Corso di dottorato di ricerca in
MATEMATICA

Ciclo del corso
XXXIV

Titolo della tesi

On the design and cryptanalysis of Isogeny-Based Public Key Encryption schemes

Candidato:
Tako Boris Fouotsa Firma:_____

Relatore di tesi:
Christophe Petit Firma:_____

Relatore di tesi:
Fabrizio Barroero Firma:_____

Coordinatore:
Alessandro Giuliani Firma:_____

Anno Accademico 2021-2022,
Dipartimento di Matematica e Fisica, Università Degli Studi di Roma Tre
Largo San Leonardo Murialdo 1, 00146, Roma

On the design and cryptanalysis of Isogeny-Based Public Key Encryption schemes

Tako Boris Fouotsa

Advisors:

Christophe Petit
ULB Belgium and UoB United Kingdom
and
Fabrizio Barroero
Università degli Studi Roma Tre

23rd February, 2022

Email: fouotsabcrb@gmail.com
Website: https://borisfouotsa.github.io
Orcid ID: 0000-0003-1821-8406

First edition February 2022

This research was funded by Università degli Studi di Roma Tre through its regular PhD scholarship program.

This thesis has been approved by the supervisors and the anonymous reviewers. The composition of the defense committee is as follows:

President:	Prof. Francesco Pappalardi
Members:	Prof. Giulio Codogni (Università degli Studi di Roma Tor Vergata) Prof. Andrea Ferraguti (Scuola Normale Superiore di Pisa) Prof. Michel Waldschmidt (Sorbonne Université)

To the loving memory of my father,
who saw the start but will never see the end.

Acknowledgements

I would like to use this opportunity to express my sincere gratitude to my lovely supervisors Christophe and Fabrizio for their infinite support, care, encouragements, and advice. I learned a lot from you. I am blessed to have you as PhD advisors.

My thanks to Francesco Pappalardi who, during the CIMPA school in Kinshasa 2018 where we met for the first time, advised me to apply for PhD positions in Rome. Since then, he has been very caring and supportive.

My thanks to the former Mathematics PhD coordinator, Angelo Lopez, for his kind support during my first visa request process, and for his care during this long journey.

My thanks to Emmanuel Fouotsa, a mentor, a collaborator, a friend. He is the one who suggested Isogeny-Based Cryptography to me in 2017/2018 as research interest when we first met. Since then, we have been working hand to hand to foster cryptologic research in Cameroon.

My thanks to my co-authors from who I have learned a lot during the collaborations. Working with you was very enriching. Particular thanks to Péter Kutas and Simon-Philipp Merz for the multiple fruitful discussions we have had during this journey.

My thanks to the anonymous reviewers of this thesis and the defense committee members for the time they put in to read this work and for their helpful feedback.

My thanks to my friends and colleagues at ROMA TRE. You proved to me that doing a PhD in mathematics, apart from being stressful, could also be very fun. Particular thanks to Andam, Andrea, Daniele, Davide, Edmond, Louis, Manoj, Myrla, Nilofar and Yannick.

Special thanks to my mother, Mafouoyo Fouotsa Epse Tako Beatrice, and to my brothers and sisters, for their unconditional love, care, support and assistance during this journey.

To my lovely wife, Mbogning Kelly Jodelle, I express my gratitude. Thank you for your affection and full time support despite the distance. You have been my source of motivation during these 3 years.

My thanks to all those who I have not mentioned here, but who in anyway, contributed actively or passively to the success of this work. Thanks to the almighty God.

Tako Boris Fouotsa
Rome, February 2022.

Abstract

The six and a half hours Facebook outage[1] of October 4, 2021 proved to the world that in the 21st century, life on earth without communication technologies is impossible. When utilising these technologies, our communications are protected using cryptographic protocols that provide secrecy, integrity, privacy,... Meanwhile, the security of the widely deployed cryptographic protocols we use today relies on some mathematical problems that are difficult to solve efficiently with our current computers. In 1994, Peter Shor[2] designed an algorithm that can solve these hard problems using a sufficiently large quantum computer. Since then, mathematicians, cryptographers and engineers have been working hand to hand to come up with new cryptographic protocols relying on new hard problems that we believe would remain secure in the presence of a large quantum computer.

Among the new hard problems suggested, *isogenies* (maps between elliptic curves) are particularly attracting since they offer very compact protocols (they use less bandwidth). Meanwhile, they are computationally slow. Also, the field of Isogeny-Based Cryptography is relatively young since the first isogeny-based cryptographic protocols appeared only a decade and a half ago. This suggests that more research is needed in the field: protocol design, cryptanalysis, efficiency improvement, optimized implementation, ...

This thesis focuses on the design of isogeny-based public key encryption schemes and key exchange protocols, and on the cryptanalysis of isogeny-based protocols. It reports five contributions to the field of Isogeny-based Post-Quantum Cryptography. Three of these contributions are protocol designs, while two of them are cryptanalysis results.

The first design is SimS (chapter 3): **Sim**plified **S**iGamal. SimS is an IND-CCA secure hash function free public key encryption scheme obtained by simplifying and improving SiGamal and C-SiGamal, two CSIDH based IND-CPA public key encryption schemes published by Moriya et al. at Asiacrypt 2020. The second design is SÉTA (chapter 4): **S**upersingular **E**ncryption from **T**orsion points **A**ttack, a public key encryption scheme obtained by transforming the Petit's torsion points attack into a trapdoor mechanism. Moreover, we provided a new general isogeny assumption called the *Uber Isogeny Assumption* which underlies the security of most isogeny based protocols. The third design is HealSIDH (chapter 5) **Heal**ed **SIDH**: an SIDH type interactive key exchange which enables static-static secret keys. HealSIDH is built on a direct countermeasure to the GPST adaptive attack on SIDH that we

[1] https://www.facebook.com/business/news/update-about-the-october-4th-outage
[2] http://www-math.mit.edu/~shor/

introduce. We derive two public key encryption schemes SHealS and HealS from HealSIDH, they both permit encryption key reuse.

The first cryptanalysis result (chapter 6) is a generalisation of the GPST reduction of the isogeny problem in SIDH to that of the computation of the endomorphism ring. In the GPST reduction, the secret isogeny needs to be relatively short. Our generalisation permits to have the same reduction for SIDH instances with larger isogeny degrees, BSIDH for instance. The second cryptanalysis result (chapter 7) is a new adaptive attack on SIDH which uses the Petit's torsion points attack as subroutine.

Contents

Chapter 1

Introduction

How would you feel if someone somewhere on earth was able to read all your emails, all your whatsapp/telegram/signal/... messages; had unlimited access to your bank account? As far as I am concerned, I would feel very bad.

In fact, all our emails, messages in social apps and our bank transactions are encrypted and digitally signed before being sent through the internet to their recipient. The encryption, which in this case is generally done using a Public Key Encryption (PKE) algorithm, ensures that only the authorised recipient is able to decrypt the encrypted messages and learn the hidden information. The digital signature, which is done using a Digital Signature Algorithm (DSA), ensures that anyone who has access to the message can effectively verify its integrity and the identity of the sender. This prohibits intruders to alter or change your messages, or to send messages on your behalf.

The public key encryption schemes and digital signature algorithms we use today are built on top of "computationally hard[1]" mathematical problems: the integer factorization problem and the discrete logarithm problem. The most famous and used version of the integer factorization problem is as follows: *given a composite integer n of the form $n = p * q$ with $p \approx q$, compute the primes p and q.* The discrete logarithm problem is as follows: *given a cyclic group G, a generator g of G and a random element h in g, compute an integer e such that $h = g^e$ in G.* These problems are computationally hard to solve when n is large or the order of the group G is a large prime.

It happens that a problem being "computationally hard" depends on the "computer model" in play. From their invention till nowadays, our computers treat information using *classical bits: 0 and 1.* They are hence called *classical computers.* In 1980, Paul Benioff [Ben80] proposed a quantum mechanical model of the Turing machine, that is a new computer model where information is treated using *quantum bits: linear combination of a 0 and a 1.* They are hence called *quantum computers.* Few years later, works of Richard Feynman [Fey82] and Yuri Manin [Man80] suggested that a quantum computer had the potential to perform simulations which are unfeasible on a classical computer. In 1994, Peter W. Shor [Sho94] described an algorithm that efficiently solves the integer factorization problem and the discrete logarithm problem using a quantum computer. More precisely, a later version of his paper has the following abstract.

[1] Problems that today most powerful computer will take many thousands of years to solve. Not to be confused with impossible problems.

"A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time of at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored." [Sho97]

Shor's discovery implies that the construction of large scale quantum computers would make our public key encryption schemes and digital signature algorithms insecure. Hence all our emails, private messages, bank account passwords, ... will be accessible to any malicious party possessing a large enough quantum computer. Considerable progress has been done in the understanding of quantum mechanics and the design of a quantum computer in the last two decades, hence amplifying the threat of quantum computers on the cryptographic protocols we use today.

As a response to the threat, cryptographers have been intensively working on new hard problems and new protocols that are made to work on classical computers, but will not be vulnerable even in the presence of quantum computers. These protocols are said to be *post-quantum secure*.

In December 2016, NIST[2] launched a standardization process for post-quantum secure protocols [Nat]. The aim of this process is to choose new secure post-quantum algorithms that will replace the ones we use today. We stand today at the third round of the process. The hard problems underlying the security of the schemes that are still in the competition come from *Lattices*, *Codes*, *Multivariate*, ***Isogenies*** and *properties of hash functions* (hashed-based).

1.1 — The rise of isogenies

Isogenies are quite young as a candidate for hard problems underlying the security of cryptographic protocols. Their official[3] appearance as computationally hard problem in cryptography goes back to 2006 with the Charles-Goren-Lauter (CGL) [CLG09] hash function and the Couveignes-Rostotsev-Stolbunov (CRS) [Cou06; RS06] key exchange. Since then, isogeny-based cryptography has grown rapidly, even better after the submission of SIKE (Supersingular Isogeny Key Encapsulation) to the NIST standardisation process in 2016. SIKE is the only isogeny-based scheme submitted to the competition and has made it to the third round as an alternate candidate (candidates that will go through a fourth round before finally being standardised or eliminated).

Isogeny-based protocols, despite being relatively slow when compared to other candidates, provide relatively short keys. This makes them particularly interesting since they can better fit devices with memory constraints. Also, their rich mathematical structure makes them very promising, for now the only known practically efficient

[2] National Institute of Standards and Technologies, USA.

[3] Note that an isogeny-based schemes was suggested by Couveignes [Cou06] in 1997 in a paper that was rejected at Crypto97. The result was presented at the *Séminaire de complexité et cryptographie* at Ecole Normale Supérieure, but the paper was not made public till 2006.

Post-Quantum alternative to the classic Diffie-Hellman key exchange protocol, that is CSIDH, is based on isogenies.

Since 2006, there have been an increasing amount of results in the field. Nevertheless, there is lot to be explored in terms of designing new isogeny-based primitives, cryptanalysing various assumptions and problems introduced. Moreover, theoretical, software and hardware acceleration of isogeny-based schemes is needed. In fact, being around only for about one and a half decade, the real potential of isogenies is still to be determined. On the opposite side, more cryptanalysis is needed in order better evaluate the security of isogeny based schemes and discard those of these many schemes being designed that are insecure. This thesis goes in this direction and focuses on *the design of new isogeny-based public key encryption schemes, and the cryptanalysis of existing ones.*

1.2 — Results and outline

This thesis contains three new public key encryption schemes designs: SimS [FP21c], SETA [Feo+19] and SHealS [FP21b]; and two cryptanalysis results [FKMT21] and [FP21a]. The remainder of this thesis is organised as follows.

Chapter 2. Chapter 2 surveys some mathematical background relevant for the rest of the thesis.

Chapter 3. At Asiacrypt 2020, Moriya et al. introduced two new IND-CPA secure supersingular isogeny based Public Key Encryption (PKE) protocols: SiGamal and C-SiGamal. Unlike the PKEs canonically derived from SIDH and CSIDH, the new protocols provide IND-CPA security without the use of hash functions. SiGamal and C-SiGamal are however not IND-CCA secure. Moriya et al. suggested a variant of SiGamal that could be IND-CCA secure, but left its study as an open problem.

In Chapter 3, we revisit the protocols introduced by Moriya et al. First, we show that the SiGamal variant suggested by Moriya et al. for IND-CCA security is, in fact, not IND-CCA secure. Secondly, we propose a new isogeny-based PKE protocol named SimS, obtained by simplifying SiGamal. SimS has smaller public keys and ciphertexts than (C-)SiGamal and it is more efficient. We prove that SimS is IND-CCA secure under CSIDH security assumptions and one Knowledge of Exponent-type assumption we introduce. Interestingly, SimS is also much closer to the CSIDH protocol, facilitating a comparison between SiGamal and CSIDH.

Chapter 4. In Chapter 4, we present *Séta*,[4] a new family of public-key encryption 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 so called *torsion attacks* on SIDH to compute an isogeny between supersingular elliptic curves given an endomorphism of the starting curve and images of torsion points. We prove the OW-CPA security of Séta and present an IND-CCA variant using the post-quantum OAEP transformation. Several variants for key generation are explored together with their impact on the selection of parameters, such as the base prime of the scheme. We furthermore formalise an

[4] To be pronounced [ʃeːtɒ] meaning "walk" in Hungarian.

"uber" isogeny assumption framework which aims to generalize computational isogeny problems encountered in schemes including SIDH, CSDIH, OSIDH and ours. Finally, we carefully select parameters to achieve a balance between security and run-times and present experimental results from our implementation.

Chapter 5. In 2016, Galbraith et al. presented an adaptive attack on the SIDH key exchange protocol. In SIKE, one applies a variant of the Fujisaki-Okamoto transform to force Bob to reveal his encryption key to Alice, which Alice then uses to re-encrypt Bob's ciphertext and verify its validity. Therefore, Bob cannot reuse his encryption keys. There have been two other proposed countermeasures enabling static-static private keys: k-SIDH and its variant by Jao and Urbanik. These countermeasures are relatively expensive since they consist in running multiple parallel instances of SIDH.

In Chapter 5, firstly, we propose a new countermeasure to the GPST adaptive attack on SIDH. Our countermeasure does not require key disclosure as in SIKE, nor multiple parallel instances as in k-SIDH. We translate our countermeasure into a key validation method for SIDH-type schemes. Secondly, we use our key validation to design HealSIDH, an efficient SIDH-type static-static key interactive exchange protocol. Thirdly, we derive a PKE scheme SHealS using HealSIDH. SHealS uses larger primes compared to SIKE, has larger keys and ciphertexts, but only 4 isogenies are computed in a full execution of the scheme, as opposed to 5 isogenies in SIKE. We prove that SHealS is IND-CPA secure relying on a new assumption we introduce and we conjecture its IND-CCA security. We suggest HealS, a variant of SHealS using a smaller prime, providing smaller keys and ciphertexts.

As a result, HealSIDH is a practically efficient SIDH based (interactive) key exchange incorporating a "direct" countermeasure to the GPST adaptive attack.

Chapter 6. It has recently been rigorously proven (and was previously known relying on certain heuristics) that the general supersingular isogeny problem reduces to the supersingular endomorphism ring computation problem. However, in order to attack SIDH-type schemes, one requires a particular isogeny which is usually not returned by the general reduction. At Asiacrypt 2016, Galbraith et al. presented a polynomial-time reduction of the problem of finding the secret isogeny in SIDH to the problem of computing the endomorphism ring of a supersingular elliptic curve. Their method exploits the fact that secret isogenies in SIDH are short, and thus it does not extend to other SIDH-type schemes, where this condition is not fulfilled.

In Chapter 6, we present a more general reduction algorithm that generalises to all SIDH-type schemes. The main idea of our algorithm is to exploit available torsion point images together with the KLPT algorithm to obtain a linear system of equations over a certain residue class ring. We show that this system will have a unique solution that can be lifted to the integers if some mild conditions on the parameters are satisfied. This lift then yields the secret isogeny. One consequence of this work is that the choice of the prime p in B-SIDH is tight.

Chapter 7. The SIDH key exchange is the main building block of SIKE, the only isogeny based scheme involved in the NIST standardization process. In 2016, Galbraith et al. presented an adaptive attack on SIDH. In this attack, a malicious party

manipulates the torsion points in his public key in order to recover an honest party's static secret key, when having access to a key exchange oracle. In 2017, Petit designed a passive attack (which was improved by de Quehen et al. in 2020) that exploits the torsion point information available in SIDH public key to recover the secret isogeny when the endomorphism ring of the starting curve is known.

In Chapter 7, firstly, we generalize the torsion point attacks by de Quehen et al. Secondly, we introduce a new adaptive attack vector on SIDH-type schemes. Our attack uses the access to a key exchange oracle to recover the action of the secret isogeny on larger subgroups. This leads to an unbalanced SIDH instance for which the secret isogeny can be recovered in polynomial time using the generalized torsion point attacks. Our attack is different from the GPST adaptive attack and constitutes a new cryptanalytic tool for isogeny based cryptography. This result proves that the torsion point attacks are relevant to SIDH parameters in an adaptive attack setting. We suggest attack parameters for some SIDH primes and discuss some countermeasures.

Chapter 8. Here we summarise the thesis and discuss some further work.

1.3 — Relevance of the thesis contribution

Being built from a trapdoor one way function, Séta is fundamentally different from SIDH and CSIDH which are Diffie-Hellman type schemes. This suggests that Séta may be suitable as building block in some advanced schemes which could not be constructed using CSIDH or SIDH.

Since 2016, year at which the GPST adaptive attack on SIDH was published, to the best of our knowledge, all the suggested countermeasures suggested till date are very costly. Our countermeasure is less costly. It appears to be the first trial to "directly" counter the attack. In fact, previous attempts are more generic since they involve key disclosure (in SIKE [Jao+20]), multiple parallel instances (in k-SIDH [AJL17]) or signing the public key with a slow and large signature (Proof of Isogeny knowledge [FDGZ21]). We believe our suggested countermeasure has a strong potential and that after further refinements, it may be used to design isogeny-based public key encryption schemes that compete with SIKE in terms of efficiency and key sizes. Such schemes would be amazingly interesting in the sense that as a plus, they will be compatible with static-static keys. The hence obtained SIDH type key exchange, as Séta, may also be a suitable building block for advanced schemes.

Our cryptanalysis results foster the understanding of the security of SIDH. Our reduction presented in Chapter 6 completes the reduction of the isogeny problem with torsion points (in SIDH type schemes) to the endomorphism ring computation problem, reduction which was established for SIDH primes by Galbraith-Petit-Shani-Ti [GPST16]. Our new adaptive attack on SIDH proves that any SIDH-type scheme becomes vulnerable to Petit's torsion points attack (and improvements) in an adaptive setting, regardless of the parameters used in the scheme. Therefore, in a setting where adaptive security matters, any SIDH type scheme needs to be protected against these torsion points attacks. This implies that in SIDH type schemes where one does not use the Fujisaki-Okamoto transform or the public keys are not signed using the Proof of Isogeny Knowledge presented in [FDGZ21], one needs to set the starting curve as a random supersingular curve with unknown endomorphism ring. Up to date, the later can only be done through a trusted setup that will generate the curve and

forget its endomorphism ring. In fact, generating supersingular curves with unknown endomorphism ring is a hard problem.

Chapter 2

Preliminaries

This thesis focuses on the design and the cryptanalysis of isogeny-based protocols. This chapter provides some general background on Public key cryptography, elliptic curves, isogenies, endomorphism rings of elliptic curves, isogeny graphs, and the central problems in isogeny-based cryptography.

2.1 — Public key cryptography

In general, for two parties to securely communicate through a public channel (such as the internet, ...) we expect them to have established some shared secret s which is used to encrypt and decrypt the messages that are sent through the public channel. This type of encryption falls into the stream of *Symmetric Cryptography* (or *private key cryptography*). In symmetric cryptography, the keys used to encrypt and decrypt messages are identical. This implies that the two parties need to agree on the key before their very first communication. One easy method for agreeing on the secret key to be used is to set up a meeting, both parties travel to the meeting point, agree on the key they will use, then each party returns to his residence. But this option is very costly. One solves this issue using Public Key Cryptography. In public key cryptography, each party chooses his secret key sk which is always kept secret, and his public key pk which is made public. This secret key/public key pair can then be later used to establish shared secrets (to be used in symmetric encryption schemes) through *key exchange protocols*, to decrypt received encrypted messages or to encrypt messages to other parties using their public key through a *public key encryption scheme*. In public key cryptography, besides key exchange protocols that are used to establish shared secrets through an insecure channel and public key encryption schemes that are used to encrypt messages and hence provide message secrecy, there are digital signature schemes that are used to provide message integrity and authenticity with respect to the sender. Key exchange protocols and public key encryption schemes are relevant for our thesis, we hence provide some general background about them.

2.1.1 – Key exchange protocol. A *key exchange protocol* is a cryptographic protocol involving two parties A as Alice and B as Bob, who secretly choose some uniformly random secret keys sk_A and sk_B respectively, then use this secret keys to compute some public keys pk_A and pk_B respectively. The public keys are exchanged through a public channel, then each party does some computation to recover some secrets s_A and s_B. More formally, a key exchange protocol is a description of three probabilistic polynomial time algorithms Setup, Key Generation and Key Exchange such that

- Setup takes the security parameter[1] λ as input and returns a set pp of public parameters;
- Key Generation takes the set of public parameters as input and returns the a pair $(\mathsf{sk}, \mathsf{pk})$ where sk is the secret key and pk is the pubic key;
- Key Exchange takes one party's secret key sk and another party's public key pk' and returns a secret value s, say s_A for the secret value computed by Alice and s_B for the secret value computed by Bob.

As its name *key exchange protocol* indicates, at the end of the process, the secret value computed by both parties using Key Exchange should be the same, that is $\mathsf{s}_A = \mathsf{s}_B$. In this case we say that the key exchange protocol is *correct*. This secret value $\mathsf{s} = \mathsf{s}_A = \mathsf{s}_B$ is the shared secret (the key that was exchanged).

The very first key exchange protocol was proposed by Diffie and Hellman in 1976 [DH76] and is known today as the Diffie-Hellman key exchange. The Diffie-Hellman key exchange is one of the most important protocols in public key cryptography. Its publication marked the beginning of *modern cryptography*[2] era. The idea of the construction is quite simple. The Diffie-Hellman key exchange is as follows.

Setup: let G be a cyclic (multiplicative) group of prime order q and let g be a generator of G. The public parameters are G, g and q.

Key Generation: Choose a uniformly random integer $a \in \{0, 1, \cdots, q-1\}$ and compute g^a. The secret key is $\mathsf{sk} = a$ and the public key is $\mathsf{pk} = g^a$.

Key Exchange: To establish a shared secret with Bob, Alice retrieves Bob's public key $\mathsf{pk}_B \in G$ and computes $\mathsf{s}_A = \mathsf{pk}_B^{\mathsf{sk}_A}$. Bob also retrieves Alice's public key $\mathsf{pk}_A \in G$ and computes $\mathsf{s}_B = \mathsf{pk}_A^{\mathsf{sk}_B}$. We have $\mathsf{s}_A = \mathsf{pk}_B^{\mathsf{sk}_A} = g^{\mathsf{sk}_A \mathsf{sk}_B} = \mathsf{pk}_A^{\mathsf{sk}_B} = \mathsf{s}_B$.

Note that for the Diffie-Hellman scheme to be efficient, the exponentiation in the group G needs to run in polynomial time in the size of the exponent and of the group order q. Now let us discuss the hard problems underlying the security of the Diffie-Hellman key exchange. There are several ways of attacking a key exchange protocol.

Directly recovering one party's secret key. Here, the adversary tries to recover the secret key of one of the parties from the knowledge of the public parameters and the public key. In the Diffie-Hellman key exchange, this corresponds to inverting the exponentiation done during the Key Generation. This is in fact the Discrete Logarithm Problem (DLP) in the group G.

Problem 2.1.1 (DLP)**.** *Let G be a cyclic group of prime order q and let g be a generator of G. Given a uniformly random element $h \in G$, compute $x \in \{0, 1, \cdots, q-1\}$ such that $h = g^x$.*

[1] The security parameter λ of a cryptographic protocol is in general the logarithm of the time (or work load) needed to break the scheme. For example, to break a cryptographic scheme with security parameter 128, you are expected to perform at least 2^{128} mathematical operations.

[2] Modern cryptography era (from the 1970's upward) refers to the era where the algorithms used in cryptographic protocols are public and their security relies on mathematical problem that are expected to be hard to solve. As opposed to *ancient cryptography*, where the security of cryptographic algorithms mostly relied on the secrecy of the algorithms themselves. We refer to [KL07, Chapter 1] for more details.

Recovering the shared key. Here, the adversary tries to recover the shared secret key from the knowledge of G, g, $\mathsf{pk}_A = g^a$ and $\mathsf{pk}_B = g^b$ where $\mathsf{sk}_A = a$ and $\mathsf{sk}_B = b$. This problem is known as the Computational Diffie-Hellman (CDH) problem.

Problem 2.1.2 (CDH)**.** *Let G be a cyclic group of prime order q and let g be a generator of G. Given uniformly random elements $g^a, g^b \in G$, compute g^{ab}.*

Distinguishing the shared secret from a random group element. Depending on subsequent use of the shared secret, one may not only require that it should be hard for an adversary to compute the shared key, but also, adversaries should not be able to distinguish the shared secret from a uniformly random group element. This problem is known as the Decisional Diffie-Hellman (DDH) problem.

Problem 2.1.3 (DDH)**.** *Let G be a cyclic group of prime order q and let g be a generator of G. Let g^a, $g^b \in G$ be two uniformly random elements. Given an element $z \in G$, determine whether $z = g^{ab}$ or not with probability non negligibly greater that $1/2$.*

Clearly, if the DDH problem is hard in G, then the CDH problem is hard in G and, if the CDH problem is hard in G, then the DLP is hard in G. There are several classical algorithms for solving the discrete logarithm problem in a generic group. The Baby-step-giant-step algorithm which runs in time $O(\sqrt{q})$ and uses $O(\sqrt{q})$ memory, the Pollard Rho algorithm which runs in time $O(\sqrt{q})$ and uses constant memory, and the index calculus algorithm which runs in sub-exponential time $O(2^{\sqrt{\log q \log\log q}})$ to solve to discrete log problem in $\mathbb{Z}/q\mathbb{Z}$ (q prime). When the order q of G is smooth[3], the Pohlig-Hellman algorithm [PH78] can be used to compute discrete logarithms in G is polynomial time. We refer to [KL07] and [Gal12] for more details about the classical discrete logarithm computation algorithms.

2.1.2 – Public Key Encryption scheme. Rather than first establishing a shared secret and then using this shared secret as private key in a symmetric encryption scheme, Alice can directly encrypt messages to Bob using Bob's public key through a Public Key Encryption scheme. Bob then uses his secret key to decrypt the ciphertext and recover the plaintext message. More formally, a public key encryption scheme is a description of four probabilistic polynomial time algorithms Setup, Key Generation, Encryption and Decryption such that

- Setup takes the security parameter λ as input and returns a set pp of public parameters;
- Key Generation takes the set of public parameters as input and returns the a pair $(\mathsf{sk}, \mathsf{pk})$ where sk is the secret key and pk is the pubic key;
- Encryption takes a public key pk and a plaintext m, and returns a ciphertext $\mathsf{c} = \mathsf{Encryption}_{\mathsf{pk}}(\mathsf{m})$;
- Decryption takes a ciphertext c and the secret key sk corresponding to the public key pk used during Encryption, and returns a plaintext $\mathsf{m}' = \mathsf{Decryption}_{\mathsf{sk}}(c)$.

[3] An integer n is said to be smooth if its primes factors are small. More precisely, n is said to be B-smooth for a given bound B if all the primes divisors of n are smaller than B.

We say that the public key encryption scheme is *correct* if for every key pair $(\mathsf{sk}, \mathsf{pk})$ and for every plaintext m in the message space $\mathcal{M}$,

$$\mathsf{Decryption}_{\mathsf{sk}}(\mathsf{Encryption}_{\mathsf{pk}}(\mathsf{m})) = \mathsf{m}.$$

The correctness of the public key encryption scheme assures that valid ciphertexts always decrypt to the original plaintext message. There are many security requirements one may want a public key encryption scheme to fulfill, but we will briefly describe only the ones relevant for our thesis: OW-CPA security, IND-CPA security and IND-CCA security. Before we get into the security requirements of a public key encryption scheme, let us first discuss the notion of *negligibility*.

Definition 2.1.4. *A function $f : \mathbb{N} \to \mathbb{R}_+$, $\lambda \mapsto f(\lambda)$ is* negligible *(in λ) if for any polynomial $p \in \mathbb{Z}[X]$, there exists $\lambda_0 \in \mathbb{N}$ such that for all $\lambda \geq \lambda_0$ we have $f(\lambda) < \frac{1}{p(\lambda)}$.*

Definition 2.1.5.

We use the notation $\mathsf{negl}(\lambda)$ to designate the image of a security parameter λ through a negligible function. Note that from Definition 2.1.4, a function f is non negligible (in λ) if there exist a polynomial $p \in \mathbb{Z}[X]$ and $\lambda_0 \in \mathbb{N}$ such that for all $\lambda \geq \lambda_0$ we have $f(\lambda) \geq \frac{1}{p(\lambda)}$.

One Wayness under Chosen Plaintext Attacks (OW-CPA). The very first requirement that every public key encryption scheme should fulfill is one-wayness: no adversary not having the decryption key (the secret key of the recipient) should be able to decrypt the ciphertext. In fact, if there was an efficient algorithm that recovers the plaintext message without requiring the decryption key then the scheme would not offer secrecy. This security requirement is formalized as follows.

Definition 2.1.6 (OW-CPA secure)**.** *A public key encryption scheme $\mathcal{P}_\lambda$ having security parameter λ is* OW-CPA secure *if for every probabilistic polynomial time adversary $\mathcal{A}$,*

$$Pr\left[\mathsf{m} = \mathsf{m}' \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Key\ Generation}(\lambda), \mathsf{m} \xleftarrow{\$} \mathcal{M}, \\ \mathsf{c} \leftarrow \mathsf{Encryption}_{\mathsf{pk}}(\mathsf{m}), \mathsf{m}' \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{c}) \end{array}\right] < \mathsf{negl}(\lambda),$$

where $\mathsf{m} \xleftarrow{\$} \mathcal{M}$ means uniformly sampling m from $\mathcal{M}$.

Indistinguishability under Chosen Plaintext Attacks (IND-CPA). Now let us suppose that a referendum is organised at an institution and each employee has to encrypt his ballot to the director who then decrypts the ballots and counts the votes. Here, an adversary who wants to attack the scheme, say a colleague who wants to learn your vote, does not need to decrypt the encrypted ballot, but to distinguish if it is the encryption of a "Yes" ballot or that of a "No" ballot. In this context, if encryptions of "Yes" ballots are distinguishable from those of "No" ballots, then the scheme is not secure for this purpose, since an adversary will be able to learn the vote of each participant looking only at his encrypted ballot. So we want them to be indistinguishable. This security requirement is known as Indistinguishability under Chosen Plaintext Attacks: for any plaintext pair $(\mathsf{m}_0, \mathsf{m}_1)$, ciphertexts of m_0 should not be distinguishable from those of m_1. This is formalized as follows.

Definition 2.1.7 (IND-CPA secure)**.** *A public key encryption scheme $\mathcal{P}_\lambda$ having security parameter λ is* IND-CPA secure *if for every probabilistic polynomial time adversary $\mathcal{A}$,*

$$Pr\left[b = b^* \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Key\ Generation}(\lambda), \mathsf{m}_0, \mathsf{m}_1 \leftarrow \mathcal{A}(\mathsf{pk}, \mathcal{M}), \\ b \xleftarrow{\$} \{0,1\}, \mathsf{c} \leftarrow \mathsf{Encryption}_{\mathsf{pk}}(\mathsf{m}_b), b^* \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{c}) \end{array}\right] = \frac{1}{2} + \mathsf{negl}(\lambda).$$

Indistinguishability under Chosen Ciphertext Attacks (IND-CCA). Still considering the vote described above, suppose now that the colleague willing to learn your vote is the vice-director and there are some ciphertexts that the director received in the company and had to discuss the corresponding plaintexts with the vice-director. Moreover, after the vote, the director will continue to receive these ciphertexts and share their corresponding plaintexts with the vice-director. Nevertheless, the director is not authorised to share the plain (decrypted) ballots with the vice-director. For the votes to be secret, the vice-director should still not be able to distinguish if an encrypted ballot is the encryption of a "Yes" ballot or that of a "No" ballot. In the cryptographic context, this translates to the scenario where the adversary has access to a decryption oracle which he can query with any ciphertext different from the one he wants to decrypt, then the decryption oracle returns the corresponding plaintext. We want the ciphertexts to remain indistinguishable when the adversary is provided this decryption oracle. This is formalized as follows.

Definition 2.1.8 (IND-CCA secure)**.** *A public key encryption scheme $\mathcal{P}_\lambda$ having security parameter λ is* IND-CCA secure *if for every probabilistic polynomial time adversary $\mathcal{A}$,*

$$Pr\left[b = b^* \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Key\ Generation}(\lambda), \mathsf{m}_0, \mathsf{m}_1 \leftarrow \mathcal{A}^{O(\cdot)}(\mathsf{pk}, \mathcal{M}), \\ b \xleftarrow{\$} \{0,1\}, \mathsf{c} \leftarrow \mathsf{Encryption}_{\mathsf{pk}}(\mathsf{m}_b), b^* \leftarrow \mathcal{A}^{O(\cdot)}(\mathsf{pk}, \mathsf{c}) \end{array}\right] = \frac{1}{2} + \mathsf{negl}(\lambda),$$

where $O(\cdot)$ is a decryption oracle that when given a ciphertext $\mathsf{c}' \neq \mathsf{c}$, outputs $\mathsf{Decryption}_{\mathsf{sk}}(\mathsf{c}')$ or $\perp$ if the ciphertext c' is invalid.

One can easily verify that every IND-CCA secure public key encryption scheme is IND-CPA secure and every IND-CPA secure public key encryption scheme is OW-CPA secure. There are several generic transforms [FO99; Ham12; BR94] that help obtain an IND-CCA secure public key encryption scheme from an IND-CPA or OW-CPA secure public key encryption scheme.

We now describe two famous public key encryption schemes: RSA and El Gamal.

The RSA cryptosystem. The very first public key encryption scheme is the RSA [RSA78] cryptosystem of Rivest, Shamir and Adleman published two years after the Diffie-Hellman key exchange paper, in which the ideas of public key encryption and digital signature were suggested but without any concrete construction. The RSA cryptosystem construction uses high school modular arithmetic and its security relies on the hardness of factoring large integers. The RSA cryptosystem is as follows.

Setup: no setup needed.

Key Generation: Choose two random primes p and q of the same size, set $N = p * q$. Choose an integer $e > 1$ coprime to $\varphi(N)$ where φ is the Euler function. Compute d such that $d * e \equiv 1 \mod \varphi(N)$. The secret key is $\mathsf{sk} = (N, e)$ and the public key is $\mathsf{pk} = (N, d)$.Encryption: Let $\mathsf{m} \in \mathbb{Z}_N^\times$ (the multiplicative group of the ring $\mathbb{Z}_N$, that is the set of integers between 1 and $N-1$ that are coprime to N) be a plaintext. Given a public key (N, e), compute the ciphertext $\mathsf{c} = \mathsf{m}^e \mod N$.

Decryption: Given the secret key (N, d) and a ciphertext c, compute $\mathsf{m}' = \mathsf{c}^d \mod N$.

The correctness of the RSA cryptosystem follows from Euler's Theorem

$$x^{\varphi(N)} \equiv 1 \mod N \quad \text{for } x \in \mathbb{Z}_N^\times.$$

Recall that $d * e \equiv 1 \mod \varphi(N)$, hence $d * e = 1 + k\varphi(N)$ for some integer k. Let $\mathsf{m} \in \mathbb{Z}_N^\times$ be a plaintext and let $\mathsf{c} = \mathsf{m}^e \mod N$ be the corresponding ciphertext. Then

$$\begin{aligned} \mathsf{c}^d &\equiv \mathsf{m}^{d*e} \mod N \\ &\equiv \mathsf{m}^{1+k\varphi(N)} \mod N \\ &\equiv \mathsf{m} * (\mathsf{m}^{\varphi(N)})^k \mod N \\ &\equiv \mathsf{m} \mod N. \end{aligned}$$

Factoring the RSA modulus N breaks the RSA encryption. The RSA factorisation problem is as follows.

Problem 2.1.9 (RSA factorisation problem)**.** *Let $N = pq$ be the product of two generic cryptographic size primes p and q such that $p \approx q$. Given N, determine p and q.*

Note that RSA encryption scheme is deterministic. Therefore, one can easily distinguish if a given ciphertext c is that of m_0 or m_1: one simply encrypts m_0 and m_1 and compares the obtained ciphertext to c. Hence RSA is not IND-CPA secure. One uses the OAEP transform [BR94] to derive the OAEP-RSA which is IND-CCA secure.

The El Gamal cryptosystem.. In 1985, Taher El Gamal [ElG85] derived a public key encryption scheme from the Diffie-Hellman key exchange protocol. This scheme today bears his name: El Gamal encryption. The El Gamal encryption is designed as follows.

Setup: let G be a cyclic (multiplicative) group of prime order q and let g be a generator of G. The public parameters are G, g and q.

Key Generation: choose a uniformly random integer $a \in \{0, 1, \cdots, q-1\}$ and compute g^a. The secret key is $\mathsf{sk} = a$ and the public key is $\mathsf{pk} = g^a$.

Encryption: Given a plaintext $\mathsf{m} \in G$, a public key pk and the public parameters G, g and q, generate a uniformly random integer $b \in \{0, 1, \cdots, q-1\}$ and compute $\mathsf{c}_1 = g^b$, $\mathsf{c}_2 = m * \mathsf{pk}^b$. The ciphertext is $\mathsf{c} = (\mathsf{c}_1, \mathsf{c}_2)$.

Decryption: Given the secret key a and a ciphertext $\mathsf{c} = (\mathsf{c}_1, \mathsf{c}_2)$, compute $\mathsf{m}' = \mathsf{c}_2 / \mathsf{c}_1^a$.

The correctness of the El Gamal public key encryption scheme follows from that of the Diffie-Hellman key exchange. The IND-CPA security of the El Gamal encryption relies on the DDH problem in the group G.

Theorem 2.1.10 ([ElG85; KL07]). *If the DDH problem in G is hard, then the El Gamal public key encryption scheme is IND-CPA secure.*

The El Gamal cryptosystem suggests a generic construction of IND-CPA secure public key encryption schemes: design a Diffie-Hellman type key exchange for which the DDH type assumption holds, then derive an El Gamal type public key encryption scheme. This type of construction is widely used in practice. For instance, the Diffie-Hellman key exchange and the El Gamal encryption scheme were later instantiated with the group of points of elliptic curves (Miller 1985 [Mil85] and Koblitz 1987 [Kob87]) to obtain the Elliptic Curve Diffie-Hellman (ECDH) key exchange and the Elliptic Curve Encryption Scheme.

Note that the El Gamal encryption is not IND-CCA secure, in fact, ciphertexts are malleable: given a ciphetext c for some plaintext m, we can efficiently derive a ciphertext $\mathsf{c}' \neq \mathsf{c}$ of some message m' related to m. When ciphertexts are malleable, the IND-CCA attack, when given a ciphertext c to decrypt, consists in deriving a new ciphertext $\mathsf{c}' \neq \mathsf{c}$ such that the plaintext m' corresponding to c' is related to the plaintext m correponding to c. One then queries the decryption oracle with c' to recover m' and use the relation between m and m' to recover m. Sometimes, one may not be able to totally recover m, but as far as the relation between m and m' permits to distinguish m from some other ciphertext m^*, it suffices to break the IND-CCA security requirement. In the case of the El Gamal encryption, the ciphertexts are malleable in the following way:

if $\mathsf{c} = (\mathsf{c}_1, \mathsf{c}_2)$ is a ciphertext for m, then for every $\alpha \in G$, $\mathsf{c}' = (\mathsf{c}_1, \alpha\mathsf{c}_2)$ is a ciphertext for $\alpha\mathsf{m}$.

One uses hash functions and Message Authentication Codes (MAC) to construct the Diffie-Hellman Integrated Encryption Scheme (DHIES) [BR97] which is IND-CCA secure.

2.2 — Elliptic curves

Elliptic curves play an important role in cryptography: they offer better[4] groups (compared to finite fields and $\mathbb{Z}/q\mathbb{Z}$) for the Diffie-Hellman key exchange, for the El Gamal encryption and El Gamal signature schemes. This contributed to Elliptic Curve Cryptography being widely deployed from 2005. Moreover, elliptic curves are the objects on which isogenies, our main interest in this thesis, are built. This section surveys elliptic curves. We refer to the books of Siverman [Sil09; Sil94] and the book of Washington [Was08] for more material about elliptic curves. Also, Panny's thesis [Pan21] provides a nice introduction to isogeny based cryptography. The Bristol isogeny-based cryptography school [MP21] provides background on nearly all the aspects of isogeny-based cryptography.

Definition 2.2.1. *Let K be a field. An elliptic curve over K is a pair (E, O) where E is a smooth projective curve over K of genus one and O is a k-rational point on E, the base point.*

[4] more secure and more efficient

The base point O is usually omitted. When the field of definition of the considered curve is not implicitly defined, we explicitly write E/K to precise that the field of definition of the curve E is K.

When the characteristic of the field K is not 2 or 3, every elliptic curve defined over K is isomorphic to a short Weierstrass curve.

Proposition 2.2.2. *Let K be a field whose characteristic is not 2 or 3. Every elliptic curve (E, O) defined over K is isomorphic over K to a short projective Weierstrass curve defined by an equation*

$$E : Y^2Z = X^3 + aXZ^2 + bZ^3 \tag{2.1}$$

with $a, b \in K$ such that the discriminant $\Delta = -16(4a^3 + 27b^2)$ is non-zero.

The unique projective point [0:1:0] of E (2.1) having $Z = 0$ is called the point at infinity and is denoted by ∞ (∞ is in fact the image of O through the above mentioned isomorphism). The affine part of E is hence defined by the short affine Weierstrass equation

$$E : y^2 = x^3 + ax + b. \tag{2.2}$$

From now on, all our curves are defined over fields of characteristic not 2 and 3, and are defined by a short affine Weierstrass equation. Always keep in mind that these are in fact projective curves and that there is an implicit point at infinity (somewhere up there at infinity).

The most relevant invariant of elliptic curves in isogeny-based cryptography is the j-invariant, it parametrises the isomorphism classes of elliptic curves over the algebraic closure $\overline{K}$ of K.

Definition 2.2.3. *Let $E/K : y^2 = x^3 + ax + b$ be an elliptic curve. The j-invariant of E, denoted by $j(E)$, is the field element*

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

Proposition 2.2.4. *Two elliptic curves $E/K : y^2 = x^3 + ax + b$ and $E'/K : y^2 = x^3 + a'x + b'$ are isomorphic over $\overline{K}$ if and only they have the same j-invariant.*

Even more, for a given j-invariant $j \in K$, when $char(K) \neq 2, 3$, the elliptic curve $E(j)$ defined by

$$E(j) : y^2 = x^3 - 3j(j - 11728)x - 2j(j - 1728)^2$$

has j-invariant j.

Even though the curve E is defined over K, the points of E, which are solutions of the short affine Weierstrass equation (Equation 2.2), have their coordinates in the algebraic closure $\overline{K}$ of K. Concretely, the set of points of E is

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

For every field extension $K \subset L \subset \overline{K}$, we say a point (x, y) of E is *L-rational* when $x, y \in L$. The set of L-rational points of E is

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

The set of points of every elliptic curve E has a unique (additive) abelian group structure for which the point at infinity is the neutral element. More interestingly, this abelian group structure is geometrically defined using the following rule :

"Three points on E sum to ∞ if and only if there exists a line that intersects E exactly at these points counted with their respective multiplicities".

This rule is highlighted in Figure 2.1. The point at infinity is on every vertical line.

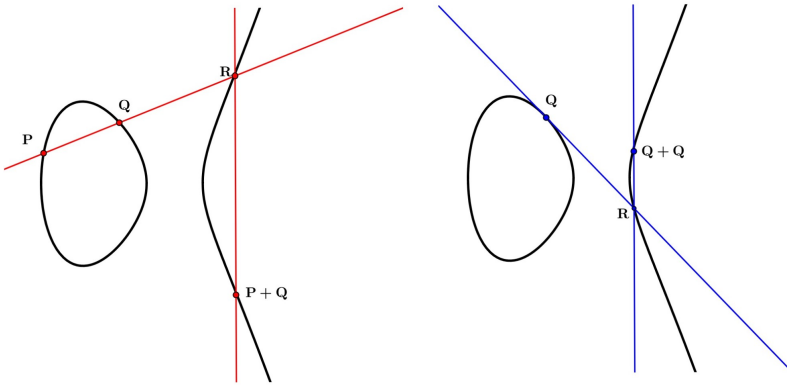


Figure 2.1: Group law of elliptic curves in the Weierstrass model.

For every field extension $K \subset L \subset \overline{K}$, the set $E(L)$ of L-rational points of E is a subgroup of E. Naturally, this additive group law comes with a scalar multiplication of points by integers.

Definition 2.2.5. *For any integer $n \in \mathbb{Z}\backslash\{0\}$, let $[n] : E \to E$, $P \mapsto [n]P$ be the scalar multiplication by n on E, defined by adding together n copies of P if $0 < n$ or $-n$ copies of $-P$ if $n < 0$. The scalar multiplication by n is a group endomorphism of E and its kernel, denoted by $E[n]$, is called the n-torsion subgroup of E.*

The group structure of $E[n]$ is given by the following proposition.

Proposition 2.2.6. *Let E/K be an elliptic curve and n a non-zero integer.*

- *If $char(K) = 0$, then $E[n] \cong \mathbb{Z}/n\mathbb{Z} \oplus \mathbb{Z}/n\mathbb{Z}$.*
- *If $char(K) = p > 0$, write $n = m \cdot p^r$ where m and p are coprime. Then*

$$E[n] \cong \mathbb{Z}/m\mathbb{Z} \oplus \mathbb{Z}/n\mathbb{Z} \quad \text{or} \quad E[n] \cong \mathbb{Z}/m\mathbb{Z} \oplus \mathbb{Z}/m\mathbb{Z}.$$

In particular, either $E[p] \cong \mathbb{Z}/p\mathbb{Z}$ or $E[p] \cong \{0\}$, and if $p \nmid n$ then $E[n] \simeq \mathbb{Z}/n\mathbb{Z} \oplus \mathbb{Z}/n\mathbb{Z}$.

From Proposition 2.2.6, it follows that when the characteristic p of the field K is non-zero, we can regroup elliptic curves defined over K into two sets: those for which the $p-$torsion is the trivial group and those for which it is isomorphic to $\mathbb{Z}/p\mathbb{Z}$.

Definition 2.2.7. *Le K be a field of characteristic $p > 0$ and let E be an elliptic curve defined over K. If $E[p] \cong \mathbb{Z}/p\mathbb{Z}$, we say that E is ordinary. If not, then $E[p] \cong \{0\}$ and we say that E is supersingular.*

There are many fundamental differences between ordinary curves and supersingular curves. They range from the number of rational points to the endomorphism ring structure passing through the field of definition of the curves and that of the rational maps between them.

The number of points a an elliptic curve defined over a finite field $\mathbb{F}_q$ is bounded by the Hasse bound.

Theorem 2.2.8 (Hasse Theorem)**.** *Let $E/\mathbb{F}_q$ be an elliptic curve, then*

$$\#E(\mathbb{F}_q) = q + 1 - t \quad \text{with} \quad |t| \leq 2\sqrt{q}.$$

Given $E/\mathbb{F}_q$ the integer t, called the *trace of the curve*, can be computed in polynomial time using Schoof's Algorithm [Sch85]. Supersingular curves can be distinguished from ordinary ones by their trace.

Theorem 2.2.9 ([Was08, Proposition 4.31])**.** *Let $E/\mathbb{F}_q$ $(q = p^n)$ be an elliptic curve such that $\#E(\mathbb{F}_q) = q + 1 - t$. Then E is supersingular if and only $t \equiv 0 \mod p$, which is if and only if $\#E(\mathbb{F}_q) \equiv 1 \mod p$.*

Corollary 2.2.10. *Let $E/\mathbb{F}_p$ where $p > 3$ is a prime be an elliptic curve. Then E is supersingular if and only $t \equiv 0 \mod p$, which is if and only if $\#E(\mathbb{F}_p) = p + 1$.*

Note that supersingular curves are defined only in fields of positive characteristic. Moreover, there exists a finite number of supersingular curves given a fixed characteristic p, and these curves are all isomomorphic to curves defined over $\mathbb{F}_{p^2}$. Concretely, we have the following theorem.

Theorem 2.2.11. *Let E be a supersingular curve defined over a field of characteristic $p > 0$. Then $j(E) \in \mathbb{F}_{p^2}$. In particular, E is isomorphic to a curve defined over $\mathbb{F}_{p^2}$.*

In characteristic p, there are exactly $\lfloor \frac{p}{12} \rfloor + \epsilon$ supersingular j-invariants (or isomorphism classes of supersingular curves), where $\epsilon \in \{0, 1, 2\}$. Moreover, for all but at most six supersingular elliptic curves E defined over $\mathbb{F}_{p^2}$, we have

$$\#E(\mathbb{F}_{p^2}) = (p + \delta)^2 \quad \text{and} \quad E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/(p + \delta)\mathbb{Z}) \oplus (\mathbb{Z}/(p + \delta)\mathbb{Z})$$

where $\delta = \pm 1$.

Beside the Weierstrass model, there are several other models of elliptic curves. The most used ones are the Montgomery model and the Edwards model. Each model has its advantages depending on the use. Montgomery curves have nice and efficient x-coordinate only addition formulas, and come with the famous *Montgomery ladder* [Mon87] for scalar multiplication. Edwards curves have similar advantages over Weierstrass curve, moreover, their addition formulas are complete[5]. We refer to [BL11] for further details about Edwards curves and their addition formulas. In this thesis, we only use Montgomery curves.

[5]The same formula is used for point addition and point doubling.

Definition 2.2.12. *Let K be a field of characteristic not 2. A Montgomery curve over K is a projective curve defined by the affine equation*

$$By^2 = x^3 + Ax^2 + x$$

where $A, B \in K$ and $B(A^2 - 4) \neq 0$. When $B = 1$ (as it will be the case in this thesis), A is called the Montgomery coefficient of the curve. The $j-$invariant of the Montgomery curve $E : By^2 = x^3 + Ax^2 + x$ is given by

$$j(E) = \frac{256(A^2 - 3)^3}{A^2 - 4}.$$

As we will see in Section 2.3, they offer particularly efficient formulas for isogenies of degree 2 and 3.

2.3 — Isogenies

Now we describe the central object used in this thesis: *isogenies of elliptic curves.*

Definition 2.3.1. *An* isogeny *between two elliptic curves E/K and E'/K is a non constant rational map*

$$\phi : E \to E'$$

which is also a group homomorphism. An isogeny is defined over K if it can be written using fractions of polynomials with coefficients in K. Two curves E and E' are said to be isogenous when there exists an isogeny $\phi : E \to E'$.

Let $Hom_K(E, E')$ denote the set of all isogenies $E \to E'$ defined over K, together with the constant morphism $0 : E \to E', P \mapsto \infty$. For isogenies defined over $\overline{K}$, we write $Hom(E, E') = Hom_{\overline{K}}(E, E')$. $Hom(E, E')$ has an abelian group structure inherited from the group structure of E': $(\phi + \psi)(P) = \phi(P) + \psi(P)$.

The scalar multiplications are examples of isogenies. When the curve E is defined over a finite field $\mathbb{F}_q$ ($q = p^n$), the *Frobenius endomorphism* of E, denoted π and given by

$$\pi : E \to E, \quad (x, y) \mapsto (x^q, y^q),$$

is an isogeny. Also, the p^{th}*-Frobenius* defined by

$$\pi_p : E \to E^p, \quad (x, y) \mapsto (x^p, y^p)$$

is an isogeny from $E : y^2 = x^3 + ax + b$ to $E^p : y^2 = x^3 + a^p x + b^p$. Note that π_p is an endomorphism if and only if E is defined over $\mathbb{F}_p$.

Proposition 2.3.2. *Let $\phi : E \to E'$ be an isogeny defined over K. Then there exist two rational functions r_1, r_2 such that*

$$\phi(x, y) = (r_1(x), y \cdot r_2(x)).$$

Write $r_1(x) = f(x)/g(x)$, then the integer

$$\deg(\phi) = \max\{\deg(f), \deg(g)\}$$

is called the degree *of ϕ.*

The rational functions r_1 and r_2 have the same poles and these poles are the x-coordinates of the kernel points of ϕ. We have

$$\ker \phi = \{(x, y) \in E \mid g(x) = 0\} \cup \{\infty\}.$$

As a consequence, isogenies have finite kernels.

Definition 2.3.3. *Let $\phi : E \to E'$ be an isogeny. If*

$$\deg(\phi) = \#\ker(\phi),$$

we say that ϕ is separable. *If not, then E and E' are defined over a finite field $\mathbb{F}_q$, $\phi = \phi_1 \circ \pi_p^r$ where ϕ_1 is separable and $\deg(\phi) = p^r \deg(\phi_1)$. In the latter case, we say that ϕ is* inseparable. *When $\phi = \pi_p^r$ for some integer r, we say that ϕ is* purely inseparable.

Being isogenous is an equivalence relation and the degree is multiplicative: $\deg(\phi \circ \psi) = \deg(\phi)\deg(\psi)$.

Proposition 2.3.4. *Let $\phi : E \to E'$ be an isogeny. Then there exists an isogeny $\widehat{\phi} : E' \to E$ such that $\phi \circ \widehat{\phi} = [\deg(\phi)]_{E'}$ and $\widehat{\phi} \circ \phi = [\deg(\phi)]_E$. The isogeny $\widehat{\phi}$ is called the* dual *of ϕ. Moreover, we have the following rules:*

- $\widehat{\widehat{\phi}} = \phi$;
- $\widehat{\phi + \psi} = \widehat{\phi} + \widehat{\psi}$ *for* $\phi \neq -\psi$;
- $\widehat{\phi \circ \psi} = \widehat{\psi} \circ \widehat{\phi}$;
- $\ker \widehat{\phi} = \phi(E[\deg(\phi)])$.

Determining whether two given curves E and E' are isogenous over K is quite easy and efficient: one computes $\#E(K)$ and $\#E'(K)$ using Schoof's algorithm, then by Tate's Theorem (Theorem 2.3.5) one checks whether $\#E(K) = \#E'(K)$. Meanwhile, computing an isogeny between isogenous curves is not easy at all, it is believed that even quantum computers can't help here. Nevertheless, when the kernel of a separable isogeny is provided, then this kernel defines the separable isogeny up to isomorphism. If the size of the kernel is smooth, then the isogeny can be effectively computed in polynomial time using Vélu's formula [Vél71].

Theorem 2.3.5 (Tate)**.** *Let K be a finite field. Two curves E/K and E'/K are isogenous over K if and only if $\#E(K) = \#E'(K)$.*

Proposition 2.3.6. *Let E be an elliptic curve defined over K and let G be a finite subgroup of E defined*[6] *over K. Then there exist an elliptic curve E' and a separable isogeny $\phi_G : E \to E'$, both defined over K, such that $\ker(\phi_G) = G$. The pair (E', ϕ_G) is unique up K-isomorphism of E'. The curve E' is denoted by E/G.*

[6] G is defined over K means that for every automorphism σ of $\overline{K}$ fixing K ($\sigma(x) = x$ for all $x \in K$), we have $\sigma(G) = G$.

Theorem 2.3.7 (Vélu [Vél71]). *Let $E : y^2 = x^3 + ax + b$ be a Weierstrass curve over a field K and let G be a finite subgroup of E. For any point $P \in E$, define*

$$f_x(P) = x(P) + \sum_{Q \in G \setminus \{\infty\}} (x(P+Q) - x(Q))$$

and

$$f_y(P) = y(P) + \sum_{Q \in G \setminus \{\infty\}} (y(P+Q) - y(Q)).$$

Then the map

$$\phi : E \to E/G \quad P \mapsto (f_x(P), f_y(P)),$$

where poles of f_x and f_y get mapped to the point at infinity, is a separable isogeny with kernel G. The codomain is a Weierstrass curve whose equation can be efficiently recovered.

Vélu's formulas can be translated to any other curve model. For example, let $E : By^2 = x^3 + Ax^2 + x$ be a Montgomery curve, let $P_2 = (x_2, 0) \in E$ and $P_3 = (x_3, y_3) \in E$ be points of order 2 and 3 respectively, with $x_2 \neq 0$. Then the groups $G_2 = \langle P_2 \rangle$ and $G_3 = \langle P_3 \rangle$ define respectively two separable degree 2 and degree 3 isogenies $\phi_2 : E \to E_2 = E/G_2$ and $\phi_3 : E \to E_3 = E/G_3$ where:

$$\phi_2(x, y) = \left(x\frac{xx_2 - 1}{x - x_2}, y\frac{x^2x_2 - 2xx_2^2 + x_2}{(x - x_2)^2} \right),$$

$$E_2 : (Bx_2)y^2 = x^3 + \left(2(1 - x_2^2)\right) x^2 + x$$

and

$$\phi_3(x, y) = \left(x\frac{(xx_3 - 1)^2}{(x - x_3)^2}, y\frac{(xx_3 - 1)(x^2x_3 - 3xx_3^2 + x + x_3)}{(x - x_3)^3} \right)$$

$$E_2 : (Bx_3^2)y^2 = x^3 + \left(x_3(Ax_3 - 6x_3^2 + 6)\right) x^2 + x.$$

Vélu formulas have complexity $O(\deg(\phi))$. Recently, Bernstein et al. [BDLS20] designed an algorithm, denoted $\sqrt{\text{élu}}$ (square root Vélu), whose asymptotic complexity is $O(\sqrt{\deg(\phi)})$. Note that for small primes $\ell < 110$, the Vélu formulas outperform the $\sqrt{\text{élu}}$ algorithm in terms of efficiency [BDLS20, Appendix A.3]. Hence when designing cryptographic isogeny-based protocols, the Vélu formulas are used for those small primes while the $\sqrt{\text{élu}}$ algorithm is used for larger primes. Both are not efficient for computing separable isogenies of generic large degree. Nevertheless, when the degree is smooth, the isogeny can be computed as a composition of isogenies of small degree.

Proposition 2.3.8. *Let E be an elliptic curve and let G be a finite subgroup of E. For any subgroup G' of G, the isogeny $\phi : E \to E/G$ can be decomposed as*

$$\phi : E \xrightarrow{\varphi_1} E/G' \xrightarrow{\varphi_2} E/G$$

where $\ker(\varphi_1) = G'$ and $\ker(\varphi_2) = \varphi_1(G)$.

Proposition 2.3.8 suggests that to compute an isogeny ϕ of degree $\prod_{i=1}^{r} \ell_i^{e_i}$, we can decompose ϕ as

$$\phi = \varphi_{11} \circ \cdots \circ \varphi_{1e_1} \circ \cdots \circ \varphi_{r1} \circ \cdots \circ \varphi_{re_r}$$

where φ_{ij} has degree ℓ_i for $1 \leq i \leq r$ and $1 \leq j \leq e_i$. Hence the cost of computing ϕ boils down to that of computing e_i isogenies of degree ℓ_i for $1 \leq i \leq r$. This enables us to efficiently compute isogenies of smooth degree with given kernel.

2.4 — Endomorphism rings and isogeny graphs

Definition 2.4.1. *The endomorphism ring of an elliptic curve E, denoted by* $\mathrm{End}(E)$, *is the ring* $\mathrm{End}(E) = Hom(E, E)$, *whose elements are isogenies from E to E, to which one adds the constant map* $0 : E \to E, P \to \infty$.

There are only three possibilities for the structure of the endomorphism ring of an elliptic curve.

Proposition 2.4.2 ([Was08, Theorem 3.2])**.** *Let E be an elliptic curve defined over a field K.*

- *If $char(K) = 0$, then either* $\mathrm{End}(E) = \mathbb{Z}$ *or* $\mathrm{End}(E)$ *is isomorphic to an order in a quadratic imaginary field.*
- *If $char(K) = p$ and $K = \mathbb{F}_q$, then* $\mathrm{End}(E)$ *is isomorphic either to an order in a quadratic imaginary field*[7] *or to a maximal order in the quaternion algebra*[8] *ramified at p and at infinity.*

In positive characteristic, the structure of the endomorphism ring is directly related to the type (ordinary or supersingular) of the curve.

