathcal{A}$ as the public key. When $\mathcal{A}$ returns an alternative secret key sk', $\mathcal{B}$ checks that it is valid and returns it as a solution.

□

Scheme	Security	Assumption	Tightness	Model
SIDH encryption	IND-CPA	SSDDH	ε	Standard
SIDH enc. (SIKE spec.)	IND-CPA	SSCDH	$2q\varepsilon$	ROM
$\mathsf{SÉTA}_{\mathsf{OW\text{-}CPA}}$	OW-CPA	RCSSI*	ε	Standard
$\mathsf{SÉTA}_{\mathsf{IND\text{-}CCA}}$	IND-CCA	RCSSI*	$(2q)^{15/8}\varepsilon^{1/8}$	ROM
SIKE	IND-CCA	SSCDH	$\frac{q}{2^r} + 6q\varepsilon$	ROM
$\mathsf{SÉTA} + \mathsf{QFO}^{\not\perp}_m$	IND-CCA	RCSSI*	$8q^{3/2}\varepsilon^{1/4}$	QROM
$\mathsf{SÉTA} + \mathsf{SXY}$	IND-CCA	RCSSI*	$\frac{q}{2^r} + \varepsilon + \varepsilon'$	QROM

Table 6.3: Security comparison of schemes. Our instance of RCSSI (with larger torsions) does not formally imply the instances of SSDDH and SSCDH. The tightness column gives (simplified) upper bounds on the advantage against the security of the scheme; ε denotes the advantage for the underlying problem, ε' denotes the sparseness of the encryption scheme, q denotes the number of queries to hash functions, $r = \Theta(\lambda)$.

Table 6.3 summarises the security comparison between our schemes and the SIDH encryption variants.

Key encapsulation mechanisms. Most of the differences between our KEM and SIKE are inherited from those between our encryption scheme and the encryption scheme derived from SIDH. In particular the security of our KEM relies on different problems, as discussed above.

Generic transformations are used both by SIKE and SÉTA to achieve IND-CCA security. SIKE makes use of those in [98], which work out of the box, and we do the same in Section 6.4.2. However, we study security in the QROM, whereas SIKE focuses on ROM security. Most QROM transformations are highly non-tight, so we also consider another transformation from [147] which provides tightness at the expense of a stronger starting property. To the best of our knowledge, this approach has not yet been applied to SIKE. This security comparison is also summarised in Table 6.3.

6.6.2 Efficiency tradeoffs

In the choices for security-efficiency trade-offs, SIKE tends to aim for the latter whereas we tend to the former. Here we briefly discuss relevant design options

applicable to both SIDH and SÉTA.

Using special primes improves efficiency as points are defined over a smaller torsion; however the impact on security is not known. SIKE uses special primes but could use generic primes at a significant practical cost. On the other hand, SÉTA could use special primes to improve efficiency.

Shorter random walks also improve efficiency and allow smaller torsion, but they directly reduce security. In SIKE, the curves are relatively close, as only square root of all curves can be reached with the random walk. One could use larger walks, as in SÉTA, at the cost of using larger isogenies or extensions. (B-SIDH [45] offers significant improvements in that direction.)

Petit's attack only works when the endomorphism ring of the curve is known. SIKE uses such a curve, although it could start from a random curve at some efficiency cost, whereas SÉTA uses a random curve by design. We leave the analysis of the efficiency and implementation of these trade-offs for further work.

6.7 Conclusion

This chapter introduced a new trapdoor mechanism for isogeny-based cryptography which constructively uses Petit's techniques of computing secret isogenies using torsion point information. Public-key encryption schemes and key encapsulation mechanisms are then derived and other transformations are proven secure in the quantum random oracle model. Compared to protocols derived from SIDH [7, 54], our protocols rely on computational problems that may be more likely to withstand future cryptanalysis. In particular, key recovery security reduces to the original isogeny problem for supersingular elliptic curves.

Chapter 7

Trapdoor DDH groups

This chapter is based on the paper 'Trapdoor DDH groups from pairings and isogenies' [121], which is a joint work with Péter Kutas and Christophe Petit, and was published at SAC 2020.

The hardness of computing discrete logarithms and related problems (including the computational and decisional Diffie–Hellman problems in various groups) has supported the security of numerous cryptographic protocols for more than 40 years. While the decisional Diffie–Hellman (DDH) problem can be solved by solving a discrete logarithm problem, the converse is not known to be true. There are instances of groups equipped with bilinear pairings, where the discrete logarithm problem is believed to be hard but the decisional Diffie–Hellman problem can be solved efficiently.

Trapdoor DDH groups are a cryptographic primitive introduced by Dent–Galbraith in 2006 [59]. Formally, a trapdoor DDH group involves *two* descriptions of a single group. With either description of the group, the usual group operations, including inversion, can be computed efficiently, and solving the discrete logarithm problem and computational Diffie–Hellman problem must be hard. Crucially, the *decisional* Diffie–Hellman problem must also be hard to solve when provided only with the first description of the group, and easy with the second description. The second description can then be used as a trapdoor in a cryptographic protocol, conferring to its owner the power to solve DDH instances.

To the best of our knowledge, there are only two constructions of trapdoor

DDH groups in the literature. Dent–Galbraith [59] use supersingular elliptic curves with equations $y^2 = x^3 + x$ defined over RSA rings $\mathbb{Z}_N$. Another construction by Dent–Galbraith was broken in [131]. Seurin [148] uses the group of quadratic residues modulo N^2 where again N is an RSA modulus.

Two more constructions based on the RSA and factoring assumptions are provided by Seurin [148], but these are *static* trapdoor DDH group constructions, where the trapdoor can only solve DDH challenges involving a fixed pair of group elements g, g^x.

Trapdoor DDH groups have been used by Dent–Galbraith to build an identification scheme [59], and by Prabhakaran–Xue to build statistically hiding sets [144]. Seurin further constructs convertible undeniable signature schemes with delegatable verification from *static* trapdoor DDH groups [148]. In his paper, Seurin identifies several features that existing constructions (including his) are lacking, and which could be key to enable "powerful applications of trapdoor DDH groups" [148, Section 1.4].

Our results. We provide a new construction of trapdoor DDH groups which has all the features identified by Seurin. Our construction uses a *random* supersingular curve with a large prime as the group order, and an isogeny between this curve and a curve with a known distortion map as a trapdoor. Security relies on the hardness of solving the Decisional Diffie–Hellman problem on a random supersingular elliptic curve, and the hardness of solving the Computational Diffie–Hellman problem when the trapdoor is known. Interestingly, hardness of DDH implies both hardness of the discrete logarithm problem on the curve and hardness of computing an isogeny between a random supersingular curve and a "special" one, with a known distortion map.[1] Our construction solves all open problems of Seurin [148]: the group has public and prime order, hashing onto the group is efficient, and the trapdoor DDH solver always outputs the correct result.

We also provide attacks on the parameters suggested by Dent–Galbraith in their remaining construction, when used in specific applications. We explain how to increase the parameters or modify the scheme to thwart the attack. While these counter-measures defeat both our attacks and previous attacks, we argue that choosing secure parameters for this construction remains a delicate task.

[1] We stress that DDH is easy for a supersingular curve with a known distortion map, but finding a distortion map on a random curve is believed to be a hard problem [61, 140]. See also Section 7.1.1.

As an additional contribution, we formally define a notion of *trapdoor pairings* which was only implicit in the work of Dent–Galbraith. A trapdoor pairing construction immediately leads to a trapdoor DDH construction, and our new trapdoor DDH groups are in fact trapdoor pairings. However by using trapdoor pairings we are able to improve the efficiency of an identification protocol provided in [59] as an application, while relying on a seemingly weaker computational assumption.

Related works. Pairings and isogeny problems have both considerable applications in cryptography, and since they are both built on elliptic curves, combining them to construct further protocols is a natural idea.

The first work in that direction is due to Koshiba and Takashima [119]. They provided a framework and security definitions for cryptographic protocols involving pairings and isogenies, called *isogenous pairing groups*. They also present key-policy attribute-based encryption schemes based on their framework. We remark that our trapdoor DDH construction does not entirely fit in Koshiba and Takashima's framework: in our construction the pairing is "hidden" and hard to evaluate, whereas in their framework the pairing can be publicly evaluated. Besides, the framework implicitly uses an asymmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with $\mathbb{G}_1 \neq \mathbb{G}_2$, while we use a symmetric pairing. Finally, we remark that their framework seems to be built with the publicly computable Weil pairing in mind (the Weil pairing is degenerate when $\mathbb{G}_1 = \mathbb{G}_2$), and our construction uses a modified Weil pairing instead.

More recently, De Feo, Masson, Petit and Sanso have constructed a Verifiable Delay Function (VDF) that also uses both pairings and isogenies [56]. Similar ideas have been used to build a Delay Encryption scheme [30]. As in our new trapdoor DDH group, the VDF and Delay Encryption use an isogeny from a "special" supersingular elliptic curve to another "random" curve, and a pairing on the image curve. These constructions crucially differ from ours as their isogeny is not secret and it is of extremely large degree (and of course VDFs, Delay Encryption schemes and Trapdoor DDH groups are different primitives). The pairing used is also the Weil pairing, so it cannot be used to solve DDH instances.

Additionally, the patent [122] presents various trapdoor DDH constructions. The first few are based on the ideas of Dent and Galbraith [59], but the last one actually uses a secret isogeny to produce a trapdoor DDH mechanism. The high-level idea is similar to ours, in particular using an isogeny as the trapdoor information used to compute a pairing. However, they do not provide details

on a concrete instantiation, or formal security and efficiency analyses.

Organization. This chapter is organized as follows. In section 7.1 we provide preliminary background on trapdoor DDH groups and related notions, and previous constructions. In section 7.2 we describe our new trapdoor DDH group and we introduce the definition of trapdoor pairing, which the new construction satisfies, and briefly discuss applications. In section 7.3, we describe two concrete instantiations of our construction. We also analyze security and suggest concrete parameters. We describe our attacks on Dent–Galbraith's first construction in section 7.4, and we conclude the chapter in section 7.6.

7.1 Preliminaries

7.1.1 Computational assumptions

Discrete logarithm problem. Menezes-Okamoto-Vanstone [130] proposed the following method (referred to as MOV reduction) for reducing the discrete logarithm problem on elliptic curves to the discrete logarithm problem on finite fields. Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ and let P be a point of order n. Let Q be a point from the subgroup generated by P. Recall that use implicit group notation, that is, we denote the generator P by $[1]$ and Q by $[x]$, where $x = \log_P Q$.

In order to find x, the idea is to find the smallest integer k (called the embedding degree) for which $E[n] \subseteq E(\mathbb{F}_{q^k})$ and to use the Weil pairing on E to reduce the elliptic curve discrete logarithm instance to a discrete logarithm instance on $\mathbb{F}_{q^k}$. For general elliptic curves, this reduction does not run in polynomial time as k may be too large. However, for supersingular curves it can be proven that $k \leq 6$ and the reduction does run in polynomial time [130]. This means that for supersingular curves there exist subexponential algorithms for solving the discrete logarithm problem.

