



**UNIVERSITY  
OF TURKU**

ON THE POST-QUANTUM FUTURE OF ELLIPTIC CURVE  
CRYPTOGRAPHY

Rayen Lucaroni

MSc Thesis  
May 2023

DEPARTMENT OF MATHEMATICS AND STATISTICS

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service

UNIVERSITY OF TURKU  
Department of Mathematics and Statistics

LUCARONI, RAYEN:

On the post-quantum future of Elliptic Curve Cryptography

MSc Thesis, 40 pages, 16 appendix pages

Mathematics

May 2023

---

This thesis is a literature study on current published quantum-resistant isogeny-based key exchange protocols.

Here we cover the topic from foundations. Chapters 1 and 2 discuss classical computation models, algorithm complexity, and how these concepts support the security of modern elliptic curve cryptography methods, such as ECDH and ECDSA.

Next, in Chapters 3 to 5, we present quantum computation models, and how Shor's algorithm on quantum computers presents a threat to the future security of classical asymmetric cryptography. We explore the foundations of isogeny-based cryptography, and two key exchange protocols of this kind: SIDH and CSIDH.

Appendices A and B are provided for readers wanting more in-depth background explanations on the algebraic geometry of elliptic curves, and quantum mechanics respectively.

Keywords: cryptography, isogeny-based cryptography, elliptic curves, ECDH, ECDSA, quantum computing, Shor's algorithm, key exchange SIDH, CSIDH.



# Contents

<b>Introduction</b>	<b>1</b>
<b>1 Preliminaries</b>	<b>3</b>
1.1 Problem Complexity . . . . .	3
1.2 One-way functions . . . . .	6
1.3 The cipher machine . . . . .	7
1.4 Digital signatures . . . . .	10
1.5 Two example cryptosystems . . . . .	11
1.5.1 Diffie-Hellman key exchange protocol . . . . .	11
1.5.2 RSA . . . . .	12
<b>2 Classical Elliptic Curve Cryptography</b>	<b>15</b>
2.1 The group of points on an Elliptic Curve . . . . .	15
2.2 ECDH key exchange protocol . . . . .	16
2.3 ECDSA . . . . .	17
2.4 Benefits of ECC . . . . .	18
<b>3 Quantum machines</b>	<b>21</b>
3.1 Quantum computation . . . . .	21
3.1.1 Quantum circuits . . . . .	23
3.2 Quantum algorithms . . . . .	24
3.2.1 Quantum parallelism . . . . .	24
3.2.2 The quantum Fourier transform . . . . .	25
3.2.3 Shor's factoring algorithm . . . . .	25
<b>4 SIDH: a broken protocol</b>	<b>29</b>
4.1 Supersingular elliptic curves . . . . .	29
4.2 Supersingular Isogeny Diffie Hellman . . . . .	31
4.2.1 Set-up . . . . .	32
4.2.2 Key exchange protocol . . . . .	32
4.3 The Castryck-Decru attack . . . . .	32
4.3.1 Preliminaries . . . . .	33
4.3.2 The attack . . . . .	34
<b>5 CSIDH</b>	<b>37</b>
5.1 The class-group action . . . . .	37
5.2 CSIDH . . . . .	39

5.3	Discussion . . . . .	40
<b>A</b>	<b>Geometric principles of Elliptic Curves</b>	<b>41</b>
A.1	The projective plane . . . . .	41
A.2	Elliptic curves . . . . .	43
A.2.1	The Weierstrass equation . . . . .	43
A.3	Tangent and secant lines . . . . .	46
A.4	Isogenies . . . . .	48
A.4.1	The group of isogenies . . . . .	50
A.4.2	The dual isogeny . . . . .	51
<b>B</b>	<b>A brief discussion of quantum physics</b>	<b>53</b>
B.1	Foundational principles of quantum mechanics . . . . .	53
B.2	Qubit systems . . . . .	55
B.3	Measuring a quantum system . . . . .	56
	<b>Bibliography</b>	<b>59</b>

# Introduction

There are few areas of mathematics that have impacted the course of human history quite like cryptography. Since the dawn of societies war, intrigue and trade have demanded a need for secret communication. It is the computer, however, that has brought a new age of cryptography, where the evolution of the former spurs that of the latter.