Proposition 2.4.3. *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. Then the endomorphism ring of E is an order in a quadratic imaginary field if and only if E is an ordinary curve.*

Proposition 2.4.4. *Let $\theta \in \mathrm{End}(E)$ be an endomorphism of an elliptic curve E and let $\overline{\theta}$ be its conjugate in the corresponding quadratic field or quaternion algebra. Its norm $N(\theta) = \theta\overline{\theta} \in \mathbb{Z}$ and its trace $tr(\theta) = \theta + \overline{\theta} \in \mathbb{Z}$ are such that*

$$\theta^2 - tr(\theta)\theta + N(\theta) = 0.$$

The degree of the actual isogeny ϕ_θ corresponding to θ is in fact $\deg(\phi_\theta) = N(\theta)$ and its dual $\widehat{\phi_\theta}$ corresponds to the conjugate $\overline{\theta}$ of θ.

From now on, unless when stated otherwise, the elliptic curves considered are defined over a finite field $\mathbb{F}_q$ with $q = p^n$.

Let $E/\mathbb{F}_q$ be an elliptic curve. Then the Frobenius endomorphism π of E is an endomorphism of E and $\mathbb{Z}[\pi] \subset \mathrm{End}(E)$. Moreover, $N(\pi) = q$ and $tr(\pi) = t$ where $\#E(\mathbb{F}_q) = q + 1 - t$, and the characteristic equation of π is $X^2 - tX + q = 0$. When $\pi \notin \mathbb{Z}$, we get that $\mathbb{Z}[\pi]$ is isomorphic to $\mathbb{Z}[\sqrt{t^2 - 4q}]$, which is an order in the quadratic

[7] We refer to the book of David Cox [Cox14] for background on quadratic fields.
[8] We refer to the book of John Voight [Voi18] for background on quaternion algebras.

imaginary field $\mathbb{Q}(\sqrt{t^2-4q})$. In this case, the knowledge of t (equivalently $\#E(\mathbb{F}_q)$) gives rise to a non trivial quadratic suborder of $End(E)$. When $\pi \in \mathbb{Z}$ (as it will be the case for supersingular curves over $\mathbb{F}_{p^2}$), the knowledge of t does not come with any further information than the cardinality of the curve. This constitutes a huge difference between ordinary curves (where $\pi \notin \mathbb{Z}$) and supersingular curves over[9] $\mathbb{F}_{p^2}$ (where $\pi \in \mathbb{Z}$). This difference is visible at the level of the ordinary isogeny graph and the supersingular isogeny graph.

We recall that the curves are defined over finite fields $\mathbb{F}_q$ of characteristic $p > 0$ and that only separable isogenies are considered.

Definition 2.4.5. *Let $\ell \neq p$ be a (small) prime. The ℓ-isogeny graph $G_\ell(\mathbb{F}_q)$ is the undirected graph with vertex set $\mathbb{F}_q$ (seen as the set of j-invariants) and edges (j_1, j_2) correspond to $\ell-$isogenies (up to isomorphism) defined over $\mathbb{F}_q$ between the curves $E(j_1)$ and $E(j_2)$.*

From the Tate Theorem (Theorem 2.3.5), we know that two isogenous curves have the same trace of Frobenius. By Theorem 2.2.9, we see that ordinary traces are distinct from supersingular traces. Hence an ordinary curve cannot be isogenous to a supersingular curve. This implies that the isogeny graph has ordinary components and supersingular components.

Ordinary graphs. Recall that ordinary curves have complex multiplication. Even though we restricted ourselves to curves defined over finite fields, most of the facts and results presented here in the ordinary case are valid for curve defined over positive characteristic fields. We refer to [Cox14] for prerequisites on orders in quadratic fields.

Definition 2.4.6. *An $\ell-$volcano is a connected undirected graph whose vertices are partitioned into one or more levels $V_0, \cdots, V_d$ such that the following hold:*

1. *The subgraph on V_0 is a regular graph of degree at most 2.*
2. *For $i > 0$, each vertex in V_i has exactly one neighbor in level V_{i-1}.*
3. *For $i < d$, each vertex in V_i has degree $\ell+1$ and exactly ℓ neighbors in level V_{i+1}.*

The levels V_0 and V_d are called the surface (or crater) and the floor of the volcano respectively. The integer d is called the depth of the volcano.

The ordinary subgraph of the $\ell-$isogeny graph, or simply the ordinary $\ell-$isogeny graph has many connected components that have the same ℓ-volcano structure. They are called *Isogeny volcanoes*. Lets us briefly describe theses $\ell-$isogeny volcanoes.

Theorem 2.4.7 ([Sut17, Lecture 23, Thm 23.3])**.** *Let $\phi : E/\mathbb{F}_q \to E'/\mathbb{F}_q$ be an ℓ-isogeny defined over $\mathbb{F}_q$ between two ordinary curves and set $\mathrm{End}(E) = \mathcal{O}$, $\mathrm{End}(E') = \mathcal{O}'$. Then $\mathcal{O}$ and $\mathcal{O}'$ are orders in the same imaginary quadratic field $\mathbb{Q}(\sqrt{t^2-4q})$ (where $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q) = q+1-t$) and one of the following holds:*

(i) $\mathcal{O} = \mathcal{O}'$, we say ϕ is horizontal;

[9] Note that the field of definition of the supersingular curve is important here. In fact, as precised in Chapter 3, when a supersingular curve E is defined over $\mathbb{F}_p$ instead, $\pi = \pi_p \notin \mathbb{Z}$ and $\mathbb{Z}[\pi]$ is non trivial.

(ii) $[\mathcal{O} : \mathcal{O}'] = \ell$, *we say* ϕ *is descending;*
(iii) $[\mathcal{O}' : \mathcal{O}] = \ell$, *we say* ϕ *is ascending.*

In general, the existence of horizontal, descending and ascending $\ell-$isogenies with domain E depends on the fields of definition $\mathbb{F}_q$ of the curve E, $\mathrm{End}(E)$ and ℓ.

Set t coprime to p such that $t \leq 2\sqrt{q}$. For any order $\mathcal{O}$ in $\mathbb{Q}(\sqrt{t^2-q})$, let $\mathrm{Ell}_q(\mathcal{O})$ be the set of isomorphism classes of ordinary curves $E/\mathbb{F}_q$ such that $\mathrm{End}(E) = \mathcal{O}$.

Proposition 2.4.8. *Let* $\mathcal{O}$ *be an order of discriminant* Δ *with* Δ *coprime to* q. *The set* $\mathrm{Ell}_q(\mathcal{O})$ *is either empty or has cardinality*[10] $h(\Delta)$. *If* $\mathrm{Ell}_q(\mathcal{O})$ *is non empty, then* $\mathrm{Ell}_{\mathcal{O}'}(\mathbb{F}_q)$ *is non empty for every imaginary quadratic order* $\mathcal{O}'$ *containing* $\mathcal{O}$.

Let $\mathcal{O}_{\max}$ be the maximal order of $\mathbb{Q}(\sqrt{t^2-q})$ and $\Delta_{\max}$ its discriminant. Fix an order $\mathcal{O}$ in $\mathbb{Q}(\sqrt{t^2-q})$ of discriminant Δ. Then $\mathcal{O} \subset \mathcal{O}_{\max}$, $[\mathcal{O}_{\max} : \mathcal{O}] = f$ such that $\Delta = f^2\Delta_{\max}$ and $\mathcal{O} = \mathbb{Z} + f\mathcal{O}_{\max}$. The integer f is called the *conductor* of $\mathcal{O}$.

Let $E/\mathbb{F}_q$ be an ordinary curve such that $\mathrm{End}(E) = \mathcal{O}$ has conductor f and the trace of Frobenius of E is t. Let f_π be the conductor of $\mathbb{Z}[\pi]$. Then its discriminant Δ_π satisfies $\Delta_\pi = t^2 - 4q = f_\pi^2\Delta_{\max}$. Write $f_\pi = \ell^d f_0$ where ℓ and f_0 are coprime. Set $\mathcal{O}_0 = \mathbb{Z} + f_0\mathcal{O}_{\max}$, then $[\mathcal{O}_0 : \mathbb{Z}[\pi]] = \ell^d$ and $\mathcal{O}_0$ has conductor f_0. We have $f = \ell^e f_0$ for some $0 \leq e \leq d$, and $[\mathcal{O}_0 : \mathcal{O}] = \ell^e$, $[\mathcal{O} : \mathbb{Z}[\pi]] = \ell^{d-e}$. Then the component of the ordinary supersingular $\ell-$isogeny graph containing E is structured as follows.

- If $\mathcal{O} = \mathcal{O}_0$, equivalently $e = 0$, the curve E admits $1 + \left(\frac{\Delta_0}{\ell}\right)$ (where $\left(\frac{\cdot}{\cdot}\right)$ is the Legendre symbol) horizontal ℓ-isogenies, no ascending $\ell-$isogeny, and $\ell - \left(\frac{\Delta_0}{\ell}\right)$ descending $\ell-$isogenies if $0 < d$.
- If $1 < d$ and $0 < e < d$, then the curve E admits no horizontal ℓ-isogeny, one ascending $\ell-$isogeny, and ℓ descending $\ell-$isogenies.
- If $e = d$ and $0 < d$, then the curve E admits no horizontal ℓ-isogeny, one ascending $\ell-$isogeny, and no descending $\ell-$isogeny.

Looking back at Definition 2.4.6, one notices that these components are $\ell-$volcanoes where the vertices in the level V_i of the volcano are the curves E for which $e = i$. Figure 2.2 illustrates the volcano structure of the ordinary components.

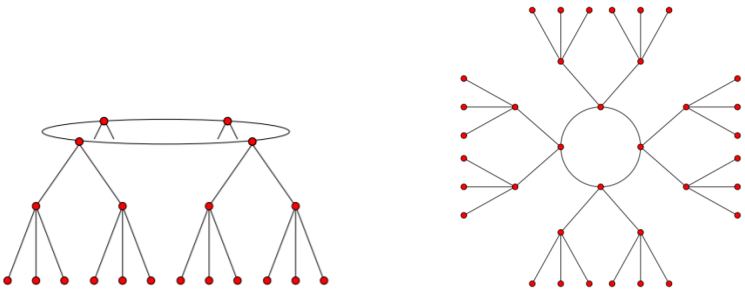


Figure 2.2: Example of 3-volcano on the left, and a component of depth 2 of the ordinary 3-isogeny graph on the right. Images taken from [Sut17].

[10] $h(\Delta)$ or $h(\mathcal{O})$ is the class number of $\mathcal{O}$.

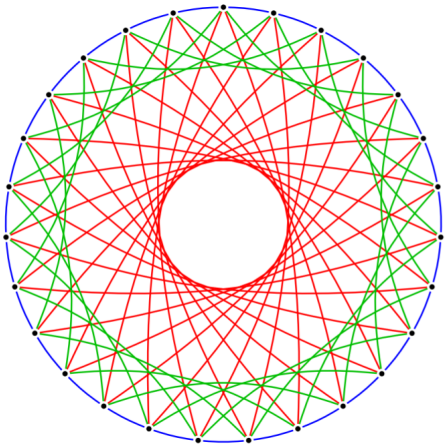


Figure 2.3: Example of isogeny star where three different isogeny degrees are used: 3, 5, 7. Beautiful image taken from [Pan21].

We refer to Kohel's thesis [Koh96] or [Sut13; Sut17] for more details about endomorphism rings of ordinary elliptic curves and ordinary graphs. Now let us have a closer look at each level of the volcano.

All the curves in the same level V_e (with $0 \leq e \leq d$) of the ordinary $\ell-$isogeny graph have the same endomorphism ring $\mathcal{O}_e = \mathbb{Z} + \ell^e \mathcal{O}_0$. Consider a different prime $\ell' \neq \ell$ such that $\ell' \nmid f_\pi$ and the discriminant Δ_e of $\mathcal{O}_e$ satisfies $\left(\frac{\Delta_e}{\ell'}\right) = 1$ (equivalently we have $\left(\frac{\Delta_\pi}{\ell'}\right) = 1$). In this case, each curve in V_e admits two horizontal $\ell'-$isogenies and 0 vertical $\ell'-$isogeny. That is at each level V_e, we have a cycle, a depth 0 $\ell'-$volcano. Taking many different such primes ℓ', we get more edges creating shortcuts in the cycle. These graphs were named *isogeny stars* by Rostovtsev and Stolbunov [RS06]. For sufficiently large values of q, an isogeny star is expander[11] when all such primes below $(\log 4q)^2$ are used [JMV09]. Figure 2.3 illustrates an example of isogeny star.

The beautiful structure of isogeny stars (as opposed to the messiness of supersingular graphs, more details in the following paragraphs) does not fall from heaven. In fact there is a class group action in the background from which the horizontal isogenies arise.

Class group action. Let $\mathcal{O}$ be an order in some quadratic imaginary field $\mathbb{Q}(\sqrt{\Delta})$ ($\Delta = t^2 - 4q < 0$). Recall that $\mathrm{Ell}_q(\mathcal{O})$ was the set of isomorphism classes of ordinary curves $E/\mathbb{F}_q$ such that $\mathrm{End}(E) = \mathcal{O}$. A *fractional ideal* of $\mathcal{O}$ is an $\mathcal{O}-$submodule $\mathfrak{a} \subset \mathbb{Q}(\sqrt{\Delta})$ such that there exists $d \in \mathcal{O}\backslash\{0\}$, $d\mathfrak{a} \subset \mathcal{O}$. A fractional ideal $\mathfrak{a}$ of $\mathcal{O}$ is *integral* if $\mathfrak{a} \subset \mathcal{O}$. A fractional ideal $\mathfrak{a}$ of $\mathcal{O}$ is *invertible* if there exists another fractional ideal $\mathfrak{b}$ of $\mathcal{O}$ such that $\mathfrak{ab} = \mathcal{O}$. The inverse $\mathfrak{a}^{-1}$ of an invertible fractional

[11] Expander graphs are strongly connected graphs: any small subset of vertices have a large boundary.

ideal $\mathfrak{a}$ of $\mathcal{O}$ is $\mathfrak{a}^{-1} = \frac{1}{N(\mathfrak{a})}\overline{\mathfrak{a}}$ where

$$\overline{\cdot} : \mathbb{Q}(\sqrt{\Delta}) \to \mathbb{Q}(\sqrt{\Delta}), \quad a + b\sqrt{\Delta} \mapsto \overline{a + b\sqrt{\Delta}} = a - b\sqrt{\Delta}$$

is the usual involution in $\mathbb{Q}(\sqrt{\Delta})$, and $N(\mathfrak{a})$ is the norm of the ideal $\mathfrak{a}$. The set $I(\mathcal{O})$ of invertible ideals of $\mathcal{O}$ forms an abelian group under ideal multiplication. The *ideal class group* of $\mathcal{O}$, denoted by $\mathsf{cl}(\mathcal{O})$, is the quotient of the group of invertible ideals of $\mathcal{O}$ modulo the subgroup $P(\mathcal{O})$ of principal fractional ideals of $\mathcal{O}$:

$$\mathsf{cl}(\mathcal{O}) = I(\mathcal{O})/P(\mathcal{O}).$$

Fractional ideals define finite subgroups of curves in $\mathrm{Ell}_q(\mathcal{O})$, which in turn define separable isogenies.

Proposition 2.4.9. *Let E be an elliptic curve in $\mathrm{Ell}_q(\mathcal{O})$ and let $\mathfrak{a}$ be an invertible integral ideal of $\mathcal{O}$ generated by $\alpha_1, \alpha_2 \in \mathfrak{a}$. Then $\mathfrak{a}$ defines a finite subgroup of E denoted $E[\mathfrak{a}]$ and given by*

$$E[\mathfrak{a}] = \bigcap_{\alpha \in \mathfrak{a}} \ker(\alpha) = \ker(\alpha_1) \cap \ker(\alpha_2).$$

Factor the ideal $\mathfrak{a}$ as $(\pi_p\mathcal{O})^r\mathfrak{a}_s$ where $\mathfrak{a}_s \not\subseteq \pi_p\mathcal{O}$. Then $\#E[\mathfrak{a}] = \#E[\mathfrak{a}_s] = N(\mathfrak{a}_s)$.

As a finite subgroup of E, $E[\mathfrak{a}]$ defines a separable isogeny

$$\phi_{\mathfrak{a}} : E \to \mathfrak{a}E := E/E[\mathfrak{a}].$$

Ideal multiplication corresponds to composition of isogenies. The principal ideals correspond to endomorphisms. Hence for two equivalent[12] ideals $\mathfrak{a}$ and $\mathfrak{b}$, we have $\mathfrak{a}E = \mathfrak{b}E$. Therefore, denoting by $[\mathfrak{a}]$ the ideal class of an invertible ideal $\mathfrak{a}$ of $\mathcal{O}$, the curve $[\mathfrak{a}]E := \mathfrak{b}E$ where $\mathfrak{b} \in [\mathfrak{a}]$ is well defined. We obtain an action of the class group $\mathsf{cl}(\mathcal{O})$ of $\mathcal{O}$ on $\mathrm{Ell}_q(\mathcal{O})$.

Theorem 2.4.10. *Let $\mathcal{O}$ be an order in $\mathbb{Q}(\sqrt{t^2 - 4q})$ such that $\mathrm{Ell}_q(\mathcal{O})$ is not empty. Then the ideal class group $\mathsf{cl}(\mathcal{O})$ acts freely on $\mathrm{Ell}_q(\mathcal{O})$ via the map*

$$\begin{array}{rcl} \mathsf{cl}(\mathcal{O}) \times \mathrm{Ell}_q(\mathcal{O}) & \to & \mathrm{Ell}_q(\mathcal{O}) \\ ([\mathfrak{a}], E) & \mapsto & [\mathfrak{a}]E. \end{array}$$

Whenever p is inert in $\mathcal{O}$, there are two orbits of cardinality $h(\mathcal{O})$ each. Otherwise, the action is transitive and the unique orbit has cardinality $h(\mathcal{O})$.

As we will briefly discuss in the following paragraph, a similar class group action can also be constructed in the supersingular case [Cas+18; CK20] when an embedding of $\mathcal{O}$ into the endomorphism ring of some supersingular curve E is provided. For ordinary curves, this class group action is always transitive. In fact, since t and p are coprime, then

$$\left(\frac{\Delta_{\mathcal{O}}}{p}\right) = \left(\frac{\Delta_{\pi}}{p}\right) = \left(\frac{t^2 - 4q}{p}\right) = 1,$$

[12] That is $\mathfrak{a}$ and $\mathfrak{b}$ have the same class $[\mathfrak{a}] = [\mathfrak{b}]$, equivalently the ideal $\mathfrak{a}\mathfrak{b}^{-1}$ is principal.

so p splits in $\mathcal{O}$.

In $\mathsf{cl}(\mathcal{O})$, each ideal class has integral ideal representative $\mathfrak{a}$ of norm $\ell_1^{e_1}\ell_2^{e_2}\cdots\ell_m^{e_m}$ where the primes ℓ_i with $1 \leq i \leq m$ split in $\mathcal{O}$, this is equivalent to $\left(\frac{\Delta_\mathcal{O}}{\ell_i}\right) = 1$. For $1 \leq i \leq m$, fix a root a_i of $X^2 - tX + q = 0 \mod \ell_i$ and let b_i be the other root. Write $\ell_i\mathcal{O} = \mathfrak{l}_i\overline{\mathfrak{l}_i}$ where $\mathfrak{l}_i$ is the prime ideal above ℓ_i generated by $(\ell_i, \pi - a_i)$ and of norm ℓ_i. Then for each elliptic curve $E \in \mathrm{Ell}_q(\mathcal{O})$, the two horizontal ℓ_i-isogenies of domain E correspond in fact to the actions of $[\ell_i]$ and $[\overline{\ell_i}] = [\ell_i]^{-1}$ on E. Their kernels are given by

$$\ker(\phi_{[\mathfrak{l}_i]}) = E[\ell_i] \cap \ker(\pi - a_i) \quad \ker(\phi_{[\mathfrak{l}_i]^{-1}}) = E[\ell_i] \cap \ker(\pi - b_i).$$

In the class group $\mathsf{cl}(\mathcal{O})$, the integral ideal $[\mathfrak{a}]$ factors as $[\mathfrak{a}] = [\mathfrak{l}_1]^{e_1} \cdots [\mathfrak{l}_m]^{e_m}$. Hence the action of the ideal class $[\mathfrak{a}]$ can be evaluated by consecutively evaluating for each $1 \leq i \leq m$, $|e_i|$ times the action of the ideal classes $[\mathfrak{l}_i]$ or $[\mathfrak{l}_i]^{-1}$ depending on the sign of e_i. This class group action evaluation is efficient as far as the primes ℓ_i are small enough. Chapter 3 provides further details about the computation of the class group action.

Supersingular graph. The supersingular $\ell-$isogeny graph is less structured compared to the ordinary $\ell-$isogeny graph. Recall that in characteristic $p > 3$, there are about $\frac{p}{12}$ isomorphism classes of supersingular curves and each class has a representative E defined over $\mathbb{F}_{p^2}$. More surprisingly, all supersingular isogenies between supersingular curves defined over $\mathbb{F}_{p^2}$ are defined over $\mathbb{F}_{p^2}$. This implies that supersingular isogeny graph over $\mathbb{F}_{p^2}$ is equivalent[13] to the full supersingular isogeny graph, that is the supersingular isogeny graph over $\overline{\mathbb{F}_p}$.

There are in fact 5 possible traces for supersingular curves defined over $\mathbb{F}_{p^2}$: $t = 0$, $t = \pm p$ and $t = \pm 2p$. Meanwhile, the number of isomorphism classes of supersingular elliptic curves effectively having trace $t = 0$ or $t = \pm p$ is at most 6 depending on the congruence class of p modulo 12 [Sch87; AAM19]. Those components[14] are less interesting as far as cryptography is concerned. The remaining curves E have trace $t = 2p$, in which case $\#E(\mathbb{F}_{p^2}) = (p-1)^2$, or $t = -2p$, in which case $\#E(\mathbb{F}_{p^2}) = (p+1)^2$. They correspond to two identical components (in terms of structure) and are isomorphic to the full isogeny graph. Therefore, to study the full supersingular $\ell-$isogeny graph, one can restrict to the supersingular component whose vertices have trace $t = -2p$ (or $t = 2p$). In general, this component is what is usually referred to when using the terms *supersingular $\ell-$isogeny graph.*

The supersingular $\ell-$isogeny graph is connected, $\ell+1$ regular, and is an expander graph [Piz90]. As a consequence, given any two supersingular $j-$invariants j_1 and j_2, there exists a relatively short ℓ-power degree isogeny $\phi : E(j_1) \to E(j_2)$. The supersingular $\ell-$isogeny graph looks very messy and unstructured. Figure 2.4 illustrates the supersingular $2-$isogeny graph over the field $\mathbb{F}_{2521^2}$.

Even though the supersingular $\ell-$isogeny graph looks highly unstructured, one can still manage to attach some structure to it. In the ordinary case, the Frobenius endomorphism of an ordinary curve E readily provides a non trivial suborder $\mathbb{Z}[\sqrt{t^2-4q}]$ of $\mathrm{End}(E)$. Ordering the orders above $\mathbb{Z}[\sqrt{t^2-4q}]$, we get the volcano structure on

[13] Up to neglecting some few (at most 6) isolated vertices.

[14] They merge with the larger components when considering the graph over an extension of $\mathbb{F}_{p^2}$ of degree 24 (at most).

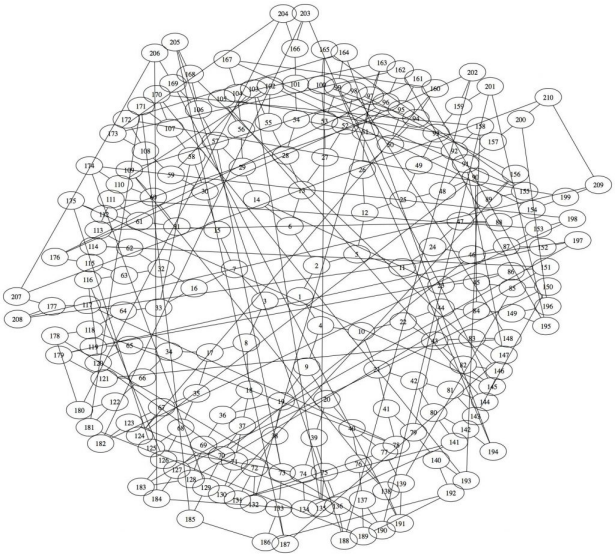


Figure 2.4: Supersingular 2−isogeny graph over $\mathbb{F}_{2521^2}$. This image give an idea on how unstructured the supersingular graph is. Image taken from [Lau17].

curves whose endomorphism contains a copy of $\mathbb{Z}[\sqrt{t^2-4q}]$. This same insight could be used to attach some volcanic structure to the supersingular isogeny graph. All one needs to do is to pick a supersingular curve E_0 together with an imaginary quadratic order $\mathcal{O}$ such that $\mathrm{End}(E_0)$ contains a copy of $\mathcal{O}$. With respect to $\mathcal{O}$, one defines horizontal, descending and ascending supersingular ℓ−isogenies. One therefore obtains a volcano structure where nodes are supersingular elliptic curves E such that $\mathrm{End}(E)$ contains a copy of $\mathcal{O}$.

Delfs-Galbraith [DG16] noticed that for supersingular curves defined over $\mathbb{F}_p$, $\mathbb{Z}[\sqrt{-p}] \subset \mathrm{End}(E)$, therefore the components of the $\mathbb{F}_p$ supersingular isogeny graph are volcanoes. The depth of these volcanoes is at most 2. Meanwhile, if one goes up to $\mathbb{F}_{p^2}$, the depth becomes infinite. In fact, the set of supersingular curves E for which $\mathrm{End}(E)$ contains a copy of the suborder $\mathbb{Z}+\ell^r\mathcal{O}$ $(0<r)$ of some imaginary quadratic order $\mathcal{O}$ is always non empty. This means that in reality, over $\mathbb{F}_{p^2}$, we obtain a sort of infinite volcano where j−invariants reappear as we descend. At each level of this infinite volcano, as in the ordinary case, there is a class group action operating on the curves. Nevertheless, in order to evaluate the class group action at a given supersingular curve E in the level r, the embedding of $\mathbb{Z}+\ell^r\mathcal{O}$ into $\mathrm{End}(E)$ needs to be known. Determining this embedding is not efficient in general. Coló-Kohel [CK20] introduce this class group action on supersingular graphs and name it *orientation*. They use it to design a Diffie-Hellman type key exchange protocol named OSIDH (Oriented Supersingular Isogeny Diffie-Hellman). Onuki [Onu21] provides further details about supersingular curve orientation and Chenu-Smith [CS21] introduce a generalisation of $\mathbb{F}_p$-supersingular elliptic curves.

2.5 — The central problems in isogeny-based cryptography

The first central problem in isogeny-based cryptography is that of computing isogenies between given isogenous elliptic curves, usually refered to as the pure isogeny problem.

Problem 2.5.1. *Let E and E' be two isogenous elliptic curves defined over a finite field $\mathbb{F}_q$. Compute an isogeny $\phi : E \to E'$.*

In general, when the degree of ϕ is smooth, ϕ can be described using a sequence $(E_0 = E, E_1, \cdots, E_n = E')$ of curves such that E_i and E_{i+1} are m_i-isogenous for some small integer m_i. Furthermore, when $E[\deg(\phi)] \subset E(\mathbb{F}_q)$, ϕ can be described using a generator of its kernel. It is infeasible to write down a large degree isogeny as a rational map. Isogenies ϕ of large non smooth degree are very difficult to handle. When $E[\deg(\phi)] \subset E(\mathbb{F}_q)$, ϕ can be described using $\ker\phi$ but one can not effectively compute (with current isogeny formulas) ϕ and check that its image curve is E'.

In practice, the problems underlying isogeny-based protocols are slightly different. In some cases, one needs to compute an isogeny defined over a specific subfield, or an isogeny with a specified degree. In SIDH for instance, the degree of the isogeny is fixed and the action of the isogeny on some torsion points is provided. This supplementary information has been exploited in adaptive attacks [GPST16; FP21a], in passive attacks [Pet17; Que+21; FKMT21] on imbalanced variants of SIDH.

Another more natural and more interesting problem is that of determining the endomorphism ring of a given elliptic curve.

Problem 2.5.2. *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. Compute* $\mathrm{End}(E)$.

Solving the endomorphism ring computation problem would lead to efficient attacks on SQISign [De +20] and Séta [SKPS19; Feo+19].

These two problems are central in isogeny-based cryptography. The known attempts to solve them on one hand, and the nature of the relation between the pure isogeny problem and the endomorphism ring problem on the other hand, differ considerably depending whether the curves in play are ordinary or supersingular.

The ordinary case. In general, the best known algorithms to solve the ordinary pure isogeny problem are improvements of an algorithm due to Galbraith [Gal99]. Let E and E' be two isogenous ordinary curves. Galbraith's algorithm consists in walking up to the surface of the ordinary $\ell-$isogeny graph through chains of ascending $\ell-$isogenies $\phi_1 : E \to E_1$ and $\phi_2 : E' \to E_2$, recovering an horizontal isogeny $\phi : E_1 \to E_2$ and returning $\widehat{\phi_2} \circ \phi \circ \phi_1$ as an isogeny from E to E'. In general, one can always choose ℓ such that E and E' are on the surface. The horizontal isogeny ϕ can be recovered classically in time[15] $\tilde{O}(q^{1/4})$ (and roughly the same amount of space) using the meet in the middle algorithm, or quantumly in time $2^{O(\sqrt{\log p})}$ using an algorithm Childs, Jao and Soukharev [CJS14] which reduces the problem of computing ϕ into an instance of the hidden-shift problem.

Regarding the ordinary endomorphism ring problem, the main obstacles are factoring the Frobenius discriminant $\Delta_\pi = t^2 - 4q$, and computing isogenies of degree ℓ where

[15] The $\tilde{O}$ notation here means that we ignore polylog factors.

ℓ divides the conductor f_π of $\mathbb{Z}[\pi]$, that is the square root of the square part of Δ_π. In fact, when this factorisation is known and the conductor f_π of $\mathbb{Z}[\pi]$ is smooth, for each prime ℓ dividing f_π, one can probe the depth of $\mathrm{End}(E)$ in the $\ell-$isogeny graph to recover the largest power of ℓ dividing the conductor of $\mathrm{End}(E)$. Once the conductor f of $\mathrm{End}(E)$ is computed, we get that $\mathrm{End}(E) \cong \mathbb{Z} + f\mathcal{O}_{\max}$, where $\mathcal{O}_{\max}$ is the maximal order in $\mathbb{Q}(\sqrt{t^2-4q})$. Works of Bisson and Sutherland [Bis12; BS11] propose a general algorithm with subexponential complexity.

The ordinary pure isogeny problem and the ordinary endomorphism ring problem are not quite related. In fact, knowing the endomorphism rings of the ordinary curves in play in the pure isogeny problem does not make the problem easier. Instead, its complexity remains the same given that from the discussion at the beginning of this paragraph, finding an isogeny from E to E', reduces to finding an horizontal isogeny between curves lying on the surface of the ordinary $\ell-$isogeny graph. This includes curves having the same endomorphism ring. Nevertheless, one should notice that when the class group action considered is on supersingular curves, say the $\mathbb{F}_p$ subgraph [Cas+18] or the Coló-Kohel orientation [CK20], computing the endomorphism ring is equivalent to solving the pure isogeny problem [CPV20; Wes21].

The supersingular case. The supersingular $\ell-$isogeny graph over $\mathbb{F}_{p^2}$ is connected. So any two supersingular curves E and E' defined over $\mathbb{F}_{p^2}$ are isogenous. The generic meet in the middle path finding algorithm takes $\tilde{O}(p^{1/2})$ time and space. In [DG16], Delfs and Galbraith suggest to walk down to the $\mathbb{F}_p$ subgraph, then brute force the isogeny defined over $\mathbb{F}_p$. This leads to a memory free algorithm with complexity $\tilde{O}(p^{1/2})$. Recently, Santos, Costello and Shi [SCS21] provided an implementation of the Delfs-Galbraith algorithm incorporating some speed-up in the walking down to the $\mathbb{F}_p$ subgraph step. Their algorithm still has the same asymptotic complexity but beats the original Delfs-Galbraith in practice when it comes to runtime.

The supersingular endomorphism ring problem was first studied in David Kohel's thesis [Koh96]. The general idea consists in searching for loops at the vertex E in the supersingular isogeny graph till these loops, that correspond to endomorphisms of E, generate the endomorphism ring of E. The most recent version of this algorithm is due to Eisenträger et al. [Eis+20] which runs in time $O(\log(p)^2 p^{1/2})$. Their idea is to generate endomorphisms till the suborder they generate is contained in a relatively small number of maximal orders, then the endomorphism ring is recovered by a brute force search among these maximal orders.

The relation between the supersingular isogeny problem and the supersingular endomorphism ring problem is well understood. In fact they are equivalent. Many papers prove this equivalence relying on heuristics [PL17; EHM17; Eis+18], but Wesolowski [Wes21] recently proved the equivalence relying on the Generalised Riemann Hypothesis.

Chapter 3

SimS: A Simplification of SiGamal

This chapter is for all practical purposes identical to the paper *SimS: A Simplification of SiGamal* [FP21c], authored jointly with Christophe Petit, which was published at PQCrypto 2021.

3.1 — Introduction

The construction of a large scale quantum computer would make the nowadays widely used public PKE schemes insecure, namely RSA [RSA78], ECC [Kob87] and their derivatives. As a response to the considerable progress in constructing quantum computers, NIST launched a standardization process for post-quantum secure protocols in December 2016 [Nat].

Isogeny-based protocols are in general based on the assumption that given two isogenous curves E and E', it is difficult to compute an isogeny from E to E'. This hard problem was used by J. M. Couveignes [Cou06], Rostovtsev and Stolbunov [RS06] to design a key exchange protocol using ordinary isogenies, and by Charles, Goren and Lauter [CLG09] to design a cryptographic hash function using supersingular isogenies. In 2011, as a countermeasure to the sub-exponential quantum attack on the CRS (Couveignes-Rostovtsev-Stolbunov) scheme by Childs et al. [CJS14], Jao and De Feo designed SIDH [JD11] (Supersigular Isogeny Diffie-Hellman), a Key Exchange protocol based on supersingular isogenies. The submission of SIKE [Jao+20] (a Key Encapsulation Mechanism based on SIDH) to the NIST standardization process marked the starting point of a more active research in isogeny-based cryptography. Isogeny-based protocols are not the most efficient candidates for post quantum cryptography, but they provide the shortest keys and ciphertexts.

In 2018, Castryck et al. constructed CSIDH [Cas+18] (Commutative SIDH) using the $\mathbb{F}_p$-sub-graph of the supersingular isogeny graph. CSIDH key exchange is close to CRS but is an order of magnitude more efficient. PKE schemes based on isogeny problems include SIKE, SÉTA [SKPS19] and more recently SiGamal and C-SiGamal [MOT20]. SÉTA and the PKEs canonically derived from the key exchange protocols SIDH and CSIDH are only OW-CPA secure. They require the use of hash functions and/or generic transformations such as the Fujisaki-Okamoto [FO99] or OAEP [BR94] to fulfil higher security requirements such as IND-CPA and IND-CCA security ([SKPS19, §2.4],[Jao+20, §1.4], [MOT20, §3.3]). This motivated Moriya, Onuki and Tagaki to introduce the SiGamal [MOT20, §5] and C-SiGamal [MOT20, §6] PKE schemes derived from CSIDH. SiGamal and C-SiGamal provide IND-CPA security under new assumptions they introduce. The authors noticed that neither SiGamal nor C-SiGamal is IND-CCA secure. In Remark 7 of [MOT20], they sug-

gest a slightly modified version of SiGamal that from their point of view could be IND-CCA secure, but they left its study as open problem.

Contributions. In this chapter, we prove that the variant of SiGamal suggested by Moriya et al. in Remark 7 of their paper is not IND-CCA secure by exhibiting a simple and concrete attack. We then modify SiGamal to thwart this attack, and obtain a new isogeny-based PKE scheme which we call SimS. We prove that SimS is IND-CPA secure relying on CSIDH security assumptions (Assumption 2). This is a considerable improvement on SiGamal whose IND-CPA security relies on new assumptions. We then introduce a "knowledge of Exponent" type assumption (Assumption 3) under which we prove that SimS is IND-CCA secure. This assumption may have other applications in isogeny-based cryptography.

We adapt the Magma code for SiGamal [Mor20] to run a proof of concept implementation of SimS using the SiGamal primes p_{128} and p_{256}. For the prime p_{128}, SimS is about 1.13x faster than SiGamal and about 1.19x faster than C-SiGamal. For the prime p_{256}, we get a 1.07x speedup when compared to SiGamal and a 1.21x speedup when compared to C-SiGamal.

For the same set of parameters, SimS has smaller private keys, public keys and ciphertexts compared to SiGamal and C-SiGamal. SimS is simple, sits between SiGamal and CSIDH, helps to better understand the relation between SiGamal and CSIDH while providing IND-CCA security and being more efficient compared to SiGamal. Table 3.1 best summarizes our contributions.

	CSIDHpke	SimS	SiGamal	C-SiGamal
Private key	$[\mathfrak{a}]$	$[\mathfrak{a}]$	$\mathfrak{a}$	$\mathfrak{a}$
Size of plaintext	$\log_2 p$	$r-2$	$r-2$	$r-2$
Size of Alice's public key	$\log_2 p$	$\log_2 p$	$2\log_2 p$	$2\log_2 p$
Size of ciphertexts (or Bob's public key)	$2\log_2 p$	$2\log_2 p$	$4\log_2 p$	$2\log_2 p$
Class group cost for p_{128} compared to CSIDH	x1.00	x1.30	x1.50	x1.50
Class group cost for p_{256} compared to CSIDH	x1.00	x2.31	x2.57	x2.57
Enc + Dec cost for p_{128} compared to CSIDHpke	x1.00	x1.38	x1.57	x1.65
Enc + Dec cost for p_{256} compared to CSIDHpke	x1.00	x2.62	x2.82	x3.17
Security	OW-CPA	IND-CCA	IND-CPA	IND-CPA

Table 3.1: Comparison between CSIDHpke, SimS, SiGamal and C-SiGamal. CSIDHpke uses the csidh-512 prime, while SimS, SiGamal and C-SiGamal use the primes p_{128} and p_{256} which are SiGamal primes that provide the same security level as the csidh-512 prime.

Outline. The remainder of this chapter is organized as follows: in Section 3.2, we recall the main ideas of the class group action and the CSIDH key exchange protocol. In section 3.3, we present the SiGamal PKE scheme and we show that the variant suggested in [MOT20, Remark 7] is not IND-CCA secure. Section 3.4 is devoted to SimS and its security arguments. In section 3.5 we present the outcome of a proof-of-concept implementation and compare SimS to CSIDH and (C-)SiGamal in Section 3.6. We conclude the paper in Section 3.7.

3.2 — Preliminaries

3.2.1 – Class group action on supersingular curves defined over $\mathbb{F}_p$. We refer to [Sil09; Was08] for general mathematical background on supersingular elliptic curves and isogenies, to [Cas+18; DG16] for supersingular elliptic curves defined

over $\mathbb{F}_p$ and their $\mathbb{F}_p$-endomorphism ring, and to [CH; Ren18] for isogenies between Montgomery curves.

Let $p \equiv 3 \mod 4$ be a prime greater than 3. The equation $By^2 = x^3 + Ax^2 + x$ where $B \in \mathbb{F}_p^*$ and $A \in \mathbb{F}_p \backslash \{\pm 2\}$ defines a Montgomery curve E over $\mathbb{F}_p$. The curve $E : By^2 = x^3 + Ax^2 + x$ is isomorphic (over $\mathbb{F}_p$) to the curve defined by the equation $y^2 = x^3 + Ax^2 + x$ (resp. $-y^2 = x^3 + Ax^2 + x$) when B is a square in $\mathbb{F}_p$ (resp. B is not a square in $\mathbb{F}_p$). The curve E is said to be supersingular if $\#E(\mathbb{F}_p) \equiv 1 \mod p$, otherwise E is said to be ordinary. If E is a supersingular curve defined over $\mathbb{F}_p$ with $p > 3$, then $\#E(\mathbb{F}_p) = p + 1$. All the elliptic curves we consider in this paper are supersingular curves defined by an equation of the form $y^2 = x^3 + Ax^2 + x$ where $A \in \mathbb{F}_p$ is called the Montgomery coefficient of the curve. In the rest of this section, we briefly describe the class group action used in CSIDH.

Let E be a supersingular curve defined over $\mathbb{F}_p$ and let π be the Frobenius endomorphism of E. The $\mathbb{F}_p$-endomorphism ring $\mathcal{O}$ of E is isomorphic to either $\mathbb{Z}[\pi]$ or $\mathbb{Z}[\frac{1+\pi}{2}]$ [DG16]. As in the ordinary case, the class group $\mathsf{cl}(\mathcal{O})$ of $\mathcal{O}$ acts freely and transitively on the set $\mathcal{E}\ell\ell_p(\mathcal{O})$ of supersingular elliptic curves defined over $\mathbb{F}_p$ and having $\mathbb{F}_p$-endomorphism ring $\mathcal{O}$. We have the following theorem.

Theorem 3.2.1. *[Cas+18, Theorem 7] Let $\mathcal{O}$ be an order in an imaginary quadratic field such that $\mathcal{E}\ell\ell_p(\mathcal{O})$ is non empty. The ideal class group $\mathsf{cl}(\mathcal{O})$ acts freely and transitively on the set $\mathcal{E}\ell\ell_p(\mathcal{O})$ via the map*

$$\begin{array}{ccc} \mathsf{cl}(\mathcal{O}) \times \mathcal{E}\ell\ell_p(\mathcal{O}) & \to & \mathcal{E}\ell\ell_p(\mathcal{O}) \\ ([\mathfrak{a}], E) & \mapsto & [\mathfrak{a}]E = E/E[\mathfrak{a}], \end{array}$$

where $\mathfrak{a}$ is an integral ideal of $\mathcal{O}$ and $E[\mathfrak{a}] = \cap_{\alpha \in \mathfrak{a}} \ker \alpha$.

From now on, we will consider the quadratic order $\mathbb{Z}[\pi]$ and the action of its class group $\mathsf{cl}(\mathbb{Z}[\pi])$ on the set $\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi])$. We represent $\mathbb{F}_p$-isomorphism classes of curves in $\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi])$ using the Montgomery coefficient A [CD20, Proposition 3].

The efficiency of the computation of an isogeny with known kernel essentially depends on the smoothness of its degree. In [Cas+18], the authors work with a prime p of the form $p = 4\ell_1 \cdots \ell_n - 1$. This implies that for $i \in \{1, \cdots, n\}$, $\left(\frac{-p}{\ell_i}\right) = 1$ and by the Kummer decomposition theorem [Kum47], $(\ell_i) = \mathfrak{l}_i\overline{\mathfrak{l}_i}$ in $\mathsf{cl}(\mathbb{Z}[\pi])$, where $\mathfrak{l}_i = (\ell_i, \pi - 1)$ and $\overline{\mathfrak{l}_i} = (\ell_i, \pi + 1)$ are integral ideals of prime norm ℓ_i. It follows that $[\mathfrak{l}_i][\overline{\mathfrak{l}_i}] = [\ell_i] = [1]$ in $\mathsf{cl}(\mathbb{Z}[\pi])$, hence $[\mathfrak{l}_i]^{-1} = [\overline{\mathfrak{l}_i}]$. Since the primes ℓ_i are small, then the action of the ideal classes $[\mathfrak{l}_i]$ and $[\mathfrak{l}_i]^{-1}$ can be computed efficiently using Vélu formulas for Montgomery curves [CH; Ren18]. In reality, the kernel of the isogeny corresponding to the action of the prime ideal $\mathfrak{l}_i = (\ell_i, \pi - 1)$ is generated by a point $P \in E(\mathbb{F}_p)$ of order ℓ_i, while that of the isogeny corresponding to the action of $\mathfrak{l}_i^{-1} = (\ell_i, \pi + 1)$ is a point $P' \in E(\mathbb{F}_{p^2}) \backslash E(\mathbb{F}_p)$ of order ℓ_i such that $\pi(P') = -P'$. The computation of the action of an ideal class $\prod[\mathfrak{l}_i]^{e_i}$ where $(e_1, \cdots, e_n) \in \{-m, \cdots, m\}^n$ can be done efficiently by composing the actions of the ideal classes $[\mathfrak{l}_i]$ or $[\mathfrak{l}_i]^{-1}$ depending on the signs of the exponents e_i. Since the prime ideals $\mathfrak{l}_i$ are fixed, then the vector $(e_1, \cdots, e_n)$ is used to represent the ideal class $\prod[\mathfrak{l}_i]^{e_i}$. From the discussion in [Cas+18, §7.1], m is chosen to be the least positive integer such that

$$(2m + 1)^n \geq |\mathsf{cl}(\mathbb{Z}[\pi])| \approx \sqrt{p}.$$

3.2.2 – CSIDH. CSIDH [Cas+18] stands for Commutative Supersingular Isogeny Diffie-Hellman and is a Diffie-Hellman type key exchange protocol. The base group in Diffie-Hellman protocol is replaced by the unstructured set $\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi])$ and the exponentiation is replaced by the class group action of $\mathsf{cl}(\mathbb{Z}[\pi])$ on $\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi])$. Concretely, CSIDH is designed as follows.

Setup. Let $p = 4\ell_1 \cdots \ell_n - 1$ be a prime where $\ell_1, \cdots, \ell_n$ are small distinct odd primes. The prime p and the supersingular elliptic curve $E_0 : y^2 = x^3 + x$ defined over $\mathbb{F}_p$ with $\mathbb{F}_p$-endomorphism $\mathbb{Z}[\pi]$ are the public parameters.

Key Generation. The private key is an n-tuple $e = (e_1, \cdots, e_n)$ of uniformly random integers sampled from a range $\{-m, \cdots, m\}$. This private key represents an ideal class $[\mathfrak{a}] = \prod [\mathfrak{l}_i]^{e_i} \in \mathsf{cl}(\mathbb{Z}[\pi])$. The public key is the Montgomery coefficient $A \in \mathbb{F}_p$ of the curve $[\mathfrak{a}]E_0 : y^2 = x^3 + Ax^2 + x$ obtained by applying the action of $[\mathfrak{a}]$ on E_0.

Key Exchange Suppose Alice and Bob have successfully computed pairs of private and public key (e, A) and (e', B) respectively. Upon receiving Bob's public key $B \in \mathbb{F}_p \backslash \{\pm 2\}$, Alice verifies that the elliptic curve $E_B : y^2 = x^3 + Bx^2 + x$ is a supersingular curve, then applies the action of the ideal class corresponding to her secret key $e = (e_1, \cdots, e_n)$ to E_B to compute the curve $[\mathfrak{a}]E_B = [\mathfrak{a}][\mathfrak{b}]E_0$. Bob does analogously with his own secret key $e' = (e'_1, \cdots, e'_n)$ and Alice's public key $A \in \mathbb{F}_p \backslash \{\pm 2\}$ to compute the curve $[\mathfrak{b}]E_A = [\mathfrak{b}][\mathfrak{a}]E_0$. The shared secret is the Montgomery coefficient S of the common secret curve $[\mathfrak{a}][\mathfrak{b}]E_0 = [\mathfrak{b}][\mathfrak{a}]E_0$.

The security of the CSIDH key exchange protocol relies on the following assumptions.

Let λ be the security parameter and let $p = 4\ell_1 \cdots \ell_n - 1$ be a prime where $\ell_1, \cdots, \ell_n$ are small distinct odd primes. Let E_0 be the supersingular elliptic curve $y^2 = x^3 + x$ defined over $\mathbb{F}_p$, let $[\mathfrak{a}]$, $[\mathfrak{b}]$ and $[\mathfrak{c}]$ be uniformly random ideal classes in $\mathsf{cl}(\mathbb{Z}[\pi])$.

Assumption 1. *The CSSICDH (Commutative Supersingular Isogeny Computational Diffie-Hellman) assumption holds if for any probabilistic polynomial time (PPT) algorithm $\mathcal{A}$,*

$$Pr[E = [\mathfrak{b}][\mathfrak{a}]E_0 \mid E = \mathcal{A}(E_0, [\mathfrak{a}]E_0, [\mathfrak{b}]E_0)] < \mathsf{negl}(\lambda).$$

Assumption 2. *The CSSIDDH (Commutative Supersingular Isogeny Decisional Diffie-Hellman) assumption holds if for any PPT algorithm $\mathcal{A}$,*

$$Pr\left[b = b^* \;\middle|\; \begin{array}{l} [\mathfrak{a}], [\mathfrak{b}], [\mathfrak{c}] \leftarrow \mathsf{cl}(\mathbb{Z}[\pi]), b \xleftarrow{\$} \{0,1\}, \\ F_0 := [\mathfrak{b}][\mathfrak{a}]E_0, F_1 = [\mathfrak{c}]E_0, \\ b^* \leftarrow \mathcal{A}(E_0, [\mathfrak{a}]E_0, [\mathfrak{b}]E_0, F_b) \end{array}\right] = \frac{1}{2} + \mathsf{negl}(\lambda).$$

In [CSV20], Castryck et al. show that Assumption 2 does not hold for primes $p \equiv 1 \bmod 4$. This does not affect primes $p \equiv 3 \bmod 4$, which are used in CSIDH, SiGamal and in our proposal SimS.

An IND-CPA insecure PKE from CSIDH. A PKE scheme can be canonically derived from a key exchange protocol. For the case of CSIDH, this PKE scheme is sketched as follows. Suppose that Alice has successfully computed her key pair (e, A). In order to encrypt a message $\mathsf{m} \in \{0,1\}^{\lceil \log p \rceil}$, Bob computes a random key pair (e', B)

and the binary representation S_{01} of the corresponding shared secret S. He sends $(B, \mathsf{c} = S_{01} \oplus \mathsf{m})$ to Alice as the ciphertext. For the decryption, Alice computes the shared secret S and its binary representation S_{01}, then recovers $\mathsf{m} = S_{01} \oplus \mathsf{c}$. In the comparison in Section 3.6, the term CSIDHpke will be used to refer to the previous PKE each time the precision is needed.

The above PKE scheme is not IND-CPA secure. In fact, given two distinct plaintexts m_0 and m_1, if (B, c) is a ciphertext for m_i, then $S_{01}^i = \mathsf{c} \oplus \mathsf{m}_i$ is the binary representation of the Montgomery coefficient of a supersingular curve while $S_{01}^{1-i} = \mathsf{c} \oplus \mathsf{m}_{1-i}$ is that of an ordinary curve with overwhelming probability. Hence an adversary can efficiently guess if the ciphertext (B, c) is that of m_0 or m_1. In practice, a hash function h is used to mask the supersingular property of the shared secret S, the ciphertext becomes $(B, \mathsf{c} = h(S_{01}) \oplus \mathsf{m})$.

3.3 — Another look at SiGamal protocol

3.3.1 – SiGamal protocol and variants. Let $p = 2^r \ell_1 \cdots \ell_n - 1$ be a prime such that $\ell_1, \cdots, \ell_n$ are small distinct odd primes. Let E_0 be the elliptic curve $y^2 = x^3 + x$ and let $P_0 \in E(\mathbb{F}_p)$ be a point of order 2^r. Recall that for every small odd prime ℓ_i dividing $p+1$, there are two prime ideals $\mathfrak{l}_i = (\ell_i, \pi - 1)$ and $\overline{\mathfrak{l}_i} = (\ell_i, \pi + 1)$ above ℓ_i in $\mathsf{cl}(\mathbb{Z}[\pi])$. Also, the isogenies $\phi_{\mathfrak{l}_i}$ and $\phi_{\overline{\mathfrak{l}_i}}$ of domain E_0 correspond to the isogenies with kernel generated by $P_{\mathfrak{l}_i} \in E_0[\ell_i] \cap ker(\pi - 1) \backslash \{0\}$ and $P_{\overline{\mathfrak{l}_i}} \in E_0[\ell_i] \cap ker(\pi + 1) \backslash \{0\}$ respectively. The points $\mathfrak{l}_i P_0$ and $\overline{\mathfrak{l}_i} P_0$ are images of the point P_0 trough these isogenies respectively. Let $\mathfrak{a} = (\alpha)\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n} \in \mathsf{cl}(\mathbb{Z}[\pi])$ where α is an integer then point $\mathfrak{a}P_0$ is the image of P_0 by the composition of the isogenies $\phi_{\mathfrak{l}_i}$ if $e_i > 0$ or $\phi_{\overline{\mathfrak{l}_i}}$ if $e_i < 0$, and the multiplication by α. For a given integer k, we denote by $[k] \circ \mathfrak{b}$ the composition of the isogeny corresponding to the ideal class $\mathfrak{b}$ and the scalar multiplication by k, and the point $[k] \circ \mathfrak{b} P_0$ denotes the image of P_0 trough this isogeny.

The SiGamal PKE scheme can be summarized as follows.

Key Generation. Let $p = 2^r \ell_1 \cdots \ell_n - 1$ be a prime such that $\ell_1, \cdots, \ell_n$ are small distinct odd primes. Let E_0 be the elliptic curve $y^2 = x^3 + x$ and let $P_0 \in E(\mathbb{F}_p)$ be a point of order 2^r. Alice takes a random integral ideal $\mathfrak{a} = (\alpha)\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}$ where α is a uniformly random element of $\mathbb{Z}_{2^r}^\times$, computes $E_1 := [\mathfrak{a}]E_0$ and $P_1 := \mathfrak{a}P_0$. Her public key is $(E_1, x(P_1))$ and her private key is $(\alpha, e_1, \cdots, e_n)$. Let $\mathbb{Z}_{2^{r-2}} = \mathbb{Z}/2^{r-2}\mathbb{Z}$ be the message space.

Encryption. Let $\mathsf{m} \in \mathbb{Z}_{2^{r-2}}$ be a plaintext, Bob embeds m in $\mathbb{Z}_{2^r}^\times$ via $\mathsf{m} \mapsto M = 2\mathsf{m} + 1$. Bob takes a random integral ideal class $\mathfrak{b} = (\beta)\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}$ where β is a uniformly random element of $\mathbb{Z}_{2^r}^\times$. Next, he computes $[M]P_1$, $E_3 = [\mathfrak{b}]E_0$, $P_3 := \mathfrak{b}P_0$, $E_4 = [\mathfrak{b}]E_1$ and $P_4 := \mathfrak{b}([M]P_1)$. He sends $(E_3, x(P_3), E_4, x(P_4))$ to Alice as the ciphertext.

Decryption. Upon receiving $(E_3, x(P_3), E_4, x(P_4))$, Alice computes $\mathfrak{a}P_3$ and solves a discrete logarithm instance between P_4 and $\mathfrak{a}P_3$ using the Pohlig-Hellman algorithm [PH78]. Let $M \in \mathbb{Z}_{2^r}^\times$ be the solution of this computation. If $2^{r-1} < M$, then Alice changes M to $2^r - M$. She computes the plaintext $\mathsf{m} = (M-1)/2$.

In C-SiGamal, a compressed version of SiGamal, one replaces the point $\mathfrak{a}\mathfrak{b}P_0$ by a distinguished point $P_{E_4} \in E_4$ of order 2^r, which then does not need to be transmitted.

The scheme integrates an algorithm that canonically computes a distinguished point of order 2^r on a given supersingular curve defined over $\mathbb{F}_p$ where $p = 2^r l_1 \cdots l_n - 1$. We refer to [MOT20] for more details on the SiGamal and C-SiGamal.

Moriya et al. prove that SiGamal and C-SiGamal are IND-CPA secure relying on two assumptions they introduce. However, they point out that SiGamal is not IND-CCA secure since one can efficiently compute a valid encryption of $3\mathsf{m}+1$ from a valid encryption of m. Indeed, given $([\mathfrak{b}]E_0, \mathfrak{b}P_0, [\mathfrak{b}]E_1, [2\mathsf{m}+1]\mathfrak{b}P_1)$ one easily computes $([\mathfrak{b}]E_0, \mathfrak{b}P_0, [\mathfrak{b}]E_1, [3][2\mathsf{m}+1]\mathfrak{b}P_1) = ([\mathfrak{b}]E_0, \mathfrak{b}P_0, [\mathfrak{b}]E_1, [2(3\mathsf{m}+1)+1]\mathfrak{b}P_1)$. A similar argument applies for C-SiGamal as well.

As a remedy, Moriya et al. suggest to omit the curve $[\mathfrak{b}]E_1$ in the ciphertext (see [MOT20, Remark 7]). We now show that this variant is still vulnerable to IND-CCA attacks.

3.3.2 – An IND-CCA attack on Moriya et al.'s variant. In this version of SiGamal, a ciphertext for m is of the form $([\mathfrak{b}]E_0, \mathfrak{b}P_0, [2\mathsf{m}+1]\mathfrak{b}P_1)$ and the decryption process is identical to that of the original SiGamal. We prove the following lemma.

Lemma 3.3.1. *Let* $\mathsf{c} = ([\mathfrak{b}]E_0, \mathfrak{b}P_0, [2\mathsf{m}+1]\mathfrak{b}P_1)$ *be a ciphertext for a plaintext* m*, then* $\mathsf{c}' = ([\mathfrak{b}]E_0, [1/3]\mathfrak{b}P_0, [2\mathsf{m}+1]\mathfrak{b}P_1)$ *is a ciphertext for* $\mathsf{m}' = 3\mathsf{m}+1$.

Proof. To decrypt c', Alice computes $[\mathfrak{a}][\mathfrak{b}]E_0$ and $\mathfrak{a}([1/3]\mathfrak{b}P_0) = [1/3]\mathfrak{a}\mathfrak{b}P_0$, then she solves a discrete logarithm problem between $[2\mathsf{m}+1]\mathfrak{b}P_1 = [2\mathsf{m}+1]\mathfrak{a}\mathfrak{b}P_0$ and $[1/3]\mathfrak{a}\mathfrak{b}P_0$. The solution to this discrete logarithm problem is

$$M' = \pm 3(2\mathsf{m}+1) = \pm(2(3\mathsf{m}+1)+1) = \pm(2\mathsf{m}'+1).$$

It follows that the corresponding plaintext (after changing M' to $2^r - M'$ when necessary) is $(M'-1)/2 = 3\mathsf{m}+1 = \mathsf{m}'$. □

Corollary 3.3.2. *The variant of SiGamal suggested by Moriya et al. in [MOT20, Remark 7] is not IND-CCA secure.*

3.4 — SimS

We now introduce a new protocol that resists the previous attack. We name our protocol SimS (**Sim**plified **S**iGamal), which highlights the fact that our scheme is a simplification of SiGamal.

3.4.1 – Overview. We observe that the attack presented in the previous section is effective because the ciphertext contains the curve $\mathfrak{b}E_0$ and its 2^r-torsion points $\mathfrak{b}P_0$.

SimS is obtained by adjusting SiGamal in such a way that when a curve is part of the ciphertext, then none of its points are, and the other way around. In order to achieve this, we replace the point $\mathfrak{a}\mathfrak{b}P_0$ in the (C)SiGamal protocol by a canonical point $P_{E_4} \in E_4 = [\mathfrak{a}][\mathfrak{b}]E_0$. More concretely, in SimS, Alice's secret key is an ideal class $[\mathfrak{a}]$, and her public key is the curve $E_1 = [\mathfrak{a}]E_0$. To encrypt a message m, Bob chooses a uniformly random ideal class $[\mathfrak{b}]$, he computes $E_3 = [\mathfrak{b}]E_0$, $E_4 = [\mathfrak{b}]E_1$ and he then canonically computes a point $P_{E_4} \in E_4(\mathbb{F}_p)$ of smooth order $2^r | p+1$. He sends E_3 and $P_4 = [2\mathsf{m}+1]P_{E_4}$ to Alice. In order to recover m, Alice computes $E_4 = [\mathfrak{a}]E_3$

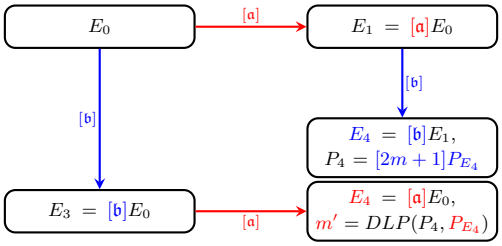


Figure 3.1: SimS scheme. The elements in black are public, while those in blue are known only by Bob and those in red only by Alice.

and P_{E_4}, then solves a discrete logarithm instance in a group of order 2^r using the Pohlig-Hellman algorithm. Figure 3.1 depicts the scheme.

The IND-CCA attack presented in Section 3.3.2 is no more feasible in SimS since no point of the curve E_3 nor the curve E_4 are part of the ciphertexts.