Decisional Diffie–Hellmann problem. On special curves, and more generally when we know a distortion map for a curve, we can build a pairing for which $e([1],[1]) \neq 1$. In such cases we can solve the DDH problem on the curve using the observation that a tuple $[1],[a],[b],[z]$ is a DDH tuple if and only if $e([a],[b]) = e([z],[1])$.

It is somewhat folklore belief that the Decisional Diffie–Hellman problem is easy for all supersingular curves (see e.g. [159, Theorem 6]), however we stress

that this is only known to hold when provided with a distortion map for the curve. Without this distortion map, the Weil pairing is useless to solve the DDH problem on a curve since $e([1],[x]) = 1$ for any x, and DDH remains a plausible hard problem. As discussed above, computing a distortion map for a uniformly random curve is also believed to be hard.

Computational Diffie–Hellman Problem. While a pairing and distortion maps together can help to solve the Decisional Diffie–Hellman problem on a curve, the Computational Diffie–Hellman problem remains a plausible hard problem in this context. When DDH is easy, the assumption that CDH is hard has been called the gap Diffie–Hellman assumption in the cryptography literature [22].

7.1.2 Trapdoor DDH groups

Trapdoor DDH groups were first introduced by Dent–Galbraith [59]. Intuitively, trapdoor DDH groups are a cryptographic construction in which knowledge of the trapdoor gives its owner the ability to solve DDH instances which are otherwise intractable. Formal definitions have appeared in Dent–Galbraith [59], Seurin [148] and Prabhakaran–Xue [144], with different security requirements in all papers. Here we recall the definition provided in [148].

We denote by $\mathsf{DDH}_{\mathbb{G}}$ the set of DDH tuples of a group $\mathbb{G}$, and we denote the order of $g \in \mathbb{G}$ by $|g|$.

Definition 22. *A trapdoor DDH group is a pair of algorithms* $(\mathsf{Gen}, \mathsf{Solve})$ *with the following properties. The trapdoor DDH group generator algorithm* Gen *is a PPT algorithm which takes as input a security parameter* 1^λ *and outputs a tuple* $(\mathbb{G}, P, \tau)$ *where* $\mathbb{G}$ *is a group,* $[1] \in \mathbb{G}$ *is a group element of order* $k = 2^{\Theta(\lambda)}$*, and* τ *is a trapdoor, such that:*

(i) Hardness of DDH without the trapdoor: the DDH problem is hard for the group generator Gen' *which outputs only* $(\mathbb{G}, [1])$*.*

(ii) Hardness of CDH with the trapdoor: the CDH problem is hard for Gen*.*

Solve *is a DPT algorithm which takes as input* $(\mathbb{G}, [1], \tau)$ *and a tuple* $[x, y, z, t] \in \mathbb{G}^4$*, either accepts (outputs 1) or rejects (outputs 0), and satisfies the following:*

(iii) Completeness: for all $(\mathbb{G}, [1], \tau)$ *possibly output by* Gen*,* Solve *always accepts if* $[x, y, z, t] \in \mathsf{DDH}_{\mathbb{G}}$*.*

(iv) Soundness: for any PPT adversary $\mathcal{A}$, we have that:

$$\mathsf{Pr}\left[\begin{array}{c|c}(\mathbb{G},P,\tau)\leftarrow\mathsf{Gen}(1^\lambda);x,y,z\leftarrow\mathbb{Z}_k; & 1\leftarrow\mathsf{Solve}(\mathbb{G},P,\tau;[x,y,z,t])\\ [t]\leftarrow\mathcal{A}(\mathbb{G},P;[x,y,z]) & \wedge\,[x,y,z,t]\notin\mathsf{DDH}_{\mathbb{G}}\end{array}\right]\approx 0.$$

We say that the trapdoor DDH group has perfect soundness when Solve *always rejects on input a non-DH tuple $[x,y,z,t]$, i.e. the above probability is zero.*

The definitions of Seurin and Dent–Galbraith are almost identical, except that the hardness of CDH with the trapdoor is not required explicitly in the definition of Dent–Galbraith. Nevertheless, their constructions satisfy this property. Prabhakaran–Xue additionally impose a Strong RSA assumption and a Diffie–Hellman knowledge of exponent assumption on the trapdoor DDH group [144]. These extra assumptions seem plausible for the specific construction of Dent–Galbraith [59] and needed for their application, but they also seem to restrict the range of possible constructions. For example, the Strong RSA assumption does not hold in a group of known order. Obtaining a trapdoor DDH group of known order is actually among the open problems left by Seurin, and in particular the Strong RSA assumption does not hold for our new construction in section 7.2.

7.1.3 Previous constructions

We briefly sketch previous constructions of trapdoor DDH groups. The first one is the most relevant one for this work.

Dent–Galbraith's "hidden pairing" construction [59]. Choose p_1, p_2 two large primes congruent to $3 \bmod 4$, such that there are large primes $r_i \mid p_i + 1$. Let $N = p_1 p_2$ and let E be an elliptic curve defined by the equation $y^2 = x^3 + x$ over the ring $\mathbb{Z}_N$. Note that the curve is supersingular, with a well-known distortion map

$$\phi : (x, y) \longmapsto (-x, \iota y),$$

where $\iota^2 = -1$. The number of points of E over $\mathbb{Z}_N$ is $(p_1+1)(p_2+1)$. Let P be a point of order $r_1 r_2$ and $\mathbb{G}$ be the group generated by P. The key observation is that a quadruple $[1, a, b, z]$ in $E(\mathbb{Z}_N)$ is a valid DDH tuple if and only if it reduces to a valid DDH tuple in $E(\mathbb{F}_{p_1})$ and $E(\mathbb{F}_{p_2})$. The DDH trapdoor in this construction is the factorization of N: given p_1 and p_2 one can solve the

DDH problem using the modified Weil pairing described in Section 7.1.1, since a distortion map on E is known. On the other hand, it seems that without the factorization of N the DDH problem on $E(\mathbb{Z}_N)$ is hard. In Section 7.4, we will show that in certain contexts, factorization is easier, forcing an increase of the parameters.

Dent–Galbraith's second construction [59]. A second construction was proposed in Dent–Galbraith's paper, based on Frey's idea of disguising an elliptic curve with a Weil descent. However, this construction was subsequently broken in [131].

Seurin's construction based on composite residuosity [148]. Choose two safe primes p_1 and p_2, namely $p_1 = 2p_1' + 1$ and $p_2 = 2p_2' + 1$ where p_1', p_2' are prime. The group $\mathbb{G}$ is the group of quadratic residues modulo N^2, where $N = p_1 p_2$. The trapdoor is the factorization of N. The group $\mathbb{G}$ is cyclic of order $Np_1'p_2'$. Let $[1]$ be a generator of $\mathbb{G}$. Given $[y] \in \mathbb{G}$, the *partial* discrete logarithm problem asks for the discrete logarithm of $[y]$ modulo N (and not modulo $Np_1'p_2'$). As shown by Paillier [137], one can solve *partial* discrete logarithms in $\mathbb{G}$ given the factorization of N, hence one can also solve Diffie–Hellman problems. On the other hand, the security of the construction is based on the hardness of the CDH problem in $\mathbb{G}$ given the factorization of N, as well as on the DDH and partial CDH problems in $\mathbb{G}$ [148].

Seurin [148] also introduced the definition of a *static* trapdoor DDH scheme where the trapdoor can only be used to solve the DDH problems involving a specific pair of elements $[1, x] \in \mathbb{G}^2$.

Seurin's static trapdoor DDH construction based on the RSA problem [148]. Let p_1, p_2, N be the same as in the previous construction. Let J_N denote the subgroup of $\mathbb{Z}_N$ consisting of those elements whose Jacobi symbol is 1. This is a cyclic group of order $m = (p_1 - 1)(p_2 - 1)/2$. Let g be a generator of J_N. Generate a random $x \in [0; m - 1]$. The trapdoor is $(m, 1/x \mod m)$, or equivalently, x and the factorization of N. Using the trapdoor one can recognize DDH instances of the form $[1, x, y, z]$ where $[1, x]$ are fixed beforehand. Indeed, $[1, x, y, z]$ is a DDH tuple if and only if $\frac{1}{x}[z] = [y]$. However, without the knowledge of the trapdoor this is RSA inversion, which seems to be a hard problem.

Seurin's static trapdoor DDH construction based on signed quadratic residues Let $N = p_1 p_2$, where p_1 and p_2 are safe primes congruent to 3 modulo 4. Let $J_N^+ = J_N/\{1, -1\}$. The group J_N^+ is cyclic of order $m = (p_1 - 1)(p_2 - 1)/4$ and let g be a generator of J_N^+. Let $x \in [0; m - 1]$. The trapdoor is $t := 2x \pm m$ (note that the computation of m is equivalent to factoring N). Then an instance $[1, x, y, z]$ is a DDH tuple if and only if $t[y] = 2[z]$, as squaring in J_N^+ is injective.

7.1.4 Seurin's open problems

In his "open problems" section [148, Section 1.4], Seurin highlighs some shortcomings of previous trapdoor DDH constructions:

> *"Two key features of trapdoor DDH groups are perfect soundness (the property that the algorithm for solving the DDH problem with the trapdoor perfectly distinguishes DH tuples from non-DH tuples), and the possibility to securely hash into the group [...]. However, none of the two candidates for TDDH groups (the hidden pairing-based proposal of [59], and [Seurin's construction]) fulfills both requirements. We think that providing a plausible candidate possessing both properties is the key to enable powerful applications of TDDH groups.*
>
> *A related open problem is whether there exists a plausible construction of a trapdoor DDH group with publicly known (ideally prime) order, since they are usually simpler to use in cryptography."*

In section 7.4 we will highlight further issues with Dent–Galbraith's construction, namely attacks on the parameters suggested, in the context of some applications. Interestingly, our new trapdoor DDH group construction will both avoid these issues and solve all of Seurin's open problems.

7.2 New trapdoor DDH groups from pairings and isogenies

In this section, we first describe our new trapdoor DDH construction. We then provide our new security definition of "trapdoor pairing" satisfied by both our construction and Dent–Galbraith's one.

7.2.1 Our construction

Fix a generator $P = [1] \in \mathbb{G}$. As is widely known, a *non-degenerate symmetric* pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ can be used to solve a DDH instance $[1, a, b, z] \in \mathbb{G}^4$ by checking whether

$$e([1], [z]) = e([a], [b]).$$

Let us now consider an elliptic curve E and the Weil pairing $e : E[m] \times E[m] \to \boldsymbol{\mu}_m$, where $\boldsymbol{\mu}_m \subseteq \mathbb{F}_{p^k}^*$ is the group of m-th roots of unity. The Weil pairing is degenerate, meaning that $e([1], [1]) = 1$, and so by itself it is not useful to solve DDH problems. This has been solved by using a distortion map, that is, an endomorphism $\phi : E \to E$ such that $\phi(P) \notin \langle P \rangle$. We then define a new pairing as

$$\hat{e}([x], [y]) = e([x], \phi[y]),$$

which is used instead of the Weil pairing, where $\phi[y]$ denotes the image of $[y]$ through the isogeny ϕ.