## Counting by hand

Atbash is an example of one of the very first encryption protocols. It was known and used by Hebrew scholars as far back as 600 BC, its legacy immortalised in the Old Testament. Atbash is an instance of the affine cipher, and simply works by mapping the 22-letter Hebrew alphabet to its reverse.

$$\text{Atbash : } E(x) = D(x) = -(x + 1) \pmod{22}$$

Before the 19<sup>th</sup> century, substitution and transposition ciphers were the only methods used for encryption. In antiquity these protocols were exceedingly simple by modern standards. While there are examples of them being used for the purpose of secret communication, famously by Julius Caesar, most were designed to amuse, add mystique or for superstitious purposes.

At this time, in terms of cryptanalysis, arguably the single most important advancement was made around the year 800, by the philosopher, scientist and mathematician Al-Kindi, who is credited with inventing frequency analysis [19]. After him, there have been many skilled cryptanalysts, usually employed in the courts of kings and queens, but it was not until 1850 that the link between these simple ciphers and modular arithmetic was found. This discovery came from Charles Babbage, interestingly the very same person credited with creating the first computer, and inventing the definition of a programmable computer.

## The invention of computers

"I was instructed to destroy all the records, which I did. I took all the drawings and the plans and all the information about Colossus on paper and put it in the boiler fire. And saw it burn."

- Tommy Flowers

This was the fate of the very first iteration of a programmable computer, Colossus, made at Bletchley Park for the purpose of breaking German teleprinter stream ciphers, known collectively as Fish, during World War II.

Colossus could do  $500 \times 10^3$  floating point operations per second (FLOPS), completing a set of runs for a message tape in as little as two minutes. The Hewlett Packard Enterprise Frontier supercomputer, which began deployment in 2021,<sup>1</sup> is capable of up to  $1.6 \times 10^{18}$  FLOPS [14].

It is clear to see that the maximum performance of computers has seen an explosion in the last 80 years; the first computers can no longer compete even against the average personal computer or mobile phone. With most people in the world having multiple orders of magnitude more than the power of Colossus at their fingertips, in an environment of constant massive data sharing, it only seems natural that cryptography would be irrevocably transformed as well.

This transformation began in 1945, with Claude E. Shannon's work on mathematically provable secrecy, but the real turning point came in 1976 and 1978, with the invention of the Diffie-Hellman key exchange protocol, and the RSA cryptosystem. This was the dawn of public key cryptography, which addressed the problem of key distribution, but at the same time generated an arms race between computational power and complexity of cryptosystems.

## The quantum future

In 1998 the world saw the first working quantum computer, the result of decades of work. Today, there are still few quantum computers in the world, and none of them can outperform classical computers yet, but with every passing year, the quantum future seems less of a science fiction dream and more a reality.

In this thesis we explore the foundations of public key cryptography, the current use of elliptic curves in public key protocols on classical computers, and how, the potential advent of quantum computing will break such algorithms. We also present the post-quantum elliptic curve cryptography key exchange protocols SIDH and CSIDH, and the Castryck-Decru attack on SIDH.

---

<sup>1</sup>Considered the most powerful as of June 2022, according to the Top500 project.



# Chapter 1

## Preliminaries

1. Some problems are more difficult than others.
2. Some problems are more difficult than their inverse.

These two deceptively simple statements are the foundations of all public key cryptography. In order to be able to rigorously reason about, firstly whether these statements are indeed true, and secondly how they can help us to design secure systems, we must first express them in a precise manner.

### 1.1 Problem Complexity

There are various different ways to answer the question of how complex it is to solve a problem. One way is to analyse how many resources would be used by a machine to find a solution to an instance of the problem. The issue with this, is that a machine uses several different resources (e.g. time, memory, space), and even these are variable depending on the particular machine and problem instance we choose.

Thankfully for us, there is a model for computation that is both mathematically relatively simple, and is able to simulate all our currently used computational methods in an efficient manner<sup>1</sup>, that is the Turing machine.