3.4.2 – The SimS public key encryption protocol. Now let us concretely describe the key generation, encryption and decryption processes. We use the Algorithm 12 to canonically compute the point $P_E \in E(\mathbb{F}_p)$ of order $2^r | p+1$.

Before we describe the protocol, let us notice that revealing P_4 or its x-coordinate may leak too much information about the curve E_4. In fact $x(P_4)$ is the root of the 2^r division polynomial of E_4. Moreover, one could easily derive $x(P_4 + (0,0)) = \frac{1}{x(P_4)}$ by a simple inversion in $\mathbb{F}_p$, which would affect the IND-CCA security of the scheme. To avoid this, we make use of a randomizing function[1] $f_E : \mathbb{F}_p \to F = Im(f_E)$, indexed by supersingular curves defined over $\mathbb{F}_p$, satisfying the following conditions:

- P1: f_E is bijective, f_E and its inverse $g_E = f_E^{-1} : F \to \mathbb{F}_p$ can be efficiently computed when E is given;
- P2: for every element $x \in \mathbb{F}_p$, any PPT adversary having no access to x and E cannot distinguish $f_E(x)$ from a random element of $F = Im(F_E)$;
- P3: for every element $x \in \mathbb{F}_p$, for every non identical rational function $R \in \mathbb{F}_p(X)$, any PPT adversary having no access to x and E cannot compute $f_E(R(x))$ from $f_E(x)$.

Example 3.4.1. *In the proof of concept implementation in Section 3.5, we use the randomizing function $f_E : x \mapsto \mathsf{bin}(x) \oplus \mathsf{bin}(A_E)$ where $\mathsf{bin}(\cdot)$ takes an element in $\mathbb{F}_p$ and returns its binary representation. In Appendix A.3, we argue that f_E satisfies (P1), (P2) and (P3), with respect to the parameters suggested in Section 3.5.*

Having such a function, SimS is designed as follows.

Key Generation: Let $p = 2^r \ell_1 \cdots \ell_n - 1$ be a prime such that $\ell_1, \cdots, \ell_n$ are small distinct odd primes and $\lambda + 2 \leq r \leq \frac{1}{2} \log p$ where λ is the security parameter. Let E_0 be the elliptic curve $y^2 = x^3 + x$. Alice takes a random ideal class $[\mathfrak{a}] \in \mathsf{cl}(\mathbb{Z}[\pi])$, computes $E_1 := [\mathfrak{a}]E_0$. Her public key is E_1 and her private key is $[\mathfrak{a}]$. The plaintext space is the set $\mathcal{M} = \mathbb{Z}_{2^{r-2}}$.

[1] In general, one will have $Im(f_E) \subset \{0,1\}^{\lceil \log_2 p \rceil}$.

Encryption: Let $\mathsf{m} \in \mathbb{Z}_{2^{r-2}}$ be a plaintext, Bob embeds m in $\mathbb{Z}_{2^r}^\times$ via $\mathsf{m} \mapsto 2\mathsf{m}+1$. Bob takes a random ideal class $[\mathfrak{b}] \in \mathsf{cl}(\mathbb{Z}[\pi])$ and computes $E_3 = [\mathfrak{b}]E_0$, $E_4 = [\mathfrak{b}]E_1$ and $P_4 = [2\mathsf{m}+1]P_{E_4}$. He sends $(E_3, x' = f_{E_4}(x(P_4)))$ to Alice as the ciphertext.

Decryption: Upon receiving (E_3, x'), Alice verifies that E_3 is a supersingular curve, computes $E_4 = [\mathfrak{a}]E_3$ and P_{E_4}. If $g_{E_4}(x')$ is not the x-coordinate of a 2^r-torsion point on the curve E_4, then Alice aborts. She solves the discrete logarithm instance between $P_4 = (g_{E_4}(x'), -)$ and P_{E_4} using the Pohlig-Hellman algorithm. Let $M \in \mathbb{Z}_{2^r}^\times$ be the solution of this computation. If $2^{r-1} < M$, then Alice changes M to $2^r - M$. She computes the plaintext $(M-1)/2$.

Theorem 3.4.2. *If f_{E_4} satisfies (P1), then SimS is correct.*

Proof. Since f_{E_4} satisfies (P1), then f_{E_4} is bijective, f_{E_4} and its inverse $g_{E_4} = f_{E_4}^{-1}$ can be efficiently computed by Alice since she has access to E_4.
As in CSIDH, the Montgomery coefficients of the curves $[\mathfrak{a}][\mathfrak{b}]E_0$ and $[\mathfrak{b}][\mathfrak{a}]E_0$ are equal. Therefore Alice and Bob obtain the same distinguished point P_{E_4}. Since the points P_{E_4} and $P_4 = [2\mathsf{m}+1]P_{E_4}$ have order 2^r, then the Pohlig-Hellman algorithm can be implemented on their x-coordinates $x(P_4) = g_{E_4}(x')$ and $x(P_{E_4})$ only to recover $M \equiv \pm(2\mathsf{m}+1) \bmod 2^r$. Since $\mathsf{m} \in \mathbb{Z}_{2^{r-2}}$, then $2\mathsf{m}+1 < 2^{r-1}$. Alice changes M to $2^r - M$ if $2^{r-1} < M$, then she computes the plaintext $(M-1)/2 = \mathsf{m}$. □

Remark 3.4.3. *Instantiating SimS with SIDH would lead to a PKE scheme which is not IND-CCA secure because SIDH is vulnerable to adaptive attacks [GPST16].*

3.4.3 – Security arguments. We prove that the IND-CPA security of SimS relies on Assumption 2. We also prove that SimS is IND-CCA secure under a Knowledge of Exponent-type assumption which we introduce.

Theorem 3.4.4. *If Assumption 2 holds and f_{E_4} satisfies (P2), then SimS is IND-CPA secure.*

Proof. We adapt the proof of [MOT20, Theorem 8] to our setting. Let us suppose that SimS is not IND-CPA secure, then there exists a PPT adversary $\mathcal{A}$ that can successfully distinguish whether a given ciphertext (E_3, x') was encrypted from a plaintext m_0 or m_1 with probability $\frac{1}{2} + \gamma$. Below, we use $\mathcal{A}$ to construct a PPT CSSIDDH solver $\mathcal{A}'$ whose success probability is $\frac{1}{2} + \frac{1}{2}\gamma$.

Given a CSSIDDH instance input $(E_0, [\mathfrak{a}]E_0, [\mathfrak{b}]E_0, F_b)$ as in Assumption 2, we choose $\bar{b} \in \{0,1\}$ uniformly at random, we compute $\mathsf{c} = ([\mathfrak{b}]E_0, f_{F_b}(x([2\mathsf{m}_{\bar{b}}+1]P_{F_b})))$. Let $b^* = \mathcal{A}(E_0, [\mathfrak{a}]E_0, \mathsf{c})$.

The CSSIDDH solver $\mathcal{A}'$ returns 1 if $b^* = \bar{b}$, and 0 if $b^* \neq \bar{b}$.

Now let's compute the advantage of $\mathcal{A}'$. Note that b is a uniformly random bit, so $b = 0$ with probability $\frac{1}{2}$ and $b = 1$ with probability $\frac{1}{2}$.

When $b = 0$, then $b^* = \bar{b}$ with probability $\frac{1}{2} + \gamma$.

When $b = 1$, $[\mathfrak{a}][\mathfrak{b}]E_0 \neq F_b$ and the ciphertext c is invalid. Since $\mathcal{A}$ does not have access to E and $x([2\mathsf{m}_b+1]P_{F_b})$, and that f_{F_b} satisfies (P2), then $\mathcal{A}$ can not distinguish $x' = f_{F_b}(x([2\mathsf{m}_b+1]P_{F_b}))$ from a random element of $Im(f_{F_b})$. Hence the output b^* of the query independent of b. We get that $b^* = \bar{b}$ with probability $\frac{1}{2}$.

Therefore, CSSIDDH solver $\mathcal{A}'$ succeeds with probability

$$\frac{1}{2}(\frac{1}{2}+\gamma)+\frac{1}{2}\times\frac{1}{2}=\frac{1}{2}+\frac{1}{2}\gamma.$$

□

Compared to the IND-CPA game setting, the adversary also has access to a decryption oracle $O(\cdot)$ in the IND-CCA game setting. To prove that SimS is IND-CCA secure, it is sufficient to prove that the decryption oracle is useless. This immediately follows if we assume that no PPT adversary having access to E_0, E_1 and a valid ciphertext c, can produce a brand new valid ciphertext c' unless she encrypts c' herself. This is formalized in the following assumption.

Assumption 3. *The CSSIKoE (Commutative Supersingular Isogeny Knowledge of Exponent) assumption is stated as follows.*

Let λ be a security parameter, let $p=2^r\ell_1\cdots\ell_n-1$ be a prime such that $\lambda+2\leq r\leq\frac{1}{2}\log p$. Let $[\mathfrak{a}]$, $[\mathfrak{b}]$ be a uniformly sampled elements of $\mathsf{cl}(\mathbb{Z}[\pi])$. Let $(f_E)_{E\in\mathsf{cl}(\mathbb{Z}[\pi])}$ be a family of randomizing functions as defined in Section 3.4.2 such that each of these functions satisfies (P3).

Then for every PPT adversary $\mathcal{A}$ that takes E_0, $[\mathfrak{a}]E_0$ and $([\mathfrak{b}]E_0, f_{[\mathfrak{a}][\mathfrak{b}]E_0}(x(P)))$ where $P\in[\mathfrak{a}][\mathfrak{b}]E_0$ is a point of order 2^r as inputs, and returns a couple $([\mathfrak{b}']E_0, f_{[\mathfrak{a}][\mathfrak{b}']E_0}(x(P')))\neq([\mathfrak{b}]E_0, f_{[\mathfrak{a}][\mathfrak{b}]E_0}(x(P)))$ where $P'\in[\mathfrak{a}][\mathfrak{b}']E_0$ is a point of order 2^r, there exists a PPT adversary $\mathcal{A}'$ that takes the same inputs and returns $([\mathfrak{b}'],[\mathfrak{b}']E_0, f_{[\mathfrak{a}][\mathfrak{b}']E_0}(x(P')))$.

Theorem 3.4.5. *Let us suppose that SimS is IND-CPA secure, and that Assumption 3 holds. Then SimS is IND-CCA secure.*

Proof. Let us suppose that Assumption 3 holds and SimS is not IND-CCA secure, and let us prove that SimS is not IND-CPA secure.

Since SimS is not IND-CCA secure, then there exists a PPT adversary $\mathcal{A}^{O(\cdot)}=(\mathcal{A}_1,O(\cdot))$ (where $O(\cdot)$ is the decryption oracle) that successfully determines if a given ciphertext c is that of a plaintext m_0 or m_1 with a non negligible advantage γ.

Suppose that the adversary $\mathcal{A}^{O(\cdot)}$ queries the decryption oracle $O(\cdot)$ with some valid ciphertexts $\mathsf{c}_1=(F_1,x_1),\cdots,\mathsf{c}_n=(F_n,x_n)$ computed by $\mathcal{A}_1$. By Assumption 3, there exists a polynomial time algorithm $\mathcal{A}_2$ that when outputting $\mathsf{c}_1=(F_1,x_1),\cdots,\mathsf{c}_n=(F_n,x_n)$ also outputs the ideal classes $[\mathfrak{b}_1],\cdots,[\mathfrak{b}_n]$ such that $F_i=[\mathfrak{b}_i]E_0$ for $i\in\{1,\cdots,n\}$. From the knowledge of the ideal classes $[\mathfrak{b}_1],\cdots,[\mathfrak{b}_n]$ and $[\mathfrak{a}]E_0$, the adversary $\mathcal{A}_2$ successfully decrypts $\mathsf{c}_1,\cdots,\mathsf{c}_n$.

Replacing the decryption oracle $O(\cdot)$ by $\mathcal{A}_2$, we obtain an adversary $\mathcal{A}'=(\mathcal{A}_1,\mathcal{A}_2)$ that successfully determines if a given ciphertext c is that of m_0 or m_1 with advantage γ (which is non negligible) and without making any call to the decryption oracle. This contradicts SimS's IND-CPA security. □

Remark 3.4.6. *In all this section, we have assumed that the ideal classes $[\mathfrak{a}]$ and $[\mathfrak{b}]$ were uniformly sampled elements of $\mathsf{cl}(\mathbb{Z}[\pi])$. Strictly speaking, in order to uniformly sample elements in $\mathsf{cl}(\mathbb{Z}[\pi])$, one needs to compute the class group structure and its generators. Computing the class group $\mathsf{cl}(\mathbb{Z}[\pi])$ requires sub-exponential time in its*

discriminant [BKV19, §1]. The class group structure for the CSIDH-512 prime was computed in [BKV19] with a lot of computational effort. As in the preliminary version of CSIDH or instantiations of CSIDH using different primes for which the class group is unknown, we assume that the many small prime ideals $\mathfrak{l}_i$ used to sampled elements in $\mathsf{cl}(\mathbb{Z}[\pi])$ (see Section 3.2.1) generate the entire class group or a sufficiently large subgroup of the class group such that the sampled ideals are close to being uniformly random. See [Cas+18, §7.1] for more details.

Remark 3.4.7. *The secret vectors $(e_1, \cdots, e_n) \in [-m, m]^n$ used to sample ideals $\mathfrak{a} = \mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n} \in \mathsf{cl}(\mathbb{Z}[\pi])$ can be seen as analogous to exponents in discrete logarithm-based protocols, and Assumption 3 is in that sense analogous to the "knowledge of exponent" assumption (see Appendix A.1) introduced by Damgård in the context of discrete logarithm-based cryptography [Dam92] and also used in [HT98]. If ever the class group $\mathsf{cl}(\mathbb{Z}[\pi])$ were computed for the SimS primes, then the analogy would be more immediate.*

3.5 — Implementation results

Here we present the experimentation results obtained by adapting the code of SiGamal [Mor20]. The implementation is done using the two primes proposed by Moriya et al. for SiGamal.

SiGamal prime p_{128}. Let p_{128} be the prime $2^{130} \cdot \ell_1 \cdots \ell_{60} - 1$ where ℓ_1 through ℓ_{59} are the smallest distinct odd primes, and ℓ_{60} is 569. The bit length of p_{128} is 522. The private key bound is $m = 10$.

SiGamal prime p_{256}. Let p_{256} be the prime $2^{258} \cdot \ell_1 \cdots \ell_{43} - 1$ where ℓ_1 through ℓ_{42} are the smallest distinct odd primes, and ℓ_{43} is 307. The bit length of p_{256} is 515. The private key bound is $m = 32$.

All the costs (number of field multiplications, where $1\mathbf{S}{=}0.8\mathbf{M}$ and $1\mathbf{a}{=}0.05\mathbf{M}$) of CSIDH presented are done with the csidh-512 prime (of 512 bits) while those of SimS, SiGamal and C-SiGamal are with p_{128} and p_{256}. The costs presented in Table 3.2 and Table 3.3 are the average costs of 20,000 rounds of key generation, encryption and decryption of each scheme.

Prime	csidh-512	p_{128}		p_{256}	
Scheme	CSIDH	SimS	(C)SiGamal	SimS	(C)SiGamal
Costs	441,989	576,124	663,654	1,023,400	1,140,189

Table 3.2: Cost (number of field multiplications, where $1\mathbf{S}{=}0.8\mathbf{M}$ and $1\mathbf{a}{=}0.05\mathbf{M}$) of class group action for CSIDH with the csidh-512 prime, SimS, SiGamal and C-SiGamal with p_{128} and p_{256}.

Remark 3.5.1. *In this proof of concept implementation, the class group algorithm considered does not take into account the improvements in [CDV20], [BFLS20], [CD20].*

3.6 — Comparison with SiGamal and CSIDH

Here we compare SimS, (C-)SiGamal and CSIDH (or CSIDHpke more precisely). The comparison is done at four levels: design, security, keys and ciphertext sizes, and efficiency.

	p_{128}			p_{256}		
	KGen	Enc.	Dec.	KGen	Enc.	Dec.
C-SiGamal	663,594	1,433,805	767,176	1,151,447	2,685,714	1,528,020
SiGamal		1,326,856	760,861		2,208,530	1,536,829
SimS	576,124	1,159,533	679,733	1,023,827	2,057,297	1,417,401

Table 3.3: Computational costs (number of field multiplications, where $1\mathbf{S}=0.8\mathbf{M}$ and $1\mathbf{a}=0.05\mathbf{M}$) for C-SiGamal, SiGamal and SimS with p_{128} and p_{256}.

Design. At the design level, SimS sits between (C)SiGamal and CSIDH. SimS's private keys are ideal classes, as in CSIDH, while in (C)SiGamal they are integral ideals. In the class group action in (C-)SiGamal, a point has to be mapped through the isogeny as well, as opposed to CSIDH and SimS.

Securiy. Security-wise, SimS IND-CPA security relies on CSIDH assumptions, contrarily to SiGamal whose IND-CPA security relies on new assumptions. Moreover, SimS is IND-CCA secure.

Keys and ciphertext sizes. The size of SimS's ciphertexts is equal to that of C-SiGamal's ciphertexts, and is half that of SiGamal ciphertexts. The size of SimS's public keys is half that of the public keys in SiGamal and C-SiGamal. The size of the private key in (C)SiGamal, compared to that of SimS, is augmented by r bits that are used to store the integer α such that the secret ideal $\mathfrak{a}$ is in the form $\mathfrak{a} = (\alpha)\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}$.

Efficiency. SimS is more efficient compared to SiGamal and C-SiGamal when using the same primes. From the results in Table 3.2, we have that for the prime p_{128}, the SimS class group action computation is 1.15x faster than that of (C)SiGamal and is 1.30x slower than that of CSIDH; and for the prime p_{256}, it is 1.11x faster than that of (C)SiGamal and is 2.31x slower than that of CSIDH. For Encryption and decryption with the prime p_{128}, SimS is about 1.13x faster than SiGamal and about 1.19x faster than C-SiGamal. For the prime p_{256}, we get a 1.07x speedup when compared to SiGamal and a 1.21x speedup when compared to C-SiGamal.

We summarize the comparison in Table 3.1. Note that the encryption in CSIDHpke is essentially two CSIDH class group computations and the decryption is one class group computation.

3.7 — Conclusion

In this chapter, we revisited the protocols introduced by Moriya et al. at Asiacrypt 2020, and obtained several results. We proved that the variant of SiGamal suggested by Moriya et al. is not IND-CCA secure. We construct a new isogeny based PKE scheme SimS by simplifying SiGamal in such a way that it resists the IND-CCA attack on SiGamal and its variants. SimS is more efficient than SiGamal and it has smaller private keys, public keys and ciphertexts. We prove that SimS is IND-CPA secure relying on CSIDH assumptions. We introduce a Knowledge of Exponent assumption in the isogeny context. Relying on the later assumption, we prove that SimS is IND-CCA secure. Interestingly, SimS is also closer to CSIDH than SiGamal was, allowing for a better comparison between those two protocols.

We leave a better study of the Knowledge of Exponent assumption and further cryptographic applications of this assumption to future work.

Acknowledgments. We thank Tomoki Moriya, Hiroshi Onuki and Tsuyoshi Takagi for sharing the SiGamal magma code with us. We thank Ankan Pal for his help in running our magma code. We thank Serge Fehr, Tomoki Moriya and the anonymous reviewers for their useful feedback.

Chapter 4

Séta: Supersingular Encryption from Torsion Points Attacks

This chapter is for all practical purposes identical to the paper *Séta: Supersingular Encryption from Torsion Points Attacks* [Feo+19], authored jointly with Luca De Feo, Cyprien Delpech de Saint Guilhem, Péter Kutas, Antonin Leroux, Christophe Petit, Javier Silva and Benjamin Wesolowski, which was published at Asiacrypt 2021.

4.1 — Introduction

Isogeny-based cryptography. Recent years have seen an increasing interest in cryptosystems based on supersingular isogeny problems as appropriate candidates for post-quantum cryptography. The latter has received greater focus due to the recent standardization process initiated by NIST.[1]

More precisely, the central problem of isogeny-based cryptography is, given two elliptic curves, to compute an isogeny between them. For the right choice of parameters, the best quantum algorithms for solving this problem still run in exponential time [BJS14]. Variants of this problem have been used to build primitives such as hash functions [CLG09], encryption schemes [JD11; Aza+20], key encapsulation mechanism (KEM)s [Aza+20] and signatures [GPS20; De +20].

Encryption schemes. The first key agreement and public-key encryption (PKE) scheme based on isogenies of ordinary elliptic curves was independently discovered by Couveignes [Cou06] and Rostovtsev and Stolbunov [RS06; Sto10]. It follows a "Diffie–Hellman-like" structure: Alice and Bob start from a public curve E_0 and choose random secret isogenies φ_A, φ_B to reach curves E_A, E_B. They then send the curves to each other and finally use their respective secrets to arrive at a common curve E_{AB}. It is then immediate to transform the key agreement into a CPA-secure PKE by following El Gamal's blueprint.

In 2011, Jao and De Feo [JD11] introduced SIDH, a key agreement protocol based on isogenies of supersingular curves, inspired both by the Couveignes–Rostovtsev–Stolbunov scheme and by the hash function of Charles, Goren and Lauter [CLG09]. In the supersingular case, however, isogenies do not have a natural commutative property, meaning that, for example, the result of applying Bob's isogeny φ_B to Alice's curve E_A cannot be meaningfully defined without some extra constraints. To solve this, Jao and De Feo proposed sending additional information in the protocol

[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 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, the isogeny problem upon which the security of the scheme is based now differs from the original problem in certain ways. Most importantly, the adversary has access to the image of certain torsion points under a secret isogeny. Galbraith, Petit, Shani and Ti [GPST16] were the first to exploit this extra information in an active attack showing that one cannot use static keys in SIDH. Then, two further works studied the generic problem of finding isogenies if the action of the isogeny on some torsion is known [Pet17; Que+21]. These look at two different scenarios:

1. The starting curve is $E_0 : y^2 = x^3 + x$;

2. The starting curve is chosen by the adversary;

Let p be a prime number; for simplicity we restrict to supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. Let A be the degree of some secret isogeny φ and let B be the order of a torsion group on which the action of φ is known. In the first case [Que+21] gives a polynomial-time algorithm to compute φ whenever $B > \sqrt{p}A^2$. In the second case it shows how to construct special starting curves (called *backdoor curves*) for which backdoor information is known, in the form of an endomorphism of the curve, which enables a polynomial-time algorithm to compute φ whenever $B > A^2$.

In SIDH one has $A \approx B \approx \sqrt{p}$ so these algorithms do not lead to an attack. However [Que+21] also shows that, if an adversary is allowed to choose the starting curve, then even in the SIDH setting it is possible to mount key-recovery attacks which take exponential time, yet are faster than known algorithms [Que+21, Corollary 32]. In anticipation of potential further cryptanalysis progress, it is desirable to design alternative cryptographic protocols that rely on different isogeny problems. An example of this is the CSIDH scheme [Cas+18] (and its variants [MOT20; FP21c]), 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.

These results show that 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.

Contributions of this chapter.. Our main contribution is to turn the attack described in [Que+21] into a PKE by using the special starting curves mentioned above as public keys. The associated secret key can be derived from an endomorphism of the curve with a specific minimal polynomial. More precisely, one can use any special curve whose endomorphism ring has a particular quadratic order embedded into it. Using such a starting curve, one can design a PKE where a message corresponds to an isogeny and a ciphertext contains the codomain of the isogeny together with images of the torsion points under the isogeny. Decryption is then performed using the algorithm which recovers the secret isogeny using the techniques developed in [Pet17] and [Que+21].

Choosing parameters for our scheme is not obvious due to the following reason. Even though trapdoor curves can be constructed in polynomial time, in practice this

can be very costly. This is acceptable for a backdoor, but not for a PKE for which key generation should be routine computation. The expensive step is to generate a supersingular elliptic curve with a prescribed endomorphism ring. We utilize techniques from SQISign [De +20] where one uses special primes to substantially speed up the procedure of generating starting curves. Furthermore, the worst-case complexity of torsion-point attacks is dependent on the number of prime factors of the isogeny degree. We therefore impose extra conditions on the quadratic order to avoid timing attacks that this could imply.

We also present variants for constructing backdoor curves which allow for slightly different decryption mechanisms. Namely one can either construct the starting curve directly and then compute a backdoor, or instead choose a secret backdoor curve first and then apply a secret walk to it. We discuss trade-offs between security, key size and speed in this context.

We emphasize that just knowing the equation of the starting curve and a description of the quadratic order embedded in it does not seem to be helpful without the concrete knowledge of an endomorphims realizing this embedding. We formalize this idea in what we call the *uber isogeny problem* or $\mathfrak{O}$-UIP (Problem 4.5.1): suppose that one knows that a certain quadratic order $\mathfrak{O}$ is embedded in the endomorphism ring of two curves E_0, E_s, and that and that a concrete embedding of E_0 is also given in input, the problem is to find an isogeny between E_0 and E_S corresponding to a $\mathfrak{O}$-ideal. The formulation of this $\mathfrak{O}$-UIP is inspired from the key recovery problem in CSIDH [Cas+18, Problem 10]. We show that SIDH, OSIDH [CK20] and our PKE scheme also rely implicitly on various instances of this assumption. We also provide an analysis on the difficulty of this problem.

Finally, we present an implementation of our scheme which includes searching for an appropriate base prime and measuring key generation and encryption/decryption speeds. Written in C, our implementation reuses some of the codebase of SQISign and improves the efficiency of several steps crucial for Séta computations.

In Section 4.2 we recall basic properties of supersingular elliptic curves and the SIDH protocol. Furthermore, we discuss backdoor curves (which in this context we rename as trapdoor curves) in more detail. In Section 4.3 we introduce our one-way function and PKE Séta. In Section 4.4 we show how one can generate keys efficiently for Séta. In Section 4.5 we introduce the uber isogeny assumption, discuss its relation to other studied isogeny problems and provide some analysis of its hardness. In Section 4.6 we provide details of our implementation.

4.2 — Preliminaries

We denote the computational security parameter by λ. We write PPT for probabilistic polynomial time. The notation $y \leftarrow \mathcal{A}(x; r)$ means that the algorithm $\mathcal{A}$, with input x and randomness r, outputs y. The notation Pr[sampling : event] means the probability of the event on the right happening after sampling elements as specified on the left. Given a set $\mathcal{S}$, we denote sampling a uniformly random element x of $\mathcal{S}$ by $x \xleftarrow{\$} \mathcal{S}$. A probability distribution X has min-entropy $H_\infty(X) = b$ if any event occurs with probability at most 2^{-b}. Given an integer $n = \prod_i \ell_i^{e_i}$, where the ℓ_i are its prime factors, we say that n is *B-powersmooth* if $\ell_i^{e_i} < B$ for all i. We denote by $\mathbb{Z}_n$ the set of residue classes modulo n.

4.2.1 – Supersingular elliptic curves. We recall definitions and results related to supersingular elliptic curves.

Let q be a power of p and let E_1, E_2 be elliptic curves defined over a finite field $\mathbb{F}_q$. An isogeny $\varphi : E_1 \to E_2$ is a surjective morphism which sends the point at infinity of E_1 to the point of infinity at E_2. An isogeny is also a group homomorphism from $E_1(\overline{\mathbb{F}_q})$ to $E_2(\overline{\mathbb{F}_q})$ with a finite kernel. The degree of the isogeny is its degree as a finite map of curves. If the isogeny φ is separable, then $\#\ker\varphi = \deg\varphi$. If there exists an isogeny φ from E_1 to E_2, then there exists a unique isogeny $\hat{\varphi}$ from E_2 to E_1 with the property that $\varphi \circ \hat{\varphi} = [n]$ where n is the degree of the isogeny and $[n]$ denotes the multiplication by n map on E_2. Such isogenies φ and $\hat{\varphi}$ are called dual of each other. We call two curves isogenous if there exists an isogeny between them. By the previous remark, this relation is symmetric.

Let E be an elliptic curve defined over $\mathbb{F}_q$. An isogeny from E to itself is called an endomorphism of E. Under addition and composition, endomorphisms of E form, together with the zero map, a ring denoted $\mathrm{End}(E)$. A theorem of Deuring states that such an endomorphism ring is either an order in an imaginary quadratic field (such curves are called ordinary) or a maximal order in a quaternion algebra (such curves are called supersingular).

It is a well-known theorem of Tate that two curves defined over $\mathbb{F}_q$ are isogenous by an isogeny defined over $\mathbb{F}_q$ if and only if their number of $\mathbb{F}_q$-rational points is equal. Isogenous curves have isomorphic endomorphism rings thus supersingularity is preserved under an isogeny. Supersingular curves can always be defined (up to isomorphism) over $\mathbb{F}_{p^2}$ and a curve is supersingular if and only if the number of points is congruent to $1 \bmod p$.

Supersingularity is thus preserved under isogenies.

Kernels of isogenies and Vélu's formulas. An isogeny is a group homomorphism whose kernel is a finite subgroup of the starting curve. Moreover, let E be an elliptic curve defined over finite field $\mathbb{F}_q$ and let G be a finite subgroup of $E(\overline{\mathbb{F}_q})$. Then there exists a unique (up to automorphisms of the target curve) separable isogeny whose kernel is exactly G. Due to this uniqueness property we will denote the image curve by E/G. Furthermore, given a subgroup G whose order is powersmooth, the curve E/G can be computed efficiently using Vélu's formulas [Vél71].

Elliptic curve j-invariant. An elliptic curve E defined over $\mathbb{F}_{p^2}$ can always be written in short Weierstrass form $E : y^2 = x^3 + Ax + B$, for $A, B \in \mathbb{F}_{p^2}$. We can therefore identify any curve with its two coefficients: $E \sim (A, B)$. Given such a curve, its j-invariant is defined as $j(E) = 1728\frac{4A^3}{4A^3+27B^2}$. As its name suggests, this quantity is invariant under any isomorphism over $\overline{\mathbb{F}_{p^2}}$. In this work, we denote by $\mathcal{J}_p$ the set of j-invariants of supersingular curves defined over $\mathbb{F}_{p^2}$. We then identify the set of isomorphism classes of supersingular elliptic curves over $\mathbb{F}_{p^2}$ with $\mathcal{J}_p$.

Twists of elliptic curves. As presented in [Aza+16, Section 2.4], two curves $E_1 \sim (A_1, B_1)$ and $E_2 \sim (A_2, B_2)$ are isomorphic over $\overline{\mathbb{F}_{p^2}}$ if and only if there is some $u \in \overline{\mathbb{F}_{p^2}}\backslash\{0\}$ such that $A_1 = u^4 A_2$ and $B_1 = u^6 B_2$. It happens that two curves defined over $\mathbb{F}_{p^2}$ are isomorphic over $\overline{\mathbb{F}_{p^2}}$ but not over $\mathbb{F}_{p^2}$; such curves are *twists* of one another.n For $p \neq 2, 3$, a quadratic twist of $E \sim (A, B)$ is any curve of the form $E^t \sim (t^2 A, t^3 B)$ for $t \in \mathbb{F}_{p^2}\backslash\mathbb{F}_{p^2}^2$ (i.e. t is not a square in $\mathbb{F}_{p^2}$). Curves with j-invariant equal to 0 or 1728 are treated separately and we refer to [Aza+16, Section 2.4].

Canonical curves. We take the same approach as [GPS20, Appendix A] to fix a canonical choice of curve for each j-invariant. Given $j \in \mathbb{F}_{p^2}$, we define the curve E_j as $E_j \sim (0,1)$ when $j = 0$, $E_j \sim (1,0)$ when $j = 1728$ and $E_j \sim (\frac{3j}{1728-j}, \frac{2j}{1728-j})$ otherwise.
Isogeny graphs. Let $\ell \neq p$ be a prime number. Define the graph $G_\ell = G_\ell(\mathbb{F}_{p^2})$ to have vertex set $V = \mathcal{J}_p$. We have that $\#V = \lfloor \frac{p}{12} \rfloor + k$, where $k \in \{0,1,2\}$. Given two vertices $j_1, j_2 \in V$, with representative curves E_1, E_2 such that $j(E_i) = j_i$, there is an edge in G_ℓ between j_1 and j_2 if and only if there is an equivalence class of ℓ-isogenies between E_1 and E_2, where two isogenies $\varphi, \psi : E_1 \to E_2$ are equivalent if there exists an automorphism α of E_2 such that $\psi = \alpha\varphi$.

Edges of $G_\ell(\mathbb{F}_{p^2})$ can also be defined by the modular polynomial [Sil94] $\Phi_\ell(x,y) \in \mathbb{Z}[x,y]$. It is symmetric, meaning that $\Phi_\ell(x,y) = \Phi_\ell(y,x)$, and is of degree $\ell+1$ in both x and y. It holds that $\Phi_\ell(j_1, j_2) = 0$ if and only if there is an ℓ-isogeny equivalence class between two curves with j-invariants j_1 and j_2, and thus an edge in G_ℓ. Therefore, given a vertex $j \in V$, its neighbours are exactly those j-invariants which are roots of the univariate polynomial $\Phi_\ell(x, j)$. As Φ_ℓ is of degree $\ell+1$ in x and all the j-invariants are in $\mathbb{F}_{p^2}$, we see that G_ℓ is an $(\ell+1)$-regular graph.

4.2.2 – Quaternion algebras and endomorphism rings of supersingular elliptic curves. A quaternion algebra is a four-dimensional central simple algebra over a field K. When the characteristic of K is not 2, then A admits a basis $1, i, j, ij$ such that $i^2 = a,\ j^2 = b, ij = -ji$ where $a, b \in K\backslash\{0\}$. The numbers a, b characterise the quaternion algebra up to isomorphism, thus we denote the aforementioned algebra by the pair (a, b). A quaternion algebra is either a division ring or it is isomorphic to $M_2(K)$, the algebra of 2×2 matrices over K.

Let A be a quaternion algebra over $\mathbb{Q}$. Then $A \otimes \mathbb{Q}_p$ is a quaternion algebra over $\mathbb{Q}_p$ (the field of p-adic numbers) and $A \otimes \mathbb{R}$ is a quaternion algebra over the real numbers. A is said to split at p (resp. at ∞) if $A \otimes \mathbb{Q}_p$ (resp. $A \otimes \mathbb{R}$) is a full matrix algebra. Otherwise it is said to ramify at p (resp. at ∞). A quaternion algebra over $\mathbb{Q}$ is split at every but finitely many places, and the list of these places defines the quaternion algebra up to isomorphism. An order in a quaternion algebra over $\mathbb{Q}$ is a four-dimensional $\mathbb{Z}$-lattice which is also a subring containing the identity (it is the non-commutative generalization of the ring of integers in number fields). A maximal order is an order that is maximal with respect to inclusion.

The endomorphism ring of a supersingular elliptic curve over $\mathbb{F}_{p^2}$ is a maximal order in the quaternion algebra $B_{p,\infty}$, which ramifies at p and at ∞. Moreover, for every maximal order in $B_{p,\infty}$ there exists a supersingular elliptic curve whose endomorphism ring is isomorphic to it.

It is easy to see that, when $p \equiv 3 \pmod 4$, this quaternion algebra is isomorphic to the quaternion algebra $(-p, -1)$. In that case, the integral linear combinations of $1, i, \frac{ij+j}{2}, \frac{1+i}{2}$ form a maximal order $\mathcal{O}_0$ which corresponds to an isomorphism class of supersingular curves, namely the class of curves with j-invariant 1728 (e.g. the curve $E : y^2 = x^3 + x$). It is easy to see that all elements $ai + bj + cij + d$ with $a, b, c, d \in \mathbb{Z}$ are contained in $\mathcal{O}_0$.

4.2.3 – Class group action on the set of supersingular curves . We briefly recall the main definitions and properties related to the class group of quadratic imaginary orders and their link with supersingular elliptic curves. We say that a

curve E admits an embedding of a quadratic imaginary order $\mathfrak{O}$, if there exists a subring of $\mathrm{End}(E)$ that is isomorphic to $\mathfrak{O}$. We say this embedding is *primitive* or *optimal* if this isomorphism cannot be extended to a super-order of $\mathfrak{O}$. We write $\mathcal{E}_{\mathfrak{O}}$ for the set of supersingular elliptic curves admitting a primitive embedding of $\mathfrak{O}$ (up to isomorphisms). Following [CK20], we also call a primitive embedding of $\mathfrak{O}$ in $\mathrm{End}(E)$ an $\mathfrak{O}$-*orientation* on E. Through the usual Deuring correspondence, $\mathfrak{O}$-ideals can be identified with isogenies. For any such ideal $\mathfrak{a}$, we write $\varphi_{\mathfrak{a}} : E \to \mathfrak{a} \star E$ for the corresponding isogeny. The property that $\mathfrak{a} \star E \cong \mathfrak{b} \star E$ when $\mathfrak{a}$ and $\mathfrak{b}$ are in the same ideal class proves that $\star$ defines a group action of the class group $\mathrm{Cl}(\mathfrak{O})$ on $\mathcal{E}_{\mathfrak{O}}$. The class number $h(\mathfrak{O})$ is the cardinality of $\mathrm{Cl}(\mathfrak{O})$. In full generality, we cannot say much more on $\#\mathcal{E}_{\mathfrak{O}}$ than the classical Proposition 4.2.1.

Proposition 4.2.1. *Let K be a quadratic imaginary field and let $\mathfrak{O}$ be a quadratic order inside K. When p does not split in K, the number of distinct embeddings of $\mathfrak{O}$ inside maximal orders of the quaternion algebra $\mathcal{B}_{p,\infty}$ is exactly $\mathrm{Cl}(\mathfrak{O})$. In particular, $\#\mathcal{E}_{\mathfrak{O}} \leq h(\mathfrak{O})$.*

In general, Proposition 4.2.1 does not help in estimating $\#\mathcal{E}_{\mathfrak{O}}$ precisely because we do not know how to estimate the number of different embeddings of $\mathfrak{O}$ into the same maximal order in $\mathcal{B}_{p,\infty}$. We provide examples of cases where more precise properties can be stated in Sections 4.5.2 and 4.5.3.

When p splits in the field K, then $\mathcal{E}_{\mathfrak{O}}$ is empty (the curves admitting an $\mathfrak{O}$-orientation are ordinary). In the remaining of this article, we consider that we are never in this case to simplify the notations and statements.

Any quadratic order $\mathfrak{O}$ can be written as $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_0$ where $\mathfrak{O}_0$ is another quadratic order (not necessarily distinct from $\mathfrak{O}$) and f is often called the conductor of $\mathfrak{O}$. When the conductor is one, we say that the quadratic order is *maximal*. In [LB20], it was shown that these conductors can be tied to isogenies.

Proposition 4.2.2. *Let $\mathfrak{O} = \mathbb{Z}+f\mathfrak{O}_0$ be a quadratic order and let E be a supersingular curve defined over $\mathbb{F}_{p^2}$. If E is in $\mathcal{E}_{\mathfrak{O}}$, then there exists an isogeny of degree f between E and a supersingular curve $E_0 \in \mathcal{E}_{\mathfrak{O}_0}$. Conversely, when there exists an isogeny of degree f between E and a supersingular curve $E_0 \in \mathcal{E}_{\mathfrak{O}_0}$, then E is in $\mathcal{E}_{\mathbb{Z}+f'\mathfrak{O}_0}$ for some f' dividing f.*

In Proposition 4.2.2, we say that the isogeny $\varphi : E_0 \to E$ of degree f is *descending* when $f' = f$. Let $\varphi : E_0 \to E$ be a descending isogeny of degree f, the embedding of $\mathfrak{O}$ in $\mathrm{End}(E)$ in Proposition 4.2.2 is obtained with endomorphisms of the form $[d] + \varphi \circ \alpha_0 \circ \hat{\varphi}$ with $d \in \mathbb{Z}$ and α_0 in the embedding of $\mathfrak{O}_0$ inside $\mathrm{End}(E_0)$. Similar endomorphisms are constructed in torsion point attacks against SIDH variants [Pet17; Que+21], and they underlie the decryption mechanism of the Séta encryption scheme.

4.2.4 – SIDH and SIKE. Here we give a high level description of SIDH and SIKE. We start with the original SIDH protocol of Jao and De Feo [JD11]. In the setup one chooses two small primes ℓ_A, ℓ_B and a prime p of the form $p = \ell_A^{e_A}\ell_B^{e_B}f - 1$, where f is a small cofactor and e_A and e_B are large (in SIKE [Aza+20] they use $\ell_A^{e_A} = 2^{216}$, $\ell_B^{e_B} = 3^{137}$ and $f = 1$). Let E be a fixed supersingular curve, for example,

assuming $p = 3 \bmod 4$, the elliptic curve with j-invariant 1728[2]. Let P_A, Q_A be a basis of $E[\ell_A^{e_A}]$ and let P_B, Q_B be a basis of $E[\ell_B^{e_B}]$. The protocol is as follows:

1. Alice chooses a random cyclic subgroup of $E[\ell_A^{e_A}]$ generated by $A = [x_A]P_A + [y_A]Q_A$ and Bob chooses a random cyclic subgroup of $E[\ell_B^{e_B}]$ generated by $B = [x_B]P_B + [y_B]Q_B$.

2. Alice computes the isogeny $\varphi_A : E \to E/\langle A\rangle$ and Bob computes the isogeny $\varphi_B : E \to E/\langle B\rangle$.

3. Alice sends the curve $E/\langle A\rangle$ and the points $\varphi_A(P_B)$ and $\varphi_A(Q_B)$ to Bob, and Bob similarly sends $(E/\langle B\rangle, \varphi_B(P_A), \varphi_B(Q_A))$ to Alice.

4. Alice and Bob both use the images of the torsion points to compute the shared secret which is the curve $E/\langle A, B\rangle$ (e.g. Alice can compute $\varphi_B(A) = [x_A]\varphi_B(P_A) + [y_A]\varphi_B(Q_A)$ and $E/\langle A, B\rangle = E_B/\langle\varphi_B(A)\rangle$).

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 $\varphi_A : E \to E/\langle A\rangle$ and her public key be the curve $E/\langle A\rangle$ together with the images of the torsion points $\varphi_A(P_B)$ and $\varphi_A(Q_B)$. Encryption and decryption work as follows:

1. To encrypt a bitstring m, Bob chooses a random subgroup generated by $B = [x_B]P_B + [y_B]Q_B$ and computes the corresponding isogeny $\varphi_B : E \to E/\langle B\rangle$. He computes the shared secret $E \to E/\langle A, B\rangle$ and hashes the j-invariant of $E/\langle A, B\rangle$ to a binary string s. The ciphertext corresponding to m is the tuple $(E/\langle B\rangle, \varphi_B(P_A), \varphi_B(Q_A), c := m \oplus s)$

2. In order to decrypt Bob's message, Alice computes $E/\langle A, B\rangle$ and from this information computes s. Then she retrieves the message by computing $c \oplus s$.

This PKE scheme is IND-CPA secure [JD11; Aza+20]. In the SIKE submission [Aza+20], it is transformed using the constructions in [HHK17, Section 3] to produce an IND-CCA secure KEM in the random oracle model (ROM).

4.2.5 – Trapdoor curves. Let E_1, E_2 be supersingular elliptic curves over $\mathbb{F}_{p^2}$ and let $\phi : E_1 \to E_2$ be an isogeny of degree D. First we recall the following algorithmic problem:

Problem 4.2.3 (SSI-T). *Let D and N be smooth coprime integers. Let $\phi : E_1 \to E_2$ be a secret isogeny of degree D. Assume that we know the action of ϕ on $E_1[N]$. Compute ϕ.*

Remark 4.2.4. *The SSI-T problem is a generalization of the CSSI introduced in [JD11] (Problem 4.5.6) where D and N are prime powers of the same size.*

[2] Jao and De Feo do not specify a particular curve, and recommend to pick one using Bröker's algorithm [Brö09], however there appears to be no advantage in doing so, and thus SIKE opts for $j = 1728$ for simplicity.

The SSI-T problem makes sense for any D, N which are coprime and sufficiently smooth. However, in many cases the size of the input is superlinear in p thus has no practical relevance. Thus from now on we restrict to instances where the D and N-torsion are efficiently representable:

Definition 4.2.5. *Let N be an integer and let p be a prime number. Let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. We call $E[N]$ efficiently representable if representing points in $E[N]$ requires polynomial space in $\log p = O(\lambda)$.*

Remark 4.2.6. *In particular $E[N]$ is efficiently representable whenever N is powersmooth or N divides $p^c - 1$ for some small c. In this paper we will mainly consider instances where N is smooth and divides $p^2 - 1$.*

We recall (slightly modified version of)[Que+21, Theorem 3] how finding a certain endomorphism of E_2 relates to finding the secret isogeny ϕ:

Theorem 4.2.7. *Let $\phi : E_1 \to E_2$ be a secret isogeny of degree D. Assume that $E[N]$ and $E[D]$ are efficiently representable for any supersingular curve E and that the action of ϕ on $E_1[N]$ is given. Suppose furthermore, that we know $\theta \in \mathrm{End}(E_1)$ and $d, e \in \mathbb{Z}$ such that the trace of θ is 0 and $\deg(\phi \circ \theta \circ \hat{\phi} + [d]) = N^2 e$. Let M be the largest divisor of D such that $E_2[M] \subset \ker(\phi \circ \theta \circ \hat{\phi}) \cap E_2[D]$. Let k be the number of distinct prime divisors of M. Then we can compute ϕ in time $O^*(2^k \sqrt{e})$.*

Proof. We sketch the proof of the theorem. Let $\tau = \phi \circ \theta \circ \hat{\phi} + [d]$. Then if $\ker(\tau)$ is cyclic, then $\tau = \psi' \circ \eta \circ \psi$ where $\deg(\psi) = \deg(\psi') = N$ and $\deg(\eta) = e$ and the kernels of ψ and ψ' are cyclic. In [Que+21, Theorem 3] it is shown that $\ker(\tau)$ is always cyclic if N is odd and if N is even then $\tau = \psi' \circ \eta \circ \psi \circ [K]$ where $\deg(\psi) = \deg(\psi') = N/K$, $\deg(\eta) = e$ and $K = 1$ or $K = 2$.

Then one can compute ψ and K using the torsion point information and ψ' using the observation that $\ker(\hat{\psi}') = \tau(E_2[B])$. The isogeny η can be computed by a meet-in-the-middle algorithm. Once τ is computed, one can compute ϕ by looking at $G = \ker(\phi \circ \theta \circ \hat{\phi}) \cap E_2[D]$. If $M = 1$ then G is cyclic and can be recomputed easily. If not, then one can use [Section 4.3][Pet17] to recover τ. The cost of this step is $O^*(2^k)$ where k is the number of prime factors of M.

□

Remark 4.2.8. *Theorem 4.2.7 in particular implies that one can recover ϕ in $O^*(\sqrt{e})$ whenever the number of distinct prime divisors of D (and hence M) is smaller than $\log \log p$. In Section 4.3.3, we introduce a condition on the quadratic order $\mathbb{Z}[\theta]$ to ensure that M is always equal to 1.*

The key ingredient to Theorem 4.2.7 is the knowledge of θ. When $M = 1$ (which will be the case for the concrete inversion procedure in Algorithm 1), all we really need is the action of θ on $E_1[N]$. Indeed, from the sketch of proof of Theorem 4.2.7, we see that in that case θ is only used to compute the kernel of the two isogenies ψ and ψ' of degree N. These kernels are computed by evaluating the N-torsion $\tau = \phi \circ \theta \circ \hat{\phi} + [d]$ which can be done with the action of θ and ϕ on $E_1[N]$.

Note the action of θ on $E_1[N]$ is hard to recover from E_1 only. This motivates a notion of (D, N)-trapdoor T to encompass any kind of information that enables the computation described in the proof of Theorem 4.2.7.

Definition 4.2.9. *Let p be a prime number and let D and N be coprime smooth integers. Then a tuple (E,T) is called a (D,N)-trapdoor curve if one can use T to solve any instance of the SSI-T problem (with parameters D,N,p) with starting curve E in polynomial time. We call T the trapdoor. In this chapter, we will have $T=(\theta,d,e)$ where $\theta\in\mathrm{End}(E)$, d and e are as in Theorem 4.2.7.*

In [Que+21] the authors introduce a polynomial-time algorithm for constructing (D,N)-trapdoor curves whenever $N>D^2$ and the number of prime divisors of $D<\log\log p$. The main idea is to reproduce the set-up of Theorem 4.2.7. Thus, if one can construct a supersingular elliptic curve E together with an endomorphism $\theta\in\mathrm{End}(E)$ verifying the requirements of Theorem 4.2.7, and compute the action of this endomorphism θ on $E[N]$, then one can solve SSI-T in polynomial time (by finding an e which is sufficiently small).

The conditions put on θ in Theorem 4.2.7 are essentially conditions on the minimal polynomial of θ, meaning that every trace zero element in the quaternion algebra whose norm is $\frac{B^2e-d^2}{A^2}$ can be used as a suitable θ. This implies that potential (D,N)-trapdoor curves are obtained from curves in $\mathcal{E}_{\mathfrak{O}}$ for quadratic order $\mathfrak{O}$ of the form $\mathbb{Z}\left[\sqrt{\frac{N^2e-d^2}{D^2}}\right]$.

We briefly sketch how θ can be generated. Since $\mathrm{Tr}(\theta)=0$, it can be written as $ci+bj+aij$ over $\mathcal{B}_{p,\infty}$. Then the degree of τ is $D^2(p^2a+p^2b+c^2)+d^2$. Observe that a,b,c can be rational numbers but since θ is an integral element its norm $p^2a^2+p^2b^2+c^2$ must be an integer. So one has to find d,e such that N^2e-d^2 is divisible by D^2 and is positive.

This can be achieved when $N>D^2$. Let $\Delta=N^2e-d^2$. Then one has to find a rational solution to the equation $p^2a^2+p^2b^2+c^2=\Delta$, which exists whenever Δ is a quadratic residue modulo p (if that is not the case one chooses a different d and e). A solution can be found using Denis Simon's algorithm [Sim05]. From there, we can find a maximal order $\mathcal{O}$ containing θ and then compute a supersingular elliptic curve whose endomorphism ring is isomorphic to $\mathcal{O}$ (see Algorithm 3 in Section 4.4.2). After that, the action of θ on the N-torsion can be found using an explicit representation of $\mathcal{O}$. All these operations can be done in polynomial time (see Algorithms 2 and 3 for more details), leading to the following theorem:

Theorem 4.2.10. *Let p be a prime number and let D and N be smooth coprime integers such that $N>D^2$ and the number of distinct prime divisors of D is smaller than $\log\log p$. Then there exists a polynomial-time algorithm which outputs a (D,N)-trapdoor curve E with the following information:*

- *The j-invariant of E.*
- *Integers d,e with $e=O(\log(p))$.*
- *A basis P,Q of $E[N]$ and the points $\theta(P),\theta(Q)$ for a trace 0 endomorphism θ such that $\deg([D]\theta+[d])=N^2e$.*

4.2.6 – Post-quantum OAEP transformation. We present here the post-quantum OAEP generic transformation we used in Section 4.3.5.

Let

$$f:\{0,1\}^{\lambda+k_1}\times\{0,1\}^{k_0}\to\{0,1\}^{n_c}$$

be an invertible injective function. The function f is the public key of the scheme, its inverse f^{-1} is the secret key. The scheme makes use of three hash functions

$$G : \{0,1\}^{k_0} \to \{0,1\}^{k-k_0},$$
$$H : \{0,1\}^{k-k_0} \to \{0,1\}^{k_0},$$
$$H' : \{0,1\}^{k} \to \{0,1\}^{k},$$

modelled as random oracles, where $k = \lambda + k_0 + k_1$. Given those, the encryption scheme is defined as follows:

- Enc: given a message $m \in \{0,1\}^{\lambda}$, choose $r \xleftarrow{\$} \{0,1\}^{k_0}$ and set

$$s = m||0^{k_1} \oplus G(r), \qquad t = r \oplus H(s),$$
$$c = f(s,t), \qquad d = H'(s||t),$$

  and output the ciphertext (c,d).
- Dec: given a ciphertext (c,d), use the secret key to compute $(s,t) = f^{-1}(c)$. If $d \neq H'(s||t)$ output $\perp$. Otherwise, compute $r = t \oplus H(s)$ and $\overline{m} = s \oplus G(r)$. If the last k_1 bits of $\overline{m}$ are 0, output the first n bits of $\overline{m}$, otherwise output $\perp$.

4.3 — Séta trapdoor one way function and public key encryption scheme

In this section we describe a general trapdoor one-way function where the main idea is to turn the attacks from [Que+21] into a trapdoor mechanism.

We first generalise the CGL hash function and we describe a trapdoor sub-family of this generalization. We then provide more details on key generation, evaluation and inversion. We finally describe the Séta public key encryption scheme and its CCA version.

4.3.1 – Generalised Charles-Goren-Lauter hash function. We generalise the CGL hash function family introduced in [CLG09]. To select a hash function from this family, one selects a j-invariant $j \in \mathcal{J}_p$ which canonically fixes a curve $E/\mathbb{F}_{p^2}$ with $j(E) = j$. There are $\ell + 1$ isogenies of degree ℓ connecting E to other vertices. These $\ell + 1$ vertices can be ordered in a canonical way and a canonical one of them can be ignored. Then, given a message $m = b_1 b_2 \dots b_n$, with $b_i \in [\ell]$, 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 walk 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. First, 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 $b_1 \in [\ell + 1]$ and $b_i \in [\ell]$ for $2 \leq i \leq n$; this introduces a one-to-one correspondence between inputs and cyclic isogenies of degree ℓ^n originating from E.

Secondly, we consider a generalisation where the walk takes place over multiple graphs G_{ℓ_i}. Given an integer $D = \prod_{i=1}^{n} \ell_i^{e_i}$ where the ℓ_i are prime factors, we introduce the notation $\mu(D) := \prod_{i=1}^{n} (\ell_i+1) \cdot \ell_i^{e_i-1}$. We then take the message m to be an element of

$$[\mu(D)] = \left\{ (m_1, \ldots, m_n) \;\middle|\; \begin{array}{l} m_i = b_{i1} b_{i2} \ldots b_{ie_i}, b_{i1} \in [\ell_i + 1], b_{ij} \in [\ell_i] \\ \text{for } 2 \leq j \leq e_i, \text{ for } 1 \leq i \leq n \end{array} \right\}$$

where each m_i is hashed along the graph G_{ℓ_i}. 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 $j = j_0$ parameterizes the overall hash function. As before, this generalization returns the final j-invariant $j_n = h_{j_{n-1}}(m_n)$ as the hash of m.

Thirdly, we also modify the CGL hash function to return the images of two canonically defined torsion points P_j and Q_j of order N under the D-isogeny $\varphi_m : E_j \to E_{j_n}$.

We call the resulting hash function family *generalized CGL* or G-CGL, and we denote it by $\mathcal{H}^{p,D,N}$, namely

$$\mathcal{H}^{p,D,N} = \left\{ h_j^{D,N} : m \mapsto (j(E_n), \varphi_m(P_j), \varphi_m(Q_j)) \mid j \in \mathcal{J}_p \right\}.$$

4.3.2 – A trapdoor function family from the G-CGL family. Given p, D and N, let $\mathcal{J}_{T,p} \subset \mathcal{J}_p$ be the set of j-invariants of (D, N)-trapdoor curves defined over $\mathbb{F}_{p^2}$ (see Definition 4.2.9). By definition of a trapdoor curve, for any $j_T \in \mathcal{J}_{T,p}$, the hash function $h_{j_T}^{D,N}$ can be inverted using the trapdoor information. We hence obtain the following family of trapdoor functions:

$$\mathcal{F}_T^{p,D,N} = \left\{ f_{j_T}^{D,N} : m \mapsto (j(E_n), \varphi_m(P_{j_T}), \varphi_m(Q_{j_T})) \mid j_T \in \mathcal{J}_{T,p} \right\},$$

where $f_{j_T}^{D,N} := h_{j_T}^{D,N}$.

Injectivity. We observe that, for a proper choice of parameters, the functions are injective.

Lemma 4.3.1. *Let $N^2 > 4D$. Then for any $j_T \in \mathcal{J}_{T,p}$, $f_{j_T}^{D,N}$ is injective.*

Proof. Let $N^2 > 4D$ and $j_T \in \mathcal{J}_{T,p}$, suppose that a function $f_{j_T}^{D}$ is not injective, i.e. that there are two distinct isogenies φ and φ' of degree D from E_{j_T} to E_c, corresponding to two distinct messages, with the same action on $E_{j_T}[N]$, implied by the colliding images of P_{j_T} and Q_{j_T}. Then, following [MP19, Section 4], their difference is also an isogeny between the same curves whose kernel contains the entire N-torsion. This, together with [Sil09, Lemma V.1.2], implies that $4D \geq \deg(\varphi - \varphi') \geq N^2$. Taking $N^2 > 4D$ ensures that in fact $\varphi = \varphi'$ and therefore that $f_{j_T}^{D,N}$ is injective. □

One-wayness. One-wayness of our function family relies on Problem 4.3.2 below. This problem is a variant of the CSSI problem introduced in [JD11], with the difference that the starting j-invariant is chosen at random from $\mathcal{J}_{T,p}$ (instead of being fixed) and only the min-entropy of the distribution is specified.

Problem 4.3.2 (Trapdoor computational supersingular isogeny (TCSSI) problem)**.** *Given p and integers D and N, let j_T be a uniformly random element of $\mathcal{J}_{T,p}$ and $\varphi_m : E_{j_T} \to E_m$ be a random isogeny of degree D sampled from a distribution X with min-entropy $H_\infty(X) = O(\lambda)$. Let $\{P_{j_T}, Q_{j_T}\}$ be a basis of the torsion group $E_{j_T}[N]$. Given $E_{j_T}, P_{j_T}, Q_{j_T}, E_m, \varphi_m(P_{j_T})$ and $\varphi_m(Q_{j_T})$, compute φ_m.*

Lemma 4.3.3. *Let j_T be a uniformly random element of $\mathcal{J}_{T,p}$. Then the function $f_{j_T}^{D,N} \in \mathcal{F}_T^{p,D,N}$ is (quantum) one-way under the (quantum) hardness of Problem 4.3.2.*

Proof. It is easy to check that the distribution of isogenies resulting from hashing a uniform $m^* \xleftarrow{\$} [\mu(D)]$ has the required entropy; hence the reduction is immediate. □

4.3.3 – Inversion. In this section, we concretely show how to use methods from [Que+21] to invert a given function $f_{j_T}^{D,N} \in \mathcal{F}_T^{p,D,N}$ with trapdoor information T. We assume that D is odd and that $\gcd(D, N) = 1$. We take E_{j_T} a supersingular curve inside $\mathcal{E}_{\mathfrak{O}}$ where $\mathfrak{O}$ is the quadratic order $\mathbb{Z}[\sqrt{(N^2e - d^2)/D^2}]$ for some integers d, e. We write θ for the endomorphism of $\mathrm{End}(E_{j_T})$ such that $\mathbb{Z}[\theta] \cong \mathfrak{O}$. Let us also take a basis P_{j_T}, Q_{j_T} of $E_{j_T}[N]$. If we define T as $e, d, P_{J_T}, Q_{j_T}, \theta(P_{j_T}), \theta(Q_{j_T})$, then E_{j_T}, T is a (D, N)-trapdoor curve as produced in Theorem 4.2.10.

To make the inversion mechanism efficient on all inputs, we require the additional condition that the discriminant Δ of $\mathfrak{O}$ is a quadratic nonresidue modulo every prime divisor of D. The concrete statement can be found in Lemma 4.3.4. We explain how to generate $E_{j,T}$, $\mathfrak{O}$ and T in Sections 4.4.1 and 4.4.2. We are given (j_m, P_m, Q_m) as the output of $f_{j_T}^{D,N}$ for some input m, which we want to recover. Let the isogeny corresponding to m be denoted by ϕ_m. We assume that $P_m = \phi_m(P_{j_T})$ and $Q_m = \phi_m(Q_{j_T})$. Let $\tau := \phi_m \circ \theta \circ \hat{\phi_m} + [d]$ and let $G := \ker(\tau - [d]) \cap E_m[D]$.