The key observation of our new construction is that the ability to compute a non-degenerate symmetric pairing relies on the knowledge of a distortion map. Moreover for a random supersingular elliptic curve obtaining this map is a hard problem, and so it constitutes a suitable trapdoor for a trapdoor pairing group.

More precisely, the Gen algorithm works as follows. Assume that we have a curve E_0 with a known distortion map $\phi : E_0 \to E_0$. We choose an isogeny $\varphi : E_0 \to E$ to perform a walk in the isogeny graph. We assume that we can efficiently perform a walk such that the output curve is essentially uniform. In section 7.3 we will discuss the specifics for each instantiations, and ensure that the walks are indeed efficient and random enough.

The public group G is given by the curve E and the trapdoor information τ is some representation of the isogeny φ. The Solve algorithm has access to the trapdoor, and thus can evaluate the pairing $\hat{e} : E \times E \to \boldsymbol{\mu}_q \subseteq \mathbb{F}_{p^k}$ defined as

$$\hat{e}([x], [y]) = e(\hat{\varphi}[x], \phi\hat{\varphi}[y]),$$

where e is the Weil pairing on E_0, and use this to solve DDH instances on E.

7.2.2 Trapdoor pairings

In our new construction, the trapdoor does not only allow to solve DDH instances, but also the ability to evaluate a non-degenerate symmetric pairing. We now formalize this property with a new definition.

We first identify a computational problem that is harder than DDH and better captures the power of being able to compute a pairing. Essentially, given group elements, a pairing allows a multiplication of their discrete logarithms. This translates into solving decisional problems which consist of checking a quadratic equation in the exponent. Note that although the corresponding computational problems remain hard, they are easy if we allow the output to be in the target group of the pairing. In particular, we consider the following computational problem.

Definition 23. *Let $\mathbb{G}$ be a group and $P = [1] \in \mathbb{G}$, and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a pairing. We call the* Target Computational Diffie–Hellman (Target CDH) problem *the problem consisting on, given $\mathbb{G}, [1, a, b]$ for uniformly random a, b, computing*

$$[1, ab]_T \in \mathbb{G}_T,$$

where $[1]_T \neq 1$ is a generator for a subgroup of $\mathbb{G}_T$, and must be output before receiving $[a, b]$.[1]

Note that a symmetric non-degenerate pairing can be used to solve the Target CDH problem by computing $[1]_T = e([1], [1])$ and $[ab] = e([a], [b])$. This implies that both Dent–Galbraith's first construction and our new construction are not only trapdoor DDH groups, but also trapdoor pairings.

Breaking the Target CDH problem implies breaking the DDH problem in $\mathbb{G}$, so the Target CDH problem is at least as hard as the DDH problem, but nevertheless it is still easy given an efficiently computable pairing.

We now formalize the idea of trapdoor pairings by mimicking the previous trapdoor DDH definition, but replacing the requirement that DDH should be solvable with the trapdoor with our harder problem.

Definition 24. *A trapdoor Target CDH group is a pair of algorithms* $(\mathsf{Gen}, \mathsf{Solve})$ *with the following properties. The trapdoor pairing group generator algorithm* Gen *is a PPT algorithm which takes as input a security parameter 1^λ and outputs a tuple $(\mathbb{G}, \mathbb{G}_T, [1], \tau)$ where $\mathbb{G}$ and $\mathbb{G}_T$ are the descriptions of two group, $[1] \in \mathbb{G}$ is a group element of order $k = 2^{\Theta(\lambda)}$, and τ is a trapdoor information, such that:*

(i) Hardness of DDH without the trapdoor: the DDH problem is hard for the group generator Gen' which outputs only $(\mathbb{G}, \mathbb{G}_T, [1])$, both in $\mathbb{G}$ and $\mathbb{G}_T$.

[1] The reason for asking for g is that, since the pairing will not be available to all parties, it is not immediate to produce a canonical generator of $\mathbb{G}_T$ from the generator of $\mathbb{G}$. We ask for it in advance so that it does not depend on $[a, b]$.

(ii) Hardness of CDH with the trapdoor: the CDH problem is hard for Gen*, both in* $\mathbb{G}$ *and* $\mathbb{G}_T$*.*

Solve *is a DPT algorithm which takes as input* $(\mathbb{G}, \mathbb{G}_T, [1], \tau)$ *and a tuple* $[a, b] \in \mathbb{G}^2$*, and outputs* $[1, z]_T \in \mathbb{G}_T^2$*, where* $[1]_T \neq 1$*, and satisfies the following:*

(iii) Completeness: for all $(\mathbb{G}, \mathbb{G}_T, [1], \tau)$ *possibly output by* Gen*,* Solve *always outputs* $[1, z]_T \in \mathbb{G}_T^2$ *such that* $z = ab$*.*

(iv) Soundness: for any PPT adversary $\mathcal{A}$*, the we have that*

$$\mathsf{Pr}\left[\begin{array}{l}(\mathbb{G},\mathbb{G}_T, P, \tau) \leftarrow \mathsf{Gen}(1^\lambda); a, b \leftarrow \mathcal{A}(1^\lambda, \mathbb{G}, \mathbb{G}_T, [1]); \\ \quad [1, z]_T \leftarrow \mathsf{Solve}(\mathbb{G}, \mathbb{G}_T, P, \tau; [a, b])\end{array} : z \neq ab\right] \approx 0.$$

We say that trapdoor Target CDH group has perfect soundness when the above probability is zero.

An alternative, perhaps more natural definition could require the Target CDH problem to be hard without the trapdoor, as opposed to the DDH problem in definition 23. We chose to require hardness of DDH (implying hardness of Target CDH) so that trapdoor pairings are naturally trapdoor DDH groups as well. The only difference between them lies in the power provided by the trapdoor: a DDH solver in definition 22, and a stronger Target CDH solver in definition 23.

7.2.3 Security of our new construction

We now prove that our new construction is a trapdoor pairing in the above sense (hence it is also a trapdoor DDH group).

Theorem 48. *Suppose that the distribution of the curve* E *output by algorithm* Gen *is statistically equivalent to the uniform distribution. Then, if the DDH problem in* E *is hard, and the CDH problem in* E *is hard given the trapdoor, the construction above is a secure trapdoor pairing group.*

Proof. It is clear from the discussion above that the Target CDH problem can be solved efficiently when the trapdoor is known, and by assumption the CDH problem is hard.

Without the trapdoor, solving DDH in $\mathbb{G}$ is exactly the DDH problem on the curve E. While E is not a uniformly random curve, it is the output of a random walk, which is close to uniformly random so that the two problems are equivalent. □

We now argue that the DDH and CDH assumptions of Theorem 48 are plausible. First, the DDH has been widely studied and used in the literature, and is believed to hold when a symmetric pairing is not available, and as discussed in Section 7.1.1, the DDH problem is easy for supersingular curves only when a distortion map is known.

We remark that constructing a curve with a distortion map is easy: one can choose a special curve, or do a random walk from one of these special curves as in our trapdoor pairing construction. On the other hand given a randomly chosen supersingular curve, computing a distortion map appears to be a difficult problem, as discussed in Section 7.1.1. Conversely, given the endomorphism ring of a curve E, one can also compute an isogeny between E_0 and E (see [61,140]), and any such isogeny can be used as a trapdoor in our scheme.

While DDH is easy on E with the trapdoor, the CDH problem still appears to be hard on E. Indeed this is formalized by the so-called Gap-CDH assumption in pairing-based cryptography. Moreover, given the trapdoor the CDH problems on the curves E_0 and E are equivalent, as we can use a trapdoor $\varphi : E_0 \to E$ to send a CDH instance $[1, a, b]$ in E_0 to $(\varphi[1], \varphi[a], \varphi[b])$. Note that scalar multiplication commutes with any isogeny, so this is a CDH instance on E.

The assumption that the output of the group generation algorithm is close to uniformly random will be discussed for the particular instantiations of the algorithm, in section 7.3, as the argument is different in each case.

7.3 Two concrete instantiations

In [148, Section 2.4], Seurin requests the following useful features for a trapdoor DDH group:

- The group order can be publicly revealed.
- The group order is a prime number.
- There is an efficient hashing algorithm into the group.

We note that no previous construction has achieved these properties at the same time. In particular, all of them use composite-order groups.

We consider two instantiations of our idea, one using curves over $\mathbb{F}_{p^2}$, and another using curves over $\mathbb{F}_p$. The first one satisfies the first property, and either the second or the third, but not both simultaneously. The second instantiation achieves the three properties at the same time.

7.3.1 Curves over $\mathbb{F}_{p^2}$

We start by stating a simple result that ensures that our isogenies will be defined over $\mathbb{F}_{p^2}$, and we will not need to move to extension fields.

Lemma 49. *Let $p \geq 3$ be a prime, and let E be a supersingular elliptic curve such that $\#E(\mathbb{F}_{p^2}) = (p+1)^2$. Then the 2-isogenies from E are $\mathbb{F}_{p^2}$-rational.*

Proof. Since E has $(p+1)^2$ points over $\mathbb{F}_{p^2}$, we have that $E(\mathbb{F}_{p^2})$ is isomorphic to $\mathbb{Z}_{p+1} \times \mathbb{Z}_{p+1}$ [51, Theorem 54]. Since p is an odd prime, we have that $2 \mid (p+1)$, and so $\mathbb{Z}_{p+1} \times \mathbb{Z}_{p+1}$ contains a copy of $\mathbb{Z}_2 \times \mathbb{Z}_2$, which is necessarily the 2-torsion of the curve. Thus the 2-torsion is $\mathbb{F}_{p^2}$-rational, and so are the 2-isogenies. □

We also recall once again theorem 20 and lemma 21, which ensure that, by taking random walks of length $2(1+\epsilon)\log_\ell p$ in the ℓ-isogeny graph, the output is statistically indistinguishable from uniformly random.

Let p be a prime such that $p = 3 \bmod 4$ and $p+1 = qf$, where q is also prime and f is a small cofactor. We consider the curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_{p^2}$. This curve is in the conditions of Lemma 49 above, has j-invariant $j = 1728$, and its endomorphism ring is known (see e.g. [140]).

To generate the trapdoor DDH group, we take a random walk $\varphi : E_0 \to E$ composed of 2-isogenies, long enough to ensure that the output curve E is statistically uniform in the graph. Since isogenies preserve the supersingular property and the number of points, any curve that we reach from E_0 through 2-isogenies is also in the conditions of the lemma, and therefore every step of the walk is defined over $\mathbb{F}_{p^2}$.

At this point, we have two options:

- We consider $E(\mathbb{F}_{p^2})$ as the trapdoor group. The group order is $(p+1)^2$, and is public, and we can efficiently hash into the group using standard techniques [101], but the group is not of prime order. In fact, the group is not even cyclic.

- We consider a subgroup of $E(\mathbb{F}_{p^2})$ of order q as the trapdoor group. It is easy to find a point of order $p+1$ in $E(\mathbb{F}_{p^2})$ and multiply it by f to obtain a point of order q, which is close in size to p. In this case, the group order is public and prime, but there is no obvious way to securely hash into the group.