**Definition 1.1.1.** (Deterministic Turing machine) A deterministic Turing machine is a 7-tuple  $M = \langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$ , defined in the following way

- $Q$  is a finite, non-empty set of states
- $\Gamma$  is a finite, non-empty set of symbols, called the alphabet
- $b \in \Gamma$  is the blank symbol
- $\Sigma \subset \Gamma \setminus b$  is the set of input symbols
- $q_0 \in Q$  is the initial state
- $F \subset Q$  is the set of final or accepting states

---

<sup>1</sup>See definition 1.1.4 for what is meant by "efficient".

- $\delta : (Q \setminus F) \times \Gamma \rightharpoonup Q \times \Gamma \times \{L, R\}$  is the transition function, a partial function mapping a state, symbol tuple to another, and performing a left shift( $L$ ) or right shift ( $R$ )

In more intuitive terms, we can represent a Turing machine as a cellular tape, where each cell  $i$  contains some symbol  $\gamma_i$  from our alphabet  $\Gamma$ , and on this tape the head of the machine travels left or right at each step, performing some transformation of the current symbol and its internal state  $q \in Q$ .<sup>2</sup>

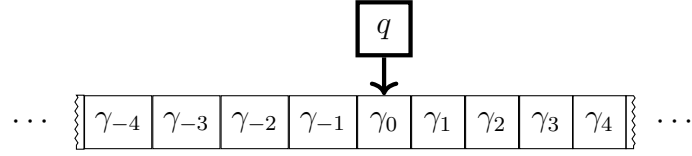


Figure 1.1: A visual representation of a Turing machine.

Using our model, we can now formally describe algorithms to carry out simple computations.

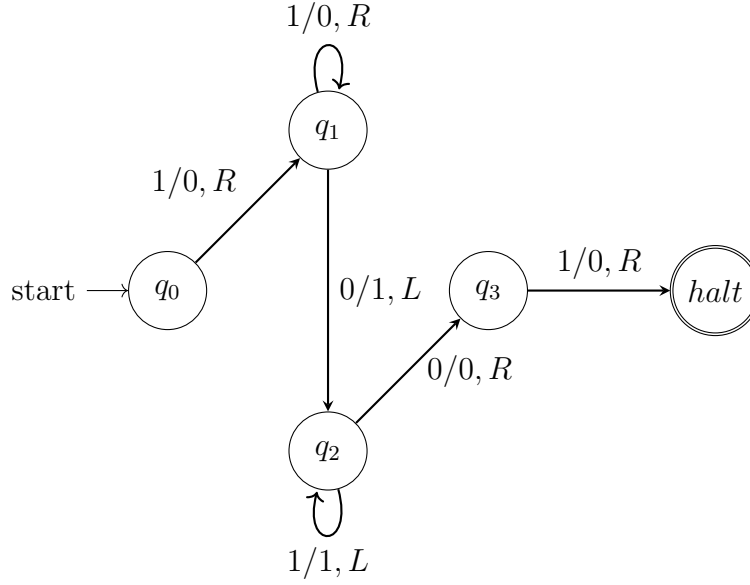
**Example 1.1.1.** *Let us define the Turing machine  $+$ , with symbol alphabet  $\Gamma = \{0 = b, 1\}$ , state set  $Q = \{q_0, q_1, q_2, q_3, \text{halt}\}$  and accepting set  $F = \{\text{halt}\}$ . We can implement addition of two positive integers in unary notation by choosing the following transition function rule:*

	0	1
$q_0$	-	$0, q_1, R$
$q_1$	$1, q_2, L$	$0, q_1, R$
$q_2$	$0, q_3, R$	$1, q_2, L$
$q_3$	-	$0, \text{halt}, R$

*Or as a transition diagram:*

---

<sup>2</sup>I greatly recommend reading Turing's original paper [22] for a more in-depth and historically meaningful account of Turing machines.



If we let our machine run on a tape containing the symbols 11110111 ( $3 + 2$ ), it will return 111111 (5).

Just as we have defined a Turing machine to compute the addition of two positive integers, it can be proven that it is possible to model in this way any algorithm that can be written in some natural or programming language<sup>3</sup>. Furthermore, the Turing machine provides a very natural way to count how long it takes for an algorithm to be computed, as it takes a discrete number of steps for each computation.