Lemma 4.3.4. *If $\Delta = \mathrm{Disc}\ \mathfrak{O}$ is a non-quadratic residue modulo every prime dividing D, the group G is cyclic and equal to $\ker(\hat{\phi})$.*

Proof. It is clear that $\ker(\hat{\phi_m}) \subset G$ since it is contained in $\ker(\phi_m \circ \theta \circ \hat{\phi_m})$ and in $E_m[D]$ as well. We now show that G is cyclic. Let M be the largest divisor of D such that $E_m[M] \subset G$. Then ϕ_m can be decomposed as $\phi_{D/M} \circ \phi_M$. Then by [Pet17, Lemma 5] the kernel of ϕ_M is fixed by θ. In the proof of [Pet17, Lemma 6] it is shown that a subgroup of $E_{j_T}[M]$ can only be fixed by an endomorphism θ if $\mathrm{Tr}(\theta)^2 - 4\deg(\theta) = \mathrm{Disc}\ \mathbb{Z}[\theta] = \Delta$ is a square modulo M. Thus, the quadratic residuosity condition on Δ 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 $\ker(\hat{\phi_m})$ which is a group of order D. This implies that $G = \ker(\hat{\phi_m})$. □

The group $G = \ker(\hat{\phi})$ can be computed by solving a double discrete logarithm problem, which is efficient as D is smooth. We summarize the steps needed for inverting the one-way function in Algorithm 1.

In [Que+21] it is shown that Algorithm 1 runs in polynomial time whenever $E_m[D]$ is efficiently representable and $\Delta = \mathrm{Disc}\ \mathbb{Z}[\theta]$ is as in Lemma 4.3.4.

Algorithm 1 Computing inverses

Require: $j_T \in \mathcal{J}_{T,p}$, a trapdoor T and c.
Ensure: $m \in [\mu(D)]$ such that $f_{j_T}^{D,N}(m) = \mathsf{c}$.

1: Parse c as $(j_m, P_m, Q_m) \in \mathbb{F}_{p^2} \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2$.
2: Parse T as $e, d, P_{J_T}, Q_{j_T}, \theta(P_{j_T}), \theta(Q_{j_T})$.
3: Compute the canonical curve E_m having j-invariant j_m.
4: Let $\tau = \phi_m \circ \theta \circ \hat{\phi}_m + [d] \in \mathrm{End}(E_m)$. ▷ Choices of θ and d ensure $\deg \tau = N^2 e$.
5: Compute τ as described in the proof of Theorem 4.2.7.
6: Compute $\ker(\phi_m \circ \theta \circ \hat{\phi}_m) \cap E_m[D] = \ker(\tau - [d]) \cap E_m[D] = \ker(\hat{\phi_m})$.
7: Compute $\ker(\phi_m)$ using $\ker(\hat{\phi}_m)$.
8: **return** $m \in [\mu(D)]$ that corresponds to $\ker(\phi_m)$.

4.3.4 – Séta Public Key Encryption. We now build Séta, a Public Key Encryption scheme using the trapdoor one-way function family of Section 4.3.2, and we show that it is OW-CPA secure. Concretely, we define the Séta PKE scheme as the tuple $(\mathsf{Key\ Generation}, \mathsf{Encryption}, \mathsf{Decryption})$ of PPT algorithms described below.

Parameters. Let λ denote the security parameter. Let p be a prime such that $p^2 - 1 = DNf$ where D, N are smooth integers and f is a small co-factor such that $2^{2\lambda} < D$, $D^2 < N$. We let $\mathsf{params} = (\lambda, p, D, N)$.

Key generation. The $\mathsf{Key\ Generation}(\mathsf{params})$ algorithm proceeds as follows:

1. Compute a uniformly random (D, N)-trapdoor supersingular elliptic curve (E_{j_T}, T) defined over $\mathbb{F}_{p^2}$ using Algorithms 2 and 3 (see Section 4.4).
2. Set $\mathsf{pk} \coloneqq (j_T)$ and $\mathsf{sk} \coloneqq T$.
3. Return $(\mathsf{pk}, \mathsf{sk})$.

Encryption. The $\mathsf{Encryption}(\mathsf{params}, \mathsf{pk}, m)$ algorithm proceeds as follows. For a given $m \in \{0,1\}^{n_m}$, where $n_m = \lfloor \log_2 \mu(D) \rfloor$, first cast m as an integer in the set $[\mu(D)]$ and then:

1. Parse $\mathsf{pk} = j_T \in \mathcal{J}_{T,p}$.
2. Compute $(j_m, P_m, Q_m) \leftarrow f_{j_T}^{D,N}(m)$.
3. Return $\mathsf{c} = (j_m, P_m, Q_m)$.

Decryption. The $\mathsf{Decryption}(\mathsf{params}, \mathsf{pk}, \mathsf{sk}, \mathsf{c})$ algorithm proceeds as follows:

1. Given $\mathsf{params}, \mathsf{sk}$ and c, parse c as $(j_c, P_c, Q_c) \in \mathbb{F}_{p^2} \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2$; if that fails, return $\perp$.
2. Follow Algorithm 1 to recover $\tilde{m} \in [\mu(D)]$; if this fails, set $\tilde{m} = \perp$.
3. If $\perp$ was recovered, return $\perp$.

4. Otherwise, from $\tilde{m} \in [\mu(D)]$, recover $m \in \{0,1\}^{n_m}$ and return it.

Theorem 4.3.5. *Let p be a prime, let D and N be integers such that $D^2 < N$. Suppose that the output distribution of Algorithm 3 is statistically close to uniform. Let E_{j_T} be an output of Algorithm 3. If Problem 4.3.2 with p, D, N, E_{j_T} and X such that $H_\infty(X) = \lambda$ is hard for quantum PPT adversaries, then the PKE scheme above is one-way chosen-plaintext attack (OW-CPA) post-quantum secure.*

Proof. Let $\mathcal{M} = \{0,1\}^{n_m}$ denote the message space of the encryption scheme, with $n_m = O(\lambda)$. We see that a randomly sampled $m \xleftarrow{\$} \mathcal{M}$ directly embedded as an integer $m \in [\mu(D)]$ yields a distribution Y with min-entropy $H_\infty(Y) \geq \lambda$ on isogenies of degree D starting from E_{j_T}. The challenge of opening a given ciphertext c then reduces to recovering the secret isogeny of Problem 4.3.2 with $X = Y$.

□

4.3.5 – IND-CCA encryption scheme. We obtain an IND-CCA secure PKE scheme by applying the generic post-quantum OAEP transformation [TU16, Section 5] (see Appendix 4.2.6) to Séta, for which we prove that our function $f_{j_T}^{D,N}$ is quantum partial-domain one-way.

Definition 4.3.6. *Let k_1, k_0 and n_c be integers. A family $\mathcal{F}$ of functions $f : \{0,1\}^{\lambda+k_1} \times \{0,1\}^{k_2} \to \{0,1\}^{n_c}$ is* partial domain one-way *if for any polynomial time adversary $\mathcal{A}$, the following advantage is negligible in λ:*

$$\mathrm{Adv}_\lambda(\mathcal{A}) = \Pr\left[s' = s; s' \leftarrow \mathcal{A}(1^\lambda, y), y \leftarrow f(s,t), (s,t) \xleftarrow{\$} A \times B, f \leftarrow \mathcal{F}\right]$$

Lemma 4.3.7. *Let j_T be a uniformly random element of $\mathcal{J}_{T,p}$. The function $f_{j_T}^{D,N}$ defined in Section 4.3.2 is a quantum partial-domain one-way function, under the hardness of Problem 4.3.2.*

Proof. We note that in our case, partial domain inversion is the same as domain inversion where only the first part of the path is required. More precisely, factor D as $D_1 \cdot D_2$ such that $\gcd(D_1, D_2) = 1$, $2^{\lambda+k_1} \leq \mu(D_1)$ and $2^{k_0} \leq \mu(D_2)$ (where $\lambda + k_0 + k_1$ is the bit-length of input strings) and then embed each of s and t into $\mu(D_1)$ and $\mu(D_2)$ respectively. Then we can set $f_{j_T}^{D,N}(s,t) := f_{j_1}^{D_2,N}(t)$ where $(j_1, P_1, Q_1) = f_{j_T}^{D_1,N}(s)$ and $f_{j_1}^{D_2,N}$ uses $\{P_1, Q_1\}$ as basis of $E_{j_1}[N]$. Since $2^{\lambda+k_1} \leq \mu(D_1)$, then recovering s from $y = f_{j_T}^{D,N}(s,t)$ is hard under the same assumption as Theorem 4.3.5 with D replaced by D_1.

□

Theorem 4.3.8 ([TU16], Theorem 2)**.** *If $f_{j_T}^{D,N}$ is a quantum partial-domain one-way function, then the OAEP-transformed scheme is IND-CCA secure in the quantum random oracle model (QROM).*

Algorithm 2 Computing the integers d, e

Require: D, N, p as above. Let S be the product of primes dividing D.
Ensure: (d, e) such that $-\frac{N^2e-d^2}{D^2} < 0$ is a quadratic non-residue modulo every prime dividing D and is a quadratic non-residue modulo p.

1: Set $e = 1$.
2: Find u such that $u^2 \equiv N^2 e \pmod{D^2}$.
3: **for** every prime ℓ_i dividing D **do**
4: Let s_{ℓ_i} be a quadratic non-residue modulo ℓ_i.
5: $r_i \leftarrow (s_{\ell_i} - \frac{-N^2e+u^2}{D^2})(2u)^{-1} \pmod{\ell_i}$.
6: **end for**
7: Compute a residue r modulo S with the property that $r \equiv r_i \pmod{\ell_i}$.
8: $\ell \leftarrow 0$.
9: $d \leftarrow D^2(S\ell + r) + u$.
10: $A \leftarrow \frac{N^2e-d^2}{D^2}$.
11: **if** $A < 0$ **then**
12: **return** $\perp$
13: **end if**
14: **if** A is not a square modulo p **then**
15: $\ell \leftarrow \ell + 1$.
16: **go to** Step 9.
17: **end if**
18: **return** (d, e)

4.4 — Key generation variants

In this section we describe various methods for generating keys for Séta. We first describe Algorithm 2, which can generate integers d, e so that $\Delta = \text{Disc } \mathfrak{O}$, where $\mathfrak{O} = \mathbb{Z}[\sqrt{(N^2e - d^2)/D^2}]$, satisfies the quadratic residuosity conditions imposed Section 4.3.3. Then, we present two options for generating a uniformly random supersingular elliptic curve inside $\mathcal{E}_{\mathfrak{O}}$ together with the remaining part of the trapdoor information T. Algorithm 3 treats the generic case, and Algorithm 4 focuses on computing a (DD_s, N)-trapdoor curve from a (D, N)-trapdoor curve and a random walk of degree D_s.

4.4.1 – Computing the trapdoor information. We recall that the required condition is that $\Delta = \text{Disc } \mathfrak{O} = -4\frac{N^2e-d^2}{D^2}$ must be negative and a quadratic non-residue modulo every prime dividing D and also modulo p. For simplicity, we fix $e = 1$ and look for d of a special form. This is described in Algorithm 2.

Lemma 4.4.1. *If d, e is the output of Algorithm 2, then $\frac{N^2e-d^2}{D^2}$ is a quadratic non-residue modulo all ℓ_i.*

Proof. Let r_i, s_{ℓ_i}, T and u be as in Algorithm 2. Let r be an integer such that $r \equiv r_i \pmod{\ell_i}$. Then we show that for every i, the integer $\frac{-N^2e+(D^2r+u)^2}{D^2}$ is not a quadratic residue modulo ℓ_i which implies that $-\frac{N^2e-d^2}{D^2}$ 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

$$\frac{-N^2e + (D^2r + u)^2}{D^2} = \frac{-N^2e + u^2}{D^2} + D^2r^2 + 2ur.$$

By our choice of r we have that

$$\frac{-N^2e + u^2}{D^2} + D^2r^2 + 2ur \equiv \frac{-N^2e + u^2}{D^2} + 2ur_i \equiv s_{\ell_i} \pmod{\ell_i},$$

which is a quadratic nonresidue by the choice of s_{ℓ_i}.

□

Lemma 4.4.2. *Let S be the product of all primes dividing D. If $N > D^2S$, then Algorithm 2 returns a correct pair (d, e) with probability higher than $1 - 2^{-\frac{N}{SD^2}+1}$ under plausible heuristic assumption.*

Proof. Since u is found by solving an equation modulo D^2, we obtain $u < D^2$. Similarly we have $r < S$. Under plausible heuristic assumptions, we can estimate to $1/2$ the probability that the quadratic reduosity condition on A is satisfied. Thus, we obtain a bound on the failure probability by counting how many values ℓ can be tried before A becomes negative. With the conservative bound that $D^2r + u \approx D^2S$, we obtain that we can try $\frac{N - D^2S}{DS^2}$ different values for small d, which gives the result.

Correctness of the result follows from Lemma 4.4.1. □

4.4.2 – Trapdoor curve generation. Now we focus on generating a random supersingular elliptic curve whose endomorphism ring contains an embedding of $\mathfrak{O} = \mathbb{Z}[\sqrt{(N^2e - d^2)/D^2}$ for d, e outputs of Algorithm 2. In [Que+21, Section 5.1] it is discussed how one can generate a specific curve inside $\mathcal{E}_{\mathfrak{O}}$. Essentially, this is achieved by computing a maximal order $\mathcal{O}$ containing the suborder $\mathfrak{O}$ (with [Voi13, Algorithm 7.9]) and then computing a supersingular elliptic curve whose endomorphism ring is isomorphic to $\mathfrak{O}$ (with [Eis+18, Algorithm 12]). This procedure can be made concretely efficient with the algorithms from [De +20] under some conditions on the prime p that partly underlie the choice of prime described in Section 4.6.2. However, this procedure is essentially deterministic, so an adversary knowing the quadratic order $\mathfrak{O}$ can just recompute the same trapdoor curve. The point of this subsection is to show how to randomize the procedure.

We obtain randomization by first generating a curve with the deterministic procedure and then applying the action of a random class group element to derive another random curve with the same embedding. This operation would be costly if it required to compute a lot of isogenies. However, we can do it over the quaternions at a negligible cost before applying the translation algorithm from maximal orders to elliptic curves.

For concrete randomization, we use the fact (see [JMV09]) that there exists a bound B (polynomial in p) for which the graph whose vertices are curves in $\mathcal{E}_{\mathfrak{O}}$ and edges are isogenies of prime degree smaller than B is an expander graph. The fast mixing property of expander graphs implies that the distribution of curves obtained after a random walk of fixed length quickly converges to the uniform distribution as the length of the walk grows. More precisely, for any δ we can find a length ε

Algorithm 3 Generating the trapdoor curve from a quadratic order $\mathfrak{O}$

Require: A prime p, an integer N, a quadratic order $\mathfrak{O}$, a bound B, a length ε.
Ensure: A uniformly random curve $E_{j_T} \in \mathcal{E}_{\mathfrak{O}}$, a basis P_{j_T}, Q_{j_T} of $E_{j_T}[N]$, and $\theta(P_{j_T}), \theta(Q_{j_T})$ with $\theta \in \mathrm{End}(E_{j_T})$ such that $\mathbb{Z}[\theta] \cong \mathfrak{O}$.

Find a max. order $\mathcal{O} \subset \mathcal{B}_{p,\infty}$ with $\mathfrak{O}$ embedded in $\mathcal{O}$ with the alg. from [Que+21].
Compute $\ell_1, \dots, \ell_n$ the n primes split in $\mathfrak{O}$ smaller than B.
Select a random vector $(\varepsilon_1, \dots, \varepsilon_n)$ in $\mathbb{Z}^n$ with L_1 norm equal to ε.
Set $\mathcal{O}_{j_T} = \mathcal{O}$.
for $1 \leq i \leq n$ **do**
    Compute $\alpha_i \in \mathfrak{O}$ such that $\mathfrak{l}_i = \mathfrak{O}\langle \alpha_i, \ell_i \rangle$ is a prime ideal above ℓ_i.
    for $1 \leq j \leq |\varepsilon_i|$ **do**
        Compute the ideal $I = \mathcal{O}_{j_T}\langle \alpha_i, \ell_i \rangle$.
        Set $\mathcal{O}_{j_T}$ as the right order of I.
    end for
end for
Compute the curve E_{j_T} from $\mathfrak{O}_{j_T}$ with [Eis+18, Algorithm 12].
Compute a canonical basis P_{j_T}, Q_{j_T} of $E_{j_T}[N]$.
Select the correct element $\theta \in \mathcal{O}_{j_T}$ such that $\mathfrak{O} \cong \mathbb{Z}[\theta]$.
Use the representation of $\mathcal{O}_{j_T}$ obtained from the execution of [Eis+18, Algorithm 12] to compute $\theta(P_{j_T}), \theta(Q_{j_T})$.
return $E_{j_T}, P_{j_T}, Q_{j_T}$ of $E_{j_T}[N], \theta(P_{j_T}), \theta(Q_{j_T})$.

(logarithmic in the size of the graph and δ) for which the statistical distance between the random walk distribution and the uniform distribution is less than δ. So once the length ε (corresponding to a sufficiently small δ) has been set, for any starting curve E_0 in $\mathcal{E}_{\mathfrak{O}}$ the curve $\prod_{i=1}^{n} \mathfrak{l}_i^{\varepsilon_i} \star E_0$ where $\mathfrak{l}_1, \dots, \mathfrak{l}_n$ are prime ideals above the n prime $\ell_1, \dots, \ell_n$ smaller than B that are split in $\mathfrak{O}$ and $(\varepsilon_1, \dots, \varepsilon_n)$ is uniformly random among the vectors in $\mathbb{Z}^n$ such that $\sum_{i=1}^{n} |\varepsilon_i| = \varepsilon$, is statistically close to a uniformly random element in $\mathcal{E}_{\mathfrak{O}}$. This result underlies Algorithm 3.

Proposition 4.4.3. *Algorithm 3 is correct and terminates in polynomial time.*

Proof. All the sub-algorithms run in polynomial-time and by choice of B and ε, the number of iterations in the loop is also polynomial.

It is easy to verify that the ideal I corresponds through the Deuring correspondence to the isogeny $\varphi_{\mathfrak{l}_i}$. Thus, our method simulates a random walk over the graph that we described at the beginning of this section. For the reasons explained there, the curve E_{j_T} obtained in the end is statistically close to a random element in $\mathcal{E}_{\mathfrak{O}}$. □

4.4.3 – Constraints on the prime. In Séta, we compute and evaluate isogenies of degree D and N. Hence we always require that D and N are smooth and that the DN-torsion groups are efficiently representable, i.e., that they are defined on extensions of $\mathbb{F}_{p^2}$ of small degree. For example, if we require that $E[DN] \subset E(\mathbb{F}_{p^4})$, then DN must divide $p^2 - 1$. The smoothness bound B_1 of D impacts the efficiency of encryption and the smoothness bound B_2 of N impacts the efficiency of decryption.

For a given security level λ, we require $2^{2\lambda} < D$ in order to protect the scheme against the meet-in-middle attack.

Since we have the range $D^2 < D^2S < D^3$ depending on the value of S (product of primes dividing D), and that Lemma 4.4.2 implies that $N > D^2S$ then we can estimate that the value DN will be between $2^{6\lambda}$ and $2^{8\lambda}$. If we want DN dividing p^2-1, we can estimate that the minimum size for the prime p will be between 3λ and 4λ bits. The actual size will depend on the size of $(p^2-1)/DN$.

Besides encryption and decryption, key generation also restricts the types of primes to be used in Séta. Indeed, Step 12 and Step 15 of Algorithm 3 use [Eis+18, Algorithm 12], which in turn uses the KLTP Algorithm [KLPT14]. Although this algorithm runs in polynomial time, it is not practical in general; the variant introduced in [De +20] achieves much greater efficiency, provided that p^2-1 is of the form $p^2-1 = \ell^f N_2 f_2$, where ℓ is a small prime, $N_2 > p^{3/2}$ is a smooth integer co-prime to ℓ and f_2 is a cofactor. We refer to [De +20, §8] for more details; a concrete method to select Séta-friendly primes is described in Section 4.6.2.

4.4.4 – Alternative key generation. We describe an alternative method for computing trapdoor curves and suggest a variant of the key generation algorithm for Séta. The main idea is to perform a random secret walk from a publicly available trapdoor curve. The method relies on the following proposition.

Proposition 4.4.4. *Let p be a prime, let D_s, D and N be three smooth integers. Let (E_{j_T}, T) where $T = (\theta(P_{j_T}), \theta(Q_{j_T}), d, e)$ be a (D_sD, N)-trapdoor curve. Let $\phi_s : E_{j_T} \to E_s$ be an isogeny of degree D_s. Set $T' = (\theta'(P_s), \theta'(Q_s), d, e)$ where $\theta' = \phi_s \circ \theta \circ \widehat{\phi_s}$ and $\{P_s, Q_s\}$ is a canonical basis of $E_s[N]$. Then (E_s, T') is a (D, N)-trapdoor curve.*

Proof. Since we know the action of θ on the torsion group $E_{j_T}[N]$ and ϕ_s, then we can efficiently evaluate $\theta' = \phi_s \circ \theta \circ \widehat{\phi_s}$ on $E_s[N]$. Since (E_{j_T}, T) is a (D_sD, N)-trapdoor curve, then $\mathrm{Tr}(\theta) = 0$ and $\widehat{\theta} = -\theta$. Hence

$$\mathrm{Tr}(\theta') = \phi_s \circ \theta \circ \widehat{\phi_s} + \widehat{\phi_s \circ \theta \circ \widehat{\phi_s}} = \phi_s \circ \theta \circ \widehat{\phi_s} - \phi_s \circ \theta \circ \widehat{\phi_s} = 0.$$

It follows that

$$\deg([D]\theta' + [d]) = D^2 \deg(\theta') + d^2 = D^2 D_s^2 \deg(\theta) + d^2 = N^2 e.$$

By Theorem 4.2.10, (E_s, T') is a (D, N)-trapdoor curve.

□

Relying on Proposition 4.4.4, Algorithm 4 computes (D, N)-trapdoor curves when given a (D_sD, N)-trapdoor curve.

Lemma 4.4.5. *Algorithm 4 is correct and runs in polynomial time.*

Proof. The correctness of Algorithm 4 follows from Proposition 4.4.4. Step 1 of Algorithm 4 consists of a degree D_s isogeny computation. Since D_s is smooth, then Step 1 runs in polynomial time. Step 2 consists of an evaluation of $\phi_s \circ \theta \circ \widehat{\phi_s}$ on P_s and Q_s. One evaluate $\widehat{\phi_s}(P_s)$ and express it as a linear combination of P_{j_T} and Q_j to recover $\theta\left(\widehat{\phi_s}(P_s)\right)$, then on evaluates $\phi_s\left(\theta\left(\widehat{\phi_s}(P_s)\right)\right)$. Similarly, one evaluates

Algorithm 4 Computing a (D, N)-trapdoor curve from a $(D_s D, N)$-trapdoor curve where $D_s \approx 2^{2\lambda}$ is a smooth integer

Require: a $(D_s D, N)$-trapdoor curve (E_{j_T}, T) where $T = (\theta(P_{j_T}), \theta(Q_{j_T}), d, e)$.
Ensure: a (D, N)-trapdoor curve (E_s, T').
1: Sample a uniformly random isogeny $\phi_s : E_{\theta,j} \to E_s$ of degree D_s .
2: Compute $T' = (\theta'(P_s), \theta'(Q_s), d, e)$ where $\theta' = \phi_s \circ \theta \circ \widehat{\phi_s}$ and $\{P_s, Q_s\}$ is a canonical basis of $E_s[N]$..
3: **return** (E_s, T')

$\phi_s\left(\theta\left(\widehat{\phi_s}(Q_s)\right)\right)$. All these steps run in polynomial time since D_s and N are smooth integers. □

A variant of the Séta setup and key generation is described as follows.

Parameters. Let λ denote the security parameter. Let p be a prime such that $p^2 - 1 = D_s D N f$ where D_s, D, N are smooth integers and f is a small co-factor such that $2^{2\lambda} < D \approx D_s$, $D_s^2 D^2 < N$. Compute a $(D_s D, N)$-trapdoor curve (E_{j_T}, T) using Algorithm 3. We let $\mathsf{params} = (\lambda, p, D_s, D, N, E_{j_T}, T)$.

Key generation. The Key Generation(params) algorithm proceeds as follows:

1. Compute a random (D, N)-trapdoor curve (E_s, T') using Algorithm 4 with (E_{j_T}, T) as input.
2. Set $\mathsf{pk} \coloneqq (j_s)$ and $\mathsf{sk} \coloneqq T'$.
3. Return $(\mathsf{pk}, \mathsf{sk})$.

The advantage of this variant is the fact the key generation algorithm does not use Algorithm 3, hence most of the requirements on p enumerated in Section 4.4.3 can be relaxed. This implies having more freedom in the choice of D and N, for which we could opt for powers of very small primes. Mostly, less good SQISign primes would be admissible for this variant, which is not the case in the original Séta described in Section 4.3.4, since its key generation uses Algorithm 3 which requires good Séta primes in order to be practically efficient. This variant is hence a good alternative to the Séta key generation, given the fruitless search of good cryptographic size SQI-Sign primes.

On the other hand, using less good SQISign primes implies that generating the $(D_s D, N)$-trapdoor curve (E_{j_T}, T) in the parameters generation is less efficient. But since this parameter generation is run once and for all, then this does not constitute a considerable drawback.

The main drawback of this key generation method is the considerably large size of the base prime p. In fact, p needs to satisfy $p^2 - 1 = D_s D N f$ where f is a small co-factor, and $D_s \approx D \approx 2^{2\lambda}$ such that attacking the isogeny $\phi_s : E_{j_T} \to E_s$ or $\phi_m : E_s \to E_m$ are equivalent with respect to the meet in the middle attack. Considering the fact that $N > (D_s D)^2$, then $N > 2^{8\lambda}$ and $2^{12\lambda} < D_s D N \leq p^2 - 1$, as opposed to $2^{6\lambda} < N D < p^2 - 1$ in Séta (see Section 4.4.3). It follows that the bit size of $p^2 - 1$ practically doubles when we use Algorithm 4 for key generation.

4.5 — "Uber" isogeny assumption

In this section, we introduce a generic framework, which we label *Uber Isogeny assumption* in analogy to [Boy08], aiming at generalizing isogeny computation problems encountered in the main families of isogeny-based schemes such as SIDH [JD11], CSIDH [Cas+18], OSIDH [CK20] and Séta (presented in this work).

The uber isogeny problem does not directly underlie the security of these various schemes (in the sense that no formal reduction is yet known). However, for each of these protocols there exists a set of parameters for which if one can solve the uber isogeny problem, then one can break the scheme. At a higher-level, our new problem can be seen as a generic key recovery problem.

By introducing this new assumption our goal is twofold. First, we highlight the proximity between the various isogeny schemes and we provide a common target for cryptanalysis. Second, the generic attack that we describe in Section 4.5.3 gives a lower-bound on the security of any future scheme whose security may be related to our uber assumption in a similar manner as SIDH, CSIDH, OSIDH and Séta.

4.5.1 – The new generic problem. The principal mathematical structure behind the uber isogeny problem is the group action at the heart of the CSIDH protocol and all the following works. In the isogeny setting, these group actions emerge through class groups of quadratic orders. The main definitions and properties were introduced in Section 4.2.3.

Problem 4.5.1 ($\mathfrak{O}$-Uber Isogeny Problem ($\mathfrak{O}-UIP$))**.** *Let $p > 3$ be a prime and let $\mathfrak{O}$ be a quadratic order of discriminant Δ. Given $E_0, E_s \in \mathcal{E}_{\mathfrak{O}}$ and an explicit embedding of $\mathfrak{O}$ into* $\mathrm{End}(E_0)$ *(i.e the knowledge of $\alpha_0 \in \mathrm{End}(E_0)$ such that $\mathbb{Z}[\alpha_0] \cong \mathfrak{O}$), find a powersmooth ideal $\mathfrak{a}$ of norm coprime with Δ such that $[\mathfrak{a}] \in \mathrm{Cl}(\mathfrak{O})$ is such that $E_s \cong \mathfrak{a} * E_0$.*

Remark 4.5.2. *In Problem 4.5.1, the powersmoothness condition on the norm is to ensure that the resulting isogeny can always be computed in polynomial time. In some special cases where the form of the prime p enables to compute some smooth isogenies in polynomial time, this condition might be relaxed a little bit.*

4.5.2 – Relation with various isogeny-based constructions. We start with the link with CSIDH [Cas+18] which is quite obvious. We state the CSIDH key recovery problem below [Cas+18, Problem 10].

Problem 4.5.3. *Given two supersingular elliptic curves E, E_0 defined over F_p with the same F_p-rational endomorphism ring $\mathfrak{O}$, find an ideal $\mathfrak{a}$ of $\mathfrak{O}$ such that $[\mathfrak{a}] \star E = E_0$. This ideal must be represented in such a way that the action of $\mathfrak{a}$ on any curve can be evaluated efficiently, for instance a could be given as a product of ideals of small norm.*

Proposition 4.5.4. *When $p = 3 \bmod 4$ and $\Delta = -4p$, Problem 4.5.1 is equivalent to the CSIDH key recovery Problem 4.5.3.*

Proof. In the case of CSIDH, the curves admitting an embedding of $\mathbb{Z}[\sqrt{-p}] \cong \mathbb{Z}[\pi]$ in their endomorphism rings are the curves defined over $\mathbb{F}_p$ (i.e left stable by π the

Frobenius morphism). Then, it is quite clear that Problem 4.5.1 is equivalent to Problem 4.5.3. □

The OSIDH protocol [CK20] is a generalization of CSIDH where $\mathbb{Z}[\pi]$ is replaced by a larger class of quadratic orders. The link between OSIDH and Problem 4.5.1 is also straightforward. Let us fix some notations[3] for this protocol and briefly recall the principle. The OSIDH key exchange protocol starts from a descending chain of ℓ-isogenies of size n that we write $\varphi_0 : F_0 \to E_0$ where F_0 admits a $\mathfrak{O}_0$-orientation (i.e an embedding of $\mathfrak{O}_0$ inside $\mathrm{End}(E_0)$). From there, φ_0 induces an $\mathfrak{O}$-orientation on E_0. The secret keys of Alice and Bob are $\mathfrak{O}$-ideals $\mathfrak{a}, \mathfrak{b}$ whose action on E_0 will lead to curves $E_A = \mathfrak{a} * E_0$ and $E_B = \mathfrak{b} * E_0$. These curves have also a $\mathfrak{O}$-orientation which implies the existence of ℓ^n-isogenies $\varphi_A : F_0 \to E_A$ and $\varphi_B : F_0 \to E_B$ as in Proposition 4.2.2. Alice public key will be E_A together with some torsion points (which will allow Bob to compute $\mathfrak{b} \star E_A$).

Proposition 4.5.5. *When $\mathfrak{O}_0$ is a quadratic order of class number 1 and $\mathfrak{O} = \mathbb{Z} + \ell^n\mathfrak{O}_0$, then if there exists a PPT algorithm that can break Problem 4.5.1, there is a PPT algorithm that can recover the keys of the OSIDH protocol presented in [CK20].*

Proof. From the definition of the group action of $\mathrm{Cl}(\mathfrak{O})$ on the curves having an $\mathfrak{O}$-orientation (see [CK20]), finding a smooth ideal $\mathfrak{c}$ such that $E_A = \mathfrak{c} * E_0$ is enough to recover the secret key. □

Note that we do not have equivalence in Proposition 4.5.5 because the OSIDH public keys include more information than just curves. This will be the same for SIDH and Proposition 4.5.7.

For SIDH, we write[4] F_0 for the common starting curve. In SIDH, recovering the secret key from the public key is equivalent to the computational supersingular isogeny problem (CSSI), see [JD11] that we state in Problem 4.5.6.

Problem 4.5.6. *Let ℓ_A be a small prime number and $A = \ell_A^{e_A}$ for some exponent e_A. Let $\varphi_A : F_0 \to E_A$ be an isogeny whose kernel is $\langle [m_A]P_A + [n_A]Q_A \rangle$, where m_A and n_A are chosen at random from $\mathbb{Z}/A\mathbb{Z}$ (where at least one is in $\mathbb{Z}/A\mathbb{Z}^\times$. Given E_A and the values $\varphi_A(P_B), \varphi_A(Q_B)$ for P, B, Q_B a basis of $F_0[B]$ find a generator R_A of* $\ker \varphi_A$.

The proposition below requires a bit more work as the link between SIDH and group actions is less obvious.

Proposition 4.5.7. *Assume that F_0 admits an $\mathfrak{O}_0$-orientation with $\mathfrak{O}_0$ a maximal quadratic order of class number 1. If there exists a PPT algorithm solving Problem 4.5.1 for $\mathfrak{O} = \mathbb{Z} + A'\mathfrak{O}_0$ where A' divides A, then there exists a PPT algorithm that breaks the CSSI problem with overwhelming probability.*

[3] These notations do not exactly agree with the ones introduced in [CK20] because we want to hightlight the link with our $\mathfrak{O}$-IOP.

[4] Once again, we highlight that these notations are unusual and were chosen to emphasize the link with Problem 4.5.1.

Proof. First, note that A is chosen so that the kernel points of A-isogenies have a polynomial-size representation. Then, since A is also smooth, the discrete logarithms can be solved in polynomial time in the A-torsion and isogenies of degree A can be computed in polynomial time.

For the rest of this proof, let us write α the endomorphism of F_0 such that $\mathbb{Z}[\alpha]$ realizes the embedding of $\mathfrak{O}_0$ inside $\mathrm{End}(F_0)$.

If the curve E_A is A-isogenous to F_0, then E_A admits an embedding of $\mathbb{Z} + A\mathfrak{O}_0$. This embedding is not necessarily primitive but we know there exists A' dividing A such that $\mathfrak{O} = \mathbb{Z} + A'\mathfrak{O}_0$ admits a primitive embedding in $\mathrm{End}(E_A)$ (see Proposition 4.2.2). Conversely, since the class number of $\mathfrak{O}_0$ is 1, then any $\mathbb{Z} + A'\mathfrak{O}_0$-orientation on E_A implies the existence of an A'-isogeny between E_A and F_0. Let us write $\varphi_{A'} : F_0 \to E_A$ this isogeny of degree A'. Then φ_A, the secret isogeny in Problem 4.5.6 is the composition of φ_A with an endomorphism θ_A of $\mathfrak{O}_0$ of degree A/A'. Since A/A' is a power of ℓ_A, there are two possibilities for θ_A. Thus, the difficulty lies in recovering $\varphi_{A'}$.

We can generate a curve E_0 in $\mathcal{E}_{\mathbb{Z}+A'\mathfrak{O}_0}$ by generating $\varphi_0 : F_0 \to E_0$ a descending isogeny of degree A'. Any ideal $\mathfrak{a}$ such that $E_A = \mathfrak{a} * E_0$ can be interpreted as an isogeny $\varphi_\mathfrak{a} : E_0 \to E_A$ of degree $n(\mathfrak{a})$. The proof is concluded by the fact that $\ker \hat{\varphi}_{A'} = \varphi_\mathfrak{a}(\ker \hat{\varphi}_0)$, which we prove below. Once $\ker \hat{\varphi}_{A'}$ has been computed, is easy to recover $\ker \varphi_{A'} = \hat{\varphi}_{A'}(E_A[A'])$ and find a solution to the CSSI as we explained above.

To prove $\ker \hat{\varphi}_{A'} = \varphi_\mathfrak{a}(\ker \hat{\varphi}_0)$, we need to understand how the fact that $\mathfrak{a}$ is an $\mathfrak{O}$-ideal translates on the action of $\varphi_\mathfrak{a}$ on $\hat{\varphi}_0$. As explained in Proposition 4.2.2 and the following paragraph, the embedding of $\mathfrak{O}$ in E_0 (resp. E_A) is obtained as $\mathbb{Z}[\varphi_0 \circ \alpha \circ \hat{\varphi}_0] = \mathbb{Z}[\theta_0]$ (resp. $\mathbb{Z}[\varphi_{A'} \circ \alpha \circ \hat{\varphi}_{A'}] = \mathbb{Z}[\theta_{A'}]$). By definition of $\mathfrak{a}$ being an $\mathfrak{O}$-ideal, we have that $\varphi_\mathfrak{a}(\ker \theta_0) = \ker \theta_A$. Thus, we need to prove that $\ker \theta_0 \cap E_0[A'] = \ker \hat{\varphi}_0$ and $\ker \theta_{A'} \cap E_A[A'] = \ker \hat{\varphi}_A$ (note that this property is exactly what underlies the inversion mechanism in Section 4.3.3). We will do it for θ_0, the property for $\theta_{A'}$ holds for the exact same reasons. It is clear from the definition of $\theta_0 = \varphi_0 \circ \alpha \circ \hat{\varphi}_0$ that we have $\ker \hat{\varphi}_0 \subset \ker \theta_0$. Let us take $P \in E_A[A'] \setminus \ker \hat{\varphi}_0$, then $Q = \hat{\varphi}_0(P) \in \ker \varphi_0 \setminus \langle 0 \rangle$. If we assume that $P \in \ker \theta_0$, it implies that $\alpha(Q) \in \ker \varphi_0$. Since $\ker \varphi_0$ is cyclic, we have that $\alpha(Q) = \lambda Q$ for some $\lambda \in \mathbb{Z}$. This contradicts the fact that φ_0 is descending. Indeed, if we write φ_Q, the isogeny of kernel generated by Q, we have $\varphi_0 = \psi_0 \circ \varphi_Q$ for some isogeny φ_Q and the condition $\alpha(Q) = \lambda Q$ implies that φ_Q is not descending and so φ_0 would not be descending, which is a contradiction. Thus, we have proven that $\ker \theta_0 \cap E_0[A'] = \ker \hat{\varphi}_0$ and this concludes the proof asexplained above. □

We refer to Section 4.3 for the full details and notations about Séta. We write $\mathfrak{O} \cong \mathbb{Z}[\sqrt{(N^2e - d^2)/D^2}] \cong \mathbb{Z}[\theta]$ and assume that $e, d, \mathfrak{O}$ are public. This assumption is plausible as the procedure described in Algorithm 2 is essentially deterministic.

Proposition 4.5.8. *If there exists a PPT algorithm solving Problem 4.5.1 for $\mathfrak{O}$, then there exists a PPT algorithm that takes a Séta public key E_s and recovers a trapdoor T such that E_{j_T}, T is a (D, N)-trapdoor curve.*

Proof. Let E_{j_T} be a Séta public key. By applying Algorithm 3 in $\mathfrak{O}$ and adding the integers e, d a (D, N)-trapdoor curve E_0, T_0 can be found in polynomial time with $E_0 \in \mathcal{E}_\mathfrak{O}$. Thus, we can apply the PPT solver for Problem 4.5.1 on E_0 and E_{j_T} to compute

an isogeny $\varphi_{\mathfrak{a}} : E_0 \to E_{j_T}$ corresponding to a $\mathfrak{O}$ ideal $\mathfrak{a}$. If we write $\theta_0 \in \mathrm{End}(E_0)$ and $\theta \in \mathrm{End}(E_{j_T})$ the endomorphisms such that $\mathfrak{O} \cong \mathbb{Z}[\theta_0] \cong \mathbb{Z}[\theta]$. Then, by definition of $\mathfrak{O}$-ideals, we have that $\theta \circ \varphi_{\mathfrak{a}} = \varphi_{\mathfrak{a}} \circ$. So if $T_0 = e, d, P_0, Q_0, \theta_0(P_0), \theta_0(Q_0)$, then $T = e, d\varphi_{\mathfrak{a}}(P_0), \varphi_{\mathfrak{a}}(Q_0), \varphi_{\mathfrak{a}}(\theta_0(P_0)), \varphi_{\mathfrak{a}}(\theta_0(Q_0))$ is such that E_{j_T}, T is a (D, N)-trapdoor curve. □

We finish this section by proving that some instances of Problem 4.5.1 are related to the more generic isogeny problem of finding a smooth isogeny between any two supersingular curves (Problem 4.5.9 below). For that it suffices to show that there exists some quadratic order that is embedded inside the endomorphism ring of any supersingular curve.

Problem 4.5.9. *Let $p > 3$, be a prime number. Given E_1,E_2 two distinct supersingular curves over $\mathbb{F}_{p^2}$. Find $\varphi : E_1 \to E_2$, an isogeny of powersmooth degree.*

Proposition 4.5.10. *There is an absolute constant $c > 0$ such that the following holds. Let $\mathfrak{O}$ be a quadratic order of conductor ℓ^e inside $\mathfrak{O}_0$ a maximal quadratic order, such that ℓ is inert in $\mathfrak{O}_0$, and $e \geq c \log_\ell(p)$. If there exists a PPT algorithm that can break Problem 4.5.1, then there is a PPT algorithm that breaks Problem 4.5.9.*

Proof. From the fact that the ℓ-isogeny graph is Ramanujan, and the rapid mixing of non-backtracking random walks in expander graphs [ABLS07], we deduce that for $e = \Omega(\log_\ell(p))$, there exists a non-backtracking path of degree ℓ^e between any two supersingular curves in the graph.

In particular, if E_0 is any $\mathfrak{O}_0$-orientable curve, there exists a cyclic isogeny of degree ℓ^e from E_0 to any other E, and since ℓ is inert in $\mathfrak{O}_0$, this isogeny must be a sequence of descending isogenies. This implies that any E is $\mathfrak{O}$-orientable. Thus, if we write E_1 and E_2, the two curves in the generic isogeny problem, then we can construct a middle curve E_0 with an explicit embedding of $\mathfrak{O}$, then use the PPT algorithm to find paths between E_0, E_1 and E_0, E_2, and finally concatenate the two paths to obtain a path between E_1 and E_2 of powersmooth degree. □

4.5.3 – Analysis of the uber isogeny assumption. In this section we investigate the complexity of solving Problem 4.5.1. We are going to see that there are various special cases leading to various complexities.

We start by giving a generic estimate which can be seen as the worst case complexity.

A first upper bound: exhaustive search. The simplest method to solve Problem 4.5.1 is to apply an exhaustive search, for instance by selecting a set of small primes ℓ_i all split in $\mathfrak{O}$ and trying all combinations of $\prod \mathfrak{l}_i^{e_i} \star E_0$ until one is isomorphic to E_s, where each $\mathfrak{l}_i$ is a prime ideal above ℓ_i. The expected running time of this algorithm is in $O(\#\mathcal{E}_{\mathfrak{O}})$. The best generic bound on the size of this set is given in Proposition 4.2.1.

The classical estimate $h(\mathfrak{O}) = \Theta(\sqrt{\Delta})$ gives a first upper-bound on the complexity to solve Problem 4.5.1. In particular, it shows that solving Problem 4.5.1 is easy when the discriminant Δ is small. However, when Δ grows, it is harder to estimate how this bound reflects on the actual complexity of the problem.

There are some special cases for which we can be a bit more precise than Proposition 4.2.1. For instance, when the discriminant are short, the following Theorem from Kaneko [Kan89] can be applied to derive a precise statement.

Theorem 4.5.11. *Take two distinct quadratic orders $\mathfrak{O}_1, \mathfrak{O}_2$ of discriminants Δ_1, Δ_2 embedded optimally in the same maximal order inside the quaternion algebra ramified exactly at p and ∞. If we have $\mathbb{Q}(\sqrt{\Delta_1}) \cong \mathbb{Q}(\sqrt{\Delta_2})$, then $\Delta_1\Delta_2 \geq p^2$.*

Applying Theorem 4.5.11 to the discriminants $\Delta \leq p$, we see that there cannot be two distinct embeddings of $\mathfrak{O}$ inside the same maximal order, thus proving that $\#\mathcal{E}_{\mathfrak{O}} = h(\mathfrak{O})$. Thus, in that case, we know that the exhaustive search method described above has asymptotic complexity $\Theta(\sqrt{\Delta})$.

Another example is given in the proof of Proposition 4.5.10, where we saw that there are some values of Δ for which we know that $\mathcal{E}_{\mathfrak{O}}$ is exactly the set of supersingular curves. More generally, the link between the conductor of $\mathfrak{O}$ and isogenies (Proposition 4.2.2) allows us to obtain some better estimates on the size of $\mathcal{E}_{\mathfrak{O}}$ by using the expander properties of isogeny graphs.

The case of CSIDH. (Proposition 4.5.4) has received a lot of attention from the community ([Cas+18; BS20; Pei20; CCJR20] since it was the first scheme that naturally fits into this framework. In fact, there are improvements over the exhaustive search strategy in both the classical and quantum settings. The main ingredient behind these speed-ups is the ability for anyone to obtain a concrete embedding (through the Frobenius morphism) of $\mathfrak{O} = \mathbb{Z}[\sqrt{-p}]$ inside $\mathrm{End}(E)$ for any $E \in \mathcal{E}_{\mathfrak{O}}$. In particular, computing $\mathfrak{a} \star E$ becomes easy for any $E \in \mathcal{E}_{\mathfrak{O}}$ when $\mathfrak{a}$ has smooth norm. In the classical setting, this implies a quadratic speed-up over the generic exhaustive search by using a meet-in-the-middle technique (see [Cas+18]). In the quantum setting, the speed-up is even more radical, as it creates a malleability oracle (see [KMPW21]) that reduces CSIDH's security to an instance of the hidden shift problem which can be solved in quantum sub-exponential time as described in [Pei20; BS20] for instance.

Note that neither of these attacks can be used in the generic case as it seems hard to obtain this malleability oracle for other group actions. For instance, in OSIDH [CK20] the public keys are made of a curve E and some torsion points to make possible the computation of $\mathfrak{a} \star E$ for some secret ideal $\mathfrak{a}$. These additional torsion points are not needed in CSIDH because they can be easily computed.

Smooth conductor inside a maximal quadratic order. A better algorithm also exists when the conductor f of $\mathfrak{O}$ is smooth. By Proposition 4.2.2, there exists an isogeny of degree f between any curve $E \in \mathcal{E}_{\mathfrak{O}}$ and any curve in $\mathcal{E}_{\mathfrak{O}_0}$, where $\mathfrak{O}_0$ is the quadratic maximal order containing $\mathfrak{O}$. Let E_0, E_s given by in an instance of Problem 4.5.1, and let us write $\varphi_0 : F_0 \to E_0$ and $\varphi_s : F_s \to E_s$ the two isogenies of degree f.

The alternative resolution method enumerates through all possible $F_s = \mathfrak{a}_0 \star F_0$ in $\mathcal{E}_{\mathfrak{O}_0}$ then tries to find φ_s of degree f. Since f is smooth, we can apply a meet-in-the-middle technique to reduce this part to $O(\sqrt{f})$. Once $\varphi_s : F_s \to E_s$ and a $\mathfrak{O}_0$-ideal $\mathfrak{a}_0$ such that $F_s = \mathfrak{a}_0 \star F_0$ has been found, we can compute a $\mathfrak{O}$-ideal such that $E_s = \mathfrak{a} \star E_0$ as described in [CK20, Section 5.1].

If we write $\Delta = f^2\Delta_0$ where Δ_0 is the fundamental discriminant of $\mathfrak{O}_0$. The complexity of this algorithm is $\Theta(\sqrt{f}\sqrt{\Delta_0})$ which is better than $\Theta(\sqrt{\Delta}) = \Theta(f\sqrt{\Delta_0})$.

Other cases. When we are not in one of the above cases, there is no known improvement over the exhaustive search (classically or quantumly). Thus, the presumed security entirely relies on the size of $\mathcal{E}_{\mathfrak{O}}$. In that regard, the cases where the conductor of $\mathfrak{O}$ is big might give more confidence in the difficulty of Problem 4.5.1 as the size of $\mathcal{E}_{\mathfrak{O}}$ is tied to the number of isogenies of a given degree between distinct pair of curves. In comparison, the distribution of embeddings of a maximal quadratic order of big discriminant (i.e above the bound in Theorem 4.5.11) have been less studied. As of yet, there are no reason to believe that there exists such quadratic orders that would be embedded in only a small portion of all the supersingular curves but not enough work has been done on the question to reach a definitive conclusion.

4.6 — Implementation

We implemented the version of Séta where the starting curve (E_{j_T}, T) is a (D, N)-trapdoor curve, i.e, the secret key does not contain a random walk, as described in Section 4.4.2. Our implementation is written in pure C, reusing large parts of the codebase of SQISign[5]; in particular we depend on GMP 6.2.1 for integer arithmetic, Pari 2.13 for quaternion arithmetic [PAR], and we adapt the so called `velusqrt` code for isogeny evaluation [BDLS20][6]. Our code is avaible at https://github.com/seta-isogeny-encryption/seta.

4.6.1 – Main building blocks. Key generation consists of two parts. Finding a suitable θ in its quaternion form and then finding a supersingular elliptic curve whose endomorphism ring contains θ. The difficult part of this procedure in practice is a subroutine for finding a supersingular elliptic curve whose endomorphism ring is isomorphic to a particular maximal order $\mathcal{O}$. For this step we reused a substantial amount of the code used for SQISign [De +20].

Encryption consists in the evaluation of an isogeny of degree D at points of order N. In order to make this efficient we choose parameters where D has small prime factors and both D and N divide $p^2 - 1$ to avoid using extension fields.

Decryption also uses evaluations of isogenies, but here isogenies of degree N are evaluated. Furthermore, decryption requires some linear algebra modulo D (when computing the intersection $\ker(\tau - [d]) \cap E_m[D]$) and modulo N (when computing the isogenies ψ and ψ'). In these steps one uses subroutines for solving discrete logarithms but due to N and D being smooth, this step is negligible compared to other computations.

4.6.2 – Prime search. To efficiently implement Séta, it is necessary to select a prime satisfying the many constraints mentioned in Section 4.4.3. To maximise efficiency of encryption and decryption, while maintaining reasonably efficient key generation, we opted to search for a prime satisfying the following constraints: (1) $p^2 - 1 = DN$, with both D and N smooth; (2) $D \approx 2^{2\lambda}$ and $N \approx 2^{4\lambda}$; and (3) D has as few prime factors as possible.

[5] https://github.com/SQISign/sqisign
[6] https://velusqrt.isogeny.org/software.html

There are currently three known techniques to search for primes such that $p^2 - 1$ is smooth, all discussed in [CMN20]. Of these, the most apt to satisfy the constraint that D has few prime factors was introduced by Costello in [Cos20]: fix an exponent $n > 1$, and sieve the space of integers $p = 2x^n - 1$ until one is found such that both $p + 1 = 2x^n$ and $p - 1 = 2(x^n - 1)$ are smooth.

Thanks to this technique, D can be taken as a factor of $p + 1$, and has thus much fewer prime factors than a generic smooth prime of the same size. The drawback of the technique is that, as n increases, the search space decreases, to the point where no smooth integers may be found.

Concretely, for $\lambda = 128$, we fixed $n = 12$ and we sieved within the space $2^{32} < x < 2^{33}$, i.e., $2^{385} < p < 2^{397}$. This yielded four primes with largest factor bounded by 2^{25}, and three with bound 2^{26}, corresponding to $x = 4679747572$, 4845958752, 4966654633, 5114946480, 6334792777, 8176556533, 8426067021. Unfortunately, the search space was fully explored, meaning that no better primes exist for $n = 12$.

The relatively large smoothness bounds negatively affect performance of all algorithms in Séta. Unfortunately, it appears to be difficult to find better primes given current knowledge. Even dropping the constraint on the number of prime factors of D, the best algorithms known today can hardly beat a 2^{20} smoothness bound for a prime of 384 bits [CMN20, Table 3].

4.6.3 – Experimental results. We ran experiments on a 4.00GHz Quad-Core Intel Core i7, using a single core. We used the prime $p = 2 \cdot 8426067021^{12} - 1$, and the smooth factors

$$
\begin{aligned}
D &= 43^{12} \cdot 84719^{11}, \\
N &= 3^{21} \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 73 \cdot 257^{12} \cdot 313 \cdot 1009 \cdot 2857 \cdot 3733 \cdot 5519 \cdot 6961 \\
&\quad \cdot 53113 \cdot 499957 \cdot 763369 \cdot 2101657 \cdot 2616791 \cdot 7045009 \cdot 11959093 \\
&\quad \cdot 17499277 \cdot 20157451 \cdot 33475999 \cdot 39617833 \cdot 45932333.
\end{aligned}
$$

The key generation was ran only once, and took 10.43 hours. The encryption procedure took 4.63 seconds, and the decryption took 10.66 minutes, averaged over six runs. The decryption time is almost entirely devoted to the evaluation of isogenies of degrees the largest factors of N.

4.7 — Further work and conclusion

The efficiency of the scheme essentially depends on the prime factorization of D. We have managed to keep all computations within $\mathbb{F}_{p^2}$ but D still has large prime factors. In principle, one can construct trapdoor curves whenever $N > D^2$ so in particular when ND divides $p - 1$ and $N = 2^k, D = 3^l$. The bottleneck here is the generation of the trapdoor curve which is rather inefficient, despite its polynomial complexity. Note that generating the curve does not affect the speed of encryption and decryption, it only affects the speed of key generation. Thus if one devised a more efficient version of the KLPT algorithm which speeds up the maximal order to elliptic curve mapping algorithm, then one could derive a much more efficient scheme. We estimate that in the best case, one could get a scheme which is only 5 times slower than SIDH. Another interesting research direction is whether one could build upon our Séta scheme and derive more advanced primitives. The framework of Séta has certain advantages in

this context when compared to SIDH. First, Séta is based on a trapdoor one-way function which could be useful in building signature schemes. Second, SIDH-based constructions are more likely to need a trusted setup to avoid backdoor curve attacks such as the one described in [Bas+21, Section 6]. Finally, public key validation is easy in the context of Séta which could be used to build non-interactive key exchange or counteract fault attacks.

This chapter presents the OW-CPA PKE scheme Séta, built upon a generalized version of the isogeny-based CGL hash function family. To do so, we made use of a "torsion-point attack" against SIDH-like schemes [Pet17] and transformed this into a decryption mechanism which recovers a message encrypted as a secret isogeny between a trapdoor starting curve and a final ciphertext curve. An IND-CCA variant is constructed using the post-quantum OAEP transform and both security properties are proven to reduce to the TCSSI problem, derived from the CSSI problem introduced in [JD11]. We then discussed the key generation in terms of computing trapdoor information, the corresponding curve generation, and of the constraints that this does or does not place on the base prime of the scheme; we also proposed an alternative method for these computations. Of independent interest, we formalized the "uber isogeny asumption" and discussed its relation with existing isogeny-based schemes, such as CSIDH, OSDIH and SIDH, before analyzing its complexity. Finally, we presented implementation results for both the search of a well-suited base prime and for key-generation, encryption and decryption experiments.

Acknowledgments. We would like to thank the anonymous reviewers for their remarks and suggestions.

Chapter 5

SHealS and HealS: isogeny based PKEs from a key validation method for SIDH

This chapter is for all practical purposes identical to the paper *SHealS and HealS: isogeny based PKEs from a key validation method for SIDH* [FP21b], authored jointly with Christophe Petit, which was published at Asiacrypt 2021.

5.1 — Introduction

The general isogeny computational problem is the following: given two isogenous elliptic curves E and E', compute an isogeny from E to E'. This hard problem was used by J. M. Couveignes [Cou06], Rostovtsev and Stolbunov [RS06] to design a key exchange protocol using ordinary isogenies, and by Charles, Goren and Lauter [CLG09] to design a cryptographic hash function using supersingular isogenies. The CRS (Couveignes-Rostovtsev-Stolbunov) key exchange scheme is less practical in general and is vulnerable to a sub-exponential quantum attack [CJS14].

In 2011, Jao and De Feo proposed SIDH [JD11] that uses isogenies of supersingular elliptic curves. SIDH is efficient and it is not vulnerable to the sub-exponential quantum attack presented in [CJS14]. Nevertheless, a recent paper by Kutas et al. [KMPW21] proves that hidden-shift like attacks apply to variants of SIDH with considerably overstretched parameters. The isogeny computational problem underlying the security of SIDH is believed to be hard to break, even when using a quantum computer. SIKE [Jao+20] (which is the state of art implementation of SIDH [JD11; FJP14]) is the only isogeny-based Key Encapsulation Mechanism (KEM) submitted to the NIST post-quantum standardization process. Even though SIKE is not the most efficient candidate among KEMs in this competition, SIKE provides the most compact keys and ciphertexts. This has certainly contributed to its selection for the third round of the competition as an alternate candidate [Nat].

Contrarily to the ordinary case where isogenies commute, supersingular isogenies do not commute in general. In order to solve this issue in SIDH, the images of some well-chosen torsion points through the secret isogeny are computed and included in the public keys.

In 2016, Galbraith et al. [GPST16] exploited this supplementary information to develop adaptive attacks on SIDH when one party has a static secret key. The main idea of the attack is that Bob replaces the images of the torsion points in his public key by malicious ones and obtains some information on Alice's static secret when looking at the obtained shared secret. Repeating this process a polynomial number of times, Bob totally recovers Alice's private key.

In SIKE, the attack is avoided by applying a variant [HHK17] of the Fujisaki-Okamoto transform [FO99]. This transform forces Bob to reveal his encryption key to Alice. Two countermeasures enabling static-static key exchange have been proposed: k-SIDH [AJL17] and a variant by Jao and Urbanik [UJ20]. These schemes essentially consist in running k^2 parallel instances of SIDH with each party having k SIDH private keys, hence each party computes about k^2 isogenies. In [Dob+20] and in [Bas+20], it is shown that variants of the adaptive attacks still apply to these schemes, and that the attacks are exponential in k in general. Hence one needs a relatively large k, say $k = 46$ as suggested by [Dob+20], for these schemes to be secure. For $k = 46$, about $46^2 = 2116$ isogenies are computed in k-SIDH, hence the scheme is arguably not practical. To the best of our knowledge, there exists no practically efficient method to counter the adaptive attack on SIDH without revealing the encryption key and using re-encryption to verify the validity of the ciphertext.

CSIDH [Cas+18] is the perfect post-quantum alternative to the classical Diffie-Hellman key exchange due to its analogy to the later primitive. Meanwhile, its quantum security has been considerably degraded recently [Pei20], [BS20], [CCJR20] and remains to be precisely estimated. CSIDH was originally instantiated with a 512 bit prime, but due to analysis of its actual quantum security, in [CCJR20] it is suggested to use primes of up to 4000 bits to achieve the NIST level 1 security. The increase of the prime size impacts the efficiency of the scheme.

Contributions. The contributions of this chapter are fourfold.

Firstly, we propose a new countermeasure to the GPST adaptive attack on SIDH. The main idea is that Bob enables Alice to verify that his torsion points were honestly generated. Consider an SIDH setting, let $\phi_A : E_0 \to E_A$ and $\phi'_A : E_B \to E_{BA}$ be Alice's secret isogenies, $\phi_B : E_0 \to E_B$ and $\phi'_B : E_A \to E_{AB}$ be Bob's secret isogenies in an SIDH instance. In Section 5.3, we prove that if Bob publishes the action of ϕ_B on $E_0[\ell_A^{2e_A}]$ and that of ϕ'_B on $E_A[\ell_A^{2e_A}]$, then Alice can exploit this information to verify Bob's public key validity. Working with SIDH parameters where $p = \ell_A^{e_A}\ell_B^{e_B} f - 1$, the torsion points of order $\ell_A^{2e_A}$ and $\ell_B^{2e_B}$ would be defined over extensions of $\mathbb{F}_{p^2}$ of degree roughly $\ell_A^{e_A}$ and $\ell_B^{e_B}$ respectively. We hence increase the field characteristic to $p = \ell_A^{2e_A}\ell_B^{2e_B} f - 1$ (where f is a small co-factor) such that the later torsion groups are defined over $\mathbb{F}_{p^2}$. Also, we set the starting curve E_0 to be a random supersingular curve with unknown endomorphism ring to avoid improved torsion points attacks. We hence obtain an efficient key validation method which does not require key disclosure and re-encryption, as it is the case in SIKE.

Secondly, we incorporate this key validation method into a key exchange scheme: HealSIDH (**Heal**ed **SIDH**). Let $p = \ell_A^{2e_A}\ell_B^{2e_B} f - 1$ as required by the countermeasure, let $\phi_A : E_0 \to E_A$, $\phi'_A : E_B \to E_{BA}$, and $\phi_B : E_0 \to E_B$, $\phi'_B : E_A \to E_{AB}$ be Alice's and Bob' secret isogenies respectively. Alice reveals the action of ϕ_A on $E_0[\ell_B^{2e_B}]$ and that of ϕ'_A on $E_B[\ell_B^{2e_B}]$. Analogously, Bob reveals the action of ϕ_B on $E_0[\ell_A^{2e_A}]$ and that of ϕ'_B on $E_A[\ell_A^{2e_A}]$. Revealing the action of ϕ'_A and ϕ'_B on torsion points implies revealing points on the shared curve $E_{AB} = E_{BA}$. To avoid this, each party canonically generates a basis of the corresponding subgroup and reveals the coordinates of the points in this canonical basis. HealSIDH is an order of magnitude more efficient compared to k-SIDH (the existing countermeasure to the adaptive attack on SIDH) since only four isogenies are computed in HealSIDH while more than k^2 (with $46 \leq k$)

of them are computed in k-SIDH. The security of HealSIDH against key recovery relies on Problem 5.4.3 which is a variant of the Supersingular Isogeny Computational Diffie-Hellman Problem (SSICDHP), Problem 5.2.5.