The hardness of computing a distortion map then relies on the hardness of computing an isogeny between a fixed curve and a random curve over $\mathbb{F}_{p^2}$, for

Algorithm 6 Trapdoor group generation (curves over $\mathbb{F}_{p^2}$)

Require: security parameter λ.
Ensure: group description $(\mathbb{G}, P)$, trapdoor φ.
1: Choose primes $p, q = \Theta(2^\lambda)$ such that $p = 3 \bmod 4$ and $p+1 = qf$ for small f.
2: Define the curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_{p^2}$.
3: Take a random walk $\varphi : E_0 \to E$ of length $2 \log p$ composed of 2-isogenies.
4: Choose a point $Q \in E(\mathbb{F}_{p^2})$ of order $p+1$. Set $P = fQ$.
5: Output $(\langle P \rangle, P, \varphi)$.

which only exponential-time attacks are known, as discussed in Section 7.1.1. This justifies the assumptions of Theorem 48.

7.3.2 Curves over $\mathbb{F}_p$

We now present an alternative instantiation that uses curves over $\mathbb{F}_p$. In the previous section, we have seen an instantiation that uses a prime-order group which we cannot efficiently hash into. The reason is that $E(\mathbb{F}_{p^2})$ does not have a unique subgroup of order q, so the trapdoor group must be specified though a generator. With this description, there is no obvious way to hash into the group without knowing the discrete logarithm of the hash, which is undesirable for security. To solve this, we want to find a group in which we can canonically identify a subgroup of order q.

To do so, we work over $\mathbb{F}_p$ instead of $\mathbb{F}_{p^2}$, taking an approach similar to CSIDH [35]. We choose a prime $p = 3 \bmod 4$ such that $p + 1 = 4\ell_1 \dots \ell_n q$, for a large prime q, and we consider again the curve $E_0 : y^2 = x^3 + x$, now over $\mathbb{F}_p$.

The idea is again to take a random walk in the isogeny graph, using only $\mathbb{F}_p$-rational isogenies.

Lemma 50. *$E(\mathbb{F}_p)$ has a unique subgroup of order q.*

Proof. $E(\mathbb{F}_p)$ has $p+1$ points. Since $q \mid (p+1)$ but $q^2 \nmid (p+1)$, we have that there is a subgroup of order q in $E(\mathbb{F}_p)$, but the $\mathbb{F}_p$-rational curve does not contain the whole q-torsion. Then $E(\mathbb{F}_p)$ contains only one subgroup $\mathbb{G}$ of order q.

□

We will use this unique subgroup $\mathbb{G}$ as the trapdoor DDH group. Note that the embedding degree of the pairing is the smallest integer k such that

$\#\mathbb{G} \mid (p^k - 1)$, so in this case we have $k = 2$. Hashing into this group is now easy. We make it explicit in the following result.

Lemma 51. *There is an efficient algorithm to hash into* $\mathbb{G}$.

Proof. Given a string, we hash it into $\mathbb{F}_p$. We interpret the result as the x-coordinate of a point in $E(\mathbb{F}_p)$. Note that a uniformly random element of $\mathbb{F}_p$ will correspond to a point in the curve with probability roughly $1/2$. If that is not the case, we increase the x-coordinate until it corresponds to a point P. Then we compute $\frac{p+1}{q}P$, landing into the unique subgroup of order q. Note that $\frac{p+1}{q}$ is coprime to q, so given a uniformly random $P \in E(\mathbb{F}_p)$, we have that $\frac{p+1}{q}P$ is a uniformly random point in $\mathbb{G}$.

□

It only remains to specify how to move through the graph using only $\mathbb{F}_p$-rational isogenies. Note that the same argument used in Lemma 50 for q works for any ℓ_i, so $E(\mathbb{F}_p)$ contains only one subgroup of order ℓ_i for each $i = 1, \ldots, n$. We make use of the following result (see [52, Section 15] for a proof).

Lemma 52. *Let* $\ell \geq 3, p \geq 5$ *be different prime numbers, such that* $\left(\frac{D}{\ell}\right) = 1$, *where* $D = t^2 - 4p$, *and* t *is the trace of Frobenius. Let* E *be a supersingular elliptic curve. Then* $\#E(\mathbb{F}_p) = p + 1$ *and there are two* ℓ*-isogenies from* E *that are* $\mathbb{F}_p$*-rational. Moreover, these correspond to:*

- *The unique subgroup* H *of order* ℓ *of* $E(\mathbb{F}_p)$.
- *The unique subgroup* $\tilde{H}$ *of order* ℓ *of* $\tilde{E}(\mathbb{F}_p)$, *where* $\tilde{E}$ *is the quadratic twist of* E.

Then, the random walk consists of choosing exponents $e_1, \ldots, e_n \in [-B, B]$. An exponent corresponds to $|e_i|$ steps in which we use the isogeny with kernel H or $\tilde{H}$, respectively, depending on whether the sign of e_i is positive or negative. The distribution of the output of the random walk depends on the structure of the class group of $\mathrm{End}_p(E_0)$. Although for certain parameters it has been computed [18], in general we make the heuristic assumption that, for B and n large enough, the random walk reaches any point in the graph with roughly the same probability.[2]

In a similar way to the case above, the distortion map is protected by the hardness of computing an isogeny between curves over $\mathbb{F}_p$. We note that there

[2] In CSIDH [35], the authors suggest $B = 5$ and $n = 74$ for a prime p of length 512 bits.

Algorithm 7 Trapdoor group generation (curves over $\mathbb{F}_p$)

Require: security parameter λ.
Ensure: group description $(\mathbb{G}, P)$, trapdoor φ.

1: Choose primes $p, q = \Theta(2^\lambda)$ such that $p = 3 \bmod 4$, and small odd primes $\ell_1, \ldots, \ell_n$ and integers $e_1, \ldots, e_n$ such that $p + 1 = 4\ell_1 \ldots \ell_n q$, and $\prod_n \ell_i > 2\sqrt{p}$.
2: Define the curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_p$.
3: Take a random walk $\varphi : E_0 \to E$ composed of e_i isogenies of degree ℓ_i, for each $i = 1, \ldots, n$.
4: Let $\mathbb{G}$ be the unique subgroup of $E(\mathbb{F}_p)$ of order q, and let P be a generator.
5: Output $(\mathbb{G}, P, \varphi)$.

is a quantum subexponential algorithm for this problem, due to Biasse-Jao-Sankar [19]. However, our construction depends on the discrete logarithm assumption, which is broken in the quantum setting anyway, so we focus on classical security.

7.3.3 Parameter choices

Let λ be the security parameter. There are two main ways to break the security of the constructions: recovering the trapdoor or solving the discrete logarithm problem. The first approach amounts to finding a non-scalar endomorphism on E or an isogeny to E_0. Recall that for supersingular elliptic curves, the best known classical algorithm [71] has complexity $\tilde{O}(p^{\frac{1}{2}})$.

As for the discrete logarithm problem, one can apply the MOV reduction [130] to reduce any discrete logarithm problem over either E_0 or E to a discrete logarithm problem over $\mathbb{F}_{p^k}$, where $k = 3$ in the first construction and $k = 2$ in the second. Note that the reduction from E is only available if the trapdoor is known, but nevertheless we do not want a party that knows the trapdoor to be able to solve CDH. The best algorithm for computing discrete logarithms in finite fields of large characteristic is the number field sieve and its variants [107, 116], which have complexity $L_n(1/3)$, where in this case $n = p^k$. On the other hand, the best algorithms for solving the discrete logarithm directly in the curve are the generic ones, with complexity $\tilde{O}(q^{1/2})$. One should therefore choose $\log p = \Omega(\lambda^3)$ to avoid these attacks.

- For the construction over $\mathbb{F}_{p^2}$, recall that $p+1 = qf$ for a small cofactor f, so roughly $\log p = \log q$. Thus, the trapdoor group $\mathbb{G}$ is formed by $\Theta(2^{\lambda^3})$

elements over $\mathbb{F}_{p^2}$.

- For the construction over $\mathbb{F}_p$, we have that $p+1 = 4\ell_1 \dots \ell_n q$, and we require that $\prod_i \ell_i > 2\sqrt{p}$, so roughly $\log p \approx 2 \log q$. Therefore, our trapdoor group is again formed by $\Theta(2^{\lambda^3})$ elements over $\mathbb{F}_p$.

The trapdoor is easy to store, as a d-isogeny requires $\log d = O(\log p)$ bits.

7.3.4 Comparison with previous constructions.

Dent–Galbraith's construction. Since the trapdoor is the factorization of N, which in turn can be obtained from the factorization of $r_1 r_2$, as explained in Section 7.4, we need to ensure that this is hard. We must therefore choose $\log(r_1 r_2) = \Omega(\lambda^3)$ to prevent the number field sieve, and since we require $r_i < \sqrt{p_i}$, we need at least $N = p_1 p_2 = \Omega((r_1 r_2)^2)$. We refer to Section 7.4 for a discussion of the case $r_1 = r_2$ and potential further attack developments.

Seurin's construction. This construction also relies on the factorization of $N = p_1 p_2$, so we must ensure that $\log N = \Omega(\lambda^3)$. Then the trapdoor DDH group is of order $N p_1' p_2' \approx N^2$.

We note that our new construction is asymptotically comparable to the previous proposals in terms of efficiency, while satisfying a stronger definition than Seurin's construction and some desirable properties missing in previous constructions. Also, choosing parameters is more straightforward than in Dent–Galbraith's construction, as the new construction is in a prime-order group, hence we do not need to account for potential factorization attacks, as those described described in the next section.

7.4 Partial attacks on Dent–Galbraith's construction

Dent–Galbraith's hidden pairing construction uses pairings on elliptic curves defined over RSA rings. As already pointed out in [73], selecting parameters for such constructions may be tricky. We now demonstrate this by showing attacks on the construction when the group order is revealed. Note that Dent–Galbraith suggest to reveal this information in some applications, for example to allow delegation of the pairing computation.

7.4.1 Case $r_1 r_2$ known and small, $r_1 \neq r_2$

We first give a simple attack on the parameters suggested by Dent–Galbraith ($p_i \approx 2^{512}$ and $r_i \approx 2^{160}$) for their construction.

Let p_1, p_2, r_1, r_2 as in Dent–Galbraith's construction, and assume that $r_1 \neq r_2$ (this condition is not explicitly required in their paper, but it is implied by their later statement that P has order $r_1 r_2$). With $r_i \approx 2^{160}$ the product $r_1 r_2$ can be easily factored with current techniques, so we can assume knowledge of r_1 and r_2. We can then apply a technique from [73, Section 4] to factor N. Namely, we apply x-only addition and doubling formulae to compute the x-coordinate of $[r_1]P$. This leads to the point at infinity modulo p_1 but not modulo p_2, hence a factor of N can be recovered as in the elliptic curve factorization method [123].

To defeat this attack one can choose parameters such that r_1 and r_2 cannot be computed from their product $r_1 r_2$, and make sure other attacks are not feasible. One condition stated in [59] is that $r_i < \sqrt{p_i}$, so the attack requires to at least double the size of p_1 and p_2.