**Definition 1.1.2.** (Running time) Let  $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$  be a function of binary words,  $T : \mathbb{N} \rightarrow \mathbb{N}$  a function of natural numbers, and  $M$  a Turing machine. We say  $M$  computes  $f$  if, when it is provided any  $x \in \{0, 1\}^*$  as input, it halts with  $f(x)$  as output. Further, we say it computes  $f$  in time  $T(n)$  if it takes at most  $T(|x|)$  steps to compute  $f(x)$ , where  $|x|$  is the number of bits  $x$  is comprised of.

At this point, we can use our definitions to determine the complexity class of problems, by measuring the running time of algorithms that solve them.

**Definition 1.1.3.** (DTIME) Let  $T : \mathbb{N} \rightarrow \mathbb{N}$  be a function. A problem  $A$  is in the set  $\mathbf{DTIME}(T(n))$  if and only if there exists a Turing machine that runs in time  $c \cdot T(n)$  for some  $c \in \mathbb{N}_{>0}$  and computes solutions to  $A$ .

**Definition 1.1.4.** (Complexity class  $\mathbf{P}$ )

$$\mathbf{P} = \bigcup_{c \geq 1} \mathbf{DTIME}(n^c)$$

$\mathbf{P}$  is also called the polynomial-time class.

For our purposes, we say that problems that fall within  $\mathbf{P}$  are those that can be solved efficiently. Equivalently, this can be stated as, algorithms that run in

<sup>3</sup>For proofs and a more complete discussion of this see [1].

polynomial time are efficient. Thus, we can now devise a way to compare the efficiency of differing algorithms, by comparing the behaviour of their running time function as input size grows.

We say an algorithm  $A$  is *more efficient* than another algorithm  $B$ , if  $T_A(n) < T_B(n)$  for large enough  $n$ , i.e. for large enough input  $A$  runs faster than  $B$ .

We have thus arrived at a more exact formulation of statement 1 at the beginning of this chapter:

**Some algorithms are more efficient than others, with increasing input size.**

Note that, although we only considered deterministic machines here, our definitions are very easily altered to define non-deterministic machines. One such machine is the probabilistic Turing machine.

**Definition 1.1.5.** (Probabilistic Turing machine) A probabilistic Turing machine is an 7-tuple  $M = \langle Q, \Gamma, b, \Sigma, \Delta, q_0, F \rangle$ . Here all elements of the tuple are equivalent to those of the deterministic Turing machine, with the exception of  $\Delta = (\delta_1, \delta_2)$  which is a pair of transition functions. At each step  $M$  probabilistically applies either  $\delta_1$  or  $\delta_2$ , independently of previous steps.

Such machines give us analogous complexity classes to the deterministic case.

**Definition 1.1.6. (BPTIME)** Let  $T : \mathbb{N} \rightarrow \mathbb{N}$  be a function. A problem  $A$  is in the set  $\mathbf{BPTIME}(T(n))$  if and only if there exists a probabilistic Turing machine that runs in time  $c \cdot T(n)$  for some  $c \in \mathbb{N}_{>0}$  and computes solutions to  $A$ , regardless of its random choices with probability greater or equal to  $\frac{2}{3}$ .

**Definition 1.1.7.** (Complexity class **BPP**)

$$\mathbf{BPP} = \bigcup_{c \geq 1} \mathbf{BPTIME}(n^c)$$

## 1.2 One-way functions

**Definition 1.2.1.** (Negligible function) A function  $\nu : \mathbb{N} \rightarrow [0, 1]$  is said to be negligible if for all  $k \in \mathbb{N}$  and sufficiently large  $n \in \mathbb{N}$ ,  $\nu(n) < n^{-k}$ .