Thirdly, we design a PKE scheme using HealSIDH. Our PKE scheme is named SHealS: **S**tatic-static key **Heal**ed **S**IKE. The idea in SHealS is to use the points to encrypt the plaintext, in such a way that the receiver solves a discrete logarithm problem in a group of smooth order to recover the plaintext. A similar idea is used in SiGamal [MOT20] and SimS [FP21c], but our design is different. SHealS uses primes two times larger (in terms of bit size) compared to SIKE primes, has larger keys and ciphertexts, but only 4 isogenies are computed and evaluated on torsion points in a full execution ($\mathsf{Key\ Generation} + \mathsf{Encryption} + \mathsf{Decryption}$) of the scheme, as opposed to 5 isogenies in SIKE, among which 3 isogenies are evaluated on torsion points while the remaining two are not. For this reason, we believe SHealS efficiency is comparable to that of SIKE, but only an optimised implementation of SHealS would help evaluate the exact timings and do a more precise efficiency comparison. The main advantage of SHealS over SIKE is the reuse of encryption keys. In fact, since there is no key disclosure, the encryption key can remain static for a given user. Moreover, this user can use this same key as a private key in the SHealS PKE setting. We prove that SHealS is IND-CPA secure relying on one new assumption we introduce. Despite not being able to come up with a succinct proof of IND-CCA security, we conjecture that SHealS is IND-CCA secure and provide arguments to support our conjecture.

Lastly, we suggest HealS, a variant of SHealS using a smaller prime, providing the same security level, smaller keys and ciphertexts. The size of the prime used in HealS is only 1.5 times that of the prime used in SIKE. This yields a speed-up over SHealS, smaller keys and ciphertexts; hence reducing the efficiency and key sizes gap between SHealS and SIKE. The drawback of HealS compared to SHealS is that private keys can not be used as encryption keys.

As a result, beside CSIDH whose quantum security remains to be precisely estimated, HealSIDH is a new efficient interactive post-quantum key exchange scheme enabling static-static key setting. Moreover, we believe the fact that there is no key disclosure in SHealS and HealS makes of them promising PKE schemes.

Related work. While this work was under submission, an SIDH Proof of Knowledge mechanism [FDGZ21] was published online by De Feo et al. This mechanism enable any party in an SIDH instance to prove that his public key was honestly generated. The proof attached to the public key is obtained by performing an SIDH-type signature on the public key to proof the knowledge of the secret isogeny and the correctness of the torsion points. For this reason, the proof is relatively large ($O(\lambda^2)$), computing and verifying the proof are relatively time consuming compared to our schemes. Nevertheless, their proof enables the design of an SIDH based NIKE while our key exchange HealSIDH is interactive.

Outline. The remainder of this capter is organized as follows: in Section 5.2, we recall some generalities about PKE schemes, elliptic curves and isogenies. We briefly present SIDH, the improved torsion points attacks and the GPST adaptive attack. We end Section 5.2 by describing existing countermeasures to the GPST adaptive attacks. Section 5.3 is devoted to our countermeasure. In Section 5.4 we present HealSIDH key exchange and in Section 5.5 we construct the SHealS PKE scheme. In

Section 5.6, we provide a concrete instantiation of HealSIDH and SHealS, and provide a high level comparison to k-SIDH and SIKE respectively. In Section 5.7, we present HealS and in Section 5.8 we conclude the paper.

5.2 — Preliminaries

5.2.1 – Public key encryption. We recall standard security definitions related to public key encryption.

Definition 5.2.1 (PKE). *A Public Key Encryption scheme $\mathcal{P}_\lambda$ is a triple of PPT algorithms (*Key Generation, Encryption, Decryption*) that satisfy the following.*

1. *Given a security parameter λ as input, the key generation algorithm* Key Generation *outputs a public key pk, a private key sk and a plaintext space $\mathcal{M}$.*
2. *Given a plaintext $\mu \in \mathcal{M}$ and a public key pk as inputs, the encryption algorithm* Encryption *outputs a ciphertext $\mathsf{c} = \mathsf{Encryption}_{pk}(\mu)$.*
3. *Given a ciphertext c and sk as inputs, the decryption algorithm* Decryption *outputs a plain text $= \mathsf{Decryption}_{sk}(\mathsf{c})$.*

Definition 5.2.2 (Correctness). *A PKE scheme $\mathcal{P}_\lambda$ is correct if for any pair of keys (pk, sk) and for every plaintext $\mu \in \mathcal{M}$,*

$$\mathsf{Decryption}_{sk}\left(\mathsf{Encryption}_{pk}(\mu)\right) = \mu.$$

Definition 5.2.3 (IND-CPA secure). *A PKE scheme $\mathcal{P}_\lambda$ is IND-CPA secure if for every PPT adversary $\mathcal{A}$,*

$$Pr\left[b = b^* \;\middle|\; \begin{array}{l} (pk, sk) \leftarrow \mathsf{Key\ Generation}(\lambda), \mu_0, \mu_1 \leftarrow \mathcal{M}, \\ b \xleftarrow{\$} \{0,1\}, \mathsf{c} \leftarrow \mathsf{Encryption}_{pk}(\mu_b), b^* \leftarrow \mathcal{A}(pk, \mathsf{c}) \end{array}\right] = \frac{1}{2} + \mathsf{negl}(\lambda).$$

Definition 5.2.4 (IND-CCA secure). *A PKE scheme $\mathcal{P}_\lambda$ is IND-CCA secure if for every PPT adversary $\mathcal{A}$,*

$$Pr\left[b = b^* \;\middle|\; \begin{array}{l} (pk, sk) \leftarrow \mathsf{Key\ Generation}(\lambda), \mu_0, \mu_1 \leftarrow \mathcal{A}^{O(\cdot)}(pk, \mathcal{M}), \\ b \xleftarrow{\$} \{0,1\}, \mathsf{c} \leftarrow \mathsf{Encryption}_{pk}(\mu_b), b^* \leftarrow \mathcal{A}^{O(\cdot)}(pk, \mathsf{c}) \end{array}\right] = \frac{1}{2} + \mathsf{negl}(\lambda),$$

where $O(\cdot)$ is a decryption oracle that when given a ciphertext $\mathsf{c}' \neq \mathsf{c}$, outputs $\mathsf{Decryption}_{sk}(\mathsf{c}')$ or $\perp$ if the ciphertext c' is invalid.

5.2.2 – Elliptic curves and isogenies. An elliptic curve is a rational smooth curve of genus one with a distinguished point at infinity. Elliptic curves can be seen as commutative groups with respect to a group addition having the point at infinity as neutral element. When an elliptic curve E is defined over a finite field $\mathbb{F}_q$, the set of $\mathbb{F}_q$-rational points $E(\mathbb{F}_q)$ of E is a subgroup of E. For every integer N coprime with q, the N-torsion subgroup $E[N]$ of E is isomorphic to $\mathbb{Z}_N \oplus \mathbb{Z}_N$.

An isogeny from E to E' is a rational map from E to E' which is also a group morphism. The kernel of an isogeny is always finite and entirely defines the isogeny up

to powers of the Frobenius. Given a finite subgroup G of E, there exists a Frobenius free isogeny of domain E having kernel G, called a separable isogeny. Its degree is equal to the size of its kernel. The co-domain of this isogeny is denoted by E/G. The isogeny and the co-domain E/G can be computed from the knowledge of the kernel using Vélu's formulas [Sil09] whose efficiency depends on the smoothness of the isogeny degree.

An endomorphism of an elliptic curve E is an isogeny from E to E. The group structure of E is closely related to that of its endomorphism ring. When E is defined over a finite field, the endomorphism ring of E is either an order in a quadratic field, in which case we say E is ordinary, or a maximal order in a quaternion algebra in which case we say E is supersingular. The generic isogeny problem is harder to solve for supersingular curves (for which the best attacks are exponential) than ordinary curves (for which there exists a sub-exponential attack [CJS14]). SIDH is based on supersingular isogenies.

5.2.3 – SIDH. The SIDH scheme is defined as follows.

Setup. Let $p = \ell_A^{e_A}\ell_B^{e_B} - 1$ be a prime such that $\ell_A^{e_A} \approx \ell_B^{e_B} \approx \sqrt{p}$. Let E_0 be a supersingular curve defined over $\mathbb{F}_{p^2}$. Set $E_0[\ell_A^{e_A}] = \langle P_A, Q_A\rangle$ and $E_0[\ell_B^{e_B}] = \langle P_B, Q_B\rangle$. The public parameters are E_0, p, ℓ_A, ℓ_B, e_A, e_B, P_A, Q_A, P_B, Q_B.

Key Generation. The secret key sk_A of Alice is a uniformly random integer α sampled from $\mathbb{Z}_{\ell_A^{e_A}}$. Compute the cyclic isogeny $\phi_A : E_0 \to E_A = E_0/\langle P_A + [\alpha]Q_A\rangle$. The public key of Alice is the tuple $\mathsf{pk}_A = (E_A, \phi_A(P_B), \phi_A(Q_B))$. Analogously, Bob's secret key sk_B is a uniformly random integer β sampled from $\mathbb{Z}_{\ell_B^{e_B}}$ and his public key is $\mathsf{pk}_B = (E_B, \phi_B(P_A), \phi_B(Q_A))$ where $\phi_B : E_0 \to E_B = E_0/\langle P_B + [\beta]Q_B\rangle$.

Key Exchange. Upon receiving (E_B, R_a, S_a), Alice checks that $e(R_a, S_a) = e(P_A, Q_A)^{\ell_B^{e_B}}$, if not she aborts. She computes the isogeny $\phi'_A : E_B \to E_{BA} = E_B/\langle R_a + [\alpha]S_a\rangle$. Her shared key is $j(E_{BA})$. Similarly, upon receiving (E_A, R_b, S_b), Bob checks that $e(R_b, S_b) = e(P_B, Q_B)^{\ell_A^{e_A}}$, if not he aborts. He computes the isogeny $\phi'_B : E_A \to E_{AB} = E_A/\langle R_b + [\beta]S_b\rangle$. His shared key is $j(E_{AB})$.

The correctness of the key exchange follows from the fact that

$$E_A/\langle \phi_A(P_B) + [\beta]\phi_A(Q_B)\rangle \simeq E_0/\langle P_A + [\alpha]Q_A, P_B + [\beta]Q_B\rangle \simeq E_B/\langle \phi_B(P_A) + [\alpha]\phi_B(Q_A)\rangle\,.$$

The security of the SIDH key exchange protocol against shared key recovery relies on Problem 5.2.5. Furthermore, Problem 5.2.6 states that it is difficult to distinguish the shared secret from a random supersingular elliptic curve.

Problem 5.2.5 (Supersingular Isogeny Computational Diffie-Hellman)**.** *Given E_0, P_A, Q_A, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, E_B, $\phi_B(P_A)$, $\phi_B(Q_A)$ (defined as in SIDH), compute E_{AB}.*

Problem 5.2.6 (Supersingular Isogeny Decisional Diffie-Hellman)**.** *Given E_0, P_A, Q_A, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, E_B, $\phi_B(P_A)$, $\phi_B(Q_A)$ (defined as in SIDH), set $F_0 = E_{AB}$ and let F_1 be a uniformly random supersingular curve. Given $(E_0, P_A, Q_A, P_B, Q_B, E_A,$ where b is uniformly sampled from $\{0,1\}$, distinguish whether $b = 0$ or $b = 1$.*

An IND-CPA secure PKE from SIDH. One canonically derives a PKE scheme from SIDH as follows. Let $H : \mathbb{F}_{p^2} \to \{0,1\}^n$ be a cryptographic hash function.

Key Generation. Alice generates her key pair exactly as in SIDH.

Encryption. Let m be a plaintext. Bob generates a random integer $\beta \in \mathbb{Z}_{\ell_B^{e_B}}$ and executes the SIDH key exchange using Alice's public key to obtain $\mathsf{c}_0 = (E_B, \phi_B(P_A), \phi_B(Q_A))$ and $j_{AB} = j(E_{AB})$. The returned ciphertext is $(\mathsf{c}_0, \mathsf{c}_1 = H(j_{AB}) \oplus \mathsf{m})$.

Decryption. Given a ciphertext $(\mathsf{c}_0, \mathsf{c}_1)$, Alice completes the underlying SIDH key exchange to obtain $j_{BA} = j(E_{BA})$ and recovers the plaintext $\mathsf{m} = \mathsf{c}_1 \oplus H(j_{BA})$.

The above scheme is IND-CPA secure assuming Problem 5.2.6 is hard [FJP14], but it is not IND-CCA since it is vulnerable to the GPST adaptive attack [GPST16] that we present later in Section 5.2.5.

5.2.4 – Passive torsion point attacks on SIDH. The direct key recovery attack (attacking one party's secret key) in SIDH translates into solving the following *Supersingular Isogeny Problem.*

Problem 5.2.7. *Let A and B be two integers such that $\gcd(A, B) = 1$. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Set $E_0[B] = \{P, Q\}$ and let $\phi : E_0 \to E_A$ be a random isogeny of degree A. Given E_0, E_A, P, Q, $\phi(P)$ and $\phi(Q)$, compute ϕ.*

The difference between Problem 5.2.7 and the general isogeny problem is the fact that the action of ϕ on the group $E_0[B]$ is revealed. In 2017, Petit [Pet17] exploited these torsion point images to design an algorithm that solves Problem 5.2.7 for a certain choice of unbalanced ($A \ll B$) parameters when the endomorphism ring of the starting curve E_0 is public. Petit's attack has recently been considerably improved by de Quehen et al. [Que+21]. We refer to [Que+21] for more details.

To counter the attack in unbalanced SIDH instances, one sets the starting curve E_0 to be a random supersingular curve with unknown endomorphism ring. We don't know how to generate random supersingular elliptic curves for which the endomorphism ring is unknown (also to the party generating the curve). This is considered as an open problem [DMPS19]. Hence one generally relies on a trusted party to generate a random curve which is then used as a public parameter of the scheme. This will be the case for the schemes presented in this paper.

5.2.5 – GPST adaptive attack. In SIDH [FJP14] one does a pairing-based check on the torsion points $\phi_B(P_A)$ and $\phi_B(Q_A)$ returned by a potentially malicious Bob. Let E be a supersingular elliptic curve, let N be an integer and let μ_N be the group of N-roots of unity. Let $e_N : E[N] \times E[N] \to \mu_N$ be the Weil pairing [Gal12]. Let $\phi : E \to E'$ be an isogeny of degree M, then for $P, Q \in E[N]$,

$$e_N(\phi(P), \phi(Q)) = e_N(P, Q)^M$$

where the first pairing is computed on E' and the second one on E.
In SIDH, given (E_B, R, S) returned by Bob as public key, Alice checks if

$$e_{\ell_A^{e_A}}(R, S) = e_{\ell_A^{e_A}}(P_A, Q_A)^{\ell_B^{e_B}}.$$

As we will see below, this verification does not assure that the points R, S were honestly generated. More precisely, the pairing verification does not capture the GPST adaptive attack.

The GPST adaptive attack. The main idea of the Galbraith et al. adaptive attack [GPST16] is that if Bob manipulates the torsion points $\phi_B(P_A)$ and $\phi_B(Q_A)$ conveniently, then he can get some information about Alice's private key α given that he knows if the secret curve computed by Alice is equal to E_{AB} or not. Hence in the attack scenario, Bob needs to have access to the later information. This access is provided to Bob through a key exchange oracle:

$$O(E, R, S, E') \text{ which returns 1 if } j(E') = j(E/\langle R + [\alpha]S\rangle) \text{ and 0 otherwise}$$

If one supposes that $\ell_A = 2$ and $e_A = n$, then after each query, Bob recovers one bit of

$$\alpha = \alpha_0 + 2^1\alpha_1 + 2^2\alpha_2 + \cdots + 2^{n-1}\alpha_{n-1}.$$

The attack recovers the first $n-2$ bits of α using $n-2$ oracle queries, and it recovers the two remaining bits by brute force. We refer to [GPST16] for more details.

5.2.6 – Existing countermeasures to the GPST adaptive attacks . The previous section has highlighted the need for a "better" key validation method for SIDH-type schemes. We now present SIKE and k-SIDH, that are currently the two main countermeasures to the GPST attack on SIDH.

SIKE (Supersingular Isogeny Key Encapsulation). Our description is more general compared to that submitted to the third round of the NIST competition [Jao+20], and it does not include key compression features. In the following scheme, G, H and F are hash functions and n is an integer, we refer to [Jao+20] for more details.

Setup. As in SIDH.

Key Generation. Generate a secret key $\mathsf{sk} = \alpha \in \mathbb{Z}_{\ell_A^{e_A}}$ and a public key $\mathsf{pk} = (E_A, \phi_A(P_B), \phi_A(Q_B))$ as in SIDH. Sample a uniformly random integer $s \in \{0,1\}^n$ and return $(s, \mathsf{sk}, \mathsf{pk})$.

Encapsulation. Sample a uniformly random integer m from $\{0,1\}^n$. Compute $\beta = G(m||\mathsf{pk}) \in \mathbb{Z}_{\ell_B^{e_B}}$ and compute $\mathsf{c}_0 = (E_B, \phi_B(P_A), \phi_B(Q_A))$ and E_{AB} as in the SIDH, together with $\mathsf{c}_1 = F(j(E_{AB})) \oplus m$ and $K = H(m||(\mathsf{c}_0, \mathsf{c}_1))$ and return $((\mathsf{c}_0, \mathsf{c}_1), K)$.

Decapsulation. From $(\mathsf{c}_0, \mathsf{c}_1)$, compute E_{BA} as in SIDH and $m' = \mathsf{c}_1 \oplus F(j(E_{BA}))$. Re-encrypt m' to obtain $\mathsf{c}_0' = (E_B', \psi_B(P_A), \psi_B(Q_A))$. If $\mathsf{c}_0 = \mathsf{c}_0'$, return $K = H(m'||(\mathsf{c}_0, \mathsf{c}_1))$, else return $K = H(s||(\mathsf{c}_0, \mathsf{c}_1))$.

In SIKE, the adaptive attacks are not applicable since during the decapsulation, Alice recomputes Bob's encryption key $\beta' = G(m'||\mathsf{pk}) \in \mathbb{Z}_{\ell_B^{e_B}}$ and checks if the obtained key leads to the curve and torsion points sent by Bob, this enables her to detect maliciously generated public keys. Therefore, the scheme requires key disclosure to the recipient. This is a common drawback to all post-quantum PKEs engaged in the NIST standardization process. In fact, as noticed in [AJL17, §1], these schemes use

ephemeral keys or indirect validation techniques that would expose one's key in the static-static setting.

Other countermeasures to the GPST attack. As a countermeasure to the GPST attack, Azarderakhsh et al. introduced k-SIDH [AJL17]. In k-SIDH, Alice's private key is a tuple $\alpha = (\alpha_1, \cdots, \alpha_k) \in (\mathbb{Z}_{\ell_A^{e_A}})^k$ and Bob's private key is a tuple $\beta = (\beta_1, \cdots, \beta_k) \in (\mathbb{Z}_{\ell_B^{e_B}})^k$. Alice and Bob simultaneously run k^2 SIDH key exchange instances corresponding to the k^2 couples of Alice and Bob's SIDH private keys (α_i, β_j), $1 \leq i, j \leq k$. The shared secret is then obtained by applying a key derivation function to the corresponding k^2 SIDH shared secrets. The scheme quickly becomes impractical as k grows.

In [UJ20], Jao and Urbanik propose a variant of k-SIDH that they expected to be more efficient. Their variant exploits non trivial automorphisms of the starting curve E_0 when this supersingular curve has j-invariant 0 or 1728 to reduce the number k of SIDH instances in k-SIDH. For example, in the case where the starting supersingular curve E_0 has j-invariant 0, there exists a non trivial automorphism η_6 of E_0 of order 6. Given a finite subgroup $G \subset E_0$, the curves E_0/G, $E_0/\eta_6(G)$ and $E_0/\eta_6^2(G)$, are isomorphic but it is not the case for the isogenies $E_0 \to E_0/G$, $E_0 \to E_0/\eta_6(G)$ and $E_0 \to E_0/\eta_6^2(G)$. Hence when performing a key exchange, these three isogenies will lead to three distinct SIDH shared keys. Hence with $\alpha' = (\alpha_1, \cdots, \alpha_{k'}) \in (\mathbb{Z}_{\ell_A^{e_A}})^{k'}$ and $\beta' = (\beta_1, \cdots, \beta_{k'}) \in (\mathbb{Z}_{\ell_B^{e_B}})^{k'}$, Alice and Bob can derive $3k'^2$ SIDH shared secrets contrarily to k'^2 for k-SIDH.

In [Dob+20], Dobson et al. show that the GPST attack can be adapted to k-SIDH. Nevertheless, the cost of the attack (number of queries to the key exchange oracle) grows exponentially with k. Dobson et al.'s attack does not directly apply to the Jao-Urbanik variant of k-SIDH. In [Bas+20], Basso et al. present an adaptation of this attack to the Jao-Urbanik variant. Moreover, they prove that considering their attack, for the same security level, k-SIDH is more efficient compared to the Jao-Urbanik variant. From these two attacks, one concludes that for k-SIDH and the Jao-Urbanik variant to be secure against adaptive attacks, one needs k to be relatively large ([Dob+20] suggests $k = 46$ for about 128 bits of security), consequently the schemes become less practical.

To sum up, as countermeasures to the GPST adaptive attack, SIKE imposes key disclosure while k-SIDH comes with a considerable efficiency drawback. We address this in the next section by providing a new countermeasure which is more efficient compared to k-SIDH and without key disclosure.

5.3 — A new countermeasure to the GPST adaptive attack

In this section, we describe a mechanism which enables Alice, when using a static key, to decide on the correctness of the torsion points returned by Bob. We translate this point correctness mechanism into a new key validation method.

5.3.1 – Overview. In our scenario, like in SIKE, we suppose that the initiator of the communication (Bob) has to prove the validity of his torsion points to the other party (Alice). Let E_0, P_A, Q_A, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(P_B)$ be the public parameters and Alice's public key in an SIDH instance. For simplicity, we suppose

that the degree of Alice's isogeny is 2^a and that the degree of Bob's isogeny is 3^b for some integers a and b. In SIDH, Bob computes a cyclic isogeny $\phi_B : E_0 \to E_B$ of degree 3^b together with the images $\phi_B(P_A)$ and $\phi_B(Q_A)$ of P_A and Q_A. We say that the torsion points $R, S \in E_B[2^a]$ returned by Bob are *correct* if $R = [\lambda]\phi_B(P_A)$ and $S = [\lambda]\phi_B(Q_A)$ for some $\lambda \in \mathbb{Z}/2^a\mathbb{Z}^\times$. We establish a Points Correctness Verification (PCV) mechanism for Alice to determine if the torsion points computed by Bob are correct.

We start with an observation by Leonardi [Leo20]: *"in an honest SIDH, $\phi'_A \circ \phi_B = \phi'_B \circ \phi_A$"*. Composing by $\widehat{\phi'_A}$ on the left, we get

$$[2^a] \circ \phi_B = \widehat{\phi'_A} \circ \phi'_B \circ \phi_A. \tag{5.1}$$

Let $P_2, Q_2 \in E_0[2^{2a}]$ be points such that $[2^a]P_2 = P_A$ and $[2^a]Q_2 = Q_A$. Then

$$\begin{cases} \phi'_A \circ \phi_B(P_2) = \phi'_B \circ \phi_A(P_2) \\ \phi'_A \circ \phi_B(Q_2) = \phi'_B \circ \phi_A(Q_2), \end{cases} \tag{5.2}$$

hence

$$\begin{cases} \phi_B(P_A) = \phi_B([2^a]P_2) = \widehat{\phi'_A} \circ \phi'_B \circ \phi_A(P_2) \\ \phi_B(Q_A) = \phi_B([2^a]Q_2) = \widehat{\phi'_A} \circ \phi'_B \circ \phi_A(Q_2) \end{cases} \tag{5.3}$$

Equation 5.3 suggests that if Alice can successfully check the equalities in Equation 5.2, then she can verify if Bob's torsion points are correct.

The idea of the PCV mechanism is that instead of revealing the action of $\phi_B : E_0 \to E_B$ on the 2^a-torsion sub-group of E_0, Bob reveals the action of ϕ_B on the 2^{2a}-torsion sub-group of E_0 and the action of $\phi'_B : E_A \to E_{AB}$ on the 2^{2a}-torsion sub-group of E_A. In our PCV mechanism, Bob's public key (when honestly computed) is $(E_B, \phi_B(P_2), \phi_B(Q_2))$. The action of $\phi'_B : E_A \to E_{AB}$ on the 2^{2a}-torsion sub-group of E_A is provided by canonically generating a new 2^{2a}-torsion basis $\{R_A, S_A\}$ of E_A and revealing $R_{ab} = \phi'_B(R_A)$ and $S_{ab} = \phi'_B(S_A)$.

At this point, Bob can be malicious in the following three ways:

1. honestly compute $R_a = \phi_B(P_2)$ and $S_a = \phi_B(Q_2)$, and maliciously compute $R_{ab} = \phi'_B(R_A)$ and $S_{ab} = \phi'_B(S_A)$;

2. maliciously compute $R_a = \phi_B(P_2)$ and $S_a = \phi_B(Q_2)$, and honestly compute $R_{ab} = \phi'_B(R_A)$ and $S_{ab} = \phi'_B(S_A)$;

3. maliciously compute $R_a = \phi_B(P_2)$ and $S_a = \phi_B(Q_2)$, and maliciously compute $R_{ab} = \phi'_B(R_A)$ and $S_{ab} = \phi'_B(S_A)$.

In the first two cases, we say that Bob is *partially point-malicious* and in the third case we say that Bob is *doubly point-malicious.*

Remark 5.3.1. *We use the term point-malicious to highlight the fact that we focus only on the correctness of the torsion points output by Bob, not on the validity of the Bob's entire public key. Hence we are supposing that ϕ_B and ϕ'_B are cyclic isogenies of degree 3^b and only the torsion point were maliciously evaluated.*

When Bob is partially point-malicious, then either the right hand term or the left hand term in Equation 5.2 is correctly computed by Alice. Hence the partial point-maliciousness of Bob would be detected since the other term of the equation would be different. Concretely, we have the following theorem.

Theorem 5.3.2. *Let E_0, P_A, Q_A, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(P_B)$ be the public parameters and Alice's public key in an SIDH instance. Let P_2, $Q_2 \in E_0[2^{2a}]$ such that $[2^a]P_2 = P_A$ and $[2^a]Q_2 = Q_A$. Let (E_B, R_a, S_a) be Bob's public key. Moreover, let $\{R_A, S_A\}$ be a canonical basis of $E_A[2^{2a}]$ and let $\{R_{ab}, S_{ab}\}$ be its image through $\phi'_B : E_A \to E_{AB}$ output by Bob. Write $\phi_A(P_2) = [e_1]R_A + [f_1]S_A$ and $\phi_A(Q_2) = [e_2]R_A + [f_2]S_A$. Let us suppose that Bob is eventually partially point-malicious and let $\psi'_A : E_B \to E_B/\langle [2^a]R_a + [\alpha][2^a]S_a\rangle$ be the isogeny computed by Alice.*
If $e_{2^{2a}}(R_a, S_a) = e_{2^{2a}}(P_2, Q_2)^{3^b}$, $[e_1]R_{ab} + [f_1]S_{ab} = \psi'_A(R_a)$ and $[e_2]R_{ab} + [f_2]S_{ab} = \psi'_A(S_a)$, then Bob's torsion points are correct.

Proof. Noticing that $[e_1]R_{ab}+[f_1]S_{ab}$ stands for $\phi'_B\circ\phi_A(P_2)$ and $\psi'_A(R_a)$ for $\phi'_A\circ\phi_B(P_2)$, while $[e_2]R_{ab}+[f_2]S_{ab}$ stands for $\phi'_B\circ\phi_A(Q_2)$ and $\psi'_A(S_a)$ for $\phi'_A\circ\phi_B(Q_2)$, the theorem follows from the previous discussion.

□

Remark 5.3.3. *The points $\phi_A(P_2), \phi_A(Q_2) \in E_A[2^{2a}]$ are secret (known only by Alice). In fact their knowledge is equivalent to the knowledge of Alice's secret since $[2^a]P_2 = P_A$ and $[2^a]Q_2 = Q_A$.*

For the third case where Bob is *doubly point-malicious*, we provide a more involved mathematical proof in the next paragraph.

5.3.2 – The main theorem. In the previous section, we make use of points of order 2^{2a} or 3^{2b}. In SIDH parameters where $p = 2^a 3^b f - 1$, these points are defined over a large extension field of degree roughly $2^a \approx 3^b$. To make our key validation efficient, we use primes of the form $p = 2^{2a}3^{2b}f - 1$. Moreover, we evaluate isogenies of degree 2^a on points of order $3^{2b} \approx 2^{2a}$. To avoid improved torsion points attacks or any variant of it, we set the starting curve E_0 to be a random supersingular curve with unknown endomorphism ring. Figure 5.1 summarizes the key validation mechanism hence obtained.

We prove the following Theorem.

Theorem 5.3.4. *Let $p = 2^{2a}3^{2b}f - 1$ and let E_0 be a random supersingular elliptic curve with unknown endomorphism ring defined over $\mathbb{F}_{p^2}$. Let E_0, P_A, Q_A, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(P_B)$ be the public parameters and Alice's public key in an SIDH instance. Let P_2, $Q_2 \in E_0[2^{2a}]$ such that $[2^a]P_2 = P_A$ and $[2^a]Q_2 = Q_A$. Let (E_B, R_a, S_a) be Bob's public key. Moreover, let $\{R_A, S_A\}$ be a canonical basis of $E_A[2^{2a}]$ and let $\{R_{ab}, S_{ab}\}$ be its image through $\phi'_B : E_A \to E_{AB}$ output by Bob. Write $\phi_A(P_2) = [e_1]R_A + [f_1]S_A$ and $\phi_A(Q_2) = [e_2]R_A + [f_2]S_A$. Let $\psi'_A : E_B \to E_B/\langle [2^a]R_a + [\alpha][2^a]S_a\rangle$ be the second isogeny computed by Alice during the key exchange.*

If $e_{2^{2a}}(R_a, S_a) = e_{2^{2a}}(P_2, Q_2)^{3^b}$, $[e_1]R_{ab} + [f_1]S_{ab} = \psi'_A(R_a)$ and $[e_2]R_{ab} + [f_2]S_{ab} = \psi'_A(S_a)$, then Bob's torsion points are correct with probability $1 - \frac{1}{O(p^{1/4})}$.

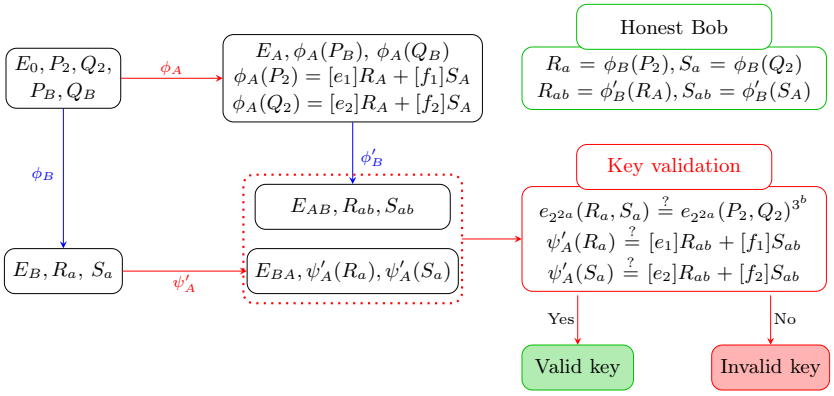


Figure 5.1: Key validation mechanism for SIDH-type schemes. The curve E_0 is a random supersingular elliptic curve with unknown endomorphism ring defined over $\mathbb{F}_{p^2}$ where $p = 2^{2a}3^{2b}f - 1$.

Proof. Let us suppose that Bob is possibly doubly point-malicious, say

$$\begin{aligned} R_a &= [x]\phi_B(P_2) + [y]\phi_B(Q_2) \\ S_a &= [z]\phi_B(P_2) + [t]\phi_B(Q_2) \\ R_{ab} &= [x']\phi'_B(R_A) + [y']\phi'_B(S_A) \\ S_{ab} &= [z']\phi'_B(R_A) + [t']\phi'_B(S_A) \end{aligned}$$

for some integers x, y, z, t, x', y', z' and t' modulo 2^{2a}.

Let us suppose that $e_{2^{2a}}(R_a, S_a) = e_{2^{2a}}(P_2, Q_2)^{3^b}$, $[e_1]R_{ab} + [f_1]S_{ab} = \phi'_A(R_a)$ and $[e_2]R_{ab} + [f_2]S_{ab} = \phi'_A(S_a)$. We prove that $x = t = x' = t' = \pm 1$ and $y = z = y' = z' = 0$, which implies that Bob's torsion points are correct. Let

$$\phi'_A : E_B \to E_{BA} = E_B / \langle \phi_B(P_A) + [\alpha]\phi_B(Q_A) \rangle = E_B / \langle [2^a]\phi_B(P_2) + [\alpha][2^a]\phi_B(Q_2) \rangle$$

be the isogeny that ought to be computed by Alice if Bob's torsion points were correct and let

$$\psi'_A : E_B \to E_B / \langle [2^a]R_a + [\alpha][2^a]S_a \rangle$$

be the isogeny effectively computed by Alice. We distinguish two cases.

Case 1: $\phi'_A \neq \psi'_A$**.** Then $E_{AB} \neq E_B / \langle [2^a]R_a + [\alpha][2^a]S_a \rangle$ with overwhelming probability. In fact, if $E_{AB} = E_B / \langle [2^a]R_a + [\alpha][2^a]S_a \rangle$ with $\phi'_A \neq \psi'_A$, then $\phi'_A \circ \widehat{\psi'_A}$ is an endomorphism of E_{AB} of degree $2^{2a} \approx \sqrt{p}$. Since E_0 is a random supersingular curve, then the curve E_{AB} which is $2^a 2^b$ isogenous to E_0 can be assimilated to a random supersingular curve. Hence the probability that E_{AB} admits a cyclic endomorphism of degree $2^{2a} \approx \sqrt{p}$ is roughly bounded by $\frac{\sqrt{p}^{3/2}}{p} \approx \frac{1}{p^{1/4}}$, this follows from the number of M-small curves where $M = \sqrt{p}$ [LB20].

Hence $R_{ab}, S_{ab} \notin E_B / \langle [2^a]R_a + [\alpha][2^a]S_a \rangle$ with overwhelming probability. Therefore $[e_1]R_{ab} + [f_1]S_{ab} \neq \psi'_A(R_a)$ and $[e_2]R_{ab} + [f_2]S_{ab} \neq \psi'_A(S_a)$ since they are points on different curves[1].

[1] In the up coming sections, rather than revealing torsion points on the shared curve, one reveals their coordinates in some canonical basis. Hence when $E_{AB} \neq E_B / \langle [2^a]R_a + [\alpha][2^a]S_a \rangle$, the points recovered by Alice are just random points on her curve and the validation fails with overwhelming probability.

Case 2: $\phi'_A = \psi'_A$**.** Then Alice computes

$$\begin{aligned}\psi'_A(R_a) = \phi'_A(R_a) &= \phi'_A([x]\phi_B(P_2) + [y]\phi_B(Q_2)) \\ &= \phi'_B \circ \phi_A([x]P_2 + [y]Q_2) \\ &= \phi'_B([x]\phi_A(P_2) + [y]\phi_A(Q_2)) \\ &= \phi'_B\left([x]([e_1]R_A + [f_1]S_A) + [y]([e_2]R_A + [f_2]S_A)\right) \\ &= \phi'_B\left([xe_1 + ye_2]R_A + [xf_1 + yf_2]S_A\right) \\ &= [xe_1 + ye_2]\phi'_B(R_A) + [xf_1 + yf_2]\phi'_B(S_A)\end{aligned}$$

and

$$\begin{aligned}\psi'_A(S_a) = \phi'_A(S_a) &= \phi'_A([z]\phi_B(P_2) + [t]\phi_B(Q_2)) \\ &= \phi'_A \circ \phi_B([z]P_2 + [t]Q_2) \\ &= \phi'_B([z]\phi_A(P_2) + [t]\phi_A(Q_2)) \\ &= \phi'_B\left([z]([e_1]R_A + [f_1]S_A) + [t]([e_2]R_A + [f_2]S_A)\right) \\ &= \phi'_B\left([ze_1 + te_2]R_A + [zf_1 + tf_2]S_A\right) \\ &= [ze_1 + te_2]\phi'_B(R_A) + [zf_1 + tf_2]\phi'_B(S_A)\end{aligned}$$

On the other hand, Alice computes

$$[e_1]R_{ab} + [f_1]S_{ab} = [x'e_1 + z'f_1]\phi'_B(R_A) + [y'e_1 + t'f_1]\phi'_B(S_A)$$

and

$$[e_2]R_{ab} + [f_2]S_{ab} = [x'e_2 + z'f_2]\phi'_B(R_A) + [y'e_2 + t'f_2]\phi'_B(S_A)$$

The integers x, y, z, t, x', y', z' and t' need to satisfy

$$\begin{cases} \psi_A(R_a) = [e_1]R_{ab} + [f_1]S_{ab} \\ \psi_A(S_a) = [e_2]R_{ab} + [f_2]S_{ab} \end{cases}$$

i.e.

$$\begin{cases} [xe_1 + ye_2]\phi'_B(R_A) + [xf_1 + yf_2]\phi'_B(S_A) = [x'e_1 + z'f_1]\phi'_B(R_A) + [y'e_1 + t'f_1]\phi'_B(S_A) \\ [ze_1 + te_2]\phi'_B(R_A) + [zf_1 + tf_2]\phi'_B(S_A) = [x'e_2 + z'f_2]\phi'_B(R_A) + [y'e_2 + t'f_2]\phi'_B(S_A) \end{cases}$$

i.e.

$$\begin{cases} xe_1 + ye_2 = x'e_1 + z'f_1 \\ xf_1 + yf_2 = y'e_1 + t'f_1 \\ ze_1 + te_2 = x'e_2 + z'f_2 \\ zf_1 + tf_2 = y'e_2 + t'f_2 \end{cases} \mod 2^{2a}$$

i.e.

$$\begin{bmatrix} e_1 & e_2 & 0 & 0 \\ f_1 & f_2 & 0 & 0 \\ 0 & 0 & e_1 & e_2 \\ 0 & 0 & f_1 & f_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} = \begin{bmatrix} e_1 & 0 & f_1 & 0 \\ 0 & e_1 & 0 & f_1 \\ e_2 & 0 & f_2 & 0 \\ 0 & e_2 & 0 & f_2 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ t' \end{bmatrix} \mod 2^{2a} \tag{5.4}$$

From Remark 5.3.3, the knowledge of e_1, e_2, f_1 and f_2 is equivalent to the knowledge of Alice's private isogeny ϕ_A. Therefore, when guessing e_1, e_2, f_1, f_2, or ϕ_A, Bob

succeeds with a probability bounded by $\frac{1}{2^a} \approx \frac{1}{p^{1/4}}$, which is negligible. Assuming that Bob does not have access neither to the matrix

$$M_1 = \begin{bmatrix} e_1 & e_2 & 0 & 0 \\ f_1 & f_2 & 0 & 0 \\ 0 & 0 & e_1 & e_2 \\ 0 & 0 & f_1 & f_2 \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}/2^{2a}\mathbb{Z}) \quad \text{nor} \quad M_2 = \begin{bmatrix} e_1 & 0 & f_1 & 0 \\ 0 & e_1 & 0 & f_1 \\ e_2 & 0 & f_2 & 0 \\ 0 & e_2 & 0 & f_2 \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}/2^{2a}\mathbb{Z}).$$

The solutions of Equation 5.4 that are independent of M_1 and M_2 satisfy

$$y = z = y' = z' = 0, \quad x = t = x' = t'.$$

Since $e_{2^{2a}}(R_a, S_a) = e_{2^{2a}}([x]\phi_B(P_2), [x]\phi_B(Q_2)) = e_{2^{2a}}(\phi_B(P_2), \phi_B(Q_2))^{x^2}$, then from the pairing equation $e_{2^{2a}}(R_a, S_a) = e_{2^{2a}}(P_2, Q_2)^{3^b}$, a needs to satisfy $x^2 = 1$, hence $x = \pm 1$.
We finally get $x = t = x' = t' = \pm 1$ and $y = z = y' = z' = 0$. □

Remark 5.3.5. *A formal proof of Theorem 5.3.2 can be obtained from that of Theorem 5.3.4 by setting $x = 1 = t$, $y = 0 = z$ or $x' = 1 = t'$, $y' = 0 = z'$ at the beginning of the proof depending on the points on which Bob decides to be partially point-malicious.*

Remark 5.3.6. *Bob can use the same key validation method to detect a malicious Alice. We set the isogeny degrees to powers of 2 and 3 just for simplicity. The key validation method generalises to any SIDH-like setup.*

5.4 — The HealSIDH (Healed SIDH) key exchange protocol

We now propose a variant of SIDH key exchange protocol which makes use of the GPST adaptive attack countermeasure we have just described. We first give the general idea behind the construction, then we concretely describe the key exchange and we finally discuss the underlying Diffie-Hellman-type hard problems.

5.4.1 – An overview of HealSIDH. The idea behind HealSIDH is to incorporate the key validation mechanism described in Section 5.3 in the SIDH key exchange.

Set $p = 2^{2a}3^{2b}f - 1$ such that $2^a \approx 3^b$, $E_0[2^{2a}] = \langle P_2, Q_2 \rangle$, $E_0[3^{2b}] = \langle P_3, Q_3 \rangle$, $P_A = [2^a]P_2$, $Q_A = [2^a]Q_2$, $P_B = [3^b]P_3$ and $Q_B = [3^b]Q_3$. Alice's secret is an integer α sampled uniformly from $\mathbb{Z}/2^a\mathbb{Z}$ while Bob's secret is an integer β sampled uniformly from $\mathbb{Z}/3^b\mathbb{Z}$. Alice computes $\phi_A : E_0 \to E_A = E_0/\langle P_A + [\alpha]Q_A \rangle$ together with $\phi_A(P_2)$, $\phi_A(Q_2)$, $\phi_A(P_3)$ and $\phi_A(Q_3)$. She canonically generates the basis $\{R_A, S_A\}$ of $E_A[2^{2a}]$ and solves for e_1, f_1, e_2 f_2 such that $\phi_A(P_2) = [e_1]R_A + [f_1]S_A$ and $\phi_A(Q_2) = [e_2]R_A + [f_2]S_A$. Her public key is $(E_A, \phi_A(P_3), \phi_A(Q_3))$ and her secret key is $(\alpha, e_1, f_1, e_2, f_2)$. Bob does the same to obtain a public key $(E_B, \phi_B(P_2), \phi_B(Q_2))$ and a secret key $(\beta, g_1, h_1, g_2, h_2)$.

If Bob wishes to establish a shared secret with Alice, he retrieves Alice's public key (E_A, R_b, S_b), computes $\phi'_B : E_A \to E_{AB} = E_A/\left\langle [3^b]R_b + [\beta][3^b]S_b \right\rangle$ together with $\phi'_B(R_A)$, $\phi'_B(S_A)$, $\phi'_B(R_b)$ and $\phi'_B(S_b)$. The yet to be confirmed shared secret is the j-invariant j_{AB} of E_{AB}. He sends $(\phi'_B(R_A), \phi'_B(S_A))$ to Alice.

Upon receiving $(\phi'_B(R_A), \phi'_B(S_A))$, Alice retrieves Bob's public key tuple (E_B, R_a, S_a). She computes $\phi'_A : E_B \to E_{BA} = E_B/\langle [2^a]R_a + [\alpha][2^a]S_a \rangle$ together with $\phi'_A(R_B)$,

$\phi'_A(S_B)$, $\phi'_A(R_a)$ and $\phi'_A(S_a)$. She then computes R_{ba} and $\widehat{\phi'_A}(\phi'_B(S_A))$.
If $e_{2^{2a}}(R_a, S_a) \neq e_{2^{2a}}(P_2, Q_2)^{3^b}$ or $[e_1]\phi'_B(R_A) + [f_1]\phi'_B(S_A) \neq \phi'_A(R_a)$ or $[e_2]\phi'_B(R_A) + [f_2]\phi'_B(S_A) \neq \phi'_A(S_a)$, Alice aborts. Otherwise, she sends $\phi'_A(R_B)$ and $\phi'_A(S_B)$ to Bob and keeps the j-invariant j_{BA} of E_{BA} as the shared secret.

Upon receiving $\phi'_A(R_B)$ and $\phi'_A(S_B)$, Bob does the key validation check.
If $e_{3^{2b}}(R_b, S_b) \neq e_{3^{2b}}(P_3, Q_3)^{2^a}$ or $[g_1]\phi'_A(R_B) + [h_1]\phi'_A(S_B) \neq \phi'_B(R_b)$ or $[g_2]\phi'_A(R_B) + [h_2]\phi'_A(S_B) \neq \phi'_B(S_b)$, Bob aborts . If not he successfully takes j_{AB} as the shared secret.

Practically, if Bob reveals the points $\phi'_B(R_A)$ and $\phi'_B(S_A)$, or Alice reveals $\phi'_A(R_B)$ and $\phi'_A(S_B)$, then an adversary can recover the curve E_{AB} since for $P \in E_{AB}$, the Montgomery coefficient $A_{E_{AB}}$ of E_{AB} satisfies

$$A_{E_{AB}} = \frac{y(P)^2 - x(P)^3 - x(P)}{x(P)^2}.$$

We avoid this by exploiting the ideas used in SIKE [Jao+20] for key compression: represent a point $P \in E[N]$ by its coordinates in a basis of $E[N]$ which can be canonically computed.

5.4.2 – HealSIDH Key Exchange. Instead of revealing the points $\phi'_B(R_A)$ and $\phi'_B(S_A)$, Bob canonically generates a basis $\{R_{AB}, S_{AB}\}$ of $E_{AB}[2^{2a}]$ and computes $e_3, f_3, e_4, f_4 \in \mathbb{Z}_{2^{2a}}$ such that

$$\phi'_B(R_A) = [e_3]R_{AB} + [f_3]S_{AB} \text{ and } \phi'_B(S_A) = [e_4]R_{AB} + [f_4]S_{AB}.$$

Similarly, Alice canonically generates a basis $\{R_{BA}, S_{BA}\}$ of $E_{BA}[3^{2b}]$ and computes $g_3, h_3, g_4, h_4 \in \mathbb{Z}_{3^{2b}}$ such that

$$\phi'_A(R_B) = [g_3]R_{BA} + [h_3]S_{BA} \text{ and } \phi'_A(S_B) = [g_4]R_{BA} + [h_4]S_{BA}.$$

Concretely, the HealSIDH Key Exchange is entirely described in Figure 5.2.

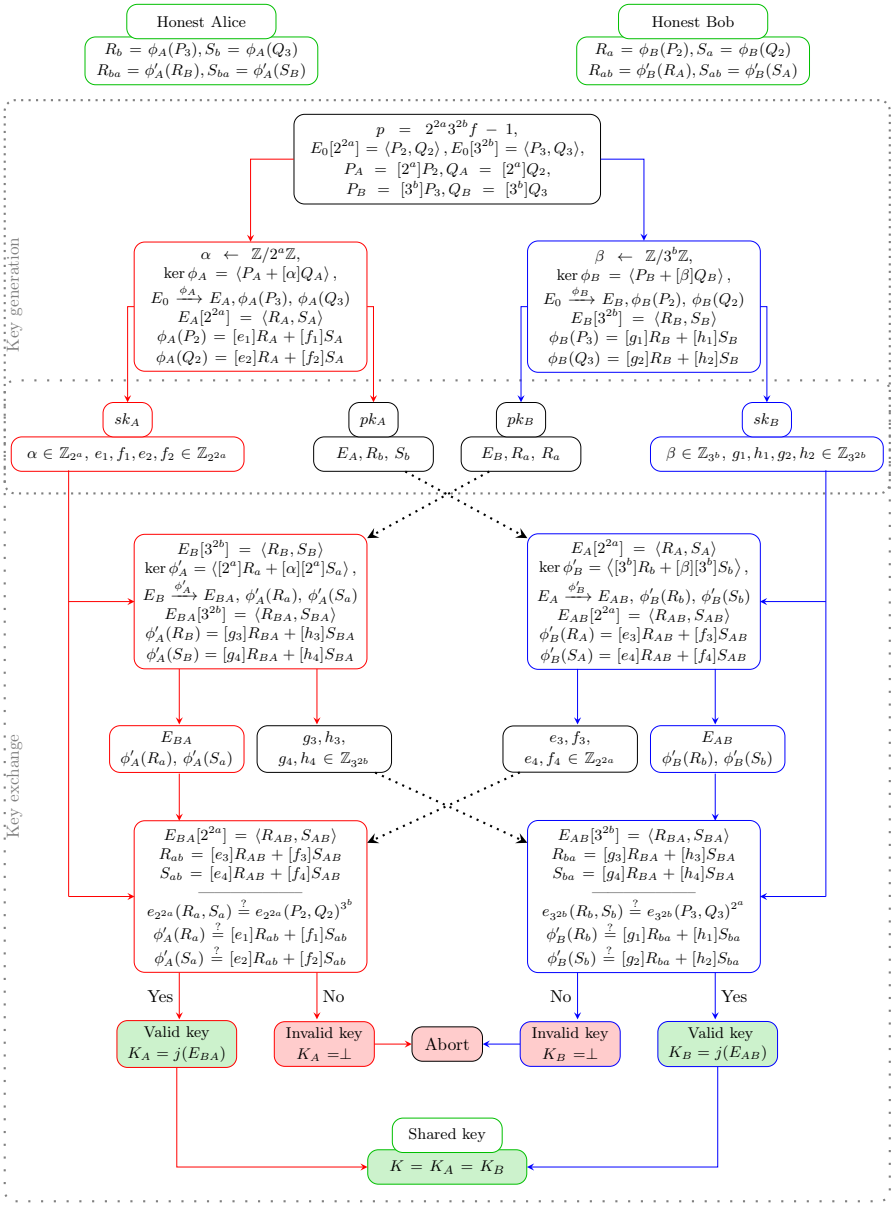
Lemma 5.4.1. *HealSIDH is correct.*

Proof. Follows from the correctness of SIDH and Theorem 5.3.4. □

Remark 5.4.2. *Two parties Alice and Bob need to run the key validation only once, during their first communication. In the subsequent communications between the two parties there is no need to revalidate the keys.*

5.4.3 – Security of HealSIDH. We present the Computational Diffie-Hellman-type problem underlying the security of HealSIDH. We argue that the Decisional variant of this problem is not hard.

Problem 5.4.3 (HealSIDH-CDHP)**.** *Let $p = 2^{2a}2^{2b}f - 1$ and E_0 a supersingular curve defined over $\mathbb{F}_{p^2}$ with unknown endomorphism ring. Let $E_0[2^{2a}] = \langle P_2, Q_2\rangle$, $E_0[3^{2b}] = \langle P_3, Q_3\rangle$, $P_A = [2^a]P_2$, $Q_A = [2^a]Q_2$, $P_B = [3^b]P_3$, $Q_B = [3^b]Q_3$. Let $\phi_A : E_0 \to E_A$, $\phi_B : E_0 \to E_B$, $\phi'_A : E_B \to E_{BA}$ and $\phi'_B : E_A \to E_{AB}$ be secret isogenies as described in SIDH-type schemes. Let $E_A[2^{2a}] = \langle R_A, S_A\rangle$, $E_B[3^{2b}] = \langle R_B, S_B\rangle$, $E_{AB}[2^{2a}] =$*



Note: the basis $\{R_A, S_A\}$, $\{R_B, S_B\}$, $\{R_{AB}, S_{AB}\}$ and $\{R_{BA}, S_{BA}\}$ are canonically generated.

Figure 5.2: HealSIDH interactive key exchange. E_0 is a random supersingular curve.

$\langle R_{AB}, S_{AB} \rangle$, $E_{AB}[3^{2b}] = \langle R_{BA}, S_{BA} \rangle$. *Let* $e_3, f_3, e_4, f_4 \in \mathbb{Z}_{2^{2a}}$ *and* $g_3, h_3, g_4, h_4 \in \mathbb{Z}_{3^{2b}}$ *such that* $\phi'_A(R_B) = [g_3]R_{BA} + [h_3]S_{BA}$, $\phi'_A(S_B) = [g_4]R_{BA} + [h_4]S_{BA}$, $\phi'_B(R_A) = [e_3]R_{AB} + [f_3]S_{AB}$ *and* $\phi'_B(S_A) = [e_4]R_{AB} + [f_4]S_{AB}$.

Given E_0, P_2, Q_2, P_3, Q_3, E_A, $\phi_A(P_2)$, $\phi_A(Q_3)$, R_A, S_A, E_B, $\phi_B(P_2)$, $\phi_B(Q_2)$,

$R_B,\ S_B,\ e_3, f_3, e_4, f_4,\ g_3, h_3, g_4, h_4,\ \textit{compute}\ E_{AB}.$

The main differences between Problem 5.4.3 and Problem 5.2.5 are as follows:

- the action of the secret isogeny ϕ_A (resp. ϕ_B) of degree 2^a (resp. 3^b) on $E_0[3^{2b}]$ (resp. $E_0[2^{2a}]$) is revealed;
- in addition to image points through ϕ_A as in SIDH, the coordinates of some image points through ϕ'_A (resp. ϕ'_B) in a canonical basis are revealed.

With respect to the first point, we reveal the action of isogenies of degree $A \approx p^{1/4}$ on a B-torsion subgroup where $B \approx p^{1/2}$. Since the endomorphism ring of the curve E_0 is unknown, then HealSIDH is protected against improved torsion attacks [Que+21].

With respect to the second point, the coordinates g_3, h_3, g_4, h_4 of $\phi'_A(R_B)$ and $\phi'_A(S_B)$ in a canonical basis of $E_{BA}[3^{2b}]$, and the coordinates e_3, f_3, e_4, f_4 of $\phi'_B(R_A)$ and $\phi'_B(S_A)$ in a canonical basis of $E_{BA}[2^{2a}]$ are revealed. We don't see how this could affect the hardness of Problem 5.4.3.

Nevertheless, revealing these coordinates implies that the decisional version of Problem 5.4.3 is not hard. In fact, suppose that you are given a random supersingular elliptic curve E and you wish to determine if $E = E_{BA}$ or $E \neq E_{BA}$. Then you can generate the canonical bases $E[3^{2b}] = \langle R_{BA}, S_{BA}\rangle$ and $E[2^{2a}] = \langle R_{AB}, S_{AB}\rangle$, perform the pairing checks

$$e_{2^{2a}}\left([e_3]R_{AB} + [f_3]S_{AB}, [e_4]R_{AB} + [f_4]S_{AB}\right) \stackrel{?}{=} e_{2^{2a}}(R_A, S_A)^{3^b}$$

and

$$e_{3^{2b}}\left([g_3]R_{BA} + [h_3]S_{BA}, [g_4]R_{BA} + [h_4]S_{BA}\right) \stackrel{?}{=} e_{3^{2b}}(R_B, S_B)^{2^a}.$$

If $E = E_{AB}$, then these checks would be successful. If $E \neq E_{AB}$, then these checks will fail with overwhelming probability since the points $[e_3]R_{AB} + [f_3]S_{AB}$, $[e_4]R_{AB} + [f_4]S_{AB}$, $[g_3]R_{BA} + [h_3]S_{BA}$ and $[g_4]R_{BA} + [h_4]S_{BA}$ would be random points of E of order 2^{2a}, 2^{2a}, 3^{2b} and 3^{2b} respectively; hence likely would not satisfy the pairing equalities.

5.5 — SHealS: a public key encryption scheme

Even though the DDH-type problem for HealSIDH is not hard, we still use HealSIDH to design a secure public key encryption scheme, which we call SHealS. We first give an overview of our construction, then we fully describe and analyze it.

5.5.1 – An overview of SHealS. Our aim is to derive a PKE scheme from HealSIDH.

A canonical way to design a PKE scheme from HealSIDH is to proceed as follows. Consider the HealSIDH setting. Alice generates her key pair $(\mathsf{sk}_A, \mathsf{pk}_A)$ where $\mathsf{sk}_A = (\alpha, e_1, f_1, e_2, f_2)$ and $\mathsf{pk}_A = (E_A, R_b, S_b)$. In order to encrypt a plaintext m of binary length n, Bob randomly samples $\beta \in \mathbb{Z}/3^b\mathbb{Z}$, computes $\mathsf{c}_0 = (E_B, R_a, S_a, e_3||f_3||e_4||f_4)$ and $\mathsf{c}_1 = H(j_{AB}) \oplus \mathsf{m}$ where $H : \mathbb{F}_{p^2} \to \{0,1\}^n$ is a cryptographic hash function. The ciphertext is $\mathsf{c} = (\mathsf{c}_0, \mathsf{c}_1)$. Decryption consists in completing the underlying HealSIDH key exchange using sk_A and c_0. If the key exchange is successful, recover $\mathsf{m} = \mathsf{c}_1 \oplus H(j_{BA})$ using the shared secret E_{BA}, else $\mathsf{m} = \perp$.

As shown in the following lemma, the resulting PKE scheme is not IND-CCA secure.

Lemma 5.5.1. *Let* $\mathsf{m} \in \{0,1\}^n$ *be a plaintext and let* $k \geq 1$ *be an integer such that the* k^{th} *bit of* m *(the coefficient of* 2^{k-1} *in the 2-adic expansion of* m*) is* 0*. If* $\mathsf{c} = (\mathsf{c}_0, \mathsf{c}_1)$ *is a ciphertext for* m*, then* $\mathsf{c}' = (\mathsf{c}_0, \mathsf{c}_1 \oplus 2^{k-1})$ *is a ciphertext for* $\mathsf{m} + 2^{k-1}$*.*

Proof. Since the k^{th} bit of m is 0, then $\mathsf{m} + 2^{k-1} = \mathsf{m} \oplus 2^{k-1}$. Hence

$$\mathsf{c}_1 \oplus 2^{k-1} = \mathsf{m} \oplus H(j_{AB}) \oplus 2^{k-1} = (\mathsf{m} \oplus 2^{k-1}) \oplus H(j_{AB}) = (\mathsf{m} + 2^{k-1}) \oplus H(j_{AB}).$$

Therefore $\mathsf{c}' = (\mathsf{c}_0, \mathsf{c}_1 \oplus 2^{k-1})$ is a ciphertext for $\mathsf{m} + 2^{k-1}$. □

This IND-CCA attack applies to all PKE schemes in which the ciphertext is of the form $(\mathsf{c}_0, H(s) \oplus \mathsf{m})$ where s and c_0 are independent of m. We choose to use points to encrypt the plaintext, as in SiGamal [MOT20] and SimS [FP21c].

5.5.2 – SHealS Public Key Encryption scheme. The plaintext space is changed to $\mathcal{M} = \mathbb{Z}_{2^{2a}}^{\times}$, the set invertible elements in the ring of integers modulo 2^{2a}. The ciphertext of a given plaintext $\mathsf{m} \in \mathcal{M}$ is $\mathsf{c} = (\mathsf{c}_0, \mathsf{c}_1)$ where $\mathsf{c}_0 = (E_B, R_a, S_a)$, $\mathsf{c}_1 = H(j_{AB}) \oplus (\mathsf{m}e_3 || \mathsf{m}f_3 || \mathsf{m}e_4 || \mathsf{m}f_4)$ and $H : \mathbb{F}_{p^2} \to \{0,1\}^{8a}$ is a cryptographic hash function.

Note that scaling e_3, f_3, e_4 and f_4 by m is equivalent to scaling the points $[e_3]R_{AB} + [f_3]S_{AB}$ and $[e_4]R_{AB} + [f_4]S_{AB}$ by $[\mathsf{m}]$. This enables Alice to recover m by solving a discrete logarithm instance in a group of order 2^{2a}.

Concretely, Figure 5.3 entirely describes SHealS PKE.

Lemma 5.5.2. *SHealS PKE is correct.*

Proof. Follows from the correctness of HealSIDH. □

Remark 5.5.3. *In SHealS, since there is no key disclosure, Bob can reuse his encryption key* β *to encrypt other plaintexts. Moreover, since the* 3^{2b} *torsion points are readily available, he can use the same* β *as a static private key.*

5.5.3 – Security analysis. We prove that SHealS is IND-CPA secure relying on Assumption 4. Next we discuss the IND-CCA security of SHealS. We conjecture that SHealS is IND-CCA secure and provide arguments to support our conjecture.