An a priori plausible alternative way to defeat the attack is to enforce $r_1 = r_2$. In this case $E(\mathbb{Z}_N)$ is the direct product of two cyclic groups of order $p_i + 1$ and similarly $\mathbb{G}$ is the direct product of two cyclic groups of order r. With this configuration, multiplying any point in $\mathbb{G}$ by r gives ∞ modulo both p_1 and p_2, hence no factor is recovered. We now consider this case more thoroughly.

7.4.2 Case $r_1 = r_2$ a known prime

The setting for a known $r := r_1 = r_2$ was in fact already studied in [73], and the best attack presented there has a complexity $O(N^{1/4}/r)$. Taking p_1 and p_2 with 512 bits and r with 160 bits leads to a cost of 2^{96} for this attack, which seems impractical today.

However, we now present an alternative attack in this setting, using Coppersmith's techniques for finding small integer roots of bivariate polynomials [41] and its generalizations by Coron [42–44]. [3] In order to factor N, we only need to find x and y such that $N = (rx - 1)(ry - 1)$, i.e., we are looking for roots of the bivariate polynomial

$$p(x, y) = 1 - N - rx - ry + r^2 xy.$$

For the parameters above there is a root (x_0, y_0) such that $|x_0| \leq 2^{352}$ and $|y_0| \leq 2^{352}$. We will use the following result.

[3] This attack can be readily extended when $r_1 \neq r_2$, but in that case the attack from the previous section will be simpler.

Theorem 53 ([41], Corollary 2)**.** *Let $p(x, y) \in \mathbb{Z}[x, y]$ be a bivariate irreducible polynomial of maximum degree δ in each variable. Let X, Y be upper bounds on the desired integer solution (x_0, y_0) and let $W = max_{i,j}\ \{|p_{ij}|X^iY^j\}$. If $XY < W^{2/(3\delta)}$, then in time polynomial in $(\log\ W, 2^\delta)$ one can find an integer solution (x_0, y_0) to the equation $p(x, y) = 0$ such that $|x_0| \leq X,\ |y_0| \leq Y$.*

An easy calculation shows that we cannot apply Theorem 53 directly here: indeed our polynomial p has degree 1 in each variable, and we have $XY \approx 2^{704}$ and $W^{2/3} \approx N^{2/3} \approx 2^{683}$. However, we can still apply the theorem by guessing a few bits of both x and y and iterating Coron's algorithm. Specifically, we set $x := 2^{12}x' + c_1$ and $y := 2^{12}y' + c_2$ where $0 \leq c_i \leq 2^{12}$ and we try to find a solution for each admissible pair (c_1, c_2). With this approach we now have bounds $X = Y = 2^{340}$ on x' and y', and we still have $W^{2/3} \approx N^{2/3} \approx 2^{683}$. As there are 2^{12} choices for each of the c_i, we only need to run the algorithm from [43] at most 2^{24} times to find p_1 and p_2.

One way to defeat this attack in practice is to increase the number of guesses needed; we now estimate the parameters needed to guarantee that this is bigger than 2^{80}. Assume r is a k bit integer and the p_i are $k + \ell$ bit primes, where k and ℓ are positive integers. Then XY is a 2ℓ bit integer and the number of bits of $W^{2/3}$ is $\frac{4}{3}(k + \ell)$. In order to achieve the desired security we need that $2\ell - 80 > \frac{4}{3}(k + \ell)$ or $\ell > 2k + 120$. When r has $k = 160$ bits, we need p_i with at least $\ell + k > 600$ bits, hence N should have at least 1200 bits.

7.4.3 Potential extensions of the attack

In the previous subsections we have merely applied existing results from the literature to demonstrate that the parameters suggested by Dent–Galbraith are insecure when the group order is revealed. We expect more elaborate and dedicated algorithms to give better results and to require further increases of the parameters.

In particular, we expect further lattice attacks to exist in the case $r_1 \neq r_2$ when $R := r_1r_2$ is known but cannot be efficiently factored (the setting originally proposed by Dent–Galbraith, but with bigger parameters). In this case we have two equations (with variables x, y, r_1, r_2):

$$\begin{cases} N = (r_1x - 1)(r_2y - 1), \\ R = r_1r_2. \end{cases}$$

One could apply multivariate generalizations of Coppersmith's method and deduce new constraints on the parameters' sizes; we leave this to further work.

As this section demonstrates, selecting parameters for Dent–Galbraith's trapdoor DDH group construction is far from trivial. Note that our new construction does not have this issue as it uses supersingular curves over $\mathbb{F}_{p^2}$ instead of $\mathbb{Z}_N$.

7.5 Applications

7.5.1 Identification scheme

By observing that we have not only a trapdoor DDH, but a more general trapdoor pairing construction, we can improve upon the Dent–Galbraith identification scheme. Essentially, in their scheme a party has a secret pairing and identifies itself by showing that it can distinguish if a challenge tuple is a DDH tuple or not. As the prover can cheat with probability $\frac{1}{2}$, this protocol must be repeated many times to ensure a negligible cheating probability. By relying on a computational problem instead, we can remove the need for repetition.

- **Setup.** Let $(\mathbb{G}, \mathbb{G}_T, [1], \tau) \leftarrow Gen(1^\lambda)$ be a trapdoor pairing group. The prover's secret key will be the trapdoor τ, which allows to compute a non-degenerate pairing
$$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T,$$
as described above. The public key is $(\mathbb{G}, \mathbb{G}_T, [1])$, where $[1] \leftarrow \mathbb{G}$.
- **Interaction.**
  - The prover picks $r \leftarrow \mathbb{F}_p$ and sends $[1]_T = re([1], [1])$ to the verifier.
  - The verifier picks $a, b \leftarrow \mathbb{F}_p$ and sends $[a, b]$ to the prover.
  - The prover computes $[u]_T = re([a], [b])$, and sends $[u]$ to the verifier.

  The verifier accepts the proof if and only if $[u]_T = [ab]_T$.

Clearly a cheating prover can solve the Target CDH problem. By assumption, this will only happen with negligible probability, so there is no need to repeat the protocol. We formalize the security in the following theorem.

Theorem 54. *The identification scheme above is complete, sound and zero-knowledge when instantiated in a trapdoor pairing group.*

Proof. Completeness is easy to check, as
$$[u]_T = re([a], [b]) = ab(r(e([1], [1])) = [ab]_T.$$

For soundness, assume that a cheating prover $\mathcal{A}$ can produce accepting proofs. We build an adversary $\mathcal{B}$ to break the Target CDH problem as follows: upon receiving $(\mathbb{G}, \mathbb{G}_T, [1])$, adversary $\mathcal{B}$ passes them to $\mathcal{A}$, who answers with $[1]_T$. $\mathcal{B}$ forwards $[1]_T$ to the challenger and receives $[a, b]$, which are again sent to $\mathcal{A}$, who answers with u. Because the proof is accepting, we have that $u = ab$, so $[u]_T$ is a valid solution for the Target CDH problem.

We argue that the scheme is zero-knowledge, that is, no information about the trapdoor pairing is leaked. To do so, we describe a simulator that produces, without knowledge of the trapdoor, transcripts indistinguishable from transcripts from honest executions of the scheme.

- **Simulator.** Pick $[1]_T \leftarrow \mathbb{G}_T$. Choose $a, b \leftarrow \mathbb{F}p$, and set $[u]_T = ab[1]_T$. The first message of the transcript is $[1]_T$, the second is $[a, b]$, and the third is $[u]_T$.

Clearly the second message is distributed as in a real execution. In the third message $[u]_T$ is correctly distributed as long as $[1]_T$ is correctly distributed, and $[1]_T$ is in both cases a uniformly random element of the target group. □

7.5.2 Breaking anonymity in ElGamal voting

We recall the ElGamal encryption scheme.[4] A group $\mathbb{G}$ and a generator $[1]$ are publicly known. A user's secret key is $sk = x \leftarrow \mathbb{F}p$ and the corresponding public key is $pk = [x]$. To encrypt a message $m \in \mathbb{F}p$, we choose randomness $r \leftarrow \mathbb{F}p$ and set

$$\mathsf{Enc}_{pk}(m; r) = [r, m + rx].$$

To decrypt a ciphertext $[c_1, c_2]$, we compute

$$\mathsf{Dec}_{sk}([c_1, c_2]) = c_2 - xc_1.$$

Note that the discrete logarithm of $[c_2 - xc_1]$ can be efficiently computed only if the set of possible messages is small. This is often the case in voting, in which the set of messages is a small set of candidates, or even just 'yes'/'no'.

We observe that an encryption, together with public information, contains a DDH tuple. Indeed, consider

$$([1], pk, [c_1], [c_2 - m]) = [1, x, r, xr].$$

[4] We present the variant known as lifted ElGamal, in which the message is an element of $\mathbb{F}p$ instead of $\mathbb{G}$.

Hence, if someone can solve the DDH problem, and the set of possible messages is small enough, it is possible to identify the message by checking whether $([1], pk, [c_1], [c_2 - \tilde{m}])$ is a DDH tuple for each possible message $\tilde{m}$, until a positive result is found.

This rules out the use of supersingular curves for electronic voting and similar purposes, as the party that sets up the group $\mathbb{G}$ potentially has access to a trapdoor that allows to open any vote. This idea extends naturally to other contexts. For example, usually zero-knowledge proofs involve using commitments, and sometimes ElGamal encryption is used as a commitment there. We note that a DDH or pairing trapdoor would allow to break the hiding property of the commitment scheme, hence compromising the security of the zero-knowledge proof and the protocols derived from it.

7.6 Conclusion and further work

In this chapter, we presented a new trapdoor DDH group construction based on supersingular elliptic curves and pairings. We also gave partial attacks on a previous trapdoor DDH group construction, and we provided a formal security definition for a related but more powerful primitive called "trapdoor pairing" (which our new construction also satisfies). Our new construction has a number of interesting properties; in particular it has all the properties identified by Seurin in his "open problems" section [148, Section 1.4] as crucial for applications.

Although trapdoor DDH groups were introduced in 2006, the number of applications of it has been so far quite limited. Seurin [148] identified some limitations of all the previous constructions (included their own), and hoped that solving these would allow for more meaningful applications. Our new construction satisfies all the properties required by Seurin, yet no obvious application seems to arise. The notions of trapdoor DDH groups and trapdoor pairings seem to fit quite naturally with the idea of a distinguished party, which would use the trapdoor to perform some special operation that is only allowed to him. This suggests that trapdoor DDH groups might be useful in constructing schemes where there is an authority figure. For example, in group signatures, members of the group can sign messages anonymously on behalf of the group. There is a group manager that is allowed to trace the signer, but is not able to produce forgeries. In this setting, a manager with a trapdoor could maybe identify a signer by noticing a DDH tuple that involves the user's public key, the message and the signature. At the same time, hardness of DDH for the rest of the parties

would keep the signatures anonymous for them. We leave the development of such a scheme to further work.