Assumption 4. *Let* E_0*,* P_2*,* Q_2*,* P_A*,* Q_A*,*P_3*,* Q_3*,* P_B*,* Q_B*,* E_A*,* R_A*,* S_A*,* $\phi_A(P_3)$*,* $\phi_A(Q_3)$*,* E_B*,* $\phi_B(P_2)$*,* $\phi_B(Q_2)$ *the public parameters and keys of an HealSIDH instance. Set* $E_{AB}[2^{2a}] = \langle R_{AB}, S_{AB} \rangle$ *where the basis* $\{R_{AB}, S_{AB}\}$ *is canonically generated, let* $\mathcal{B}_0 = \{\phi'_B(R_A), \phi'_B(S_A)\}$ *and let* $\mathcal{B}_1 = \{R, S\}$ *be a uniformly random basis of* $E_{AB}[2^{2a}]$ *such that* $e_{2^{2a}}(R,S) = e_{2^{2a}}(R_A, S_A)^{3^b}$*. Set* $\phi'_B(R_A) = [e_{03}]R_{AB} + [f_{03}]S_{AB}$*,* $\phi'_B(S_A) = [e_{04}]R_{AB} + [f_{04}]S_{AB}$*,* $R = [e_{13}]R_{AB} + [f_{13}]S_{AB}$ *and* $S = [e_{14}]R_{AB} + [f_{14}]S_{AB}$*. For any PPT algorithm* $\mathcal{A}$*,*

$$Pr\left[b = b^* \;\middle|\; \begin{array}{l} b \xleftarrow{\$} \{0,1\}, \\ b^* \leftarrow \mathcal{A}\left(\begin{array}{l} E_A, \phi_A(P_3), \phi_A(Q_3), E_B, \phi_B(P_2), \\ \phi_B(Q_2), E_{AB}, e_{b3} || f_{b3} || e_{b4} || f_{b4} \end{array} \right) \end{array} \right] = \frac{1}{2} + \mathsf{negl}(\lambda).$$

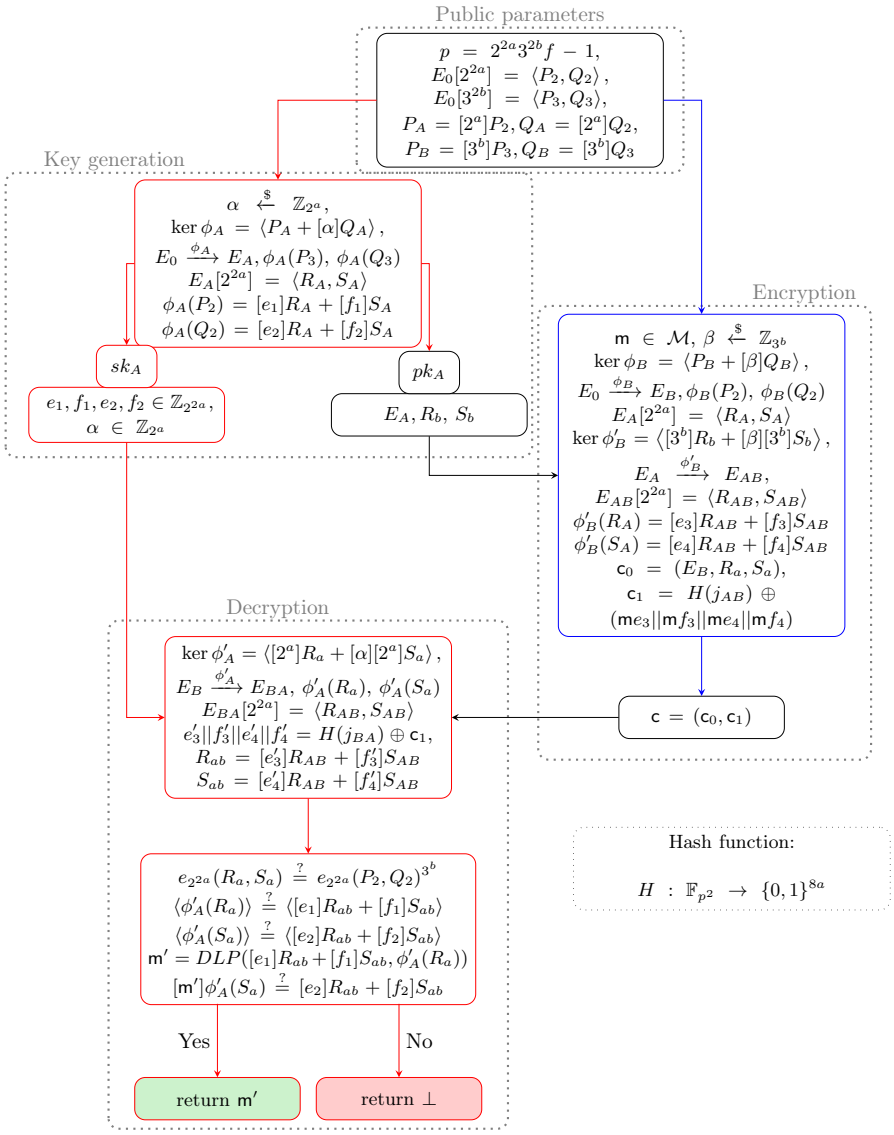


Figure 5.3: SHealS PKE. E_0 is a supersingular curve with unknown endomorphism ring.

Theorem 5.5.4. *If Assumption 4 holds and H is sampled from an entropy smoothing*[2] *hash functions family $\mathcal{H}$, then SHealS is IND-CPA secure.*

Proof. Analogous to the proof of Theorem 3.4.4. □

[2]This means that for any random hash function $H \in \mathcal{H}$ and any random element $s \in \mathbb{F}_{p^2}$, it is hard to distinguish $H(s)$ from a random element of $\{0,1\}^{8a}$ [Sho04].

Concretely, Assumption 4 states that given E_A, $\phi_A(P_3)$, $\phi_A(Q_3)$, E_B, $\phi_B(P_2)$, $\phi_B(Q_2)$, E_{AB}, it is difficult to distinguish the images points $\phi'_B(R_A), \phi'_B(S_A)$ of a basis $\{R_A, S_A\}$ of $E_A[2^{2a}]$ through ϕ'_B and a uniformly random basis $\{R, S\}$ of $E_{AB}[2^{2a}]$ such that $e_{2^{2a}}(R,S) = e_{2^{2a}}(R_A, S_A)^{3^b}$.

Concerning the IND-CCA security of SHealS, one may be tempted to use a knowledge of exponent type as Fouotsa and Petit did to prove the IND-CCA security of SimS [FP21c]. But this type of assumption does not hold for SIDH type schemes. In fact, one can not see SIDH as an analog to the classic Diffie-Hellman as it is the case in CSIDH. In CSIDH, the secret isogeny can have any degree in a well chosen key space. But in SIDH, the degree of the secret isogeny is fixed. This eliminates the idea of assimilating the secret isogenies in SIDH to "exponents".

We have not been able to come up with a succinct proof of IND-CCA security for SHealS, but we argue that SHealS is not vulnerable to any known attack on SIDH type schemes since we have countered the GPST adaptive attack [GPST16] and possible variants of it, and the improved torsion points attacks [Pet17; Que+21]. Note that we do not take side channel attacks into consideration in this analysis. We hence state the following conjecture and leave it's proof or its invalidation for future work.

Conjecture 1. *SHealS is IND-CCA secure.*

5.6 — Concrete instantiations and comparisons: HealSIDH vs k-SIDH; SHealS vs SIKE

5.6.1 – Concrete instantiation. We performed a basic Sagemath [The20] proof-of-concept implementation of our key validation method, HealSIDH and SHealS. We use the prime $p_{870} = 2^{432}3^{274}10-1$ where $a = 216$ and $b = 137$ as in SIKEp434 [Jao+20, §1.6]. Hence we expect SHealSp870 and SIKEp434 on one hand, HealSIDHp870 and k-SIDHp434 on the other hand, to provide the same security level.

The proof-of-concept implementation of SHealS is very basic and unoptimized, hence it cannot serve as a reference when comparing SHealS and SIKE in terms of efficiency. In the following paragraph, we do a high level comparison between SHealS and SIKE. We argue that the efficiency of an optimized implementation of SHealS is comparable to that of SIKE (considering instances providing the same security level).

5.6.2 – SHealS vs SIKE. We provide a high level comparison between SHealS and SIKE and argue that SHealS's efficiency is close to that of SIKE. In what follows, we suppose that in both SIKE and SHealS, an SIDH-type public key (E, P, Q) is represented by (x_P, x_Q, x_{P-Q}) as specified in [Jao+20]. Let λ be a security parameter, and let p_h and p_s respectively be the HealSIDH (or SHealS) prime and the SIKE prime providing λ bits of security. It follows that $\lceil \log p_s \rceil \approx 4\lambda$ and $\lceil \log p_h \rceil \approx 8\lambda$.

Design. At the design level, in SHealS, the encryption public key is validated through a "direct" key validation mechanism while in SIKE, the validation is done through re-encryption. For this reason, the number of isogenies computed in SIKE (Key Generation+ Encapsulation + Decapsulation) is 5 while only 4 isogenies are computed in SHealS (Key Generation + Encryption + Decryption). Nevertheless, all the 4 isogenies in SHealS are evaluated on torsion points as well, while only 3 of the 5 isogenies in SIKE are

evaluated on torsion points. In SHealS, a trusted party is needed for the generating the starting curve E_0.

Security. SHealS's IND-CCA security is conjectured while that of SIKE is inheritated from a variant Fujisaki-Okamoto transform [HHK17].

Keys sizes. In SIKE and SHealS, the secret key is α and $(\alpha, e_1, f_1, e_2, f_2)$ respectively. Since e_1, f_1, e_2, f_2 lie in $\mathbb{Z}/2^{2a}\mathbb{Z}$, then their bitsize is twice that of $\alpha \in \mathbb{Z}_{2^a}$. Hence the secret key of HealSIDH is 9 times larger compared to that of SIKE.
The public key in SIKE and SHealS are both of the form (x_P, x_Q, x_{P-Q}). Hence in SIKE the public key has roughly $3(2\lceil \log p_s \rceil) = 6\lceil \log p_s \rceil \approx 24\lambda$ bits while in SHealS it has roughly $3(2\lceil \log p_h \rceil) = 6\lceil \log p_h \rceil \approx 48\lambda$ bits. Therefore, the size of the public key in SHealS is roughly twice that of the public key in SIKE.
For the ciphertext, the bitsize of c_0 in SHealS is twice that of c_0 in SIKE, while the bit size of c_1 in ShealS is $8a = 16\lambda$, opposed to $n \in \{128, 192, 256\}$ in SIKE. It follows that the size of SHealS ciphertexts is about 2.45 times that of SIKE ciphertexts.

Efficiency. As mentioned before, only 4 isogenies are computed in SHealS while 5 isogenies are computed in SIKE. Meanwhile, the prime used in SHealS is twice as large as SIKE prime. And, in SHealS, the isogenies $\phi'_A : E_B \to E_{BA}$ and $\phi'_B : E_A \to E_{AB}$ are evaluated on two torsion points each, which is not the case in SIKE. Without an advanced implementation of SHealS, it is difficult to provide a precise efficiency comparison between both schemes.

We summarize the comparison[3] in Table 5.1. Let λ be a desired security level.

	SIKE	SHealS
Field characteristic size	$\approx 4\lambda$	$\approx 8\lambda$
Private key size	$\approx 2\lambda$	$\approx 18\lambda$
Public key size	$\approx 24\lambda$	$\approx 48\lambda$
Ciphertext size	$\approx 26\lambda$	$\approx 64\lambda$
KeyGen (isog. comp.)	1	1
Encaps/Encrypt (isog. comp.)	2	2
Decaps/Decrypt (isog. comp.)	2	1
Adaptive attacks	No	No (conjecture)
Key disclosure	Yes	No
Encryption key reuse	No	Yes
Key validation method used	re-encryption	Key val. method in § 5.3

Table 5.1: High level comparison between SHealSIDH and SIKE.

5.6.3 – HealSIDH vs k-SIDH. To the best of our knowledge, the only existing post-quantum key exchange schemes enabling static-static key setting prior to this work[4] were CSIDH [Cas+18], k-SIDH [AJL17] and its variant by Jao and Urb-

[3]Note that the comparisons in Table 5.1, 5.3 and 5.2 are really high level and do not include the discrete logarithm instances (at most six of them in dimension 2 groups) in HealSIDH, SHealS and HealS.

[4]While this work was under submission, a proof of isogeny knowledge [FDGZ21] was published online. We will provide a concrete comparison with this construction in later versions of this paper that we will make available on the IACR eprint database.

anik [UJ20]. As highlighted in Section 5.2.6, Basso et al. [Bas+20] showed that k-SIDH is preferable to the later variant from an efficiency vs security point of view. We provide a high level comparison between HealSIDH and k-SIDH since both are countermeasures to the GPST adaptive attacks.

Design. At the design level, HealSIDH comes with an incorporated key validation method, while k-SIDH mitigates the GPST adaptive attacks by running many parallel SIDH intances. This implies that more than k^2 isogenies are computed in k-SIDH (full execution of the key exchange) while only 4 isogenies are computed in HealSIDH. Nevertheless, There are two rounds in HealSIDH, as opposed to one round in k-SIDH. Note that the starting curve in HealSIDH is generated by a trusted party, which is not the case in k-SIDH.

Security. Security wise, HealSIDH is not vulnerable to the GPST adaptive attacks since it incorporates a countermeasure. In k-SIDH, one does not eliminate the attack completely, but one increases its cost in such a way that it becomes exponential in k.

Keys sizes. From the comparison made in Section 5.6.2, the secret key in HealSIDH has roughly 18λ bits. In k-SIDH, the size of the secret key is k times that of a SIKE secret key, hence $2k\lambda$. The public keys in HealSIDH have roughly 48λ bits while in k-SIDH they have about $24k\lambda$ bits.

Efficiency. As mentioned before, only 4 isogenies are computed in HealSIDH. In k-SIDH, roughly $2k^2 + 2k$ isogenies are computed. Even though the HealSIDH prime size is twice that of the k-SIDH prime, k-SIDH is still an order of magnitude less efficient compared to HealSIDH because of the relatively large number of isogenies computed.

Table 5.2 provides a high level comparison between HealSIDH and k-SIDH. We refer to [AJL17] for more details on k-SIDH.

	HealSIDH	k-SIDH
Field characteristic size	$\approx 8\lambda$	$\approx 4\lambda$
Private key size	$\approx 18\lambda$	$\approx 2k\lambda$
Public key size	$\approx 48\lambda$	$\approx 24k\lambda$
KeyGen	1	k
key exchange	2	$2k^2$
Adaptive attacks	No	exp. in k
Static-static key	yes	yes
NIKE	No	yes

Table 5.2: High level comparison between HealSIDH and k-SIDH ($46 \leq k$).

5.7 — HealS (Healed SIKE): improving the efficiency of SHealS

From the comparison in Section 5.6.2, one concludes that the prime size, the key and ciphertext sizes in SHealS are at least twice that in SIKE. In this section, our aim is to improve on this prime, key and ciphertext sizes.

5.7.1 – HealS Public Key Encryption. Having a closer look at ShealS, one notices that since Bob does not run a key validation on Alice's public key in the PKE

encryption scheme, then it is not a requisite to have the 3^{2b}-torsion points defined over $\mathbb{F}_{p^2}$. Hence when the parameters are chosen for a PKE scheme purpose only, the prime p can be relaxed to $p = 2^{2a}3^b f - 1$ where $2^a \approx 3^b$ and f is a small cofactor. Most of the scheme remains unchanged. Concretely, HealS is SHealS with a prime of the form $p = 2^{2a}3^b f - 1$.

While the base prime change when going from SHealS to HealS comes with considerable speed-up and considerable improvement on key and ciphertext sizes (see Section 5.7.2), one should notice that Bob can no more use his encryption key as secret key when receiving encrypted messages. In fact, in order to encrypt a plaintext for Bob, one needs to compute the images of torsion points of order 3^{2b}. For HealS primes, these torsion points are defined over large extensions since $p = 2^{2a}3^b f - 1$. Nevertheless, Bob can reuse the same encryption key β to encrypt other messages to other parties or the same party, only he can not use it as decryption key. This technical difference motivated us to rename the instance HealS instead of keeping the name SHealS. Appendix B.1 provides more details about the Key Generation, Encryption and Decryption algorithms in HealS.

5.7.2 – Concrete instantiation and comparison with SIKE. We instantiate HealS with the prime $p_{650} = 2^{432}3^{137} - 1$ where $a = 216$ and $b = 137$ as in SIKEp434 [Jao+20, §1.6]. Hence HealSp650 and SIKEp434 are expected to provide the same security level.

We summarise a high level comparison between HealS and SIKE in Table 5.3. We also include SHealS in this table to highlight the advantages of HealS when compared to SHealS.

	SIKE	SHealS	HealS
Field characteristic size	$\approx 4\lambda$	$\approx 8\lambda$	$\approx 6\lambda$
Private key size	$\approx 2\lambda$	$\approx 18\lambda$	$\approx 18\lambda$
Public key size	$\approx 24\lambda$	$\approx 48\lambda$	$\approx 36\lambda$
Ciphertext size	$\approx 26\lambda$	$\approx 64\lambda$	$\approx 48\lambda$
KeyGen (isog. comp.)	1	1	1
Encaps/Encrypt (isog. comp.)	2	2	2
Decaps/Decrypt (isog. comp.)	2	1	1
Adaptive attacks	No	No (conj.)	No (conj.)
Key disclosure	Yes	No	No
Encryption key reuse	No	Yes	Yes
Key validation method used	re-encryption	Key val. method in § 5.3	

Table 5.3: High level comparison between HealS, SHealS and SIKE.

Table 5.4 compares the key and ciphertext sizes of our PKE with some NIST finalists KEMs. We notice that the key sizes in HealS are more compact compared to these finalists. The ciphertext size in HealS is close to that of Kyber, NTRU and Saber, while being considerably larger compared to that of Classic McEliece.

	HealS	Kyber	NTRU	Classic McEliece	Saber
sk	288	1632	935	6492	1568
pk	576	800	699	261120	672
c	768	768	699	128	736

Table 5.4: Key and ciphertext sizes comparison between HealS and the four NIST finalists KEMs Kyber, NTRU, Classic McEliece and Saber, for 128 bits of security (NIST level 1).

5.8 — Conclusion

In this chapter, we introduced an efficient countermeasure to the GPST adaptive attack which does not require key disclosure nor re-encryption. Next, we used this countermeasure to design an efficient static-static key interactive exchange scheme: HealSIDH. HealSIDH is not vulnerable to the GPST adaptive attacks. We derive an IND-CPA secure PKE scheme with conjectured IND-CCA security SHealS from HealSIDH. The full execution of SHealS contains only 4 isogeny computations while that of SIKE contains 5 isogeny computations. For this reason, even though SHealS uses larger parameters and has larger keys, we expect its efficiency to be comparable to that of SIKE. In order to optimize the efficiency, keys and ciphertexts sizes, we suggest HealS, a variant of SHealS using a smaller prime. The main difference between SHealS and HealS is that in SHealS, a party can use his private key as encryption key when encrypting ciphertexts for other parties.

Moreover, we provided a high level comparison between HealSIDH and k-SIDH on one hand, and between SHealS, HealS and SIKE on the other hand. HealSIDH is an order of magnitude more efficient compared to k-SIDH and the keys in k-SIDH are about k times bigger compared to those of HealSIDH. The advantages of SHealS and HealS over SIKE are

- no encryption key disclosure to the recipient during encryption;
- incorporated key validation method (no re-encryption during decryption);
- encryption key reuse.

In order to evaluate the concrete efficiency of the schemes constructed in this paper, an advanced implementation of SHealS and HealS is needed. We leave this task to follow-up work. We believe the design of SHealS leaves room for considerable optimisations. These may come from the implementation, from variants of the key validation method or from redesigning the schemes.

Furthermore, there are possibly existing isogeny-based schemes that would benefit from our key validation method. Also the key validation may enables the design of new isogeny-based primitives. We also leave such an investigation for future work.

Acknowledgements. We would like to express our sincere gratitude to the anonymous reviewers of Asiacrypt 2021 for their helpful comments and suggestions.

Chapter 6

On the Isogeny Problem with Torsion Point Information

This chapter is for all practical purposes identical to the eprint *On the Isogeny Problem with Torsion Point Information* [FKMT21], authored jointly with Péter Kutas, Simon-Philipp Merz and Yan Bo Ti, which will appear at PKC 2022.

6.1 — Introduction

Practical large scale quantum computers pose a threat to most cryptosystems currently in use [Gro96; Sho97]. Recent advances in quantum computing and the need for long-term security in cryptography has led to a surge of interest in developing quantum secure replacements for these classical cryptographic algorithms. Moreover, NIST has started a procedure to determine new cryptographic standards for a post-quantum era [Nat].

Most of the standardisation candidates are based on lattices, codes or multivariate polynomial systems over finite fields. A more recent but promising area of post-quantum research is isogeny-based cryptography.

Couveignes was the first one to mention isogenies for cryptographic use in 1997 [Cou06], and the area gained traction in the following decade with new developments such as collision-resistant hashing [CLG09] and key exchange [RS06; Sto10] based on isogeny problems. After Jao and De Feo introduced supersingular isogeny Diffie-Hellman (SIDH) [JD11], a predecessor of the isogeny-based submission to NIST's standardisation procedure SIKE [Jao+17], the area has enjoyed increasing popularity.

The central problem in most of isogeny-based cryptography is to find an isogeny $\varphi : E_1 \to E_2$, i.e. a morphism both in the sense of algebraic geometry and group theory, between two given supersingular elliptic curves defined over a finite field $\mathbb{F}_q$. For two supersingular elliptic curves E_1 and E_2, the problem of computing an arbitrary isogeny between them and the problem of computing their endomorphism rings $\mathrm{End}(E_1)$ and $\mathrm{End}(E_2)$ was revently proven to be equivalent [Wes21] under the assumption that the generalized Riemann hypothesis holds.

There are infinitely many isogenies $E_1 \to E_2$, but attacking isogeny-based primitives such as SIDH requires to recover an isogeny $\varphi : E_1 \to E_2$ of a specific degree. Generic algorithms are unlikely to return an isogeny of the correct degree given the endomorphism rings. In Section 4 of [GPST16], it is shown how to recover secret isogenies in the case of SIDH. The attack exploits the observation that secret isogenies in SIDH are of particularly small degree. In the case where the isogeny one wishes to recover is not of particularly small degree, as is the case in B-SIDH [Cos20],

SÉTA [SKPS19] or an instantiation of SIDH with secret isogenies of larger degree, this observation no longer holds and the algorithm due to Galbraith et al. no longer applies.

Chapter contributions. The results of this chapter provide a polynomial-time algorithm (assuming GRH) which recovers an isogeny between two supersingular elliptic curves of a specific degree N_1, given their endomorphism rings and some torsion point images under the isogeny. More precisely, let d be the least degree of any isogeny between two isogenous supersingular elliptic curves E_1 and E_2. Then, our algorithm solves the following problem, whenever $N_1 < dN_2/16$.

Task 6.1.1. *Let N_1, N_2 be coprime integers and let $\varphi : E_1 \to E_2$ be a secret isogeny of degree N_1 between two supersingular elliptic curves. Let P_B, Q_B be a basis of $E_1[N_2]$. Given End(E_1), End(E_2), $\varphi(P_B)$, and $\varphi(Q_B)$, find an isogeny $\varphi' : E_1 \to E_2$ of degree N_1 such that $\varphi_{|E_1[N_2]} = \varphi'_{|E_1[N_2]}$.*

Since SIDH-type schemes such as B-SIDH tend to use balanced parameters, where $N_1 \approx N_2$, the condition that $N_1 < dN_2/16$ is very mild.

The main idea behind the algorithm is the following. Isogenies from E_1 to E_2 form a $\mathbb{Z}$-module M of rank 4 and a basis of M can be computed using the KLPT algorithm [KLPT14]. Then, one computes an LLL-reduced basis $\psi_1, \psi_2, \psi_3, \psi_4$ of M. We show how to evaluate $\psi_i(P_B)$, $\psi_i(Q_B)$ for every i and we know $\phi(P_B)$ and $\phi(Q_B)$. Since $\phi = x_1\psi_1 + x_2\psi_2 + x_3\psi_3 + x_4\psi_4$ for some $x_i \in \mathbb{Z}$, this yields 4 linear equations in 4 variables, x_1, x_2, x_3, x_4, modulo N_2 (torsion-point images can be represented by a 2×2 matrix with entries from $\mathbb{Z}/N_2\mathbb{Z}$ and each entry corresponds to an equation). We will show that this system of equations has a unique solution for x_i modulo N_2 which we also compute. Since the ψ_i form an LLL-reduced basis, we can bound the absolute value of the coefficients x_i by $N_2/2$ for $N_1 < dN_2/16$. This leads to a unique solution for $x_i \in \mathbb{Z}$.

The contribution of this paper can be seen as an extension of the reductions by Kohel et al. [KLPT14] and Wesolowski [Wes21] which allow to compute an isogeny (of no specific degree) between two supersingular elliptic curves, whenever the endomorphism rings of the curves are known. Note that Kohel et al. provide a heuristic polynomial-time algorithm for this reduction, whereas Wesolowski shows that this reduction works in polynomial-time in general assuming the generalized Riemann hypothesis (GRH).

Together with known results on the computation of endomorphism rings, a consequence of this work is an answer to the open question how small the size of the prime p in B-SIDH can be chosen. More precisely, this work implies that one cannot lower the size of the prime p in B-SIDH significantly while maintaining the same security level. Current parameter sets are not threatened because parameters were selected in a cautious way (i.e., were larger than necessary if one only accounted for existing attacks). Our algorithm has better a similar classical runtime to a generic meet-in-the-middle algorithm but is essentially memory-free (whereas meet-in-the-middle requires an exponential amount of memory). Furthermore, the quantum version of our attack has a much better runtime than previously known quantum attacks ($O(p^{1/4})$[Eis+20] compared to $O(p^{1/2})$[JS19]). The running time of our algorithms is dominated by the computation of the endomorphism rings.

Outline. In Section 6.2, we recall some necessary mathematical background, details of the SIDH key exchange as well as some related work. In Section 6.3, we give algorithms to evaluate non-smooth degree isogenies and to compute an isogeny of a specific degree between two supersingular elliptic curves with known endomorphism ring, if certain torsion point information is available. Moreover, we discuss the impact of this work on isogeny-based cryptography before concluding the paper in Section 6.5.

6.2 — Preliminaries

In this section, we recall some relevant background on elliptic curves and isogeny-based cryptography. For further introductory reading, we refer to Silverman [Sil09] and De Feo [De 17] respectively. Furthermore, we briefly recall some consequences of the KLPT algorithm [KLPT14] and the LLL lattice reduction [LLL82]. Moreover, we sketch a related algorithm due to Galbraith et al. [GPST16] which computes an isogeny of specific degree between two supersingular elliptic curves with known endomorphism ring, if this degree is sufficiently small.

6.2.1 – Elliptic curves and isogenies. Let E_1, E_2 be elliptic curves defined over a field K. An isogeny between E_1 and E_2 is a non-constant rational map which is also a group homomorphism (or equivalently, fixes the point at infinity). The *degree* of an isogeny is its degree as a finite map of curves, i.e. the degree of the extension of function fields. An isogeny is called *separable* if the corresponding field extension is separable. For a separable isogeny, the degree equals the size of its kernel. Furthermore, for every finite subgroup G of an elliptic curve E, there exists a unique separable isogeny whose kernel is G. We denote the corresponding curve by E/G. Given a finite subgroup $G \subset E$ the corresponding isogeny from E to E/G can be computed using Vélu's formulae [Vél71].

Let $\phi : E_1 \to E_2$ be an isogeny of degree d. Then there exists a unique isogeny $\hat{\phi}$ with the property that $\phi \circ \hat{\phi} = [d]$, where $[d]$ denotes the multiplication by d. This isogeny $\hat{\phi}$ is called the *dual* of ϕ and it is also of degree d. An isogeny from E to itself is called an *endomorphism*. Endomorphisms of E form a ring under addition and composition denoted by $\mathrm{End}(E)$.

Let E be defined over a finite field of characteristic p. Then $\mathrm{End}(E)$ is either an order in an imaginary quadratic field and E is called *ordinary*, or a maximal order in the rational quaternion algebra $B_{p,\infty}$ ramified at p and at infinity in which case E is called *supersingular*. For the rest of the paper we will restrict ourselves to supersingular elliptic curves.

For an elliptic curve $E : y^2 = x^3 + Ax + B$, its *j-invariant* is given by $j(E) = 1728\frac{4A^3}{4A^3+27B^2}$ and two curves are isomorphic if and only if they share the same j-invariant.

Example 6.2.1. *For the supersingular elliptic curve $E_0 : y^2 = x^3 + x$ the above formula yields the j-invariant $j(E_0) = 1728$. It is well-known that $\mathrm{End}(E_0)$ is the $\mathbb{Z}$-module generated by $1, \iota, \frac{1+\pi}{2}$ and $\frac{\iota+\iota\pi}{2}$, where ι denotes E_0's non-trivial automorphism, $(x, y) \mapsto (-x, iy)$, and π is the Frobenius endomorphism, $(x, y) \mapsto (x^p, y^p)$.*

Let ℓ be a prime number and define the supersingular ℓ-isogeny *graph* as follows. The vertices of the graph are isomorphism classes of supersingular elliptic curves

represented by their j-invariant and two vertices are connected by an edge if and only if they are ℓ-isogenous. The supersingular ℓ-isogeny graph is connected, $(\ell+1)$-regular and a Ramanujan expander graph. The diameter of the graph is between $\log p$ and $2\log p$. The presumed hardness of path-finding in this graph is the hardness assumption underlying isogeny-based cryptography.

Remark 6.2.2. *In the rest of this paper we will call an integer smooth if its smoothness bound is polynomial in* $\log p$.

6.2.2 – SIDH and B-SIDH. We give a brief description of SIDH [JD11] and B-SIDH [Cos20] key exchanges.

The public parameters of SIDH are two coprime smooth numbers N_1 and N_2, a prime p of the form $p = N_1N_2f - 1$, where f is a small cofactor, and a supersingular elliptic curve E_0 defined over $\mathbb{F}_{p^2}$ together with points P_A, Q_A, P_B, Q_B such that $E_0[N_1] = \langle P_A, Q_A\rangle$ and $E_0[N_2] = \langle P_B, Q_B\rangle$.

The protocol proceeds as follows:

1. Alice chooses a random cyclic subgroup of $E_0[N_1]$ as $G_A = \langle P_A + [x_A]Q_A\rangle$ and Bob chooses a random cyclic subgroup of $E_0[N_2]$ as $G_B = \langle P_B + [x_B]Q_B\rangle$.

2. Alice and Bob compute the isogeny $\phi_A : E_0 \to E_0/\langle G_A\rangle =: E_A$ and the isogeny $\phi_B : E_0 \to E_0/\langle G_B\rangle =: E_B$, respectively.

3. Alice sends the curve E_A and the two points $\phi_A(P_B), \phi_A(Q_B)$ to Bob. Mutatis mutandis, Bob sends $\big(E_B, \phi_B(P_A), \phi_B(Q_A)\big)$ to Alice.

4. Alice and Bob use the given torsion points to obtain the shared secret $j(E_0/\langle G_A, G_B\rangle)$. To do so, Alice computes $\phi_B(G_A) = \phi_B(P_A) + [x_A]\phi_B(Q_A)$ and uses the fact that $E_0/\langle G_A, G_B\rangle \cong E_B/\langle\phi_B(G_A)\rangle$. Bob proceeds analogously.

In practice N_1 and N_2 are chosen to be powers of 2 and 3, respectively, to maximize the efficiency of the scheme. However, choosing a prime of the form $N_1N_2f - 1$ with $N_1 \approx N_2$ implies that the curves E_A, E_B are much closer at E_0 than the diameter of the supersingular isogeny graph, i.e. the paths connecting E_0 with E_A and E_B are shorter than one would expect for randomly chosen isogenous curves.

In order to avoid walking only in a small subgraph and to reduce the size of the prime p, Costello introduced the variant B-SIDH [Cos20]. The main differences between SIDH and B-SIDH are

- N_1 and N_2 are smooth coprime divisors of $p-1$ and $p+1$ (or vice versa) respectively. Hence, $p+1$ and $p-1$ both need to have large smooth factors as opposed to just one of them in SIDH.
- For the best parameter choice, we have $N_1 \approx N_2 \approx p$ as opposed to $N_1 \approx N_2 \approx \sqrt{p}$ in SIDH.
- Isogenies are a priori defined over $\mathbb{F}_{p^4}$ as opposed to $\mathbb{F}_{p^2}$.

In B-SIDH the curves E_0 and E_A are no longer closer than expected in the isogeny graph, but parameter selection might be harder and it seems at first to come at the expense of working over larger field extensions. However, to every supersingular

elliptic curve E defined over $\mathbb{F}_{p^2}$, there exists a quadratic twist (i.e., a curve defined over $\mathbb{F}_{p^2}$ which is isomorphic to E over $\mathbb{F}_{p^4}$ but not over $\mathbb{F}_{p^2}$). If E has $(p+1)^2$ rational points over $\mathbb{F}_{p^2}$, then its twist has $(p-1)^2$ rational points over $\mathbb{F}_{p^2}$. Thus, when computing an isogeny of degree N_1 dividing $p+1$ one can work with the curves having $p+1$ rational points, and before computing an isogeny of degree N_2 dividing $p-1$, one switches to twists that have $p-1$ rational points. Technically, the switch makes it possible to compute the isogenies using only operations over $\mathbb{F}_{p^2}$. For more details we refer to [Cos20].

6.2.3 – KLPT and LLL lattice reduction. In this subsection, we recall some facts about the Kohel-Lauter-Petit-Tignol (KLPT) algorithm [KLPT14] and the Lenstra-Lenstra-Lovász (LLL) lattice reduction [LLL82].

Let $B_{p,\infty}$ be the quaternion algebra ramified at p and at infinity. Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be maximal orders in $B_{p,\infty}$. Then the quaternion isogeny problem asks for a left ideal I connecting $\mathcal{O}_1$ and $\mathcal{O}_2$, i.e., a left ideal I of $\mathcal{O}_1$ which is also a right ideal of $\mathcal{O}_2$. By [KLPT14, Lemma 8], we have the following result.

Lemma 6.2.3. *Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be maximal orders in $B_{p,\infty}$. Then the intersection $\mathcal{O}_1 \cap \mathcal{O}_2$ has the same index M in $\mathcal{O}_1$ and $\mathcal{O}_2$. Furthermore,*

$$I(\mathcal{O}_1, \mathcal{O}_2) = \{\alpha \in B_{p,\infty} \mid \alpha\mathcal{O}_2\overline{\alpha} \subset M\mathcal{O}_1\}$$

is a left ideal of $\mathcal{O}_1$ and a right ideal of $\mathcal{O}_2$ of reduced norm M. $I(\mathcal{O}_1, \mathcal{O}_2)$ can be computed in polynomial time.

Lemma 6.2.3 shows that one can compute a connecting ideal between two maximal orders efficiently. However, this ideal will not have smooth norm in general. In [KLPT14], the main algorithm shows how to compute an equivalent left ideal of $\mathcal{O}_1$ of norm ℓ^k where ℓ is some small prime number.

Let E_1, E_2 be supersingular elliptic curves with endomorphism rings $\mathcal{O}_1$ and $\mathcal{O}_2$ respectively. Then isogenies from E_1 to E_2 are left $\mathcal{O}_1$-modules and right $\mathcal{O}_2$-modules. In particular, they form a $\mathbb{Z}$-lattice of rank 4 [Voi18, Lemma 42.1.11]. The $\mathbb{Z}$-lattice is isomorphic to a connecting left ideal I as an $\mathcal{O}_1$-module by the following lemma.

Lemma 6.2.4. *[Voi18, p. 42.2.7] Let $Hom(E_2, E_1)$ denote the set of isogenies from E_2 to E_1 and let $\mathcal{O}_1$ and $\mathcal{O}_2$ denote the endomorphism rings of E_1 and E_2 respectively. Let I be a connecting ideal of $\mathcal{O}_1$ and $\mathcal{O}_2$ and let ϕ_I denote the corresponding isogeny. Then the map $\phi_I^* : Hom(E_1, E_2) \to I$, $\psi \mapsto \psi \circ \phi_I$ is an isomorphism of left $\mathcal{O}_1$-modules.*

Since the KLPT-algorithm computes a connecting ideal between two maximal orders, Lemma 6.2.4 implies that one can compute a $\mathbb{Z}$-basis of $\mathrm{Hom}(E_1, E_2)$. However, the degree of these isogenies might not be smooth and it is not obvious that one can evaluate them efficiently. In Algorithm 5, we will show that one can evaluate these isogenies on points efficiently using the KLPT algorithm.

Next, we recall some basic facts about lattice reduction, which aims to transform an arbitrary input basis into a basis of "higher quality". In the following, we are interested in bases that are close to orthogonal.

Let $B := (b_1, \ldots, b_n)$ be the basis of a lattice L, let π_i denote the projection onto $\mathrm{span}(b_1, \ldots, b_{i-1})$ for $i = \{1, \ldots, n\}$ and let $B^* := (b_1^*, \ldots, b_n^*)$ be the *Gram-Schmidt orthogonalization* of B, where $b_i^* = \pi_i(b_i)$. Intuitively speaking, a good basis is one in which the sequence of Gram-Schmidt norms $\|b_1^*\|, \|b_2^*\|, \ldots, \|b_n^*\|$ does not decay too fast.

The Lenstra–Lenstra–Lovász (LLL) reduction calculates a short and nearly orthogonal lattice basis for any lattice in polynomial time [LLL82]. We recall a more precise statement in the following proposition using the Gram-Schmidt coefficients $\mu_{i,j} := \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}$.

Proposition 6.2.5. *The LLL lattice reduction with factors (η, δ), where $\delta \in (0.25, 1)$ and $\eta \in [0.5, \sqrt{\delta}]$, provides in polynomial time a basis $B = (b_1, \ldots, b_n)$ that is size-reduced with $\mu_{i,j} < \eta$ for all $j < i$ and has Gram-Schmidt orthogonalization satisfying the Lovász condition $\delta \|b_i^*\|^2 \leq \|\mu_{i+1,i} b_i + b_{i+1}^*\|^2$.*

The default parameters for LLL-reduction in MAGMA, which we will use later in this paper, are $\delta = 0.75$ and $\eta = 0.501$. Since LLL-reduced bases are in some sense close to orthogonal, we can expect short vectors in the lattice to have rather small coefficients with respect to the basis. This is captured by the following lemma which is a consequence of [LLL82, Equation (1.8)] and Cramer's rule.

Lemma 6.2.6. *Let L be a full rank lattice with LLL-reduced basis $b_1, \ldots, b_n$ with factors (η, δ) and let $v := \sum_{i=1}^{n} \gamma_i b_i \in L$. Then*

$$|\gamma_i| \leq \left(\frac{4}{(4\delta - 1)}\right)^{n(n-1)/4} \frac{|v|}{|b_i|}.$$

Proof. By [LLL82, Equation (1.8)], an LLL-reduced basis $b_1, \ldots, b_n$ satisfies

$$\prod_{i=1}^{n} |b_i| \leq \left(\frac{4}{(4\delta - 1)}\right)^{n(n-1)/4} \det(L).$$

Therefore, using Cramer's rule we get

$$\begin{aligned} |\gamma_i| &= \frac{\det(b_1, \ldots, b_{i-1}, v, b_{i+1}, \ldots, b_n)}{\det(L)} \leq \frac{|b_1| \cdots |b_{i-1}| \cdot |v| \cdot |b_{i+1} \cdots |b_n|}{\det(L)} \cdot \frac{|b_i|}{|b_i|} \\ &\leq \left(\frac{4}{(4\delta - 1)}\right)^{n(n-1)/4} \cdot \frac{|v| \cdot \det(L)}{|b_i| \cdot \det(L)} = \left(\frac{4}{(4\delta - 1)}\right)^{n(n-1)/4} \cdot \frac{|v|}{|b_i|}. \end{aligned}$$

□

6.2.4 – GPST. In [GPST16, §4], Galbraith, Petit, Shani and Ti describe how to compute the secret isogeny of an SIDH instance efficiently, if the endomorphism rings of both the domain and the codomain of the isogeny are known (or can be computed). We summarize their results and we recall why the algorithm does not work as such outside of an SIDH setting.

Let $\varphi : E_1 \to E_2$ be a ℓ^n-degree isogeny one wishes to recover, given the two endomorphism rings $\mathcal{O}_1$ and $\mathcal{O}_2$ of E_1 and E_2 respectively. Since E_1 and E_2 are

supersingular curves, their endomorphism rings are maximal orders in the rational quaternion algebra $B_{p,\infty}$. By Lemma 6.2.3, one can recover an ideal connecting $\mathcal{O}_1$ and $\mathcal{O}_2$. Such an ideal corresponds to one of infinitely many isogenies between E_1 and E_2. This isogeny is in general not of degree ℓ^n and, in particular, it is not the same as φ. Yet, to attack SIDH, the isogeny needs to be of the correct degree.

The secret isogenies in SIDH are of degree approximately $\sqrt{p}$. However, a pair of random supersingular elliptic curves over $\mathbb{F}_{p^2}$ is unlikely to be connected by an isogeny of degree significantly smaller than $\sqrt{p}$. In [GPST16] the authors leverage this observation to recover the sought isogeny given the endomorphism rings of E_1 and E_2 as follows.

Given a connecting ideal I for the endomorphism rings, the authors compute a Minkowski reduced basis which is used to recover an element $\alpha \in I$ of minimal norm. By [KLPT14, Lemma 5], the ideal $I' := I\overline{\alpha}/\operatorname{Norm}(I)$ is another ideal connecting $\mathcal{O}_1$ and $\mathcal{O}_2$ of minimal norm, $\operatorname{Norm}(\alpha)$. Then, one can compute the isogeny $E_1 \to E_2$ of degree $\operatorname{Norm}(\alpha)$ corresponding to this ideal using Vélu's formulae. If the shortest isogeny between E_1 and E_2 is indeed of degree ℓ^n, this algorithm allows to recover such an isogeny of correct degree from the endomorphisms. The experimental results in [GPST16] suggest that, by trying relatively few small elements α in the previous algorithm, one recovers an isogeny that can be used to attack SIDH with overwhelming probability.

Clearly, the approach outlined above relies crucially on the fact that the degree of the isogeny one wants to recover is among the smallest possible degrees of isogenies connecting E_1 and E_2. In schemes that do not use secret isogenies of unexpectedly short degree, e.g. in B-SIDH [Cos20], SÉTA [SKPS19], or if somebody was to instantiate SIDH with secret isogenies of larger degree, renders the GPST approach infeasible.

6.3 — Computing isogenies using torsion information

In this section, we describe an algorithm to evaluate non-smooth degree isogenies; and an algorithm to compute a secret isogeny $\phi : E_1 \to E_2$ of degree N_1 between supersingular elliptic curves, provided that certain torsion images and the endomorphism rings of E_1 and E_2 are known.

6.3.1 – Evaluating non-smooth degree isogenies. In this subsection, we provide an algorithm for the following problem

Task 6.3.1. *Let E_1 and E_2 be two curves with given endomorphism rings $\mathcal{O}_1$ and $\mathcal{O}_2$ respectively. Let I be an $\mathcal{O}_1$-left and $\mathcal{O}_2$-right ideal of norm N_1 and let $P \in E_1$. Evaluate $\phi_I(P)$, where ϕ_I is the isogeny corresponding to the ideal I.*

To solve this task, we extend an algorithm due to Petit and Lauter [PL17, Algorithm 3] which evaluates endomorphisms. Note that a solution to Task 6.3.1 evaluates isogenies of non-smooth degree between curves with known endomorphism rings.

Petit-Lauter Algorithm [PL17, Alg. 3]:. Let $(E_1, \mathcal{O}_1)$ denote a supersingular curve and its endomorphism ring, and let $w \in \mathcal{O}_1$. In order to evaluate the endomorphism $\phi_{w\mathcal{O}_1}$ on a point $P \in E_1$, the algorithm by Petit and Lauter uses a curve $(E_0, \mathcal{O}_0)$

whose endomorphisms can be efficiently evaluated, e.g. the curve with j-invariant 1728 (see Example 6.2.1). The algorithm proceeds as follows.

Let $\{w_1, w_2, w_3, w_4\}$ be a basis of $\mathcal{O}_0$ and let $\{\phi_1, \phi_2, \phi_3, \phi_4\}$ be the corresponding basis of $\mathrm{End}(E_0)$. The core idea of the algorithm is to use the KLPT algorithm to compute a powersmooth isogeny $\varphi : E_1 \to E_0$ of degree N.

Then, we have $N\mathcal{O}_1 \subset \mathcal{O}_0$ and thus $Nw \in \mathcal{O}_0$. For $w = \frac{a_1w_1+a_2w_2+a_3w_3+a_4w_4}{N}$ this implies

$$\phi_{w\mathcal{O}_1} = \varphi^{-1} \circ \frac{a_1\phi_1 + a_2\phi_2 + a_3\phi_3 + a_4\phi_4}{N} \circ \varphi,$$

where $\varphi^{-1} := \frac{1}{\deg\varphi}\widehat{\varphi}$. Since all the isogenies on the right-hand side can be evaluated efficiently, this allows to evaluate $\phi_{w\mathcal{O}_1}$.

Solving Task 6.3.1:. Let $(E_2, \mathcal{O}_2)$ be a supersingular elliptic curve with its endomorphism ring, let I be an $\mathcal{O}_1$-left and $\mathcal{O}_2$-right ideal of non-smooth norm and let $P \in E_1$. We would like to evaluate the isogeny ϕ_I corresponding to the ideal I at the point P.

Using the KLPT algorithm, we compute an $\mathcal{O}_1$-right and $\mathcal{O}_2$-left ideal J whose smooth norm is coprime to that of I. Then, the ideal IJ represents an endomorphism $w \in \mathcal{O}_1$ of E_1. The element $w \in O_1$ can be recovered by computing the shortest vector in IJ. We obtain $IJ = w\mathcal{O}_1$ for some $w \in \mathcal{O}_1$. Using [PL17, Algorithm 3], we evaluate $Q = \phi_{w\mathcal{O}_1}(P)$, and compute $\phi_I(P) = \phi_J^{-1}(Q)$. We summarize the steps in Algorithm 5.

Algorithm 5 Evaluating non-smooth degree isogenies

Require: Elliptic curves E_1, E_2 with endomorphism rings $\mathcal{O}_1, \mathcal{O}_2$ and an $\mathcal{O}_1$-left and $\mathcal{O}_2$-right ideal I together with a point $P \in E_1$, an elliptic curve E_0 such that its endomorphism ring $\mathcal{O}_0$ is generated by endomorphisms $\phi_1, \phi_2, \phi_3, \phi_4$ that can be evaluated efficiently.

Ensure: $\phi_I(P)$.

1: Compute an $\mathcal{O}_1$-right and $\mathcal{O}_2$-left ideal J whose smooth norm is coprime to that of I using KLPT algorithm;
2: Compute an $\mathcal{O}_1$-left and $\mathcal{O}_0$-right ideal K of powersmooth norm N using KLPT algorithm;
3: Set $IJ = w\mathcal{O}_1$ for some $w \in \mathcal{O}_1$ and find integers a_1, a_2, a_3 and a_4 such that $Nw = a_1w_1 + a_2w_2 + a_3w_3 + a_4w_4$;
4: Evaluate $Q = \phi_{IJ}(P) = \frac{\phi_K^{-1} \circ (a_1\phi_1+a_2\phi_2+a_3\phi_3+a_4\phi_4) \circ \phi_K(P)}{N}$ using [PL17, Alg. 3];
5: **return** $\phi_J^{-1}(Q)$

Lemma 6.3.2. *Algorithm 5 runs in polynomial time.*

Proof. Since the endomorphism rings of the curves E_0, E_1 and E_2 are known, the calls of the KLPT algorithm in Step 1 and Step 2 run in polynomial time. Note that the original KLPT algorithm runs in heuristic polynomial time, but Wesolowski showed that the reduction of KLPT is polynomial time assuming only GRH [Wes21].

The ideal I ($\mathcal{O}_1$-left and $\mathcal{O}_2$-right) and J ($\mathcal{O}_1$-right and $\mathcal{O}_2$-left) have coprime norms, hence the two-sided $\mathcal{O}_1$ ideal IJ corresponds to a non trivial endomorphism

$w \in \mathcal{O}_1$ of E_1 that can be recovered by computing a Minkowski reduced basis of IJ. For lattices up to dimension 4, a Minkowski reduced basis can be computed in polynomial time [NS04]. The integers a_1, a_2, a_3 and a_4 are obtained by rewriting the quaternion Nw as an element of $\mathcal{O}_0$. Therefore, Step 3 runs in polynomial time. By hypothesis, the isogenies ϕ_1, ϕ_2, ϕ_3 and ϕ_4 can be evaluated efficiently. The ideals K and J have smooth norm, hence the isogenies ϕ_K, ϕ_K^{-1} and ϕ_J^{-1} have smooth degree and can also be evaluated efficiently. It follows that Step 4 and Step 5 run in polynomial time as well. □

6.3.2 – Main algorithm. Next, we generalise Algorithm 2 of [GPST16]. There, an isogeny ϕ between two curves E_1 and E_2 with known endomorphism rings $\mathcal{O}_1$ and $\mathcal{O}_2$ is computed, if its degree is minimal, i.e., ϕ is the isogeny of smallest degree connecting E_1 and E_2. The algorithm in [GPST16] applies to the SIDH setting where the degree of the secret isogenies are minimal with non negligible probability (or otherwise at least particularly short). Meanwhile, the torsion point information available in SIDH-like schemes is not used at all.

We will show in this section how the torsion point information in SIDH-like schemes can be exploited together with the knowledge of endomorphism rings to compute secret isogenies of arbitrary (larger but fixed) degree.

The strategy is as follows. Let $\phi : E_1 \to E_2$ be a secret isogeny, let P, Q be a basis of $E_1[N_2]$ and let $\phi(P)$, $\phi(Q)$ be the torsion information provided in SIDH-like schemes. Let $I(\mathcal{O}_1, \mathcal{O}_2)$ be a connecting ideal between the maximal orders $\mathcal{O}_1$ and $\mathcal{O}_2$. Instead of solving for a minimal norm element of the ideal $I(\mathcal{O}_1, \mathcal{O}_2)$ as in [GPST16], we compute an LLL-reduced basis $\{\psi_1, \psi_2, \psi_3, \psi_4\}$ of I.

Using Algorithm 5, the isogenies ψ_i, $i = 1, \ldots, 4$, can be evaluated at the points P and Q. Next, we want to write ϕ in terms of our LLL-reduced basis, i.e. we want to find $(x_1, \ldots, x_4) \in (\mathbb{Z}/N_2\mathbb{Z})^4$ such that

$$\phi = x_1\psi_1 + x_2\psi_2 + x_3\psi_3 + x_4\psi_4. \tag{6.1}$$

Clearly, recovering x_i allows to compute the secret isogeny ϕ.

Note that Equation 6.1 implies in particular

$$\sum_{i=1}^{4} x_i\psi_i(P) = \phi(P) \qquad \text{and} \qquad \sum_{i=1}^{4} x_i\psi_i(Q) = \phi(Q). \tag{6.2}$$

To compute x_1, x_2, x_3 and x_4, we first prove that a solution to Equation 6.2 is unique modulo N_2. Then, we use simple linear algebra methods to recover it. Finally, we will show that knowing the x_i modulo N_2 is enough to recover them as integers.

Lemma 6.3.3. *Let E_1, E_2 be supersingular elliptic curves over $\mathbb{F}_{p^2}$ and let P, Q be a basis of $E_1[N_2]$. Let $\psi_1, \psi_2, \psi_3, \psi_4$ be a $\mathbb{Z}$-basis of $Hom(E_1, E_2)$. The system*

$$\sum_{i=1}^{4} x_i\psi_i(P) = \phi(P), \qquad \sum_{i=1}^{4} x_i\psi_i(Q) = \phi(Q)$$

has a unique solution $(x_1, x_2, x_3, x_4) \in (\mathbb{Z}/N_2\mathbb{Z})^4$.

Proof. Let P', Q' be a basis of $E_2[N_2]$. Every isogeny ϕ in $\mathrm{Hom}(E_1, E_2)$ can be identified with a matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in M_2(\mathbb{Z}/N_2\mathbb{Z})$ by writing its images on $E_1[N_2]$ as follows

$$\phi(P) = aP' + cQ', \ \phi(Q) = bP' + dQ'.$$

Let $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ be a matrix in $M_2(\mathbb{Z}/N_2\mathbb{Z})$. First, we prove that for any matrix A, there exists an isogeny $\phi \in \mathrm{Hom}(E_1, E_2)$ such that representation of ϕ is A.

Let $\psi : E_1 \to E_2$ be an isogeny such that the degree of ψ is coprime to N_2. Note that such an isogeny exists as the ℓ-isogeny graph is connected for any prime ℓ. Let M be the matrix corresponding to ψ. Since the degree of ψ is coprime to N_2, it corresponds to an invertible matrix in $M_2(\mathbb{Z}/N_2\mathbb{Z})$.

It is known (see [Voi18, Theorem 42.1.9.]) that $\mathrm{End}(E_1)/N_2\,\mathrm{End}(E_1)$ is isomorphic to $M_2(\mathbb{Z}/N_2\mathbb{Z})$ (the injection is clear, surjectivity is the key result). Note that the isomorphism depends on a choice of basis of $E_1[N_2]$. Consider the isomorphism corresponding to the basis P, Q. Then, there exists an endomorphism $\theta \in \mathrm{End}(E_1)$ whose matrix representation is AM^{-1}. This implies that the matrix representation of $\phi = \theta \circ \psi$ is $AM^{-1}M = A$, i.e. there exists an isogeny from E_0 to E_1 that is represented by the matrix A.

Clearly, $\sum_{i=1}^{4} x_i\psi_i$ and $\sum_{i=1}^{4} y_i\psi_i$ are represented by the same matrix if $x_i \equiv y_i \pmod{N_2}$ for $i = 1, \ldots, 4$. Thus, there are at most $N_2^4 = |(\mathbb{Z}/N_2\mathbb{Z})^4|$ different matrices that one can obtain.

Now, the Lemma follows by a simple counting argument. Since every matrix in $M_2(\mathbb{Z}/N_2\mathbb{Z})$ is represented for an isogeny, every matrix must uniquely correspond to a sum of the form $\sum_{i=1}^{4} x_i\psi_i$ modulo N_2. Consequently, if a matrix has two different representations of the form $\sum_{i=1}^{4} x_i\psi_i$, then they are the same modulo N_2 which finishes the proof.

□

Remark 6.3.4. *Essentially the main result of the proof is that Hom(E_1, E_2) modulo N_2 is isomorphic to $M_2(\mathbb{Z}/N_2\mathbb{Z})$ as a $\mathbb{Z}/N_2\mathbb{Z}$-module. Informally, the key idea is that Hom(E_1, E_2) is a left ideal in End(E_1), hence it will be a left ideal in $M_2(\mathbb{Z}/N_2\mathbb{Z})$ modulo N_2. Since isogenies between E_1 and E_2 of degree coprime to N_2 exist, this left ideal will contain invertible matrices, hence it must be the entire matrix ring.*

Now we provide details on how to recover x_1, x_2, x_3, x_4. Given $\psi_i(P)$, $\psi_i(Q)$ for $i = 1, 2, 3, 4$ and $\phi(P)$, $\phi(Q)$, where $\{\psi_1, \psi_2, \psi_3, \psi_3\}$ is the LLL-reduced basis of $\mathrm{Hom}(E_1, E_2)$, we would like to compute $(x_1, \cdots, x_4) \in (\mathbb{Z}/N_2\mathbb{Z})^4$ such that

$$\sum_{i=1}^{4} x_i\psi_i(P) = \phi(P), \qquad \sum_{i=1}^{4} x_i\psi_i(Q) = \phi(Q).$$

Note that N_2 is a smooth integer and that $\phi(P)$ and $\phi(Q)$ form a basis of $E_2[N_2]$ as $\deg(\phi)$ and N_2 are coprime. For $i = 1, 2, 3, 4$, we can compute the integers $a_i, b_i, c_i, d_i \in \mathbb{Z}/N_2\mathbb{Z}$ such that $\psi_i(P) = [a_i]\phi(P) + [b_i]\phi(Q)$ and $\psi_i(Q) = [c_i]\phi(P) + [d_i]\phi(Q)$ by using

the Weil pairing and solving discrete logarithms in a group of smooth order. Now, the integers $(x_1, \cdots, x_4) \in (\mathbb{Z}/N_2\mathbb{Z})^4$ satisfy

$$\phi(P) = \left[\sum_{i=1}^{4} x_i a_i\right] \phi(P) + \left[\sum_{i=1}^{4} x_i b_i\right] \phi(Q)$$

and

$$\phi(Q) = \left[\sum_{i=1}^{4} x_i c_i\right] \phi(P) + \left[\sum_{i=1}^{4} x_i d_i\right] \phi(Q).$$

We obtain

$$\begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix} \cdot \begin{pmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{pmatrix}.$$

By Lemma 6.3.3, there exists a unique solution $\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix}$ to the previous equation. Hence the matrix

$$M := \begin{pmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{pmatrix}$$

is invertible and the solution is given by $\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \cdot M^{-1}$. The latter operation corresponds to adding the first and the fourth row of M^{-1}. We summarize this process in Algorithm 6.

Algorithm 6 Computing the linear system

Require: $\psi_i(P)$ and $\psi_i(Q)$ for $i = 1, \ldots, 4$, where ψ_i are a $\mathbb{Z}$-basis of $\mathrm{Hom}(E_1, E_2)$; $\phi(P)$ and $\phi(Q)$ of smooth order N_2.
Ensure: x_1, x_2, x_3, x_4 such that $\sum_{i=1}^{4} x_i\psi_i(P) = \phi(P)$, and $\sum_{i=1}^{4} x_i\psi_i(Q) = \phi(Q)$.
1: **for** $i = 1, \cdots, 4$ **do**
2: Compute $a_i, b_i, c_i, d_i \in \mathbb{Z}/N_2\mathbb{Z}$ such that $\psi_i(P) = [a_i]\phi(P) + [b_i]\phi(Q)$ and
3: $\psi_i(Q) = [c_i]\phi(P) + [d_i]\phi(Q)$;
4: **end for**
5: Set M to be the 4×4 matrix whose rows are (a_i, b_i, c_i, d_i) for $i = 1, 2, 3, 4$;
6: Compute the inverse matrix M^{-1} of M;
7: Set (x_1, x_2, x_3, x_4) to be the sum of the first and the fourth rows of M^{-1};
8: **return** x_1, x_2, x_3, x_4 such that $|x_i| \leq N_2/2$.

Lemma 6.3.5. *Algorithm 6 is correct and runs in polynomial time provided that N_2 is smooth.*

Proof. Follows from the previous discussion. □

Lemma 6.3.6 gives a condition under which the solution computed in Algorithm 6 gives a solution to Equation 6.1.

Lemma 6.3.6. *Let* $d := \min\{\deg(\varphi) \mid \varphi : E_1 \to E_2 \text{ is isogeny}\}$. *If* $\frac{N_1}{N_2} < \frac{d}{16}$, *then given the solution* $x_1, \ldots, x_4$ *to* $\sum_{i=1}^{4} x_i\psi_i(P) = \phi(P)$, $\sum_{i=1}^{4} x_i\psi_i(Q) = \phi(Q)$ *returned by Algorithm 6, we have* $\phi = \sum_{i=1}^{4} x_i\psi_i$ *in* $Hom(E_1, E_2)$.

Proof. By Lemma 6.2.6, setting $\delta = 0.75$ and $n = 4$, we have that $\phi = \sum_{i=1}^{4} \gamma_i\psi_i$ where $|\gamma_i| \leq \frac{8\deg(\phi)}{\deg(\psi_i)} \leq \frac{8N_1}{d}$. It follows that $|\gamma_i| \leq \frac{8N_1}{d} < \frac{N_2}{2}$ since $\frac{N_1}{N_2} < \frac{d}{16}$ by hypothesis.

The solution $(x_1,\ x_2,\ x_3,\ x_4)$ returned by Algorithm 6 satisfies $|x_i| < \frac{N_2}{2}$ for $i = 1, 2, 3, 4$. Moreover, by Lemma 6.3.3, this solution is unique modulo N_2. Thus, $\phi = \sum_{i=1}^{4} x_i\psi_i$ in $\mathrm{Hom}(E_1, E_2)$. □

The entire process of computing isogenies of a specific but arbitrary degree between two supersingular curves with known endomorphism ring is summarised in Algorithm 7.