Bibliography

[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 Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 418–433, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer. 19

[2] Masayuki Abe, Miguel Ambrona, Miyako Ohkubo, and Mehdi Tibouchi. Lower bounds on structure-preserving signatures for bilateral messages. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 3–22, Amalfi, Italy, September 5–7, 2018. Springer. 46

[3] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. *Journal of Cryptology*, 29(2):363–421, April 2016. 70, 75

[4] Masayuki Abe, Charanjit S. Jutla, Miyako Ohkubo, and Arnab Roy. Improved (almost) tightly-secure simulation-sound QA-NIZK with applications. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*, pages 627–656, Brisbane, Queensland, Australia, December 2–6, 2018. Springer. 46

[5] Gora Adj, Daniel Cervantes-Vázquez, Jesús-Javier Chi-Domínguez, Alfred Menezes, and Francisco Rodríguez-Henríquez. On the cost of computing isogenies between supersingular elliptic curves. In *International Conference on Selected Areas in Cryptography*, pages 322–343. Springer, 2018. 36

[6] Noga Alon, Itai Benjamini, Eyal Lubetzky, and Sasha Sodin. Non-backtracking random walks mix faster. *Communications in Contemporary Mathematics*, 9(4):585–603, 2007. 146

[7] Reza Azarderakhsh, Matthew Campagna, Craig Costello, LD Feo, Basil Hess, A Jalali, D Jao, B Koziel, B LaMacchia, P Longa, et al. Supersingular isogeny key encapsulation. *Submission to the NIST Post-Quantum Standardization project*, 2017. 152, 154, 155, 175, 178

[8] Reza Azarderakhsh, David Jao, Kassem Kalach, Brian Koziel, and Christopher Leonardi. Key compression for isogeny-based cryptosystems. In *AsiaPKC '16*, pages 1–10, New York, NY, USA, 2016. ACM. 117

[9] Robert Beals, Stephen Brierley, Oliver Gray, Aram W. Harrow, Samuel Kutin, Noah Linden, Dan Shepherd, and Mark Stather. Efficient distributed quantum computing. *Proc. R. Soc. A*, 469(2153):20120686, 2013. 21

[10] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. Compact e-cash and simulatable VRFs revisited. In *International Conference on Pairing-Based Cryptography*, pages 114–131. Springer, 2009. 45

[11] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 390–399, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press. 19

[12] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. 8

[13] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 103–128, Darmstadt, Germany, May 19–23, 2019. Springer. 71

[14] David Bernhard, Georg Fuchsbauer, and Essam Ghadafi. Efficient signatures of knowledge and DAA in the standard model. In Michael J. Jacobson Jr., Michael E. Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors, *ACNS 13*, volume 7954 of *LNCS*, pages 518–533, Banff, AB, Canada, June 25–28, 2013. Springer. 45

[15] Daniel Bernstein, Luca de Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. *arXiv preprint arXiv:2003.10118*, 2020. 173

[16] Daniel J Bernstein. Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete? *SHARCS'09 Special-purpose Hardware for Attacking Cryptographic Systems*, page 105, 2009. 20

[17] Daniel J Bernstein and Edoardo Persichetti. Towards KEM unification. *IACR Cryptol. ePrint Arch.*, 2018:526, 2018. 156, 167

[18] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *ASIACRYPT 2019, Part I*, LNCS, pages 227–247. Springer, December 2019. 193

[19] Jean-François Biasse, David Jao, and Anirudh Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In Willi Meier and Debdeep Mukhopadhyay, editors, *INDOCRYPT 2014*, volume 8885 of *LNCS*, pages 428–442, New Delhi, India, December 14–17, 2014. Springer. 36, 37, 109, 126, 170, 194

[20] Gaetan Bisson and Andrew V. Sutherland. Computing the endomorphism ring of an ordinary elliptic curve over a finite field. *Journal of Number Theory*, 131(5):815–831, 2011. 38

[21] Nir Bitansky, Ran Canetti, Henry Cohn, Shafi Goldwasser, Yael Tauman Kalai, Omer Paneth, and Alon Rosen. The impossibility of obfuscation with auxiliary input or a universal simulator. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 71–89, Santa Barbara, CA, USA, August 17–21, 2014. Springer. 2

[22] Ian F Blake, Gadiel Seroussi, and Nigel P Smart. *Advances in elliptic curve cryptography*, volume 317. Cambridge University Press, 2005. 183

[23] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238, Interlaken, Switzerland, May 2–6, 2004. Springer. 73

[24] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 327–357, Vienna, Austria, May 8–12, 2016. Springer. 19, 66, 67, 69, 70, 76, 79, 83, 101, 104

[25] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system I: The user language. *Journal of Symbolic Computation*, 24(3-4):235–265, 1997. 172

[26] Paul Bottinelli, Victoria de Quehen, Chris Leonardi, Anton Mosunov, Filip Pawlega, and Milap Sheth. The dark SIDH of isogenies. Cryptology ePrint Archive, Report 2019/1333, 2019. https://eprint.iacr.org/2019/1333. 174

[27] Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In *International Workshop on Public Key Cryptography*, pages 1–15. Springer, 2007. 45

[28] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334, San Francisco, CA, USA, May 21–23, 2018. IEEE Computer Society Press. 66, 67, 69, 70, 104

[29] Benedikt Bünz, Mary Maller, Pratyush Mishra, and Noah Vesely. Proofs for inner pairing products and applications. Technical report, Cryptology ePrint Archive, Report 2019/1177, 2019. 67, 70, 79

[30] Jeffrey Burdges and Luca De Feo. Delay encryption, 2020. https://eprint.iacr.org/2020/638. 181

[31] Jan Camenisch, Rafik Chaabouni, and abhi shelat. Efficient protocols for set membership and range proofs. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 234–252, Melbourne, Australia, December 7–11, 2008. Springer. 47

[32] Matteo Campanelli, Antonio Faonio, Dario Fiore, Anaïs Querol, and Hadrián Rodríguez. Lunar: a toolbox for more efficient universal and updatable zksnarks and commit-and-prove extensions. In *ERINT IACR, Report 2020/1069*, 2020. 101

[33] Matteo Campanelli, Dario Fiore, and Anaïs Querol. LegoSNARK: Modular design and composition of succinct zero-knowledge proofs. In *ACM CCS 2019*, pages 2075–2092. ACM Press, 2019. 70

[34] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4):557–594, 2004. 8

[35] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427, Brisbane, Queensland, Australia, December 2–6, 2018. Springer. 36, 153, 192, 193

[36] Pyrros Chaidos, Véronique Cortier, Georg Fuchsbauer, and David Galindo. BeleniosRF: A non-interactive receipt-free electronic voting scheme. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1614–1625, Vienna, Austria, October 24–28, 2016. ACM Press. 47

[37] Nishanth Chandran, Jens Groth, and Amit Sahai. Ring signatures of sub-linear size without random oracles. In *International Colloquium on Automata, Languages, and Programming*, pages 423–434. Springer, 2007. 45, 47

[38] Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, January 2009. 36, 109, 152, 158

[39] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Noah Vesely, and Nicholas P. Ward. Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In Vincent Rijmen and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, LNCS, pages 738–768. Springer, May 2020. 66, 67, 68, 71, 101

[40] Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014. 38, 151

[41] Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260, 1997. 196, 197

[42] Jean-Sébastien Coron. Finding small roots of bivariate integer polynomial equations revisited. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 492–505. Springer, 2004. 196

[43] Jean-Sébastien Coron. Finding small roots of bivariate integer polynomial equations: A direct approach. In *Annual International Cryptology Conference*, pages 379–394. Springer, 2007. 196, 197

[44] Jean-Sébastien Coron, Alexey Kirichenko, and Mehdi Tibouchi. A note on the bivariate Coppersmith theorem. *Journal of Cryptology*, pages 1–5, 2013. 196

[45] Craig Costello. B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion. Technical report, Cryptology ePrint Archive, Report 2019/1145, 2019. https://eprint.iacr.org, 2019. 174, 178

[46] Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. Efficient compression of SIDH public keys. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 679–706, Paris, France, April 30 – May 4, 2017. Springer. 174

[47] Craig Costello, Patrick Longa, and Michael Naehrig. Efficient algorithms for supersingular isogeny Diffie-Hellman. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 572–601, Santa Barbara, CA, USA, August 14–18, 2016. Springer. 109, 123

[48] Geoffroy Couteau and Dominik Hartmann. Shorter non-interactive zero-knowledge arguments and zaps for algebraic languages. 2020. 47

[49] George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square span programs with applications to succinct NIZK arguments. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 532–550, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer. 2, 65

[50] Vanesa Daza, Carla Ràfols, and Alexandros Zacharakis. Updateable inner product argument with logarithmic verifier and applications. In *PKC 2020, Part I*, LNCS, pages 527–557. Springer, 2020. 67, 69, 70

[51] Luca De Feo. Mathematics of isogeny-based cryptography. *arXiv preprint arXiv:1711.04062*, 2017. 191

[52] Luca De Feo. Isogeny graphs in cryptography. 2019. 193

[53] Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 759–789, Darmstadt, Germany, May 19–23, 2019. Springer. 152

[54] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014. 38, 152, 155, 174, 175, 176, 178

[55] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. Light post-quantum signatures from quaternions and isogenies. *personal communication*, 2020. 174

[56] Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. Technical report, Cryptology ePrint Archive, Report 2019/166, 2019. 181

[57] Cyprien Delpech de Saint Guilhem, Péter Kutas, Christophe Petit, and Javier Silva. Séta: Supersingular encryption from torsion attacks. *IACR Cryptol. ePrint Arch.*, 2019:1291, 2019. 3, 151

[58] Christina Delfs and Steven D Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Designs, Codes and Cryptography*, 78(2):425–440, 2016. 170

[59] Alexander W Dent and Steven D Galbraith. Hidden pairings and trapdoor DDH groups. In *International Algorithmic Number Theory Symposium*, pages 436–451. Springer, 2006. 3, 179, 180, 181, 183, 184, 185, 186, 196

[60] Max Deuring. Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 14:197–272, 1941. 10.1007/BF02940746. 34

[61] Kirsten Eisenträger, Sean Hallgren, Kristin E. Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 329–368, Tel Aviv, Israel, April 29 – May 3, 2018. Springer. 37, 38, 180, 190

[62] Alex Escala and Jens Groth. Fine-tuning Groth-Sahai proofs. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 630–649, Buenos Aires, Argentina, March 26–28, 2014. Springer. 22, 45, 46

[63] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147, Santa Barbara, CA, USA, August 18–22, 2013. Springer. 1

[64] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Luis Villar. An algebraic framework for Diffie-Hellman assumptions. *Journal of Cryptology*, 30(1):242–288, January 2017. 24, 51

[65] Shimon Even, A.L. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. In *Inform. and Control 61*, pages 159–173, 1984. 72

[66] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Mathematical Cryptology*, 8(3):209–247, 2014. 39, 109, 110, 119, 121, 124

[67] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer. 8, 18

[68] Georg Fuchsbauer. Commuting signatures and verifiable encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 224–245, Tallinn, Estonia, May 15–19, 2011. Springer. 45

[69] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554, Santa Barbara, CA, USA, August 15–19, 1999. Springer. 8, 156

[70] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953, 2019. https://eprint.iacr.org/2019/953. 66, 67

[71] Steven D Galbraith. Constructing isogenies between elliptic curves over finite fields. *LMS Journal of Computation and Mathematics*, 2:118–138, 1999. 36, 37, 39, 170, 194

[72] Steven D Galbraith. *Mathematics of public key cryptography*. Cambridge University Press, 2012. 26

[73] Steven D Galbraith and James F McKee. Pairings on elliptic curves over finite commutative rings. In *IMA International Conference on Cryptography and Coding*, pages 392–409. Springer, 2005. 195, 196

[74] Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 63–91, Hanoi, Vietnam, December 4–8, 2016. Springer. 40, 110, 176

[75] Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 3–33, Hong Kong, China, December 3–7, 2017. Springer. 3, 109, 152

[76] Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. *Journal of Cryptology*, 33(1):130–175, January 2020. 3, 109, 152

[77] Juan A. Garay, Philip D. MacKenzie, and Ke Yang. Strengthening zero-knowledge protocols using signatures. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 177–194, Warsaw, Poland, May 4–8, 2003. Springer. 18

[78] Sanjam Garg, Mohammad Mahmoody, Daniel Masny, and Izaak Meckler. On the round complexity of OT extension. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 545–574, Santa Barbara, CA, USA, August 19–23, 2018. Springer. 67

[79] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645, Athens, Greece, May 26–30, 2013. Springer. 2, 65, 69

[80] Rosario Gennaro and Daniel Wichs. Fully homomorphic message authenticators. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 301–320, Bengalore, India, December 1–5, 2013. Springer. 2

[81] Essam Ghadafi and Jens Groth. Towards a classification of non-interactive computational assumptions in cyclic groups. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 66–96, Hong Kong, China, December 3–7, 2017. Springer. 1, 73

[82] Essam M Ghadafi. Sub-linear blind ring signatures without random oracles. In *IMA International Conference on Cryptography and Coding*, pages 304–323. Springer, 2013. 47

[83] Steven Goldfeder, Melissa Chase, and Greg Zaverucha. Efficient post-quantum zero-knowledge and signatures (draft). Cryptology ePrint Archive, Report 2016/1110, 2016. http://eprint.iacr.org/2016/1110. 19, 20

[84] Alonso González. Shorter ring signatures from standard assumptions. In *IACR International Workshop on Public Key Cryptography*, pages 99–126. Springer, 2019. 47

[85] Alonso González, Alejandro Hevia, and Carla Ràfols. QA-NIZK arguments in asymmetric groups: New tools and new constructions. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 605–629, Auckland, New Zealand, November 30 – December 3, 2015. Springer. 2, 26, 46, 48, 49, 76, 77, 90, 94

[86] Alonso González and Carla Ràfols. Shorter pairing-based arguments under standard assumptions. LNCS, pages 728–757. Springer, December 2019. 2, 66, 67, 68, 69, 70, 71, 72, 73, 90, 93, 94

[87] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459, Shanghai, China, December 3–7, 2006. Springer. 23, 45

[88] Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 192–208, Santa Barbara, CA, USA, August 16–20, 2009. Springer. 104

[89] Jens Groth. Short non-interactive zero-knowledge proofs. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 341–358, Singapore, December 5–9, 2010. Springer. 2

[90] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340, Singapore, December 5–9, 2010. Springer. 65

[91] Jens Groth. Efficient zero-knowledge arguments from two-tiered homomorphic commitments. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 431–448, Seoul, South Korea, December 4–8, 2011. Springer. 67, 69, 70, 104

[92] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326, Vienna, Austria, May 8–12, 2016. Springer. 2, 65

[93] Jens Groth, Markulf Kohlweiss, Mary Maller, Sarah Meiklejohn, and Ian Miers. Updatable and universal common reference strings with applications to zk-SNARKs. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 698–728, Santa Barbara, CA, USA, August 19–23, 2018. Springer. 66, 96

[94] Jens Groth and Steve Lu. A non-interactive shuffle with pairing based verifiability. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 51–67, Kuching, Malaysia, December 2–6, 2007. Springer. 45

[95] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111, Santa Barbara, CA, USA, August 20–24, 2006. Springer. 45

[96] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358, St. Petersburg, Russia, May 28 – June 1, 2006. Springer. 45

[97] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432, Istanbul, Turkey, April 13–17, 2008. Springer. 45, 66, 75

[98] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 341–371, Baltimore, MD, USA, November 12–15, 2017. Springer. 154, 155, 156, 167, 177

[99] Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. *Designs, Codes and Cryptography*, 80(1):29–61, 2016. 45

[100] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43:439–561, 2006. 111

[101] Thomas Icart. How to hash into elliptic curves. In *Annual International Cryptology Conference*, pages 303–316. Springer, 2009. 191

[102] 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. 151, 152, 154, 155, 174, 175

[103] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *PQCrypto 2011*, volume 7071 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2011. 36, 38, 39, 109

[104] David Jao and Vladimir Soukharev. Isogeny-based quantum-resistant undeniable signatures. In Michele Mosca, editor, *PQCrypto 2014*, volume 8772 of *Lecture Notes in Computer Science*, pages 160–179. Springer, 2014. 109

[105] Samuel Jaques and John M. Schanck. Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In *Advances in Cryptology – CRYPTO 2019*, pages 32–61. Springer, 2019. 36

[106] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–276, September 2004. 25

[107] Antoine Joux, Reynald Lercier, Nigel Smart, and Frederik Vercauteren. The number field sieve in the medium prime case. In *Annual International Cryptology Conference*, pages 326–344. Springer, 2006. 194

[108] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20, Bengalore, India, December 1–5, 2013. Springer. 46, 69

[109] Charanjit S. Jutla and Arnab Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 295–312, Santa Barbara, CA, USA, August 17–21, 2014. Springer. 46

[110] Yael Tauman Kalai, Omer Paneth, and Lisa Yang. How to delegate computations publicly. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1115–1124, Phoenix, AZ, USA, June 23–26, 2019. ACM Press. 66

[111] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Exploring constructions of compact NIZKs from various assumptions. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 639–669, Santa Barbara, CA, USA, August 18–22, 2019. Springer. 66

[112] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Compact NIZKs from standard assumptions on bilinear maps. In Vincent Rijmen and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, LNCS, pages 379–409. Springer, May 2020. 66

[113] Kiran S. Kedlaya and Christopher Umans. Fast polynomial factorization and modular composition. *SIAM J. Comput.*, 40(6):1767–1802, 2011. 142

[114] Eike Kiltz, Jiaxin Pan, and Hoeteck Wee. Structure-preserving signatures from standard assumptions, revisited. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 275–295, Santa Barbara, CA, USA, August 16–20, 2015. Springer. 46

[115] Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128, Sofia, Bulgaria, April 26–30, 2015. Springer. 46, 48, 69, 70, 76, 77, 93, 94

[116] Taechan Kim and Razvan Barbulescu. Extended tower number field sieve: A new complexity for the medium prime case. In *Annual International Cryptology Conference*, pages 543–571. Springer, 2016. 194

[117] David Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996. 32, 37, 139, 150

[118] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17A:418–432, 2014. 38, 111, 119, 127, 133, 134, 135, 137, 150

[119] Takeshi Koshiba and Katsuyuki Takashima. Pairing cryptography meets isogeny: A new framework of isogenous pairing groups. *IACR Cryptology ePrint Archive*, 2016:1138, 2016. 181

[120] Péter Kutas, Chloe Martindale, Lorenz Panny, Christophe Petit, Katherine E. Stange, and Victoria De Quehen. Weak instances of sidh under improved torsion attacks. *personal communication*, 2020. 174

[121] Péter Kutas, Christophe Petit, and Javier Silva. Trapdoor ddh groups from pairings and isogenies. *IACR Cryptol. ePrint Arch.*, 2019:1290, 2019. 3, 179

[122] Kristin E Lauter, Denis Charles, and Anton Mityagin. Trapdoor pairings, May 15 2012. US Patent 8,180,047. 181

[123] Hendrik W Lenstra Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, pages 649–673, 1987. 196

[124] Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 514–532, Copenhagen, Denmark, May 11–15, 2014. Springer. 46

[125] Benoît Libert, Thomas Peters, and Moti Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 296–316, Santa Barbara, CA, USA, August 16–20, 2015. Springer. 46

[126] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189, Taormina, Sicily, Italy, March 19–21, 2012. Springer. 65

[127] Helger Lipmaa. Prover-efficient commit-and-prove zero-knowledge SNARKs. In David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 16*, volume 9646 of *LNCS*, pages 185–206, Fes, Morocco, April 13–15, 2016. Springer. 76

[128] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In *ACM CCS 2019*, pages 2111–2128. ACM Press, 2019. 66, 67, 68, 71, 101

[129] Chloe Martindale and Lorenz Panny. How to not break SIDH. Cryptology ePrint Archive, Report 2019/558, 2019. `https://eprint.iacr.org/2019/558`. 160

[130] Alfred J Menezes, Tatsuaki Okamoto, and Scott A Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on information Theory*, 39(5):1639–1646, 1993. 182, 194

[131] David J Mireles Morales. An attack on disguised elliptic curves. *Journal of Mathematical Cryptology*, 2(1):1–8, 2008. 180, 185

[132] Paz Morillo, Carla Ràfols, and Jorge Luis Villar. The kernel matrix Diffie-Hellman assumption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 729–758, Hanoi, Vietnam, December 4–8, 2016. Springer. 1, 25, 73

[133] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109, Santa Barbara, CA, USA, August 17–21, 2003. Springer. 66

[134] Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. Hash function requirements for schnorr signatures. *J. Mathematical Cryptology*, 3(1):69–87, 2009. 20

[135] Phong Q. Nguyen and Damien Stehlé. Low-dimensional lattice basis reduction revisited. *ACM Transactions on Algorithms*, 5(4), 2009. 134, 137

[136] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *Annual International Cryptology Conference*, pages 111–126. Springer, 2002. 8

[137] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999. 185

[138] Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 330–353, Hong Kong, China, December 3–7, 2017. Springer. 3, 40

[139] Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 330–353. Springer International Publishing, 2017. 110, 152, 161, 162, 163, 174, 176

[140] Christophe Petit and Kristin E Lauter. Hard and easy problems for supersingular isogeny graphs. Cryptology ePrint Archive, Report 2017/962, 2017. https://eprint.iacr.org/2017/962. 37, 38, 180, 190, 191

[141] Arnold K. Pizer. Ramanujan graphs and Hecke operators. *Bulletin of the American Mathematical Society*, 23(1):127–137, 1990. 32

[142] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 387–398, Saragossa, Spain, May 12–16, 1996. Springer. 18

[143] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000. 18

[144] Manoj Prabhakaran and Rui Xue. Statistically hiding sets. In *Cryptographers' Track at the RSA Conference*, pages 100–116. Springer, 2009. 180, 183, 184

[145] Carla Ràfols. Stretching groth-sahai: NIZK proofs of partial satisfiability. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 247–276, Warsaw, Poland, March 23–25, 2015. Springer. 46

[146] Carla Ràfols and Javier Silva. QA-NIZK arguments of same opening for bilateral commitments. In *AFRICACRYPT 20*, LNCS, pages 3–23. Springer, 2020. 2, 45