Algorithm 7 Computing isogeny with torsion-point information

Require: Supersingular elliptic curves E_1, E_2 with known endomorphism rings $\mathcal{O}_1, \mathcal{O}_2$ which are connected by an isogeny ϕ of degree N_1 and $\phi(P), \phi(Q)$, where P, Q are a basis of $E_1[N_2]$, such that $\frac{N_1}{N_2} < \frac{d}{16}$.

Ensure: ϕ.

1: Using KLPT, compute a basis of an $\mathcal{O}_1$-left and $\mathcal{O}_2$-right ideal I ;
2: Compute an LLL-reduced basis $\psi_1, \psi_2, \psi_3, \psi_4$ of I;
3: Compute $\psi_i(P), \psi_i(Q)$ using Algorithm 5 ;
4: Use Algorithm 6 to solve for $x_1, x_2, x_3, x_4 \in \mathbb{Z}/N_2\mathbb{Z}$ such that
5: $\sum_{i=1}^{4} x_i\psi_i(P) = \phi(P)$, $\sum_{i=1}^{4} x_i\psi_i(Q) = \phi(Q)$;
6: Compute isogeny from the relation $\phi = \sum_{i=1}^{4} x_i\psi_i$[1];
7: **return** ϕ.

Finally, we prove that Algorithm 7 succeeds in polynomial time.

Theorem 6.3.7. *Let* $d := \min\{\deg(\phi) \mid \phi : E_1 \to E_2 \text{ is isogeny}\}$. *Algorithm 7 solves Problem 6.1.1 in polynomial time, if* $\frac{N_1}{N_2} < \frac{d}{16}$.

Proof. Correctness of the algorithm follows from Lemma 6.3.6 and the preceding discussion. We are left to show the polynomial running time. Step 1 uses the reduction of the KLPT algorithm [KLPT14], which runs in polynomial time [Wes21]. Step 2 is the LLL lattice reduction algorithm which also runs in polynomial time. Step 3 and Step 4 run in polynomial time by Lemma 6.3.2 and Lemma 6.3.5 respectively. □

Remark 6.3.8. *We could also have required the condition* $\frac{N_1}{N_2} \leq \frac{d}{16}$ *and in that case we get the condition that* $|x_i| \leq N_2/2$. *However, when* N_2 *is even and* x_i *is congruent to* $N_2/2$, *then the lift to the above range is not unique (as* $-N_2/2$ *and* $N_2/2$ *represent*

[1]Note that this is an abstract representation of ϕ. In fact computing and writing down ϕ this way (as a sum of rational maps) is impossible since ϕ has large degree in general. Nevertheless, with this abstract notation, ϕ can be evaluated on any point. When its degree N_1 is smooth, its kernel can be recovered and ϕ can then be written down as a composition of isogenies of small degree.

the same residue class). This is not an issue for Algorithm 7 as one will have multiple candidates (16 of them in the worst case) for ψ that can be tested. By looking at the degrees, the correct one can be chosen efficiently. More generally, one can actually relax the statement of Theorem 6.3.7 further by allowing non-unique lifts and adding a check step at the end of Algorithm 7.

Remark 6.3.9. *As was shown in Lemma 6.3.6, Algorithm 7 requires an amount of torsion point information that depends on the degree d of the shortest isogeny between the supersingular elliptic curves E_1 and E_2.*

For many applications of cryptographic interest balanced parameters are used where $N_1 \approx N_2$. Taking $\frac{N_1}{N_2} \approx 1$, the procedure above works whenever the two curves are not connected by an isogeny of degree smaller than 16. This can be checked easily with an exhaustive search.

6.3.3 – Example. We will illustrate the attack with an example.
Consider the prime $p = 83701957499$, where we have $p + 1 = 2^2 \cdot 3^{14} \cdot 5^4 \cdot 7$. Let B be the quaternion algebra ramified at p and ∞ and generated over the rationals by i, j, k where $i^2 = -p$, $j^2 = -1$, and $k = ij$. Fix the finite field $\mathbb{F}_{p^2}$ where $\alpha^2 = -1$ generates $\mathbb{F}_{p^2}$ over $\mathbb{F}_p$.

Consider the elliptic curve given by $E_0 : y^2 = x^3 + x$ which has j-invariant 1728. The endomorphism ring of E_0 is generated by:

$$1, j, \frac{j+k}{2}, \frac{1+i}{2}.$$

We let the secret isogeny be a 3^{14}-isogeny $\theta : E_0 \to E$. We use θ to recover the endomorphism ring of E which is generated by

$$\frac{5159993 + i + 10319986j + 11800766447346k}{9565938}, \frac{2i + 6291065j + 7411685041437k}{9565938}, \frac{3j + 196249k}{2}, 1594323k.$$

Note that in the real attack, we have made the assumption that $\mathrm{End}(E)$ is known, so we have only used the secret to calculate a known quantity.

Now, using the knowledge of both endomorphism rings, we are able to compute a connecting ideal between them and also compute the reduced basis of the ideal to be

$$\frac{227049 + i + 154612j}{2}, \frac{154612 - 227049j + k}{2}, \frac{121127 - 9i + 4995744j + 14k}{2}, \frac{4995744 - 14i - 121127j - 9k}{2}.$$

We can interpret these endomorphisms and map the generators of the $E_0[5^4]$ through them.

We have chosen the points

$$P_5 = (75854242840\alpha + 62002351922, 51107649030\alpha + 19190692821),$$
$$Q_5 = (17857458337\alpha + 504604508, 77775481527\alpha + 25718537048)$$

to be the generators of $E_0[5^4]$.

In particular, by naming the reduced basis elements as $\psi_1, \psi_2, \psi_3, \psi_4$, we have that

$$\begin{aligned}
\psi_1(P_5) &= (9049577476\alpha + 26838535531, 9532248787\alpha + 18861270144)\\
\psi_1(Q_5) &= (14085392798\alpha + 75272963133, 35152660085\alpha + 3705843319)\\
\psi_2(P_5) &= (54148936824\alpha + 29574813, 27904476482\alpha + 79581351851)\\
\psi_2(Q_5) &= (6218706354\alpha + 14437916419, 19897519544\alpha + 26853032937)\\
\psi_3(P_5) &= (27253519435\alpha + 63921648196, 55371710596\alpha + 3587102479)\\
\psi_3(Q_5) &= (6221393886\alpha + 23453138168, 81414672111\alpha + 63571818133)\\
\psi_4(P_5) &= (20904892135\alpha + 45099774747, 32347928248\alpha + 14718113311)\\
\psi_4(Q_5) &= (16837240041\alpha + 11444980635, 5815630261\alpha + 82050564219)
\end{aligned}$$

Furthermore, we have the images of P_5 and Q_5 through the secret isogeny θ as given as part of the problem. Note that these ψ_i are not the same as the ones defined in the previous section as they are endomorphisms of E_0. However, they are just the original ψ_i composed with the isogeny between E_1 and E_0 coming from KLPT. We will denote the actual isogenies corresponding to them by ψ_i'. They can be evaluated at P_5 and Q_5 by applying the connecting isogeny to them and multiplying it with the inverse of its degree modulo 5^4. These are points in E, and in particular, they are in the subgroup $E[5^4]$. This allows us to express them in terms of $\theta(P_5)$ and $\theta(Q_5)$ which we are given.

This results in the following 4×4 matrix

$$\begin{pmatrix}
222 & 128 & 484 & 474\\
311 & 363 & 337 & 12\\
184 & 477 & 307 & 574\\
344 & 566 & 191 & 132
\end{pmatrix}$$

whose first row represents the four coefficients that expresses $\psi_1'(P_5)$ as a linear combination of $\theta(P_5)$ and $\theta(Q_5)$, and $\psi_1'(Q_5)$ as a linear combination of $\theta(P_5)$ and $\theta(Q_5)$. For example,

$$\psi_2'(Q_5) = [337]\theta(P_5) + [12]\theta(Q_5).$$

Inverting this matrix and summing the first and fourth rows allow us to recover the coefficients x_i's providing the expression of the secret isogeny as a linear combination of ψ_1', ψ_2', ψ_3' and ψ_4'. The result of the computation is that

$$\theta = 14\psi_1' + 9\psi_2' + \psi_4'.$$

One can check that this is correct without actually computing the ψ_i' by computing that the degree of this linear combination is indeed 3^{14} (as the action on the 5^4-torsion is already correct).

Remark 6.3.10. *As one can see in this example, the secret isogeny is not the shortest isogeny between E_0 and E, hence the algorithm from [GPST16] would not have been sufficient for finding θ.*

6.4 — Relevance to isogeny-based cryptography

We use this section to summarize how Algorithm 7 impacts different isogeny-based constructions.

First, we recall the current state-of-the-art regarding endomorphism ring computations as it is clearly the most time consuming part when attacking an isogeny-based cryptosystem using the reduction given by this paper.

Given a supersingular elliptic curve E defined over a finite field of characteristic p, the problem is to find $\mathrm{End}(E)$. The first algorithm to solve this is described in Kohel's thesis [Koh96] and was later improved by Delfs-Galbraith [DG16] to a running time of $\tilde{O}(p^{1/2})$. The most recent algorithm is due to Eisenträger et al. [Eis+20] which runs in time $O(\log(p)^2 p^{1/2})$. The best known quantum algorithm is due to Biasse, Jao and Sankar [BJS14] and has a running time of $\tilde{O}(p^{1/4})$.

The isogeny-based community for a long time considered the meet in the middle attack (MiTM) [Gal99] as best attack when addressing the security level of isogeny-based schemes. Meanwhile, this MiTM attack requires exponential storage, hence may be unrealistic. Recently, [Adj+18] and [Cos+20] considered the van Oorschot-Wiener (vOW) parallel collision finding algorithm [VW99] for the isogeny computation problem. The vOW collision search allows for a space-time trade-off in the generic MiTM, leading to a larger time complexity when limited storage is used. Estimating the security level of isogeny-based schemes using vOW, suggests that one can reduce the size of parameters that where previously fixed considering the generic MiTM attack with unrealistic memory requirements. For an SIDH-like scheme in which the secret isogenies have degree roughly N, the scheme is secured against the MiTM attack if $2^{2\lambda} < N$, where λ is the desired security level. When considering the vOW attack, N may be considerably smaller compared to $2^{2\lambda}$. See for instance a recent proposal for the reduction of parameters in SIKE by Longa et al. [LWS20]. However, one also needs to take the attack into account where one computes the endomorphism ring of curves and then uses Algorithm 7 to attack the secret isogeny. Given the classical and quantum complexity $O(\log(p)^2 p^{1/2})$ and $\tilde{O}(p^{1/4})$ respectively, this implies that the parameter p must also satisfy $2^{2\lambda} < p$.

Our attack applied against SIDH has complexity similar to the attack from [GPST16]. It does not effect parameter choices as SIDH isogenies are short and thus pathfinding algorithms are more efficient. Our algorithm has much more impact when isogeny degrees are larger (as the complexity of our algorithms depends on p and not on N_1). For B-SIDH, the proposed prime p is roughly $2^{2\lambda}$. Provided the new analysis of the vOW collision search attack in [LWS20], one may be tempted to propose smaller B-SIDH primes in order to improve on B-SIDH's efficiency. However, doing so would make the scheme vulnerable to attacks that compute endomorphism rings and use the results of this paper. This is because p would be smaller than $2^{2\lambda}$. Hence, one consequence of this paper is that the current choice of the parameter p in B-SIDH is tight. Furthermore, one can also interpret this result differently. Namely, any SIDH-like construction has to use parameters at least as large as B-SIDH, otherwise they become vulnerable to out attack. In other words, proposing schemes with longer isogeny walks than in B-SIDH does not provide any security benefit. This is not

unexpected, as walks in B-SIDH have lengths which are comparable to the diameter of the supersingular isogeny graph.

Another interpretation of our result is that when torsion point images are provided, then the problem of finding one isogeny between two supersingular elliptic curves becomes equivalent to finding an isogeny of a specific degree for a wide range of parameters.

6.5 — Conclusion

In this chapter, we showed how to compute an isogeny of a specific degree between two supersingular elliptic curves, given their endomorphism rings and the images of some torsion points through the isogeny. This can be seen as an extension of an algorithm due to Galbraith et al. [GPST16] which did not use torsion point information but required the isogeny to be of small degree.

As a consequence, this paper allows us to estimate the security of schemes like B-SIDH, SÉTA and SIDH instantiated with larger degree isogenies when considering an attack that computes endomorphism rings. In particular, our work provides a significant speed-up to existing quantum attacks on B-SIDH. We stress that this work does not allow to break any of the recommended parameter sets. However, our work shows that the prime chosen in B-SIDH cannot be lowered for the given security levels and also implies that any (reasonable) scheme that provides torsion point images has to use a 2λ-bit prime for security level λ (making B-SIDH the most compact construction that uses torsion point images).

Acknowledgements. We would like to thank Craig Costello for his useful comments on a previous draft.

Chapter 7

A New Adaptive Attack on SIDH

This chapter is for all practical purposes identical to the paper *A New Adaptive Attack on SIDH* [FP21a], authored jointly with Christophe Petit, which will appear at CT-RSA 2022.

7.1 — Introduction

The first isogeny-based cryptographic schemes are the CGL (Charles-Goren-Lauter) hash function [CLG09] and the CRS (Couveignes-Rostovtsev-Stolbunov) key exchange [RS06; Cou06]. The CRS scheme is a Diffie-Hellman type key exchange scheme using isogenies of ordinary elliptic curves. It is vulnerable to a sub-exponential quantum hidden shift like attack [CJS14] and is not practically efficient.

In 2011, Jao and De Feo proposed SIDH [JD11; FJP14] that uses isogenies of supersingular elliptic curves. SIDH is efficient and it is not vulnerable to the sub-exponential quantum attack presented in [CJS14]. Nevertheless, a recent paper by Kutas et al. [KMPW21] proves that hidden shift like attacks apply to variants of SIDH with considerably overstretched parameters. The problem of computing isogenies between given supersingular elliptic curves is arguably new in cryptography. Its relation with the supersingular endomorphism ring computation problem have been studied in [PL17; Eis+18]. A rigorous proof (under the GRH) of the equivalence between the two problems was recently proposed by Wesolowski [Wes21].

Contrarily to the ordinary case where isogenies commute, supersingular isogenies do not commute in general. In order to solve this issue in SIDH, the images of some well-chosen torsion points through the secret isogeny are computed and included in the public keys. This implies that the hard problem underlying the security of SIDH is different from the general supersingular isogeny problem. Moreover, this torsion points have been used in designing adaptive and passive attacks on SIDH and/or its (unbalanced) variants.

The most relevant adaptive attack (excluding side channel attacks) on SIDH is due to Galbraith, Petit, Shani and Ti (GPST) [GPST16]. They suppose that one honest party Alice uses a static secret key, and the other malicious party Bob performs multiple key exchanges with Alice. The main idea of the attack is that Bob replaces the images of the torsion points in his public key by malicious ones and obtains some information on Alice's static secret isogeny when looking at the obtained shared secret. Repeating this process a polynomial number of times, Bob totally recovers Alice's private key. The pairing-based key validation method present in SIDH does not detect the GPST adaptive attack. In SIKE [Jao+20] (Supersingular Isogeny Key

Encapsulation), the GPST adaptive attack is avoided by leveraging SIDH with a variant [HHK17] of the Fujisaki-Okamoto transform [FO99].

The first passive torsion points attacks are due to Petit [Pet17] and were recently improved by de Quehen et al. [Que+21]. These attacks combine the availability of the endomorphism ring of the starting curve E_0 in SIDH and the torsion point information available in SIDH public keys, to compute a suitable endomorphism of Alice's public curve E_A. The secret isogeny is then recovered using the later endomorphism. For sufficiently unbalanced SIDH parameters (the degrees of the secret isogenies of the parties are of different size), the latest version of the attack [Que+21] is more efficient compared to the generic meet in the middle and the van-Oorschot - Wiener (vOW) attack [VW99]. For balanced parameters (the degrees of the secret isogenies of both parties are approximately of the same size), the quantum version of the attack is as efficient as the best known quantum attacks [Que+21, Figure 1]. Other passive attacks exploiting the availability of torsion points in the public key are described in [FKMT21; KMPW21].

The improved torsion points attacks do not apply to SIDH and BSIDH parameters since these parameters are balanced. Therefore, one may argue that they are not relevant to SIDH, BSIDH or any other SIDH like schemes using balanced isogenies degrees.

Contributions. The contribution of this chapter is twofold.

First, we revisit the torsion point attacks. The torsion points attacks are used to recover a secret isogeny $\phi : E_0 \to E$ of degree N_A when the images of torsion points of order N_B in E_0 are provided. We prove that one can tweak the algorithm in such a way that it recovers ϕ when only the images of three cyclic disjoint groups $G_1, G_3, G_3 \subset E_0[N_B]$ of order N_B are provided. This constitutes a generalisation of the torsion point attacks and will be useful in the design of our adaptive attack.

Secondly, we design a new adaptive attack on SIDH-types schemes, including BSIDH. Our attack uses torsion point attacks as a subroutine.

Let $\phi_A : E_0 \to E_A$ be Alice's secret static isogeny in an SIDH instance. Let N_A and N_B be the isogeny degrees of Alice and Bob respectively. Our attack actively recovers the images through ϕ_A of three pairwise disjoint cyclic groups $G_1, G_2, G_3 \subset E_0[NN_B]$ of order N_BN where N is a well chosen integer coprime to N_A. This leads to an unbalanced SIDH instance for which the torsion point attacks can be used to recover the secret isogeny in polynomial time.

Our attack differs from the GPST adaptive attack as follows. In the GPST adaptive attack, the malicious Bob computes isogenies of correct degrees N_B and manipulates torsion points images. Our attack consists of computing isogenies of degrees larger than N_B and scaling the torsion point images by a suitable scalar to make the public key pass the pairing-based key validation method in SIDH. One then utilises the torsion points attack to recover the secret.

We prove that our attack runs in polynomial time. We provide specific attack parameters for SIDH primes \$IDHp182, \$IDHp217, SIDHp377, SIDHp434, SIDHp503 and SIDHp546. For these SIDH primes, the attack fully recovers Bob's secret isogeny querying a few tens of thousand times the key exchange oracle. Determining specific attack parameters for BSIDH primes is computationally intensive. We only give an example of generic attack parameters for the smallest BSIDH prime. We

suggest countermeasures among which the Fujisaki-Okamoto transform (as used in SIKE), using SIDH proof of isogeny knowledge as recently proposed in [FDGZ21] or setting the starting curve in SIDH to be a random supersingular curve with unknown endomorphism ring.

The torsion point attacks do not apply to SIDH parameters [Que+21, §1.1 Figure 1] since they do not (yet) outperform generic passive attacks such as the meet in the middle on SIDH parameters. This attack comes as an ice breaker. This result, despite being less efficient when compared to the GPST adaptive attack, it proves that the torsion point attacks become relevant to SIDH and BSIDH parameters in an adaptive attack setting. Moreover, this attack vector is the first of its kind. It exploits the fact that in an SIDH instance, the pairing check does not suffices to convince Alice that Bob effectively computed an isogeny of degree N_B. We believe this attack fosters the understanding of SIDH and is a new cryptanalytic tool for isogeny based cryptography.

Outline. The remaining of this chapter is organized as follows: in Section 7.2, we recall some generalities about elliptic curves and isogenies. We briefly present SIDH and the GPST adaptive attack. In Section 7.3, we present the torsion point attacks and describe our generalisation. In Section 7.4 we present an overview of our attack and describe the active phase. We also discuss the computation of the attack parameters and summarize the attack. In Section 7.5, we suggest attack parameters for some SIDH primes and we briefly describe some countermeasures. We conclude the paper in Section 7.6.

7.2 — Preliminaries

7.2.1 – Elliptic curves and isogenies. An elliptic curve is a rational smooth curve of genus one with a distinguished point at infinity. Elliptic curves can be seen as commutative groups with respect to a group addition having the point at infinity as neutral element. When an elliptic curve E is defined over a finite field $\mathbb{F}_q$, the set of $\mathbb{F}_q$-rational points $E(\mathbb{F}_q)$ of E is a subgroup of E. For every integer N coprime with q, the N-torsion subgroup $E[N]$ of E is isomorphic to $\mathbb{Z}_N \oplus \mathbb{Z}_N$.

An isogeny from E to E' is a rational map from E to E' which is also a group morphism. The kernel of an isogeny is always finite and entirely defines the isogeny up to powers of the Frobenius. Given a finite subgroup G of E, there exists a Frobenius free isogeny of domain E having kernel G, called a separable isogeny. Its degree is equal to the size of its kernel. The co-domain of this isogeny is denoted by E/G. The isogeny and the co-domain E/G can be computed from the knowledge of the kernel using Vélu's formulas [Sil09] whose efficiency depends on the smoothness of the isogeny degree.

An endomorphism of an elliptic curve E is an isogeny from E to E. The structure of E is closely related to that of its endomorphism ring. When E is defined over a finite field, the endomorphism ring of E is either an order in a quadratic field, in which case we say E is ordinary, or a maximal order in a quaternion algebra in which case we say E is supersingular. The generic isogeny problem is harder to solve for supersingular curves (for which the best attacks are exponential) than ordinary curves (for which there exists a sub-exponential attack [BJS14]). SIDH is based on supersingular isogenies.

7.2.2 – SIDH: Supersingular Isogeny Diffie-Hellman. The SIDH scheme is defined as follows.

Setup. Let $p = \ell_A^{e_A}\ell_B^{e_B} - 1$ be a prime such that $\ell_A^{e_A} \approx \ell_B^{e_B} \approx \sqrt{p}$. Let E_0 be a supersingular curve defined over $\mathbb{F}_{p^2}$. Set $E_0[\ell_A^{e_A}] = \langle P_A, Q_A\rangle$ and $E_0[\ell_B^{e_B}] = \langle P_B, Q_B\rangle$. The public parameters are E_0, p, ℓ_A, ℓ_B, e_A, e_B, P_A, Q_A, P_B, Q_B.

Key Generation. The secret key sk_A of Alice is a uniformly random integer α sampled from $\mathbb{Z}_{\ell_A^{e_A}}$. Compute the cyclic isogeny $\phi_A : E_0 \to E_A = E_0/\langle P_A + [\alpha]Q_A\rangle$. The public key of Alice is the tuple $\mathsf{pk}_A = (E_A, \phi_A(P_B), \phi_A(Q_B))$. Analogously, Bob's secret key sk_B is a uniformly random integer β sampled from $\mathbb{Z}_{\ell_B^{e_B}}$ and his public key is $\mathsf{pk}_B = (E_B, \phi_B(P_A), \phi_B(Q_A))$ where $\phi_B : E_0 \to E_B = E_0/\langle P_B + [\beta]Q_B\rangle$.

Key Exchange. Upon receiving Bob's public key (E_B, R_a, S_a), Alice checks[1] that $e(R_a, S_a) = e(P_A, Q_A)^{\ell_B^{e_B}}$, if not she aborts. She computes the isogeny $\phi'_A : E_B \to E_{BA} = E_B/\langle R_a + [\alpha]S_a\rangle$. Her shared key is $j(E_{BA})$. Similarly, upon receiving (E_A, R_b, S_b), Bob checks that $e(R_b, S_b) = e(P_B, Q_B)^{\ell_A^{e_A}}$, if not he aborts. He computes the isogeny $\phi'_B : E_A \to E_{AB} = E_A/\langle R_b + [\beta]S_b\rangle$. His shared key is $j(E_{AB})$.

The correctness of the key exchange follows from the fact that

$$E_A/\langle \phi_A(P_B) + [\beta]\phi_A(Q_B)\rangle \simeq E_0/\langle P_A + [\alpha]Q_A, P_B + [\beta]Q_B\rangle \simeq E_B/\langle \phi_B(P_A) + [\alpha]\phi_B(Q_A)\rangle\,.$$

The scheme is summarized in Figure 7.1.

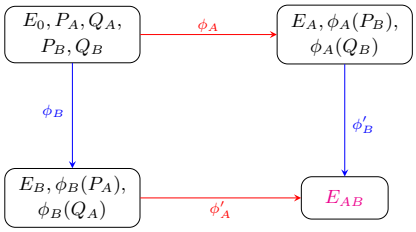


Figure 7.1: SIDH Key Exchange

The security of the SIDH key exchange protocol against shared key recovery relies on Problem 7.2.1. Furthermore, Problem 7.2.2 states that it is difficult to distinguish the shared secret from a random supersingular elliptic curve.

Problem 7.2.1 (Supersingular Isogeny Computational Diffie-Hellman)**.** *Given E_0, P_A, Q_A, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, E_B, $\phi_B(P_A)$, $\phi_B(Q_A)$ (defined as in SIDH), compute E_{AB}.*

Problem 7.2.2 (Supersingular Isogeny Decisional Diffie-Hellman)**.** *Given E_0, P_A, Q_A, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, E_B, $\phi_B(P_A)$, $\phi_B(Q_A)$ (defined as in SIDH) and a random supersingular curve E, distinguish between $E = E_{AB}$ and $E \neq E_{AB}$.*

In the rest of this paper, we denote by N_A and N_B the degree of Alice's and Bob's isogeny respectively. Since we will be supposing that Alice is honest and Bob

[1] Note that in the original SIDH [JD11], this pairing check is not part of the scheme. But, as precised in [CLN16] and [GPST16], one includes the check to discard some malformed public keys.

is potentially malicious, Alice's public key will be $(E_A, \phi_A(P_B), \phi_A(Q_B))$ while Bob's will be (E_B, R, S).

7.2.3 – GPST adaptive attack. In SIDH [FJP14] one does a pairing-based check on the torsion points $\phi_B(P_A)$ and $\phi_B(Q_A)$ returned by a potentially malicious Bob. Let E be a supersingular elliptic curve, let N be an integer and let μ_N be the group of N-roots of unity. Let $e_N : E[N] \times E[N] \to \mu_N$ be the Weil pairing [Gal12]. Let $\phi : E \to E'$ be an isogeny of degree M, then for $P, Q \in E[N]$,

$$e_N(\phi(P), \phi(Q)) = e_N(P, Q)^M$$

where the first pairing is computed on E' and the second one on E.
In SIDH, given (E_B, R_a, S_a) returned by Bob as public key, Alice checks if

$$e_{\ell_A^{e_A}}(R_a, S_a) = e_{\ell_A^{e_A}}(P_A, Q_A)^{\ell_B^{e_B}}.$$

As we will see below, this verification does not assure that the points R, S were honestly generated. More precisely, the pairing verification does not capture the GPST adaptive attack.

The GPST adaptive attack. The main idea of the Galbraith et al. adaptive attack [GPST16] is that if Bob manipulates the torsion points $\phi_B(P_A)$ and $\phi_B(Q_A)$ conveniently, then he can get some information about Alice's private key α given that he knows if the secret curve computed by Alice is equal to E_{AB} or not. Hence in the attack scenario, Bob needs to have access to the later information. This access is provided to Bob through a key exchange oracle:

$$O(E, R, S, E') \text{ which returns 1 if } j(E') = j(E/\langle R + [\alpha]S\rangle) \text{ and 0 otherwise}$$

If one supposes that $\ell_A = 2$ and $e_A = n$, then after each query, Bob recovers one bit of

$$\alpha = \alpha_0 + 2^1\alpha_1 + 2^2\alpha_2 + \cdots + 2^{n-1}\alpha_{n-1}.$$

Concretely, let us suppose that Bob has successfully recovered the first i bits of α, say $K_i = \alpha_0 + 2^1\alpha_1 + \cdots + 2^{i-1}\alpha_{i-1}$ so that

$$\alpha = K_i + 2^i\alpha_i + 2^{i+1}\alpha'$$

He generates $(E_B, \phi_B(P_A), \phi_B(Q_A))$ and computes the resulting key E_{AB}. To recover α_i, he chooses suitable integers a, b, c, d and queries the oracle O on (E_B, R, S, E_{AB}) where $R = [a]\phi_B(P_A) + [b]\phi_B(Q_A)$ and $S = [c]\phi_B(P_A) + [d]\phi_B(Q_A)$. The integers a, b, c and d are chosen to satisfy the following conditions:

1. if $\alpha_i = 1$, $\langle R + [\alpha]S\rangle = \langle \phi_B(P_A) + [\alpha]\phi_B(Q_A)\rangle$;
2. if $\alpha_i = 0$, $\langle R + [\alpha]S\rangle \neq \langle \phi_B(P_A) + [\alpha]\phi_B(Q_A)\rangle$;
3. the Weil paring $e_{2^n}(R, S)$ must be equal to $e_{2^n}(\phi_B(P_A), \phi_B(Q_A))$

The first two conditions help to distinguish the bit α_i. The third one prevents the attack from being detected by the pairing-based check presented in Section 7.2.3. When attacking the ith bit of α where $1 \leq i \leq n-2$, the attack uses the integers

$$a = \theta, \quad b = -\theta 2^{n-i-1} K_i, \quad c = 0, \quad d = \theta(1 + K_i 2^{n-i-1})$$

where $\theta = \sqrt{(1+2^{n-i-1})^{-1}}$. The attack recovers the first $n-2$ bits of α using $n-2$ oracle queries, and it recovers the two remaining bits by brute force. We refer to [GPST16] for more details.

The GPST adaptive attack exploits the fact that the pairing check does not convince Alice that the torsion points returned by Bob were honestly computed. In the rest of this paper, we will design a new adaptive attack that exploits the fact that the pairing check does not convince Alice that Bob effectively computed an isogeny of degree N_B.

7.3 — Generalizing torsion points attacks

In this section, we revisit the torsion point attacks. Firstly, we describe the torsion point attacks. Next, we provide a generalisation of these attacks that can be used to solve weaker version of Problem 7.3.1.

7.3.1 – Torsion points attacks on SIDH. The direct key recovery attack (attacking one party's secret key) in SIDH translates into solving the following *Supersingular Isogeny Problem.*

Problem 7.3.1. *Let N_A and N_B be two integers such that $\gcd(N_A, N_B) = 1$. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Set $E_0[N_B] = \langle P, Q \rangle$ and let $\phi : E_0 \to E$ be a random isogeny of degree N_A. Given E_0, E, P, Q, $\phi(P)$ and $\phi(Q)$, compute ϕ.*

The difference between Problem 7.3.1 and the general isogeny problem is the fact that the action of ϕ on the group $E_0[N_B]$ is revealed. In 2017, Petit [Pet17] exploited these torsion point images and the knowledge of the endomorphism ring of the starting curve E_0 to design an algorithm that solves Problem 7.3.1 for a certain choice of unbalanced ($N_A \ll N_B$) parameters. Petit's attack has recently been considerably improved by de Quehen et al. [Que+21].

The idea of the torsion points attacks if to find a trace 0 endomorphism $\theta \in \mathrm{End}(E_0)$ that can be efficiently evaluated on $E_0[N_B]$, an integer d and a small smooth integer e such that

$$N_A^2 \deg \theta + d^2 = N_B^2 e. \tag{7.1}$$

Writing Equation 7.1 in terms of isogenies we get

$$\phi \circ \theta \circ \widehat{\phi} + [d] = \psi_2 \circ \psi_e \circ \psi_1 \tag{7.2}$$

where ψ_1 and ψ_2 are isogenies of degree N_B, ψ_e is an isogeny of degree e. The torsion point information $\phi(P)$, $\phi(Q)$ is used to evaluate $\tau = \phi \circ \theta \circ \widehat{\phi} + [d]$ on $E[N_B]$. Knowing τ on $E_0[N_B]$, the kernels of the isogenies $\psi_1 : E \to E_1$ and $\widehat{\psi_2} : E \to E_2$ can be recovered efficiently. The isogeny $\psi_e : E_1 \to E_2$ is recovered by brute force or meet in the middle. We refer to [Que+21, § 4.1] for technical details.

Having computed $\psi_1 \circ \psi_e \circ \psi_2$, one recovers

$$\ker \widehat{\phi} = \ker \left(\psi_2 \circ \psi_e \circ \psi_1 - [d]\right) \cap E[N_A].$$

Figure 7.2 illustrates the attack.

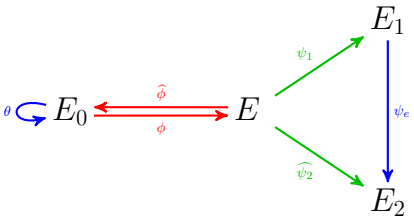


Figure 7.2: Improved torsion points attack.

The efficiency of torsion point attacks mostly depends on the imbalance between the isogeny degree N_A and the order N_B of the torsion points images.

de Quehen et al. [Que+21] show that under some heuristics, when $j(E_0) = 1728$, Problem 7.3.1 can be solved in:

1. Polynomial time when: $N_B > pN_A$ and $p > N_A$;
2. Superpolynomial time but asymptotically more efficient than meet-in-the-middle on a classical computer when: $N_B > \sqrt{p}N_A$;
3. Superpolynomial time but asymptotically more efficient than quantum claw-finding [JS20] when: $N_B > \max\{N_A, \sqrt{p}\}$.

More concretely, if $N_A \approx p^\alpha$ and $N_B \approx N_A p^\eta$, then the improved torsion points attack runs in time $\tilde{O}\left(N_A^{\frac{1+2(\alpha-\eta)}{4\alpha}}\right)$ and $\tilde{O}\left(N_A^{\frac{1+2(\alpha-\eta)}{8\alpha}}\right)$ on a classical computer and a quantum computer respectively [Que+21, §6.2 Proposition 27]. In the special case where $\alpha = \frac{1}{2}$, we get the following corollary.

Corollary 7.3.2. *Suppose that $N_A \approx p^{\frac{1}{2}}$ and $N_B \approx p^{\frac{1}{2}+\eta}$ where $1 \leq \eta$. Under some heuristics, [Que+21, Algorithm 7] solves Problem 7.3.1 in polynomial time.*

Remark 7.3.3. *SIKE parameters (for which E_0 is close to a curve having j-invariant 1728 and $N_A \approx N_B \approx \sqrt{p}$) are not affected by these improved torsion points attacks. Also, the attack does not affect any SIDH-type scheme in which the starting curve E_0 is a random supersingular curve with unknown endomorphism ring.*

In our attack setting, we will not be provided with the images of torsion points through isogenies, but with the images of cyclic torsion groups. In the next section, we generalize the torsion point attacks such that they directly apply to our setting.

7.3.2 – Generalized torsion points attacks. We consider the following problem.

Problem 7.3.4. *Let N_A and N_B be two integers such that* $\gcd(N_A, N_B) = 1$. *Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Let G_1, G_2, G_3 be three pairwise disjoint cyclic groups of E_0 of order N_B. Let $\phi : E_0 \to E$ be a random isogeny of degree N_A.*

Given E_0, G_1, G_2, G_3, E, $\phi(G_1)$, $\phi(G_2)$ and $\phi(G_3)$, compute ϕ.

The difference between Problem 7.3.4 and Problem 7.3.1 is the way the torsion point information is provided. In Problem 7.3.1, image points of a basis of the N_B-torsion group are given, while in Problem 7.3.4, only the images of three cyclic disjoint groups of order N_B are provided. This a priori represents less information, but as we show below, this is sufficient to run the improved torsion point attacks.

Let θ, d and e be such that Equation 7.1 is satisfied, set $\tau = \phi \circ \theta \circ \widehat{\phi} + [d]$. Let G_1, G_2 and G_3 be as in Problem 7.3.4. In the improved torsion point attacks, the torsion point information $(\phi(P), \phi(Q))$ is solely used to recover the action of τ on $E[N_B]$ as explained in Section 7.3.1. Hence we only need to prove that the knowledge of $\phi(G_1)$, $\phi(G_2)$ and $\phi(G_3)$ is sufficient to evaluate τ on $E[N_B]$.

First we prove that from the action of ϕ on 3 cyclic disjoint groups of order N_B, we can recover the image of a basis of $E_0[N_B]$ through $[\lambda] \circ \phi$ for some integer λ coprime to N_B. Concretely, we have the following lemma.

Lemma 7.3.5. *Let $\phi : E_0 \to E$ an isogeny of degree N_A and let N_B be a smooth integer coprime to N_A. Let $G_1 = \langle P_1 \rangle$, $G_2 = \langle P_2 \rangle$, $G_3 = \langle P_3 \rangle$ be three pairwise disjoint cyclic groups of E_0 of order N_B. Given $H_1 = \langle Q_1 \rangle$, $H_2 = \langle Q_2 \rangle$, $H_3 = \langle Q_3 \rangle$ such that $\phi(G_i) = H_i$ for $i = 1, 2, 3$; there exists an integer $\lambda \in (\mathbb{Z}/N_B\mathbb{Z})^\times$ such that we can compute λ^2 and $[\lambda] \circ \phi(P)$ for any $P \in E_0[N_B]$.*

The result in Lemma 7.3.5 partially available in [Bas+21, Lemma 1 §3.2] where Basso et. al prove that from the action of ϕ on 3 well chosen cyclic groups of smooth order N_B, one can recover the action of ϕ on any group of order N_B. Our Lemma goes a bit further and proves that we can evaluate $[\lambda] \circ \phi$ on the N_B torsion for some $\lambda \in (\mathbb{Z}/N_B\mathbb{Z})^\times$ such that λ^2 is known. Note that knowing λ^2 does not always enable us to compute λ, since when N_B is not a prime power, the equation $x^2 \equiv a^2 \mod N_B$ may have more than two solutions.

Proof of Lemma 7.3.5. For $i = 1, 2, 3$, set $\phi(P_i) = [\lambda_i]Q_i$ where $\lambda_i \in (\mathbb{Z}/N_B\mathbb{Z})^\times$. Since $G_1 \cap G_2 = \{0\}$, then $\{P_1, P_2\}$ is a basis of $E_0[N_B]$ and $\{Q_1, Q_2\}$ is a basis of $E[N_B]$. Write $P_3 = [v_1]P_1 + [v_2]P_2$ and $Q_3 = [u_1]Q_1 + [u_2]Q_2$. Then, we get

$$[\lambda_3 u_1]Q_1 + [\lambda_3 u_2]Q_2 = [\lambda_3]Q_3 = \phi(P_3) = [v_1]\phi(P_1) + [v_2]\phi(P_2) = [v_1\lambda_1]Q_1 + [v_2\lambda_2]Q_2.$$

Hence $\lambda_3 u_1 = v_1\lambda_1$, $\lambda_3 u_2 = v_2\lambda_2$ and $\lambda_i/\lambda_3 = u_i/v_i$ for $i = 1, 2$. Since $G_1 \cap G_3 = G_2 \cap G_3 = \{0\}$ and N_A is coprime to N_B, then $H_1 \cap H_3 = H_2 \cap H_3 = \{0\}$ and $u_1, u_2, v_1, v_2 \in (\mathbb{Z}/N_B\mathbb{Z})^\times$. Thus $\lambda_1 v_1/u_1 = \lambda_3 = \lambda_2 v_2/u_2$, and $\phi(P_1) = [\lambda_3]Q_1'$, $\phi(P_2) = [\lambda_3]Q_2'$ where $Q_1' = [v_1/u_1]Q_1$ and $Q_2' = [v_2/u_2]Q_2$.

We have

$$e_{N_B}(P_1, P_2)^{\deg\phi} = e_{N_B}(\phi(P_1), \phi(P_2)) = e_{N_B}([\lambda_3]Q_1', [\lambda_3]Q_2') = e_{N_B}(Q_1', Q_2')^{\lambda_3^2}.$$

We recover λ_3^2 by solving the following discrete logarithm

$$\lambda_3^2 = DLP\left(e_{N_B}(P_1, P_2)^{\deg\phi}, e_{N_B}(Q_1', Q_2')\right).$$

For any $S = [\alpha]P_1 + [\beta]P_2 \in E_0[N_B]$ we have $[\lambda_3] \circ \phi(S) = [\alpha]Q_1' + [\beta]Q_2'$.

□

Now that we can evaluate $[\lambda] \circ \phi$ point wise on $E_0[N_B]$ for some $\lambda \in (\mathbb{Z}/N_B\mathbb{Z})^\times$ such that λ^2 is provided, we show how to evaluate τ on $E[N_B]$.

Since we can evaluate $\phi_\lambda = [\lambda] \circ \phi$ on $E_0[N_B]$, then we can evaluate $\widehat{\phi_\lambda}$ on $E[N_B]$ as well. Therefore we can evaluate $\phi_\lambda \circ \theta \circ \widehat{\phi_\lambda}$ on $E[N_B]$. Meanwhile, we have

$$\phi_\lambda \circ \theta \circ \widehat{\phi_\lambda} = ([\lambda] \circ \phi) \circ \theta \circ ([\lambda] \circ \widehat{\phi}) = [\lambda^2] \circ \phi \circ \theta \circ \widehat{\phi}.$$

Since $\lambda^2 \in (\mathbb{Z}/N_B\mathbb{Z})^\times$ is provided, then we get

$$\phi \circ \theta \circ \widehat{\phi} = [\lambda^{-2}] \circ \phi_\lambda \circ \theta \circ \widehat{\phi_\lambda}$$

on $E[N_B]$. Hence $\tau = \phi \circ \theta \circ \widehat{\phi} + [d]$ can be efficiently evaluated on $E[N_B]$. This concludes our discussion.

From now on, we can translate the solutions in [Que+21] computing θ, d, e, and using the torsion point attacks to solve Problem 7.3.1 into solutions that compute θ, d, e, and solve Problem 7.3.4 in the same time and memory complexity, ignoring polylogarithmic factors.

Theorem 7.3.6 (Generalized Torsion Point Attacks)**.** *Suppose we are given an instance of Problem 7.3.4 where N_A has $O(\log\log p)$ distinct prime factors. Assume we are given the restriction of a trace-zero endomorphism $\theta \in End(E_0)$ to $E_0[N_B]$, an integer d coprime to N_B, and a smooth integer e such that*

$$\deg\left(\phi \circ \theta \circ \widehat{\phi} + [d]\right) = N_B^2 e \quad \text{or} \quad \deg\left(\phi \circ \theta \circ \widehat{\phi} + [d]\right) = N_B^2 pe.$$

Then we can compute ϕ in time $\tilde{O}(\sqrt{e})$.

Proof. Follows from the previous discussion, [Que+21, Theorem 3] and [Que+21, Theorem 5]. □

We have the following Corollary.

Corollary 7.3.7. *Suppose that $N_A \approx p^{\frac{1}{2}}$ and $N_B \approx p^{\frac{1}{2}+\eta}$ where $1 \leq \eta$. Under some heuristics, Problem 7.3.4 can be solved in polynomial time.*

In the following section, we use the revisited torsion point attacks to design a new adaptive attack on SIDH.

7.4 — A new adaptive attack on SIDH

In this section, we present our attack. First we present an overview, next we describe the active phase of our attack.

7.4.1 – Overview. In our attack, we suppose that one party is using a static secret/public key pair, and the other party runs multiple key exchanges with the honest party. He is provided with a the same oracle $O(E, R, S, E')$ described in Section 7.2.3.

The main idea of the attack is to use a key exchange oracle to recover the action of Alice's secret isogeny on a larger torsion point group. Doing so leads to an unbalanced SIDH. The malicious Bob then uses the revisited torsion point attacks, which in this case run in polynomial time, to recover Alice's secret key. Hence our attack has two phases.

Let N_A and N_B be the isogeny degrees of Alice and Bob respectively. In general, we have $N_A N_B | p+1$ in the case of SIDH schemes, $N_A | p+1$, $N_B | p-1$ or $N_B | p+1$, $N_A | p-1$ for BSIDH. Let $E_0 = E(1728)$ be the starting curve, $E_0[N_B] = \langle P_B, Q_B \rangle$, and let $(E_A, \phi_A(P_B), \phi_A(Q_B))$ be Alice's public key where her static secret key is an isogeny $\phi_A : E_0 \to E_A$ of degree N_A. Moreover, suppose that you are given some "suitable" smooth integer N coprime to N_A such that $E_0[N_B N] \subset E_0(\mathbb{F}_{p^{2k}})$ for some integer k (we will provide the requirements on N as we describe the attack in the following sections).

The two phases of the attack can be summarized as follows.

- **The active phase.** Bob uses the access to a key exchange oracle $O(E, R, S, E')$ to secretly transform Alice's static public key $(E_A, \phi_A(P_B), \phi_A(Q_B))$ into a tuple $(E_A, \phi_A(G_1), \phi_A(G_2), \phi_A(G_3))$ where $G_1 = \langle P \rangle$, $G_2 = \langle Q \rangle$, $G_3 = \langle R \rangle$ are cyclic subgroups of maximal order in $E_0[N_B N]$, such that $G_1 \cap G_2 = G_2 \cap G_3 = G_1 \cap G_3 = \{0\}$.
- **The passive phase.** Having $(E_A, \phi_A(G_1), \phi_A(G_2), \phi_A(G_3))$, Bob applies the revisited torsion point attacks to recover Alice's secret.

The passive phase is nothing else than the revisited torsion point attacks described in Section 7.3.2. In the rest of this section, we provide a full description of the active phase.

7.4.2 – Explicit description of the active phase. Let p be the base prime. Let $N = \ell_1^{v_1} \cdots \ell_n^{v_n}$ be a smooth integer coprime to N_A such that $E_0[\ell_i^{v_i}] \subset E(\mathbb{F}_{p^{2k_i}})$ and for each prime ℓ_i which is not a square modulo N_A, v_i is even. Let G_1, G_2, G_3 be cyclic subgroups of $E_0[N_B N]$ of order $N_B N$ such that $G_1 \cap G_2 = G_1 \cap G_3 = G_2 \cap G_3 = \{0\}$. The active phase of the attack consists in recovering $\phi_A(G_j)$ for $j = 1, 2, 3$.

For $j = 1, 2, 3$, we can represent G_j as $G_j = \sum_{i=1}^{r} G_{ji}$ where G_{ji} is a group of order $N_B \ell_i^{v_i}$. The action of ϕ_A on G_j is recovered by computing $\phi_A(G_{ji})$ for $i = 1, \cdots, n$. Storing $\phi_A(G_j)$ in this form enables us to perform all computations in extension fields of degree $k_1, \cdots, k_n$, instead of $LCM(k_1, \cdots, k_n)$ the full group G_j is considered. This is because all supersingular isogenies are $\mathbb{F}_{p^2}$-rational. Hence we never go to extension fields with degree beyond $\max\{k_i, i = 1, \cdots, r\}$. Let us describe how we compute $\phi_A(G_{ji})$ for $j = 1, 2, 3$ and $i = 1, \cdots, n$.

Let G be a cyclic subgroup of $E_0[N_B \ell^v]$ of order $N_B \ell^v$. Let us suppose that $\ell \equiv \mu^2 \mod N_A$ is a square modulo N_A and that $v = 1$. Note that $\phi_A([\ell]G)$ is readily provided in Alice's public key since this group has order N_B. To compute the action of ϕ_A on G of order $N_B \ell$, Bob computes the isogeny $\phi_G : E_0 \to E_G$ having kernel G together with $R = [\mu^{-1}]\phi_G(P_A)$, $S = [\mu^{-1}]\phi_G(Q_A)$. Let H be a random cyclic subgroup of

$E_A[N_B\ell]$ of order $N_B\ell$ containing $\phi_A([\ell]G)$. Let $\phi_H : E_A \to E_H$ be the isogeny of kernel H and $\phi'_A : E_G \to E_G/\phi_G(\ker(\phi_A))$ be the isogeny of kernel $\phi_G(\ker(\phi_A))$. If H is the image of the group G through ϕ_A then the diagram in Figure 7.3 commutes and $O(E_G, R, S, E_H) = 1$. In the other case, when $H \neq \phi_A(G)$, Lemma 7.4.1 shows that the oracle returns 1 with negligible probability.

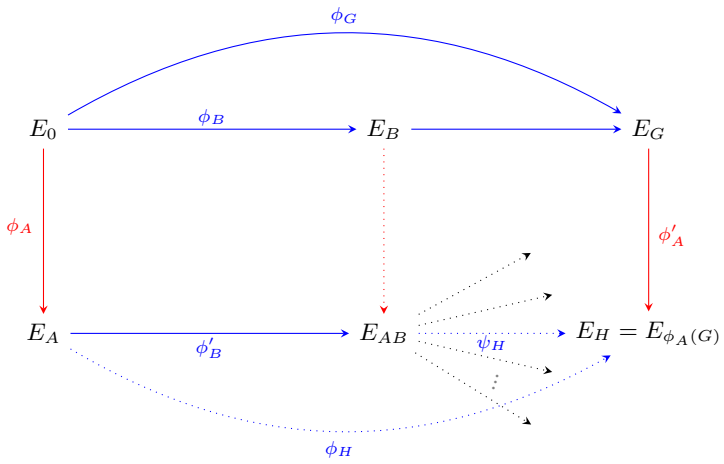


Figure 7.3: Computing the action of ϕ_A on G.

Lemma 7.4.1. *Suppose that $\ell \approx O(\log p)$ and $N_A N_B \approx p$ (or $N_A N_B > p$), and let G, H, E_H and $E_G/\phi_G(\ker(\phi_A))$ be defined as above. If $H \neq \phi_A(G)$ then $E_H = E_G/\phi_G(\ker(\phi_A))$ with negligible probability.*

Proof. Suppose that $E_H = E_G/\phi_G(\ker(phi_A))$ and let $H' = \phi_A(G)$. By construction, we get $[\ell]H = [\ell]\phi_A(G) = [\ell]H'$, and we can decompose ϕ_H and ϕ'_H as $\phi_H = \psi_H \circ \phi'_B$ and $\phi_{H'} = \psi_{H'} \circ \phi'_B$ where ϕ_H and $\phi_{H'}$ are isogenies of degree ℓ from E_{AB} to $E_G/\phi_G(\ker(\phi_A))$. Since $H \neq H'$, then $\widehat{\psi_{H'}} \neq \pm\psi_H$ and $\widehat{\psi_{H'}} \circ \psi_H$ is a non scalar endomorphism of E_{AB} of degree ℓ^2. Therefore, the curve E_{AB} is an ℓ^2-small curve as defined in [LB20].

On the other hand, since $N_A N_B \approx p$, then E_{AB} is statistically a random supersingular curve since the diameter of the supersingular isogeny graph is roughly p [GPS17]. Moreover, the number of ℓ^2-small curves is roughly ℓ^3 [LB20]. Considering the fact that the number of supersingular curves defined over $\mathbb{F}_{p^2}$ is $\frac{p}{12}$, then the probability that E_{AB} is an ℓ^2-small curve is roughly $\frac{12\ell^3}{p}$, which is negligible since $\ell \approx O(\log p)$. □

Remark 7.4.2. *We scale $\phi_G(P_A)$ and $\phi_G(Q_A)$ by μ^{-1} in order to avoid the detection by pairing computation. When scaled by μ^{-1}, we have*

$$\begin{aligned} e_{N_A}(R,S) &= e_{N_A}([\mu^{-1}]\phi_G(P_A), [\mu^{-1}]\phi_G(Q_A)) \\ &= e_{N_A}(P_A, Q_A)^{\mu^{-2}\deg\phi_G} \\ &= e_{N_A}(P_A, Q_A)^{N_B}. \end{aligned}$$

The above equation also justifies the requirement that ℓ should be a quadratic residue modulo N_A. When ℓ is not a quadratic residue modulo N_A and ℓ^2 divides N, we set the group G to have order $N_B\ell^2$ and we proceed the same way. In the later case, we scale the points $\phi_G(P_A)$ and $\phi_G(Q_A)$ by $\ell^{-1} \mod N_A$ instead.

If $1 < v$, then the process can be iterated to recover the action of ϕ_A on groups of order $N_B\ell$, $N_B\ell^2$, $\cdots$, $N_B\ell^v$ when ℓ is a square modulo N_A, respectively $N_B\ell^2$, $N_B\ell^4$, $\cdots$, $N_B\ell^v$ when ℓ is not a quadratic residue modulo N_A. Note that in the later case, v is even.

We deduce Algorithm 8 for computing the action of ϕ_A on a larger group G.

Algorithm 8 Evaluating the action of ϕ_A on a larger group G of order $N_B\ell^v$ using $O(E, R, S, E')$.

Require: E_0, P_A, Q_A, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, G.
Ensure: $\phi_A(G)$.
Set $G_0 = [\ell^v]G$;
if ℓ is a square modulo N_A **then**
    Compute $\mu = \sqrt{\ell} \mod N_A$;
    for $i = 1, \cdots, v$ **do**
        $G_i = [\ell^{v-i}]G$
        Compute $\phi_{G_i} : E_0 \to E_{G_i}$ of degree $N_B\ell^i$ and of kernel G_i, together with
         $R = [\mu^{-i}]\phi_{G_i}(P_A)$ and $S = [\mu^{-i}]\phi_{G_i}(Q_A)$;
        for H cyclic group of E_A of order $N_B\ell^i$ containing $\phi_A(G_{i-1})$ **do**
            Compute $\phi_H : E_A \to E_H$ of kernel H;
            if $O(E_{G_i}, R, S, E_H) = 1$ **then**
                Set $\phi_A(G_i) = H$;
            end if
        end for
    end for
    $G' = \phi_A(G_v)$;
else
    for $i = 1, \cdots, v/2$ **do**
        $G_i = [\ell^{v-2i}]G$
        Compute $\phi_{G_i} : E_0 \to E_{G_i}$ of degree $N_B\ell^{2i}$ and of kernel G_i, together with
         $R = [\ell^{-i}]\phi_{G_i}(P_A)$ and $S = [\ell^{-i}]\phi_{G_i}(Q_A)$;
        for H cyclic group of E_A of order $N_B\ell^{2i}$ containing $\phi_A(G_{i-1})$ **do**
            Compute $\phi_H : E_A \to E_H$ of kernel H;
            if $O(E_{G_i}, R, S, E_H) = 1$ **then**
                Set $\phi_A(G_i) = H$;
            end if
        end for
    end for
    $G' = \phi_A(G_{v/2})$;
end if
return G'.

Lemma 7.4.3. *Algorithm 8 runs in time* $\tilde{O}(k_v) = O(k_v \cdot \text{poly}(\log p))$ *time whenever* ℓ *is of polynomial size and* $E_0[N_B \ell^v] \subset E(\mathbb{F}_{p^{2k_v}})$. *The output of Algorithm 8 is* $\phi_A(G)$ *with overwhelming probability.*

Proof. Since ℓ, N_A and N_B are smooth integers, the time complexity of Algorithm 8 depends on the degree k_v of the field extension only. Hence Algorithm 8 runs in time $O(k_v \cdot \text{poly}(\log p))$. The second point of the Lemma follows from Lemma 7.4.1. □

Recall that $E_0[N_B \ell_i^{v_i}] \subset E(\mathbb{F}_{p^{2k_i}})$. Set $k^* = \max\{k_i\}$. Algorithm 9 fully describes the active phase our attack.

Algorithm 9 Recovering the action of ϕ_A on cyclic disjoint groups G_1, G_2, G_3 of order $N_B N$ using the oracle $O(E, R, S, E')$

Require: E_0, P_A, Q_A, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, N_A, N_B, $N = \ell_1^{v_1} \cdots \ell_n^{v_n}$, G_{ji} for $j = 1, 2, 3$ and $i = 1, \cdots, n$.
Ensure: $\phi_A(G_{ji})$ for $j = 1, 2, 3$ and $i = 1, \cdots, r$.
1: **for** $i = 1, \cdots, n$ **do**
2: **for** $j = 1, 2, 3$ **do**
3: Compute $\phi_A(G_{ji})$ using Algorithm 8;
4: **end for**
5: **end for**
6: **return** $\phi_A(G_{ji})$ for $j = 1, 2, 3$ and $i = 1, \cdots, n$.

Lemma 7.4.4. *Algorithm 9 runs in time* $\tilde{O}(\max\{k^*\})$ *whenever* ℓ_i *for* $i = 1, \cdots, n$, N_A, N_B *are smooth integers.*

Proof. Follows from the Lemma 7.4.3. □

This concludes our description of the active phase. In the next section, we discuss the computation of the integer N.

7.4.3 – Computing the integer N. We address the existence and the computation of the integer N. We would like to compute a smooth integer $N = \ell_1^{v_1} \cdots \ell_n^{v_n}$ coprime to N_A such that $E_0[N_B \ell_i^{v_i}] \subset E(\mathbb{F}_{p^{2k_i}})$ and for each prime ℓ_i which is not a square modulo N_A, v_i is even. Recall that by Corollary 7.3.7, the torsion point attacks run in polynomial time when $p < N$.

We start by the following Lemma which describes the group structure of supersingular curves over extension fields.

Lemma 7.4.5. *Let* $E/\mathbb{F}_{p^2}$ *be a supersingular elliptic curve such that* $E(\mathbb{F}_{p^2}) \simeq (\mathbb{Z}_{p-\epsilon})^2$ *where* $\epsilon = \pm 1$ *corresponds to the sign of the trace of Frobenius* $t = 2\epsilon p$ *of* E *over* $\mathbb{F}_{p^2}$.
Then for every natural number k, *the group structure of* E *over* $\mathbb{F}_{p^{2k}}$ *is given by*

$$E(\mathbb{F}_{p^{2k}}) \simeq (\mathbb{Z}_{p^k - \epsilon^k})^2 \tag{7.3}$$

Proof. Let k be natural number and let t_k be the trace of Frobenius of E over $\mathbb{F}_{p^{2k}}$. Then by Hasse Theorem (theorem V.1.1 of [Sil09]),

$$|E(\mathbb{F}_{p^{2k}})| = p^{2k} + 1 - t_k.$$

Over $\mathbb{F}_{p^2}$, the characteristic equation of Frobenius is given by

$$X^2 - 2\epsilon pX + p^2 = (X - \epsilon p)^2$$

By Theorem 4.12 of [Was08]

$$t_k = 2(\epsilon p)^k = 2\epsilon^k p^k$$

where ϵ^k is the sign of t_k. Hence $\quad t_k^2 = 4p^{2k} \quad$ and by lemma 4.8 of [Sch87]

$$E(\mathbb{F}_{p^{2k}}) \simeq (\mathbb{Z}_{\sqrt{p^{2k}-\epsilon^k}})^2 \simeq (\mathbb{Z}_{p^k-\epsilon^k})^2.$$

□

From Equation 7.3, we have that $E_0[N_B\ell_i^{v_i}] \subset E_0(\mathbb{F}_{p^{2k_i}})$ if and only if $N_B\ell_i^{v_i} | p^{k_i} - \epsilon^{k_i}$ where ϵ is the sign of the trace of Frobenius of E_0 as described in the proof of Lemma 7.4.5.

Let ℓ be a small prime. Then $\ell^v | p^{2k} - 1$ for some $k \leq \ell^v$. This means that for each prime ℓ_i dividing N, $k_i \leq \ell_i^{v_i}$. This heals a easy way to compute N : choose the smallest primes ℓ_i coprime to $N_A N_B$, such that $p < N = \prod \ell_i^2$. Then the largest ℓ_i is in $O(\log p)$. Moreover we have k_i at most ℓ_i^2.

To moderate the fields extension degrees, we also include in N primes ℓ that are squares modulo N_A. For this primes, we only require ℓ to divide $p^{2k} - 1$, hence obtaining a smaller field extension.

We describe the full process in Algorithm 10. The algorithm returns the list P of prime power factors of N with the list D of the corresponding extension field degrees.

Lemma 7.4.6. *Algorithm 10 runs in polynomial time and for each prime ℓ_i dividing N, $k_i \leq \ell_i^2 \approx O(\log^2 p)$.*

Proof. Follows from the previous discussion. □

Remark 7.4.7. *In all this section, we were attacking Alice's secret isogeny. To attack Bob's secret isogeny instead, one interchanges the roles of N_A and N_B. Mostly, the quadratic residuosity condition on N will depend on N_B.*

Remark 7.4.8. *In practice, one may set a bound on the extension degrees and slightly increase the size of the primes ℓ_i. This will be the case in the attack parameters we will present in Section 7.5.*

Algorithm 10 Computing N

Require: p, N_A, N_B.
Ensure: P, D.
Create the lists P and D, set $N = 1$, set $\ell = 1$;
while $N < p$ **do**
  choose the next prime ℓ coprime to $N_A N_B$;
  if ℓ is a square modulo N_A **then**
    Compute the smallest integer k such that $\ell | p^{2k} - 1$.
    Append ℓ to the list P and $2k$ to the list D;
    $N = N * \ell$;
  else
    Compute the smallest integer k such that $\ell^2 | p^{2k} - 1$.
    Append ℓ^2 to the list P and $2k$ to the list D;
    $N = N * \ell^2$;
  end if
end while
return P, D;

Algorithm 11 New Adaptive attack on SIDH

Require: E_0, P_A, Q_A, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, N_A, N_B.
Ensure: $\ker(\phi_A)$.
Compute a suitable smooth integer N using Algorithm 10.
Let G_1, G_2, G_3 cyclic disjoint subgroups of $E_0[N_B N]$ of order $N_B N$.
Compute $\phi_A(G_1)$, $\phi_A(G_2)$, $\phi_A(G_3)$ using the oracle $O(E, R, S, E')$ and Algorithm 9.
Compute ϕ_A using the revisited torsion point attacks of Theorem 7.3.6.
return $\ker(\phi_A)$.