[147] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 520–551, Tel Aviv, Israel, April 29 – May 3, 2018. Springer. 154, 157, 158, 167, 177

[148] Yannick Seurin. New constructions and applications of trapdoor DDH groups. In *International Workshop on Public Key Cryptography*, pages 443–460. Springer, 2013. 3, 180, 183, 185, 186, 190, 200

[149] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.*, 41(2):303–332, 1999. 7

[150] Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009. 26, 30, 160

[151] Anton Stolbunov. Cryptographic schemes based on isogenies. Doctoral thesis, 2012. 152

[152] Anton Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Adv. in Math. of Comm.*, 4(2):215–235, 2010. 151

[153] Seiichiro Tani. Claw finding algorithms using quantum walk. *Theoretical Computer Science*, 410(50):5285–5297, 2009. 36

[154] Oleg Taraskin, Vladimir Soukharev, David Jao, and Jason LeGrow. An isogeny-based password-authenticated key establishment protocol. Cryptology ePrint Archive, Report 2018/886, 2018. `https://eprint.iacr.org/2018/886`. 40

[155] Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 192–216, Beijing, China, October 31 – November 3, 2016. Springer. 156

[156] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015*, volume 9057 of *Lecture Notes in Computer Science*, pages 755–784. Springer, 2015. 19, 125

[157] Dominique Unruh. Post-quantum security of Fiat-Shamir. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 65–95, Hong Kong, China, December 3–7, 2017. Springer. 7, 19

[158] Paul C Van Oorschot and Michael J Wiener. Parallel collision search with cryptanalytic applications. *Journal of cryptology*, 12(1):1–28, 1999. 36

[159] Eric R Verheul. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. *Journal of Cryptology*, 17(4):277–296, 2004. 182

[160] Marie-France Vignéras. The arithmetic of quaternion algebra. `http://maths.nju.edu.cn/~guoxj/notes/qa.pdf`, 2006. 32

[161] John Voight. Quaternion algebras, 2017. `https://math.dartmouth.edu/~jvoight/quat-book.pdf`. 32

[162] Joachim von zur Gathen and Victor Shoup. Computing Frobenius maps and factoring polynomials. *Computational Complexity*, 2:187–224, 1992. 117, 142, 144

[163] Jacques Vélu. Isogénies entre courbes elliptiques. *Communications de l'Académie royale des Sciences de Paris*, 273:238–241, 1971. 117

[164] Lawrence C. Washington. *Elliptic curves: number theory and cryptography*. Chapman & Hall/CRC, 2008. 26

[165] William C. Waterhouse. Abelian varieties over finite fields. *Annales scientifiques de l'E.N.S.*, 2:521–560, 1969. 139

[166] Sun Xi, Haibo Tian, and Yumin Wang. Toward quantum-resistant strong designated verifier signature from isogenies. *International Journal of Grid and Utility Computing*, 5(2):292–296, September 2012. 109

[167] Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A post-quantum digital signature scheme based on supersingular isogenies. In *Financial Crypto*, volume 2017, 2017. 21, 110, 119, 121, 148, 150

Appendix A

Notation

$\#\mathsf{S}$: cardinality of a set S.

$x \leftarrow \mathsf{S}$: x is sampled from the uniform distribution on S.

$\mathcal{A}^{\mathcal{O}}$: algorithm $\mathcal{A}$ with oracle access to algorithm $\mathcal{O}$.

$\mathbb{Z}_n$: group of residue classes modulo n.

$\mathbb{F}_q$: finite field of size q.

$\overline{\mathbb{F}}$: algebraic closure of the field $\mathbb{F}$.

$E/\mathbb{F}$: elliptic curve defined over the field $\mathbb{F}$.

$E(\mathbb{F})$: $\mathbb{F}$-rational points of E.

$E[m]$: m-torsion of E.

∞: point at infinity (neutral element) of an elliptic curve.

$\mathbb{Q}_p$: field of p-adic numbers.

$\mathbf{B}_{p,\infty}$: quaternion algebra over $\mathbb{Q}$ ramified at p and ∞.

$\mathbf{A} \circ \mathbf{B}$: Hadamard product of the matrices $\mathbf{A}$ and $\mathbf{B}$.

DPT: deterministic polynomial time.

PPT: probabilistic polynomial time.

CRS: common reference string.

SRS: structured reference string.

ROM: random oracle model.

QROM: quantum random oracle model.

(QA)-NIZK: (quasi-adaptive) non-interactive zero-knowledge.

PKE: public-key encryption.

KEM: key encapsulation mechanism.

Appendix B

Publications

Journals

2020 Galbraith, S. D., Petit, C., & Silva, J. Identification protocols and signature schemes based on supersingular isogeny problems. *Journal of Cryptology*, 33(1), 130-175.

Abstract. We present signature schemes whose security relies on computational assumptions relating to isogeny graphs of supersingular elliptic curves. We give two schemes, both of them based on interactive identification protocols. The first identification protocol is due to De Feo, Jao and Plût. The second one, and the main contribution of the paper, makes novel use of an algorithm of Kohel, Lauter, Petit and Tignol for the quaternion version of the ℓ-isogeny problem, for which we provide a more complete description and analysis, and is based on a more standard and potentially stronger computational problem. Both identification protocols lead to signatures that are existentially unforgeable under chosen message attacks in the random oracle model using the well-known Fiat-Shamir transform, and in the quantum random oracle model using another transform due to Unruh. A version of the first signature scheme was independently published by Yoo, Azarderakhsh, Jalali, Jao and Soukharev. This is the full version of a paper published at ASIACRYPT 2017.

Conference proceedings

2020 Kutas, P., Petit, C., & Silva, J. Trapdoor DDH groups from pairings and isogenies. To appear in *SAC 2020*.

Abstract. Trapdoor DDH groups are an appealing cryptographic primitive introduced by Dent–Galbraith (ANTS 2006), where DDH instances are hard to solve unless provided with additional information (i.e., a trapdoor). In this paper, we introduce a new trapdoor DDH group construction using pairings and isogenies of supersingular elliptic curves, and present two instantiations of it. The construction solves all shortcomings of previous constructions as identified by Seurin (RSA 2013). We also present partial attacks on a previous construction due to Dent–Galbraith, and we provide a formal security definition of the related notion of "trapdoor pairings".

2020 Ráfols, C., & Silva, J. QA-NIZK arguments of same opening for bilateral commitments. In *International Conference on Cryptology in Africa* (pp. 3-23). Springer, Cham.

Abstract. Zero-knowledge proofs of satisfiability of linear equations over a group are often used as a building block of more complex protocols. In particular, in an asymmetric bilinear group we often have two commitments in different sides of the pairing, and we want to prove that they open to the same value. This problem was tackled by González, Hevia and Ràfols (ASIACRYPT 2015), who presented an aggregated proof, in the QA-NIZK setting, consisting of only four group elements. In this work, we present a more efficient proof, which is based on the same assumptions and consists of three group elements. We argue that our construction is optimal in terms of proof size.

2019 Daza, V., González, A., Pindado, Z., Ràfols, C., & Silva, J. Shorter quadratic QA-NIZK proofs. In *IACR International Workshop on Public Key Cryptography* (pp. 314-343). Springer, Cham.

Abstract. Despite recent advances in the area of pairing-friendly Non-Interactive Zero-Knowledge proofs, there have not been many efficiency improvements in constructing arguments of satisfiability of quadratic (and larger degree) equations since the publication of the Groth-Sahai proof system (JoC 2012).In this work, we address the problem of aggregating such proofs using techniques derived from the interactive setting and recent constructions of SNARKs. For certain types of quadratic equations, this problem was investigated before by Gonzáalez et al. (ASIACRYPT 2015). Compared to their result, we reduce the proof size by approximately 50% and the common reference string from quadratic to linear, at the price of using less standard computational assumptions. A theoretical motivation for our work is to investigate how efficient NIZK proofs based on falsifiable assumptions can be. On the practical side, quadratic equations appear naturally in several cryptographic schemes like shuffle and range arguments.

2017 Galbraith, S. D., Petit, C., & Silva, J. Identification protocols and signature schemes based on supersingular isogeny problems. In *International Conference on the Theory and Application of Cryptology and Information Security* (pp. 3-33). Springer, Cham. **Best paper award.**

Abstract. We present signature schemes whose security relies on computational assumptions relating to isogeny graphs of supersingular elliptic curves. We give two schemes, both of them based on interactive identification protocols. The first identification protocol is due to De Feo, Jao and Plût. The second one, and the main contribution of the paper, makes novel use of an algorithm of Kohel, Lauter, Petit and Tignol for the quaternion version of the ℓ-isogeny problem, for which we provide a more complete description and analysis, and is based on a more standard and potentially stronger computational problem. Both identification protocols lead to signatures that are existentially unforgeable under chosen message attacks in the random oracle model using the well-known Fiat-Shamir transform, and in the quantum random oracle model using another transform due to Unruh. A version of the first signature scheme was independently published by Yoo, Azarderakhsh, Jalali, Jao and Soukharev.

Preprints

2019 de Saint Guilhem, C. D., Kutas, P., Petit, C., & Silva, J. SÉTA: Supersingular encryption from torsion attacks.

Abstract. We present *SÉTA*,[1] a new family of PKE schemes with post-quantum security based on isogenies of supersingular elliptic curves. It is constructed from a new family of trapdoor one-way functions, where the inversion algorithm uses Petit's 2017 attack to compute an isogeny between supersingular elliptic curves given images of torsion points. We use this method as a decryption mechanism to first build a OW-CPA scheme; we then prove further properties to obtain IND-CCA security in the quantum random oracle model using generic transformations, both for a PKE scheme and a KEM. We compare our protocols with the NIST proposal SIKE from both security and efficiency points of view, and we discuss how further work, including on cryptanalysis, may affect this comparison.

2020 Ràfols, C., & Silva, J. Circuit satisfiability arguments from two-tiered commitments.

Abstract. We describe very efficient arguments for proving circuit satisfiability in bilinear groups based on falsifiable assumptions. When the input is public and the circuit has low depth, these constructions improve the state of the art by an order of magnitude. Our approach is to obtain first an efficient (almost) non-interactive argument for circuit satisfiability that can be verified with pairing product equations. Then we reduce communication complexity by using the recent argument for inner pairing products (IPP) of Bünz et al. (EPRINT 2019).
Our first construction applies this idea to the argument of correct circuit evaluation of González and Ràfols (ASIACRYPT 2019). For a circuit of depth d, the result is an argument for correct evaluation with a communication cost of $O(\log d)$, but with a structured reference string that depends on the circuit. Our second construction overcomes this limitation by introducing a novel interactive argument with universal and updateable structured reference string and a communication cost of $O(d)$, and secure under falsifiable assumptions. We then show how to compile this argument with the IPP argument to achieve communication complexity $O(\log d)$. The security of our constructions relies on a W-assumption, W being the width of the circuit. We extend both our constructions to the setting where the circuit input is secret and the proof is zero-knowledge, with a communication cost overhead that is logarithmic in the input.