7.4.4 – Attack summary. The full attack is summarised in Algorithm 11.

Now we evaluate the number of oracle queries. Since $N = \ell_1^{v_1} \cdots \ell_n^{v_n}$ where for each prime ℓ_i which is not a square modulo N_A, v_i is even, then we can write $N = \ell_1^{2v_1} \cdots \ell_n^{2v_n} \ell_{n+1}^{u_1} \cdots \ell_{n+m}^{u_m}$ where the primes ℓ_{n+j} for $j = 1, \cdots, m$ are squares modulo N_A. From Algorithm 8, for each prime factor ℓ_i $(1 \leq i \leq n)$ of N, the maximum number of queries to the oracle (E, R, S, E') is equal to the number of cyclic subgroups of $\left(\mathbb{Z}/\ell_i^2\mathbb{Z}\right)^2$ of order ℓ_i^2, which is $\ell_i(\ell_i + 1)$. Note that if the first $\ell_i(\ell_i + 1) - 1$ queries fail, then there is no need to perform the last query since it will succeed. Also, for each prime factor ℓ_{n+j} $(1 \leq j \leq m)$ of N, the maximum number of queries to the oracle (E, R, S, E') is equal to the number of cyclic subgroups of $(\mathbb{Z}/\ell_i\mathbb{Z})^2$ order ℓ_i, which is $\ell_i + 1$. Here also, there is no need to perform the last query when the first ℓ_i queries failed. Therefore, the maximum number of oracle queries in the attack is

$$O_q = \sum_{i=1}^{n} v_i \left[\ell_i(\ell_i + 1) - 1\right] + \sum_{j=1}^{m} u_j \ell_{n+j}.$$

Now we can state the main result of this paper.

Theorem 7.4.9. *Let p, E_0, $N_A < p$, $N_B < p$, P_A, Q_A, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$ be the public parameters and the public key of an SIDH type scheme.*

Provided a key exchange oracle $O(E, R, S, E')$, Algorithm 11 recovers ϕ_A in polynomial time.

Furthermore, Algorithm 11 performs at most

$$O_q = \sum_{i=1}^{n} v_i \left[\ell_i(\ell_i + 1) - 1\right] + \sum_{j=1}^{m} u_j \ell_{n+j}$$

queries to the key exchange oracle where $N = \ell_1^{2v_1} \cdots \ell_n^{2v_n} \ell_{n+1}^{u_1} \cdots \ell_{n+m}^{u_m}$ is the integer computed in Step 1.

Proof. By Lemma 10, Step 1 outputs a smooth integer N such that $\max\{k_i\} \approx O(\log^2 p)$. Hence by Lemma 7.4.3, Step 3 runs in time $\tilde{O}(\log^2 p) = \tilde{O}(1)$. Step 4 runs in polynomial time since $p < N$. The number of oracle queries follows from the discussion preceding Theorem 7.4.9. □

Remark 7.4.10. *In our attack, the malicious Bob computes isogenies of degree $N_B\ell^2$ or $N_B\ell$ depending on the quadratic residuosity of ℓ modulo N_A. In appendix C.1, we suggest a variant of the attack where isogenies Bob computes isogenies of degree ℓ^2 or ℓ instead. Nevertheless, this variant can be easily detected.*

7.5 — Relevance and countermeasures

In this section, we suggest some attack parameters for $IDH and SIDH primes. We discuss possible countermeasures to the attack.

7.5.1 – Attack parameters for some SIDH and BSIDH primes. We propose attack parameters for the two (non cryptographic size) primes suggested for the $IKE challenge [Cos21, §10], the SIDH primes SIDHp377 and SIDHp546 suggested by Longa et al. [LWS20], SIDHp434 and SIDHp503 as specified in SIKE [Jao+20].

As attack parameters, we provide the prime factorisation of N, the maximum field extension degree $k^* = \max\{k_i\}$, $\eta \approx N/p$ and the number O_q of oracle queries. We also precise which party is attacked: B stands for Bob and A stands for Alice.

The outcome of our investigations on the above mentioned $IDH primes and SIDH primes is summarised in Table 7.1 and Table 7.2 respectively.

When it comes to BSIDH instances, generating specific attack parameters is less trivial. We believe this may be because BSIDH primes[2] are twin primes. Using the generic attack parameters computation described in Algorithm 10, the degree of the field extensions are relatively larger compared to those used when running the attack on SIDH. For example, let us consider the smallest BSIDH prime (prime in example 6 of [Cos20])

$$p = 2 \cdot (2^3 \cdot 3^4 \cdot 17 \cdot 19 \cdot 31 \cdot 37 \cdot 53^2)^6 - 1.$$

Set $N_A = p + 1$ and $N_B = (p - 1)/2$. Then we get

$$N = 5^2 \cdot 11^2 \cdot 23^2 \cdot 29^2 \cdot 41^2 \cdot 47^2 \cdot 59^2 \cdot 61^2 \cdot 67^2 \cdot 71^2 \cdot 79^2 \cdot 83^2 \cdot 89^2 \cdot 97^2 \cdot 101^2 \cdot 107^2 \cdot 109^2 \cdot 113^2 \cdot 127^2 \cdot 131^2 \cdot 137^2$$

[2] Primes p such that both $p + 1$ and $p - 1$ are smooth.

Party	k^*	η	O_q	N
$IDHp182 prime: $p = 2^{91}3^{57} - 1$				
B	96	$\frac{185}{182}$	7251	$5^2 * 7 * 11^2 * 13 * 19 * 31 * 37 * 43 * 47^2 * 61 * 67 * 73 * 79 * 97 * 103 * 109 * 127 * 139 * 157 * 181 * 241 * 277 * 421 * 433 * 541 * 661 * 919$
$IDHp217 prime: $p = 2^{110}3^{67} - 1$				
B	96	$\frac{222}{217}$	9349	$5^2 * 7 * 13 * 19 * 31 * 37 * 43 * 61 * 67 * 73 * 79 * 97 * 109 * 157 * 163 * 181 * 193 * 199 * 211 * 223 * 229 * 271 * 277 * 307 * 337 * 571 * 631 * 1009 * 1093 * 1249 * 1381$

Table 7.1: Attack parameters for the two $IDH primes.

Party	k	η	O_q	N
SIDHp377 prime: $p = 2^{191}3^{117} - 1$				
B	120	$\frac{377}{377}$	40728	$5^2 * 7 * 11^2 * 13 * 19 * 31 * 37 * 43 * 61 * 67 * 73 * 79 * 97 * 103 * 109 * 157 * 181 * 193 * 199 * 229 * 241 * 271 * 277 * 307 * 313 * 331 * 337 * 433 * 487 * 571 * 631 * 661 * 739 * 1009 * 1021 * 1051 * 1093 * 1249 * 1993 * 2161 * 2707 * 3433 * 3529 * 4003 * 4603 * 5419$
SIDHp434 prime: $p = 2^{216}3^{137} - 1$				
B	152	$\frac{438}{434}$	66169	$5^2 * 7 * 11^2 * 13 * 17^2 * 19 * 31 * 37 * 43 * 61 * 67 * 71^2 * 73 * 79 * 97 * 103 * 109 * 127 * 139 * 151 * 181 * 193 * 211 * 277 * 373 * 409 * 421 * 433 * 457 * 547 * 601 * 613 * 739 * 751 * 757 * 1123 * 1171 * 1231 * 1489 * 1741 * 1873 * 2311 * 2593 * 2887 * 3037 * 3061 * 4357 * 5227 * 6091 * 6661 * 7621$
SIDHp503 prime: $p = 2^{250}3^{159} - 1$				
B	158	$\frac{512}{503}$	81049	$5^2 * 7 * 11^2 * 13 * 19 * 31 * 37 * 43 * 61 * 67 * 73 * 79 * 97 * 103 * 109 * 127 * 139 * 151 * 157 * 163 * 181 * 193 * 199 * 211 * 229 * 241 * 277 * 409 * 421 * 433 * 439 * 457 * 463 * 571 * 577 * 601 * 859 * 967 * 1093 * 1153 * 1171 * 1201 * 1303 * 1327 * 1741 * 2131 * 2179 * 2269 * 2371 * 2377 * 2689 * 3037 * 3169 * 4663 * 6151 * 6469 * 6529 * 8893 * 9769$
SIDHp546 prime: $p = 2^{273}3^{172} - 1$				
B	152	$\frac{551}{546}$	112441	$5^2 * 7 * 11^2 * 13 * 19 * 31 * 37 * 43 * 61 * 67 * 73 * 79 * 83^2 * 97 * 103 * 109 * 127 * 139 * 151 * 157 * 163 * 181 * 193 * 223 * 277 * 307 * 379 * 409 * 421 * 433 * 457 * 613 * 631 * 661 * 691 * 751 * 1117 * 1153 * 1249 * 1321 * 1621 * 1741 * 1753 * 1801 * 1933 * 1999 * 2053 * 2137 * 2281 * 3571 * 3823 * 5059 * 5281 * 5563 * 6373 * 6397 * 6481 * 7549 * 7639 * 8161 * 9151$

Table 7.2: Attack parameters for some SIDH primes.

and the ℓ_i^2 torsion points for ℓ_i dividing N are defined over extension fields of $\mathbb{F}_{p^2}$ of degree

$$20, 55, 253, 406, 820, 23, 3422, 15, 402, 2485, 3081, 3403, 1958,$$
$$9312, 2020, 5671, 11772, 12656, 8001, 1310, 2329,$$

the order is the same as in the prime factorisation of N. The number of oracle queries is $O_q = 152523$. Note that here, one will be working with extension fields of degree up to 12656. One may prefer to compute a different integer N for which the maximum extension field degree is relatively small, but as we mentioned before, this requires intensive computations which we could not do on a personal computer.

Remark 7.5.1. *Our attack applies to eSIDH [COR20] as well. It can be easily adapted to k-SIDH [AJL17] and it's variant by Jao and Urbanik [UJ20]. In the later case, the number of oracle queries is exponential in k.*

7.5.2 – Countermeasures to the attack. A straightforward countermeasure of the attack is to use a variant of the Fujisaki-Okamoto transform [FO99; HHK17] as in SIKE. This transform obliges Bob to disclose his secret key to Alice who will recompute Bob's public to verify its correctness. Recomputing Bob's public key will enable Alice to detect Bob's maliciousness.

A second countermeasure is that Bob uses the SIDH proof of Knowledge as recently suggested in [FDGZ21]. In this proof of knowledge, Bob proves that there exists an isogeny of degree N_B between E_0 and E_B and that the provided torsion points were not maliciously computed. Nevertheless, this countermeasure is very costly, since the proof of isogeny knowledge is nothing else than the SIDH based signature scheme, which is relatively slow and has large signatures.

Another less costly countermeasure is to set the curves E_0 to be a random supersingular elliptic curve with unknown endomorphism ring. This counters the improved torsion points attack. Hence Bob will not be able to recover Alice's secret isogeny after recovering its action on a larger torsion group. Nevertheless, one should keep in mind that this later countermeasure does not counter the GPST adaptive attack. Also, it requires a trusted party that will run the setup.

Remark 7.5.2. *Since the starting curve in HealSIDH, SHealS and Heals (presented in Chapter 5) is a random supersingular curve with unknown endomorphism ring, then this adaptive attack does not apply to those schemes.*

7.6 — Conclusion

In this chapter, we present a generalisation of the torsion point attacks and use it to design a new adaptive attack on SIDH type schemes. Our generalized torsion point attacks recover a secret isogeny when its action on three disjoint cyclic subgroups of relatively large order is provided. Our adaptive attack consists of maliciously computing isogenies of larger degrees than expected in SIDH, then using an access to a key exchange oracle to recover the action of the honest party's secret isogeny on large torsion groups. We then use this generalized torsion point attacks to recover the secret isogeny.

We provide concrete attack parameters for SIDH instances instantiated with the SIDH primes \$IDHp182, \$IDHp217, SIDHp377, SIDHp434, SIDHp546 and SIDHp503. A search of attack parameters on BSIDH primes is ongoing. We finally suggest countermeasures among which the Fujisaki-Okamoto transform (as used in SIKE), using a proof of isogeny knowledge as recently proposed in [FDGZ21] or setting the starting curve in SIDH to be a random supersingular curve with unknown endomorphism ring.

This result proves that torsion point attacks, which do not yet apply to SIDH, become relevant to SIDH parameters in an adaptive attack setting. Moreover, it introduces a new cryptanalytic tool for isogeny based cryptography.

Chapter 8

Summary and further work

In this thesis, we designed three isogeny-based public key encryption schemes and/or key exchange protocols and developed two cryptanalysis results on SIDH types schemes.

Among the schemes designed, we have SimS, SETA and HealSIDH. SimS is an IND-CCA hash function free PKE which improves on a recent work of Moriya, Onuki and Takagi. SÉTA is a new PKE obtained by transforming the Petit's attack into a trapdoor mechanism. And, HealSIDH is a new key exchange protocol obtained from a countermeasure to the GPST adaptive attack.

Our cryptanalysis results include a generalisation of the GPST reduction of the isogeny problem in SIDH instances to the endomorphism ring computation problem. Also, we design a new adaptive attack on SIDH that uses the Petit's torsion point attack as subroutine. Our attack is fundamentally different from the GPST adaptive attack.

The difference between SÉTA and previous isogeny-based PKEs (PKEs derived from SIDH or CSIDH) is that SÉTA is built from a trapdoor mechanism while these previously existing isogeny-based PKEs are El Gamal type encryption schemes, derived from the Diffie-Hellman type key exchanges CSIDH and SIDH. This suggests that SÉTA may be used as building block for some advanced schemes that were difficult to built with CSIDH or SIDH. We leave this investigation for future work.

Our countermeasure to the GPST adaptive attack in new and comes with too much overhead. We believe the countermeasure could be optimised or redesigned to reduce the overhead. Also, this countermeasure will eventually enable the design of new advanced schemes using SIDH as a subroutine. We also leave this as future work.

The last research direction we suggest is cryptanalysis. Our countermeasure to the GPST adaptive attack is new, hence will need to be properly analysed in the near future. Our new adaptive attack on SIDH leaves us with the question: how far can we reach with the Petit's torsion points attack? We expect to explore possibilities for further improvements of Petit's torsion points attack.

Appendix

A.1 — Knowledge of Exponent assumption

In the context of Discrete Logarithm-based cryptography, the Knowledge of Exponent assumption is stated as follows.

Assumption 5 (Knowledge of Exponent assumption [Nao03])**.** *Let $G = \langle g \rangle$ be a cyclic group of prime order q where q is of cryptographic size. Let x be a uniformly random exponent in $\{2, \cdots, q-1\}$ and let $h = g^x$. The adversary tries to compute $h_1, h_2 \in G$ such that $h_1 = g^z$ and $h_2 = h^z$ for some $z \in \{2, \cdots, q-1\}$.*
The knowledge of exponent assumption holds if for every polynomial time adversary $\mathcal{A}$ that when given g, q and h outputs (g^z, h^z), there exists a polynomial time adversary $\mathcal{A}'$ that for the same inputs outputs (z, g^z, h^z).

Intuitively, this assumption states that the only efficient way to compute (g^z, h^z) is to first fix z, then to compute g^z and h^z.

In SimS, the ciphertexts are of the form $\mathsf{c} = ([\mathfrak{b}]E_0, f_{[\mathfrak{b}][\mathfrak{a}]E_0}(x([2\mathsf{m}_0 + 1]P_{[\mathfrak{b}][\mathfrak{a}]E_0}))$. Assumption 3 states the only efficient way to compute a valid ciphertext is to first fix the ideal class $[\mathfrak{b}]$, then run the encryption algorithm of SimS to compute $\mathsf{c} = ([\mathfrak{b}]E_0, f_{[\mathfrak{b}][\mathfrak{a}]E_0}(x([2\mathsf{m}_0 + 1]P_{[\mathfrak{b}][\mathfrak{a}]E_0}))$.

A.2 — Generating the distinguished point of order 2^r

Here we discuss how when given a supersingular curve E defined over $\mathbb{F}_p$ where $p = 2^r \ell_1 \cdots \ell_n - 1$, one can efficiently generate a distinguished point P_E of order 2^r. The algorithm used by Moriya et al. in C-SiGamal to generate such a point mainly relies on the following result.

Theorem A.2.1. *([MOT20, Appendix A]) Let p be a prime such that $p \equiv 3 \bmod 4$ and let E be a supersingular Montgomery curve defined over $\mathbb{F}_p$ satisfying $End_{\mathbb{F}_p}(E) \cong \mathbb{Z}[\pi]$. Let $P \in E$.*
If $P \in E[\pi - 1] \backslash E[2]$, then $\quad x(P) \in (\mathbb{F}_p^*)^2 \iff P \in 2E[\pi - 1]$.
If $P \in E[\pi + 1] \backslash E[2]$, then $\quad x(P) \notin (\mathbb{F}_p^*)^2 \iff P \in 2E[\pi + 1]$.

Hence when searching for the x-coordinate of points of order 2^r in E, we need to avoid elements of $\mathbb{F}_p$ that are squares. Since $p = 2^r \ell_1 \cdots \ell_n - 1$ with $r > 1$, then $\left(\frac{-1}{p}\right) = -1$, $\left(\frac{2}{p}\right) = 1$ and $\left(\frac{\ell_i}{p}\right) = 1$ for $i \in \{1, \cdots, n\}$. Furthermore, let us suppose that $\ell_1, \cdots, \ell_{n-1}$ are the first smallest odd primes, then for every $I \subset \{0, 1, \cdots, n-1\}$, $\left(\frac{-\prod_{i \in I} \ell_i}{p}\right) = -1$ where $\ell_0 = 2$. Moriya et al.'s Algorithm [MOT20, Appendix A]

exploits this to consecutively sample x from the sequence $-2, -3, -4, \cdots$ and when x is the x-coordinate of a point in $E(\mathbb{F}_p)$, it checks if this point has order divisible by 2^r. Corollary A.2.2 proves that if a such x is the x-coordinate of a point in $E(\mathbb{F}_p)$ then the corresponding point has order divisible by 2^r, hence the check is not necessary.

Corollary A.2.2. *Let p be a prime such that $p \equiv 3 \bmod 4$ and let E be a supersingular Montgomery curve defined over $\mathbb{F}_p$ satisfying $End_{\mathbb{F}_p}(E) \cong \mathbb{Z}[\pi]$. Let $P \in E(\mathbb{F}_p)$ such that $x(P) \neq 0$.*
If $x(P) \notin (\mathbb{F}_p^*)^2$ then $[\ell_1 \times \cdots \times \ell_n]P$ is a point of order 2^r.

Proof. Since $E(\mathbb{F}_p) = E[\pi - 1]$ is a cyclic group, then there exist a point Q of order $p+1 = 2^r \ell_1 \cdots \ell_n$ such that $E(\mathbb{F}_p) = \langle Q \rangle$. Set $P = [\alpha_P]Q$. Since E is in the Montgomery form, then $E(\mathbb{F}_p) \cap E[2] = \langle (0,0) \rangle$. Since $x(P) \neq 0$, then $P \in E[\pi - 1] \backslash E[2]$. Let us suppose that $x(P) \notin (\mathbb{F}_p^*)^2$, then by Theorem A.2.1 $P \notin 2E[\pi - 1]$, hence α_P is odd. Therefore, $\gcd(p+1, \alpha_P) = \gcd(2^r \ell_1 \cdots \ell_n, \alpha_P) = \gcd(\ell_1 \cdots \ell_n, \alpha_P)$. This implies that $P = [\alpha_P]Q$ is a point of order

$$\frac{p+1}{\gcd(p+1, \alpha_P)} = 2^r \cdot \frac{\ell_1 \cdots \ell_n}{\gcd(\ell_1 \cdots \ell_n, \alpha_P)}.$$

Hence $[\ell_1 \times \cdots \times \ell_n]P$ is a point of order 2^r. □

Exploiting Corollary A.2.2 we get Algorithm 12 which improves on that used by Moriya et al. for the same purpose.

Algorithm 12 Computing the distinguished point P_E

Require: The prime $p = 2^r \ell_1 \cdots \ell_n - 1$ and Montgomery coefficient $A \in \mathbb{F}_p$ of a supersingular curve.
Ensure: $P_E \in E(\mathbb{F}_p)$ of order 2^r.
1: Set $x \leftarrow -2$
2: **while** $x^3 + Ax^2 + x$ is not a square in $\mathbb{F}_p$ and $-x \leq \ell_{n-1} + 1$ **do**
3: Set $x \leftarrow x - 1$
4: **end while**
5: **if** $-x \leq \ell_{n-1} + 1$ **then**
6: Set $P = (x, \cdot) \in E(\mathbb{F}_p)$
7: Set $P_E = [\ell_1 \times \cdots \times \ell_n]P$
8: **return** P_E
9: **else**
10: **return** $\perp$.
11: **end if**

A random element $x \in \mathbb{F}_p^* \backslash (\mathbb{F}_p^*)^2$ is the x-coordinate of a point $P \in E(\mathbb{F}_p)$ with probability $\frac{1}{2}$. The probability that Algorithm outputs $\perp$ is bounded by $\left(\frac{1}{2}\right)^{\ell_{n-1}}$. For SiGamal primes p_{256} and p_{128} (see Section 3.5), ℓ_{n-1} is 191 and 281 respectively, hence the output is $\perp$ with probability 2^{-191} and 2^{-281} respectively.

Remark A.2.3. *Algorithm 12 is deterministic, hence always outputs the same point P_E when the input in unchanged.*

A.3 — On the randomising function f_E

Given a prime p of size n ($n = \lceil \log_2 p \rceil$) and a supersingular elliptic curve $E/\mathbb{F}_p$, we consider the function

$$f_E : \mathbb{F}_p \to F = Im(f_E) \subset \{0,1\}^n, \quad x \mapsto \mathsf{bin}(x) \oplus \mathsf{bin}(A_E).$$

Clearly, f_E is bijective and satisfies (P1) with $G_E = f_E^{-1} : y \mapsto x \in \mathbb{F}_p$ such that $\mathsf{bin}(x) = y \oplus \mathsf{bin}(A_E)$. Proving that f_E satisfies (P2) and (P3) is less straightforward. Nevertheless, we give some intuitive arguments on why we believe that f_E satisfies (P2) and (P3).

Given an element $y \in F$, in order to distinguish whether $y = f_E(x)$ where x is the x-coordinate of a point of order 2^r on some supersingular curve E or just a random element of $\{0,1\}^n$, one may first fix the supersingular curve E, then check if $g_E(y)$ is the the x-coordinate of a point of order 2^r on E. This process needs to be repeated for all $O(\sqrt{p})$ supersingular elliptic curves defined over $\mathbb{F}_p$. Hence leading to an exponential adversary. Another possible way is to try all elements in the set $Y = \{(z,t) \in \mathbb{F}_p^2, \mathsf{bin}(z) \oplus \mathsf{bin}(t) = y\}$ till you get a couple (z,t) for which t is the Montgomery coefficient of a supersingular curve E and z is the x-coordinate of a point of order 2^r on E. So p has to be chosen such that the cardinality of Y is exponential (in the security parameter) for every $y \in \{0,1\}^n$.

Let k be the bit length of $p - 2^{n-1}$, that is the position of the second most significant bit of p. Then for every $y \in \{0,1\}^n$, there exist at least 2^{k-1} couples $(z,t) \in \mathbb{F}_p^2$ such that $y = \mathsf{bin}(z) \oplus \mathsf{bin}(t)$. In fact, one can write $y = b||y_1||y_0$ where b is the first bit of y, y_1 and y_0 have $n-k$ and $k-1$ bits respectively. Then for every $w \in \{0,1\}^{k-1}$, $z' = b||0\cdots 0||(w \oplus y_0)$ and $t' = 0||y_1||w$ are binary representations of elements in $\mathbb{F}_p$ and $y = z' \oplus t'$.

For the primes p_{128} and p_{256} used in Section 3.5, we have $k = n-1$. In brief, when using the above function f_E, one should avoid primes p such that $p - 2^{n-1} < 2^\lambda$ where λ is the security parameter and n is the binary length of p.

The third property (P3), intuitively, follows from the fact there is no compatibility with XOR and algebraic operations. In fact, given $a \oplus b$, it seems hard to derive $R(a) \oplus b$ where R is non identical rational function.

B.1 — HealS PKE

The HealS Public Key Encryption scheme is detailed in Figure B.1.

C.1 — A simpler, but detectable variant of the attack

We present a simpler variant of our attack, but which can be easily detected. In Section 7.4.2, we use Algorithm 8 to recover the action of ϕ_A on groups of order $N_B \ell^v$. In the case where ℓ is coprime to N_B, there is no need to consider groups of order $N_B \ell^v$ since we already know the action of ϕ_A on the N_B-torsion points. Therefore, we can directly recover the action of ϕ_A on groups of order ℓ^v.

Let d be the smallest divisor of N_B such that $N_B = dN'_B$ and N'_B is a square modulo N_A, say $N'_B \equiv \gamma^2 \mod N_A$. To recover the action of ϕ_A on a cyclic group G_1 of order ℓ where $\ell \equiv \mu^2 \mod N_A$, Bob chooses a cyclic group G_0 of order d and sets $G = G_0 + G_1$, which is a group of order $d\ell$. He computes the isogeny

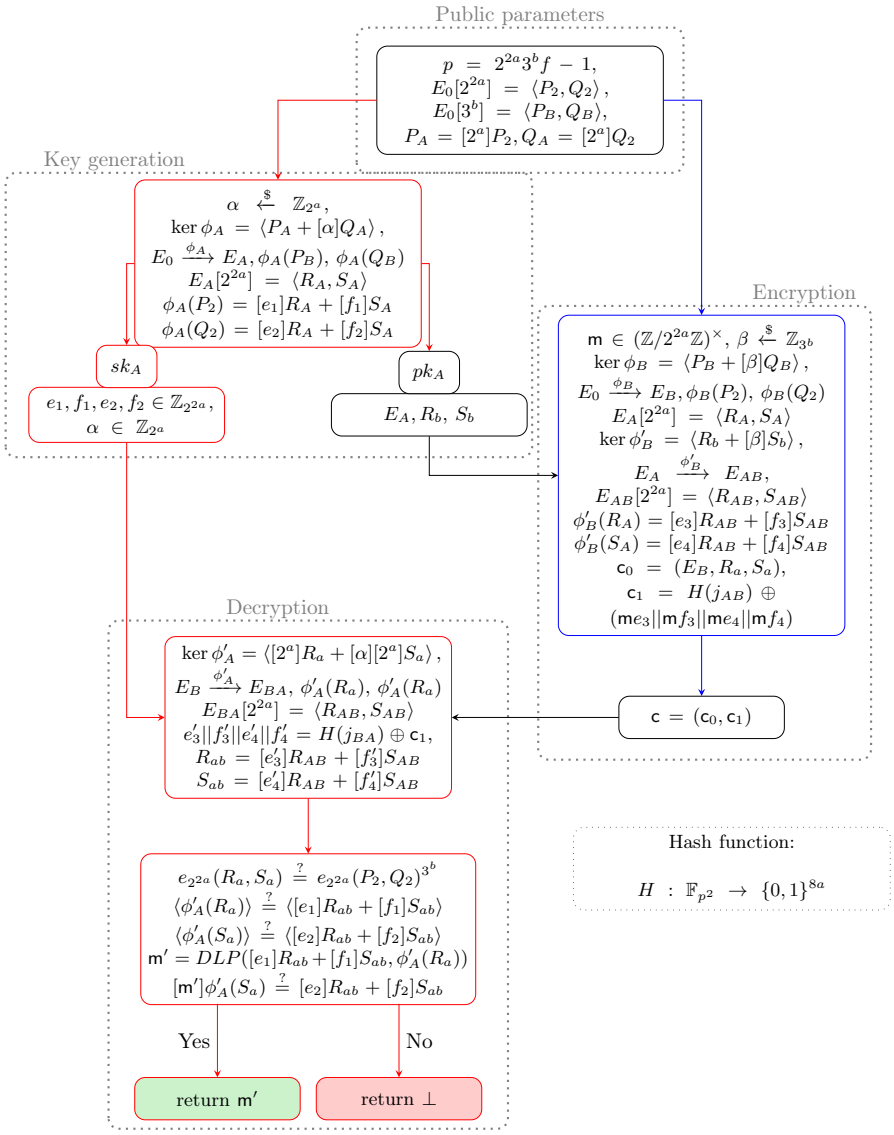


Figure B.1: HealS PKE.

$\phi_G : E_0 \to E_G = E_0/G$ together with $R = [\gamma\mu^{-1}]\phi_G(P_A)$ $S = [\gamma\mu^{-1}]\phi_G(Q_A)$. For each cyclic group $H \subset E_A[d\ell]$ containing $\phi_A(G_0)$, Bob computes $E_H = E_A/H$ and queries

the oracle (E_G, R, S, E_H). Note that

$$\begin{aligned} e_{N_A}(R,S) &= e_{N_A}([\gamma\mu^{-1}]\phi_G(P_A), [\gamma\mu^{-1}]\phi_G(Q_A)) \\ &= e_{N_A}(P_A, Q_A)^{\gamma^2\mu^{-2}\deg\phi_G} \\ &= e_{N_A}(P_A, Q_A)^{N'_B\ell^{-1}d\ell} \\ &= e_{N_A}(P_A, Q_A)^{N_B}, \end{aligned}$$

Hence the pairing check does not detect the attack. Nevertheless, when N_B is a very smooth integer (like in SIDH where $N_B = 3^b$ and $d \in \{1, \ell\}$), d is small. Hence Alice can easily check if the curves E_0 and E_G are $d\ell$-isogenous to discard such malicious public keys.

Collaborators

I have been collaborating with the following lovely researchers in the past two years. The list is in alphabetical order.

Luca De Feo

Institution : IBM Research Europe, Zürich, Switzerland
Inria, France
Homepage: https://defeo.lu

Cyprien Delpech de Saint Guilhem

Institution : imec-COSIC, KU Leuven, Leuven, Belgium
Homepage: https://www.esat.kuleuven.be/cosic/people/cyprien→

Péter Kutas

Institution : Eötvös Loránd University, Budapest, Hungary
University of Birmingham, Birmingham, UK

Antonin Leroux

Institution : LIX, CNRS, Institut Polytechnique de Paris, France
DGA
Homepage: http://www.lix.polytechnique.fr/Labo/Antonin→

Simon-Philipp Merz

Institution : Royal Holloway, University of London, UK
Homepage: https://simon-philipp.com

Christophe Petit

Institution : Université Libre de Bruxelle, Brussels, Belgium
University of Birmingham, Birmingham, UK
Homepage: http://homepages.ulb.ac.be/ chripeti/index.html

Javier Silva

Institution : Universitat Pompeu Fabra, Spain

Yan Bo Ti

Institution : DSO, Singapore

Benjamin Wesolowski

Institution : Institut de Mathématiques de Bordeaux, France
Homepage: https://www.bweso.com

Author's publications

Published or accepted papers

5. T. B. Fouotsa and P. Kutas and S.-P. Merz and Y. B. Ti, *On the isogeny problem with torsion points*. To appear at PKC 2022. Eprint, 2021.
4. T. B. Fouotsa and C. Petit, *A New Adaptive attack on SIDH*. To appear at CT-RSA 2022. Eprint, 2021.
3. T. B. Fouotsa and C. Petit, *SHealS and HealS: Isogeny-Based PKEs from a Key Validation Method for SIDH*. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 279-307, Springer, Cham, 2021. Paper Eprint, 2021.
2. L. De Feo, C. D. de Saint-Guilhem, T. B. Fouotsa, A. Leroux, P. Kutas, C. Petit, J. Silva and B. Wesolowski, *SETA: Supersingular Encryption from Torsion point Attacks*. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 249–278, Springer, Cham, 2021. Paper Eprint, 2021.
1. T. B. Fouotsa and C. Petit, *SimS: A simplification of SiGamal*. In Jung Hee Cheon and Jean-Pierre Tillich, editors,*Post-Quantum Cryptography*, pages 277–295, Cham, 2021. Springer International Publishing. Paper Eprint, 2021.

Bibliography

[AAM19] Gora Adj, Omran Ahmadi and Alfred Menezes. "On isogeny graphs of supersingular elliptic curves over finite fields". In: *Finite Fields and Their Applications* 55 (2019), pp. 268–283.

[ABLS07] Noga Alon, Itai Benjamini, Eyal Lubetzky and Sasha Sodin. "Non-backtracking random walks mix faster". In: *Communications in Contemporary Mathematics* 9.04 (2007), pp. 585–603.

[Adj+18] Gora Adj, Daniel Cervantes-Vázquez, Jesús-Javier Chi-Domínguez, A. Menezes and F. Rodríguez-Henríquez. "On the cost of computing isogenies between supersingular elliptic curves". In: *IACR Cryptol. ePrint Arch.* 2018.

[AJL17] Reza Azarderakhsh, David Jao and Christopher Leonardi. "Post-quantum static-static key agreement using multiple protocol instances". In: *International Conference on Selected Areas in Cryptography*. Springer. 2017, pp. 45–63.

[Aza+16] Reza Azarderakhsh, David Jao, Kassem Kalach, Brian Koziel and Christopher Leonardi. "Key Compression for Isogeny-Based Cryptosystems". In: *Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography*. AsiaPKC '16. Xi'an, China: ACM, 2016, pp. 1–10. ISBN: 978-1-4503-4286-5. DOI: 10.1145/2898420.2898421. URL: http://doi.acm.org/10.1145/2898420.2898421.

[Aza+20] Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Aaron Hutchinson, Amir Jalali, David Jao, Koray Karabina, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Geovandro Pereira, Joost Renes, Vladimir Soukharev and David Urbanik. *Supersingular Isogeny Key Encapsulation*. 10th Oct. 2020. URL: http://sike.org.

[Bas+20] Andrea Basso, Péter Kutas, Simon-Philipp Merz, Christophe Petit and Charlotte Weitkämper. "On Adaptive Attacks Against Jao-Urbanik's Isogeny-Based Protocol". In: *Progress in Cryptology - AFRICACRYPT 2020*. Ed. by Abderrahmane Nitaj and Amr Youssef. Cham: Springer International Publishing, 2020, pp. 195–213. ISBN: 978-3-030-51938-4.

[Bas+21] Andrea Basso, Péter Kutas, Simon-Philipp Merz, Christophe Petit and Antonio Sanso. *Cryptanalysis of an oblivious PRF from supersingular isogenies*. Cryptology ePrint Archive, Report 2021/706. https://ia.cr/2021/706. 2021.

[BDLS20] Daniel J Bernstein, Luca De Feo, Antonin Leroux and Benjamin Smith. "Faster computation of isogenies of large prime degree". In: *Open Book Series* 4.1 (2020), pp. 39–55. DOI: 10.2140/obs.2020.4.39.

[Ben80] Paul Benioff. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". In: *Journal of Statistical Physics* 22 (May 1980), pp. 563–591. DOI: 10.1007/BF01011339.

[BFLS20] Daniel J. Bernstein, Luca De Feo, Antonin Leroux and Benjamin Smith. *Faster computation of isogenies of large prime degree*. Cryptology ePrint Archive, Report 2020/341. https://eprint.iacr.org/2020/341. 2020.

[Bis12] Gaetan Bisson. "Computing endomorphism rings of elliptic curves under the GRH". In: *Journal of Mathematical Cryptology* 5.2 (2012), pp. 101–114. DOI: doi:10.1515/jmc.2011.008. URL: https://doi.org/10.1515/jmc.2011.008.

[BJS14] Jean-Franccois Biasse, David Jao and Anirudh Sankar. "A quantum algorithm for computing isogenies between supersingular elliptic curves". In: *International Conference on Cryptology in India*. Springer. 2014, pp. 428–442.

[BKV19] Ward Beullens, Thorsten Kleinjung and Frederik Vercauteren. "CSI-FiSh: Efficient Isogeny Based Signatures Through Class Group Computations". In: *Advances in Cryptology – ASIACRYPT 2019*. Ed. by Steven D. Galbraith and Shiho Moriai. Cham: Springer International Publishing, 2019, pp. 227–247.

[BL11] Daniel J Bernstein and Tanja Lange. "A complete set of addition laws for incomplete Edwards curves". In: *Journal of Number Theory* 131.5 (2011), pp. 858–872.

[Boy08] Xavier Boyen. "The Uber-Assumption Family". In: *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*. Ed. by Steven D. Galbraith and Kenneth G. Paterson. Vol. 5209. Lecture Notes in Computer Science. Springer, 2008, pp. 39–56. DOI: 10.1007/978-3-540-85538-5_3. URL: https://doi.org/10.1007/978-3-540-85538-5%5C_3.

[BR94] Mihir Bellare and Phillip Rogaway. "Optimal asymmetric encryption". In: *Workshop on the Theory and Application of Cryptographic Techniques*. Springer. 1994, pp. 92–111.

[BR97] Mihir Bellare and Phillip Rogaway. "Minimizing the use of random oracles in authenticated encryption schemes". In: *International Conference on Information and Communications Security*. Springer. 1997, pp. 1–16.

[Brö09] Reinier Bröker. "Constructing supersingular elliptic curves". In: *Journal of Combinatorics and Number Theory* 1.3 (2009).

[BS11] Gaetan Bisson and Andrew V. Sutherland. "Computing the endomorphism ring of an ordinary elliptic curve over a finite field". In: *Journal of Number Theory* 131.5 (May 2011), pp. 815–831. ISSN: 0022-314X. DOI: 10.1016/j.jnt.2009.11.003. URL: http://dx.doi.org/10.1016/j.jnt.2009.11.003.

[BS20] Xavier Bonnetain and André Schrottenloher. "Quantum Security Analysis of CSIDH". In: *Advances in Cryptology – EUROCRYPT 2020* 12106 (2020), pp. 493–522.

[Cas+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny and Joost Renes. "CSIDH: an efficient post-quantum commutative group action". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2018, pp. 395–427.

[CCJR20] Jorge Chávez-Saab, Jesús-Javier Chi-Domınguez, Samuel Jaques and Francisco Rodrıguez-Henrıquez. *The SQALE of CSIDH: Square-root vélu Quantum-resistant isogeny Action with Low Exponents*. Tech. rep. Cryptology ePrint Archive, Report 2020/1520, 2020. https://eprint.iacr. org ..., 2020.

[CD20] Wouter Castryck and Thomas Decru. "CSIDH on the surface". eng. In: *Post-quantum cryptography, 11th international conference, PQCrypto 2020*. Ed. by J. Ding and J. P. Tillich. Vol. 12100. Paris, FRANCE: Springer, 2020, pp. 111–129. ISBN: 9783030442224. URL: http://dx.doi.org/10.1007/978-3-030-44223-1_7.

[CDV20] Wouter Castryck, Thomas Decru and Frederik Vercauteren. "Radical Isogenies". In: *Advances in Cryptology – ASIACRYPT 2020*. Ed. by Shiho Moriai and Huaxiong Wang. Cham: Springer International Publishing, 2020, pp. 493–519. ISBN: 978-3-030-64834-3.

[CH] Craig Costello and Huseyin Hisil. *A simple and compact algorithm for SIDH with arbitrary degree isogenies*. In: Takagi T., Peyrin T. (eds) Advances in Cryptology – ASIACRYPT 2017. ASIACRYPT 2017. Lecture Notes in Computer Science, vol 10625. Springer, Cham. https://doi.org/10.1007/978-3-319-70697-9_11.

[CJS14] Andrew Childs, David Jao and Vladimir Soukharev. "Constructing elliptic curve isogenies in quantum subexponential time". In: *Journal of Mathematical Cryptology* 8.1 (2014), pp. 1–29.

[CK20] Leonardo Colò and David Kohel. "Orienting supersingular isogeny graphs". In: *Journal of Mathematical Cryptology* 14.1 (2020), pp. 414–437.

[CLG09] Denis X Charles, Kristin E Lauter and Eyal Z Goren. "Cryptographic hash functions from expander graphs". In: *Journal of Cryptology* 22.1 (2009), pp. 93–113.

[CLN16] Craig Costello, Patrick Longa and Michael Naehrig. "Efficient Algorithms for Supersingular Isogeny Diffie-Hellman". In: *Advances in Cryptology – CRYPTO 2016*. Ed. by Matthew Robshaw and Jonathan Katz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 572–601. ISBN: 978-3-662-53018-4.

[CMN20] Craig Costello, Michael Meyer and Michael Naehrig. *Sieving for twin smooth integers with solutions to the Prouhet-Tarry-Escott problem*. Cryptology ePrint Archive, Report 2020/1283. 2020. URL: https://eprint.iacr.org/2020/1283.

[COR20] Daniel Cervantes-Vázquez, Eduardo Ochoa-Jiménez and Francisco Rodríguez-Henríquez. *eSIDH: the revenge of the SIDH*. Cryptology ePrint Archive, Report 2020/021. https://ia.cr/2020/021. 2020.

[Cos+20] C. Costello, P. Longa, M. Naehrig, J. Renes and Fernando Virdia. "Improved Classical Cryptanalysis of SIKE in Practice". In: *Public Key Cryptography*. 2020.

[Cos20] Craig Costello. "B-SIDH: Supersingular Isogeny Diffie-Hellman Using Twisted Torsion". In: *Advances in Cryptology - ASIACRYPT 2020, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*. 2020, pp. 440–463.

[Cos21] Craig Costello. *The Case for SIKE: A Decade of the Supersingular Isogeny Problem*. Cryptology ePrint Archive, Report 2021/543. https://ia.cr/2021/543. 2021.

[Cou06] Jean-Marc Couveignes. *Hard Homogeneous Spaces*. Cryptology ePrint Archive, Report 2006/291. https://eprint.iacr.org/2006/291. 2006.

[Cox14] D.A. Cox. *Primes of the Form x2+ny2: Fermat, Class Field Theory, and Complex Multiplication*. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley, 2014. ISBN: 9781118400746. URL: https://books.google.it/books?id=NklYBAAAQBAJ.

[CPV20] Wouter Castryck, Lorenz Panny and Frederik Vercauteren. "Rational isogenies from irrational endomorphisms". In: *Advances in Cryptology–EUROCRYPT 2020* 12106 (2020), p. 523.

[CS21] Mathilde Chenu and Benjamin Smith. "Higher-degree supersingular group actions". In: *arXiv preprint arXiv:2107.08832* (2021).

[CSV20] Wouter Castryck, Jana Sotáková and Frederik Vercauteren. "Breaking the Decisional Diffie-Hellman Problem for Class Group Actions Using Genus Theory". In: *Advances in Cryptology – CRYPTO 2020*. Ed. by Daniele Micciancio and Thomas Ristenpart. Cham: Springer International Publishing, 2020, pp. 92–120. ISBN: 978-3-030-56880-1.

[Dam92] Ivan Damgård. "Towards Practical Public Key Systems Secure Against Chosen Ciphertext attacks". In: *Advances in Cryptology — CRYPTO '91*. Ed. by Joan Feigenbaum. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 445–456. ISBN: 978-3-540-46766-3.

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

[De 17] Luca De Feo. "Mathematics of isogeny based cryptography". In: *arXiv preprint:1711.04062* (2017).

[DG16] Christina Delfs and Steven D Galbraith. "Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$". In: *Designs, Codes and Cryptography* 78.2 (2016), pp. 425–440.

[DH76] Whitfield Diffie and Martin E. Hellman. "New directions in cryptography". In: *IEEE Trans. Information Theory* 22.6 (Nov. 1976), pp. 644–654.

[DMPS19] Luca De Feo, Simon Masson, Christophe Petit and Antonio Sanso. "Verifiable Delay Functions from Supersingular Isogenies and Pairings". In: *Advances in Cryptology – ASIACRYPT 2019*. Ed. by Steven D. Galbraith and Shiho Moriai. Cham: Springer International Publishing, 2019, pp. 248–277. ISBN: 978-3-030-34578-5.

[Dob+20] Samuel Dobson, Steven D. Galbraith, Jason LeGrow, Yan Bo Ti and Lukas Zobernig. "An adaptive attack on 2-SIDH". In: *International Journal of Computer Mathematics: Computer Systems Theory* 5.4 (2020), pp. 282–299. DOI: 10.1080/23799927.2020.1822446. URL: https://doi.org/10.1080/23799927.2020.1822446.

[EHM17] Kirsten Eisentraeger, Sean Hallgren and Travis Morrison. *On the Hardness of Computing Endomorphism Rings of Supersingular Elliptic Curves*. Cryptology ePrint Archive, Report 2017/986. https://ia.cr/2017/986. 2017.

[Eis+18] Kirsten Eisenträger, Sean Hallgren, Kristin Lauter, Travis Morrison and Christophe Petit. "Supersingular isogeny graphs and endomorphism rings: reductions and solutions". In: *Advances in Cryptology – EUROCRYPT 2018*. Springer. 2018, pp. 329–368.

[Eis+20] Kirsten Eisentraeger, Sean Hallgren, Chris Leonardi, Travis Morrison and Jennifer Park. "Computing endomorphism rings of supersingular elliptic curves and connections to pathfinding in isogeny graphs". In: *arXiv preprint arXiv:2004.11495* (2020).

[ElG85] Taher ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms". In: *IEEE transactions on information theory* 31.4 (1985), pp. 469–472.

[FDGZ21] Luca De Feo, Samuel Dobson, Steven D. Galbraith and Lukas Zobernig. *SIDH Proof of Knowledge*. Cryptology ePrint Archive, Report 2021/1023. https://ia.cr/2021/1023. 2021.

[Feo+19] Luca De Feo, Cyprien Delpech de Saint Guilhem, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Christophe Petit, Javier Silva and Benjamin Wesolowski. *SÉTA: Supersingular Encryption from Torsion Attacks*. Cryptology ePrint Archive, Report 2019/1291. https://ia.cr/2019/1291. 2019.

[Fey82] Richard Phillips Feynman. "Simulating physics with computers". In: *International Journal of Theoretical Physics* 21 (1982), pp. 467–488.

[FJP14] Luca De Feo, David Jao and Jérôme Plût. *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*. Pagesn 209-247. 2014.

[FKMT21] Tako Boris Fouotsa, Péter Kutas, Simon-Philipp Merz and Yan Bo Ti. *On the Isogeny Problem with Torsion Point Information*. Cryptology ePrint Archive, Report 2021/153. https://eprint.iacr.org/2021/153. 2021.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. "Secure integration of asymmetric and symmetric encryption schemes". In: *Annual International Cryptology Conference*. Springer. 1999, pp. 537–554.

[FP21a] Tako Boris Fouotsa and Christophe Petit. *A New Adaptive Attack on SIDH*. Cryptology ePrint Archive, Report 2021/1322. https://ia.cr/2021/1322. 2021.

[FP21b] Tako Boris Fouotsa and Christophe Petit. "SHealS and HealS: isogeny-based PKEs from a key validation method for SIDH". In: Springer-Verlag, 2021.

[FP21c] Tako Boris Fouotsa and Christophe Petit. "SimS: A Simplification of SiGamal". In: *Post-Quantum Cryptography*. Ed. by Jung Hee Cheon and Jean-Pierre Tillich. Cham: Springer International Publishing, 2021, pp. 277–295. ISBN: 978-3-030-81293-5.

[Gal12] Steven D. Galbraith. *Mathematics of Public Key Cryptography*. Cambridge University Press, 2012. DOI: 10.1017/CBO9781139012843.

[Gal99] Steven D Galbraith. "Constructing isogenies between elliptic curves over finite fields". In: *LMS Journal of Computation and Mathematics* 2 (1999), pp. 118–138.

[GPS17] Steven D. Galbraith, Christophe Petit and Javier Silva. "Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems". In: *Advances in Cryptology – ASIACRYPT 2017*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Springer International Publishing, 2017, pp. 3–33.

[GPS20] Steven D Galbraith, Christophe Petit and Javier Silva. "Identification protocols and signature schemes based on supersingular isogeny problems". In: *Journal of Cryptology* 33.1 (2020), pp. 130–175.

[GPST16] Steven D Galbraith, Christophe Petit, Barak Shani and Yan Bo Ti. "On the security of supersingular isogeny cryptosystems". In: *Advances in Cryptology – ASIACRYPT 2016*. Springer. 2016, pp. 63–91.

[Gro96] Lov K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*. 1996, pp. 212–219.

[Ham12] Mike Hamburg. *Fast and compact elliptic-curve cryptography*. Cryptology ePrint Archive, Report 2012/309. https://eprint.iacr.org/2012/309. 2012.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns and Eike Kiltz. "A modular analysis of the Fujisaki-Okamoto transformation". In: *Theory of Cryptography Conference*. Springer. 2017, pp. 341–371.

[HT98] Satoshi Hada and Toshiaki Tanaka. "On the existence of 3-round zero-knowledge protocols". In: *Advances in Cryptology — CRYPTO '98*. Ed. by Hugo Krawczyk. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 408–423. ISBN: 978-3-540-68462-6.

[Jao+17] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev and David Urbanik. *SIKE: Supersingular Isogeny Key Encapsulation*. http://sike.org/. 2017.

[Jao+20] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Aaron Hutchinson, Amir Jalali, Koray Karabina, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Geovandro Pereira, Joost Renes, Vladimir Soukharev and David Urbanik. *Supersingular Isogeny Key Encapsulation*. October 1, https://sike.org/files/SIDH-spec.pdf. 2020.

[JD11] David Jao and Luca De Feo. "Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies". In: *Post-Quantum Cryptography*. Ed. by Bo-Yin Yang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 19–34. ISBN: 978-3-642-25405-5.

[JMV09] David Jao, Stephen D Miller and Ramarathnam Venkatesan. "Expander graphs based on GRH with an application to elliptic curve cryptography". In: *Journal of Number Theory* 129.6 (2009), pp. 1491–1504.

[JS19] Samuel Jaques and John M. Schanck. "Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE". In: *Advances in Cryptology – CRYPTO 2019*. Springer. 2019, pp. 32–61.

[JS20] Samuel Jaques and André Schrottenloher. *Low-gate Quantum Golden Collision Finding*. Cryptology ePrint Archive, Report 2020/424. https://eprint.iacr.org/2020/424. 2020.

[Kan89] Masanobu Kaneko. "Supersingular j-invariants as singular moduli mod p". In: *Osaka Journal of Mathematics* 26.4 (1989), pp. 849–855.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007. ISBN: 978-1-58488-551-1.

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

[KMPW21] Péter Kutas, Simon-Philipp Merz, Christophe Petit and Charlotte Weitkämper. "One-way functions and malleability oracles: Hidden shift attacks on isogeny-based protocols". In: *IACR Cryptol. ePrint Arch.* 2021 (2021), p. 282.

[Kob87] Neal Koblitz. "Elliptic curve cryptosystems". In: *Mathematics of computation* 48.177 (1987), pp. 203–209.

[Koh96] David Russell Kohel. "Endomorphism rings of elliptic curves over finite fields". PhD thesis. University of California, Berkeley, 1996.

[Kum47] E. Kummer. "Zur Theorie der complexen Zahlen." In: *Journal für die reine und angewandte Mathematik (Crelles Journal)* (1847), pp. 319–326.

[Lau17] Kristin E. Lauter. "Postquantum Opportunities: Lattices, Homomorphic Encryption, and Supersingular Isogeny Graphs". In: *IEEE Security & Privacy* 15 (2017), pp. 22–27.

[LB20] Jonathan Love and Dan Boneh. "Supersingular curves with small non-integer endomorphisms". In: *Open Book Series* 4.1 (2020), pp. 7–22.

[Leo20] Christopher Leonardi. *A Note on the Ending Elliptic Curve in SIDH*. Cryptology ePrint Archive, Report 2020/262. https://eprint.iacr.org/2020/262. 2020.

[LLL82] Arjen K Lenstra, Hendrik Willem Lenstra and László Lovász. "Factoring polynomials with rational coefficients". In: *Mathematische annalen* 261.ARTICLE (1982), pp. 515–534.

[LWS20] Patrick Longa, Wen Wang and Jakub Szefer. *The Cost to Break SIKE: A Comparative Hardware-Based Analysis with AES and SHA-3*. Cryptology ePrint Archive, Report 2020/1457. https://eprint.iacr.org/2020/1457. 2020.

[Man80] I.U.I. Manin. *Vychislimoe i nevychislimoe*. Sov. radio, 1980. URL: https://books.google.cm/books?id=pAo-zgEACAAJ.

[Mil85] Victor S Miller. "Use of elliptic curves in cryptography". In: *Conference on the theory and application of cryptographic techniques*. Springer. 1985, pp. 417–426.

[Mon87] Peter L Montgomery. "Speeding the Pollard and elliptic curve methods of factorization". In: *Mathematics of computation* 48.177 (1987), pp. 243–264.

[Mor20] Tomoki Moriya. *Magma codes for SiGamal*. Online, August 14, http://tomoriya.work/code.html. 2020.

[MOT20] Tomoki Moriya, Hiroshi Onuki and Tsuyoshi Takagi. "SiGamal: A Supersingular Isogeny-Based PKE and Its Application to a PRF". In: *Advances in Cryptology – ASIACRYPT 2020*. Ed. by Shiho Moriai and Huaxiong Wang. Cham: Springer International Publishing, 2020, pp. 551–580. ISBN: 978-3-030-64834-3.

[MP19] Chloe Martindale and Lorenz Panny. *How to not break SIDH*. Cryptology ePrint Archive, Report 2019/558. https://eprint.iacr.org/2019/558. 2019.

[MP21] Chloe Martindale and Christophe Petit. *Isogeny-based cryptography school*. Online and University of Bristol, https://isogenyschool2020.co.uk. July 2021.

[Nao03] Moni Naor. "On Cryptographic Assumptions and Challenges". In: *Advances in Cryptology - CRYPTO 2003*. Ed. by Dan Boneh. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 96–109. ISBN: 978-3-540-45146-4.

[Nat] National Institute for Standards and Technology (NIST). "Post-quantum crypto standardization (2016), https://csrc.nist.gov/projects/post-quantum-cryptography". In: ().

[NS04] Phong Q. Nguyen and Damien Stehle. "Low-dimensional lattice basis reduction revisited". English. In: *ANTS 2004*. Ed. by Duncan A. Buell. United States: Springer, Springer Nature, 2004, pp. 338–357. ISBN: 9783540221562. DOI: 10.1007/978-3-540-24847-7_26.

[Onu21] Hiroshi Onuki. "On oriented supersingular elliptic curves". In: *Finite Fields and Their Applications* 69 (2021), p. 101777. ISSN: 1071-5797. DOI: https://doi.org/10.1016/j.ffa.2020.101777. URL: https://www.sciencedirect.com/science/article/pii/S1071579720301465.

[Pan21] Lorenz Panny. "Cryptography on Isogeny Graphs". PhD thesis. Technische Universiteit Eindhoven, 2021. URL: https://yx7.cc/docs/phd/thesis.pdf.

[PAR] PARI Group. *PARI/GP version `2.12.0`*. available from http://pari.math.u-bordeaux.fr/. Université de Bordeaux, 2021.

[Pei20] Chris Peikert. "He Gives C-Sieves on the CSIDH". In: *Advances in Cryptology - EUROCRYPT 2020*. 2020, pp. 463–492. DOI: 10.1007/978-3-030-45724-2_16. URL: https://doi.org/10.1007/978-3-030-45724-2%5C_16.

[Pet17] Christophe Petit. "Faster algorithms for isogeny problems using torsion point images". In: *Advances in Cryptology – ASIACRYPT 2017*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Springer International Publishing, 2017, pp. 330–353.

[PH78] Stephen Pohlig and Martin Hellman. "An improved algorithm for computing logarithms over GF (p) and its cryptographic significance (corresp.)" In: *IEEE Transactions on information Theory* 24.1 (1978), pp. 106–110.

[Piz90] Arnold Pizer. "Ramanujan graphs and Hecke operators". In: *Bulletin of the American Mathematical Society* 23 (1990), pp. 127–137.

[PL17] Christophe Petit and Kristin Lauter. *Hard and Easy Problems for Supersingular Isogeny Graphs*. Cryptology ePrint Archive, Report 2017/962. https://eprint.iacr.org/2017/962. 2017.

[Que+21] Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit and Katherine E. Stange. "Improved Torsion-Point Attacks on SIDH Variants". In: *Advances in Cryptology – CRYPTO 2021*. Ed. by Tal Malkin and Chris Peikert. Cham: Springer International Publishing, 2021, pp. 432–470.

[Ren18] Joost Renes. "Computing Isogenies Between Montgomery Curves Using the Action of (0, 0)". In: *Post-Quantum Cryptography*. Ed. by Tanja Lange and Rainer Steinwandt. Cham: Springer International Publishing, 2018, pp. 229–247. ISBN: 978-3-319-79063-3.

[RS06] Alexander Rostovtsev and Anton Stolbunov. "Public-Key Cryptosystem Based on Isogenies." In: *IACR Cryptology ePrint Archive* 2006 (2006), p. 145.

[RSA78] R. Rivest, A. Shamir and L. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Communications of the ACM, 21 (2): 120–126. http://people.csail.mit.edu/rivest/Rsapaper.pdf. 1978.

[Sch85] René Schoof. "Elliptic curves over finite fields and the computation of square roots mod ". In: *Mathematics of computation* 44.170 (1985), pp. 483–494.

[Sch87] René Schoof. "Nonsingular plane cubic curves over finite fields." In: *J. Comb. Theory, Ser. A* 46.2 (1987), pp. 183–211.

[SCS21] Maria Corte-Real Santos, Craig Costello and Jia Shi. *SuperSolver: accelerating the Delfs-Galbraith algorithm with fast subfield root detection*. Cryptology ePrint Archive, Report 2021/1488. https://ia.cr/2021/1488. 2021.

[Sho04] Victor Shoup. *Sequences of games: a tool for taming complexity in security proofs*. Cryptology ePrint Archive, Report 2004/332. https://ia.cr/2004/332. 2004.

[Sho94] P. W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134.

[Sho97] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM J. Comput.* 26.5 (1997), pp. 1484–1509.

[Sil09] Joseph H Silverman. *The arithmetic of elliptic curves*. Vol. 106. Springer Science & Business Media, 2009.

[Sil94] Joseph H Silverman. *Advanced topics in the arithmetic of elliptic curves*. Vol. 151. Graduate Texts in Mathematics. Springer-Verlag, New-York, 1994.

[Sim05] Denis Simon. "Quadratic equations in dimensions 4, 5 and more". In: *Preprint* (2005).

[SKPS19] Cyprien Delpech de Saint Guilhem, Péter Kutas, Christophe Petit and Javier Silva. *SÉTA: Supersingular Encryption from Torsion Attacks*. Cryptology ePrint Archive, Report 2019/1291. https://eprint.iacr.org/2019/1291. 2019.

[Sto10] Anton Stolbunov. "Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves." In: *Adv. in Math. of Comm.* 4.2 (2010), pp. 215–235.

[Sut13] Andrew Sutherland. "Isogeny volcanoes". In: *The Open Book Series* 1.1 (2013), pp. 507–530.

[Sut17] Andrew V. Sutherland. *Elliptic Curves*. MIT Mathematics Lectures notes, 18.783/2027. https://math.mit.edu/classes/18.783/2017/. 2017.

[The20] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.0)*. https://www.sagemath.org. 2020.

[TU16] Ehsan Ebrahimi Targhi and Dominique Unruh. "Post-quantum security of the Fujisaki-Okamoto and OAEP transforms". In: *Theory of Cryptography Conference*. Springer. 2016, pp. 192–216.

[UJ20] David Urbanik and David Jao. "New techniques for SIDH-based NIKE". In: *Journal of Mathematical Cryptology* 14.1 (2020), pp. 120–128.

[Vél71] Jacques Vélu. "Isogénies entre courbes elliptiques". In: *CR Acad. Sci. Paris, Séries A* 273 (1971), pp. 305–347.

[Voi13] John Voight. "Identifying the matrix ring: algorithms for quaternion algebras and quadratic forms". In: *Quadratic and higher degree forms*. Springer, 2013, pp. 255–298.

[Voi18] John Voight. *Quaternion algebras*. Preprint, 2018.

[VW99] Paul C Van Oorschot and Michael J Wiener. "Parallel collision search with cryptanalytic applications". In: *Journal of cryptology* 12.1 (1999), pp. 1–28.

[Was08] Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography, Second Edition*. 2nd ed. Chapman & Hall/CRC, 2008. ISBN: 9781420071467.

[Wes21] Benjamin Wesolowski. *The supersingular isogeny path and endomorphism ring problems are equivalent*. Cryptology ePrint Archive, Report 2021/919. https://ia.cr/2021/919. 2021.