**Definition 1.2.2.** (One-way function) A function  $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$  in  $\mathbf{P}$ , is a one-way function if for every polynomial-time algorithm  $A$  there is a negligible function  $\nu : \mathbb{N} \rightarrow [0, 1]$  such that for all  $n \in \mathbb{N}$ :

$$\Pr_{x \in \{0, 1\}^n} [A(y) = x' : f(x') = f(x)] < \nu(n)$$

(i.e. Any efficient algorithm that attempts to compute a pseudo-inverse for  $f$  succeeds with negligible probability)

The one-way function essentially describes a function that can be computed efficiently, but for which there is no efficient way to invert it. It is not known whether such functions exist, but there are a few candidates, that, as far as we have been able to see, may fit the description.

The one-way function is the formalisation of statement 2 in the introduction of this chapter.

**Example 1.2.1.** Let us define the set  $I$ :

$$I = \{(p, g) : p \text{ is prime and } g \text{ is a generator of } \mathbb{Z}_p^*\}$$

Consider now the function  $\exp(p, g, a) = g^a \bmod p$ , with  $(p, g) \in I$ . This is the discrete exponentiation function, and as defined it is bijective. Thus there is an inverse function  $dl(p, g, b) = (p, g, a)$  such that  $g^a \bmod p = b$ .

We know that  $\exp$  can be computed in polynomial time, however no efficient algorithm is known for computing  $dl$ , known as the Discrete Logarithm Problem (DLP).  $\exp$  is thusly considered a possible one-way function.

**Definition 1.2.3.** (Trapdoor function) A one-way function  $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$  is a trapdoor function, if there exists some information  $t$  (known as the trapdoor), such that if  $t$  and  $f(x)$  are given, then it is possible to efficiently compute  $x$ .

**Example 1.2.2.** Let us define the set  $I$ :

$$I = \{(n, e) : n \in \mathbb{Z} \text{ is the product of two primes and } \gcd(e, \varphi(n)) = 1\}$$

Consider now the function  $RSA(n, e, a) = a^e \bmod n$ , with  $(n, e) \in I$ . For every  $(n, e) \in I$  and  $b \in \mathbb{Z}_n$  there is a unique  $a$ , such that  $a^e \bmod n = b$ .

$RSA$  is considered to be a potential one-way function, as it is computable efficiently, but no efficient algorithm is known to compute its inverse. If it is a one-way function, then it is also a trapdoor function.

Let  $d \in \mathbb{Z}_n$  be chosen such that  $e \cdot d = 1 \bmod \varphi(n)$ . Knowing such a  $d$  means we can efficiently find an inverse as follows:

$$b^d = (a^e \bmod n)^d = a^{e \cdot d} \bmod n = a \bmod n$$

Thus  $d$  is the trapdoor.<sup>4</sup>

Assuming one-way functions exist, it is not known if every one-way function can be used to construct a trapdoor function.

## 1.3 The cipher machine

Having established our model of computation, we now look at how we can describe public key cryptosystems using these concepts. We remind ourselves that the goal of a cryptosystem is to preserve *confidentiality*, thus we also need a way to analyse whether our system fulfils that goal.

**Definition 1.3.1.** (Public key cryptosystem) Let  $K$ ,  $M$  and  $C$  be finite sets. A public key cryptosystem is an algorithm triple  $(G, E, D)$ , where

- $G$  is the probabilistic key generation algorithm. Its only input is randomness. It outputs  $(k, k') \in K$ , the public and private keys.

---

<sup>4</sup>This is the basis of the RSA encryption protocol of Subsection 1.5.2

- $E$  is the probabilistic encryption algorithm, which takes the public key  $k$  and message  $m \in M$  as input, and outputs the ciphertext  $c \in C$ .
- $D$  is the deterministic decryption algorithm, which takes the private key  $k'$  and a ciphertext as input and outputs a message in  $M$ , or an error symbol  $\perp$ .

We say such a system is *sound* if for all  $(k, k') \in K$ , and all  $m \in M$

$$D_{k'} \circ E_k(m) = m$$

**Example 1.3.1.** Let  $f : K_1 \times M \rightarrow C$  be a trapdoor function, such that for each  $k \in K_1$ , there is a distinct trapdoor  $k' \in K_2$ , then it can be seen how we can build a sound public key cryptosystem with the blueprint of Figure 1.2, where  $Inv$  is an algorithm computing pseudo-inverses of  $f$  using a trapdoor  $k'$

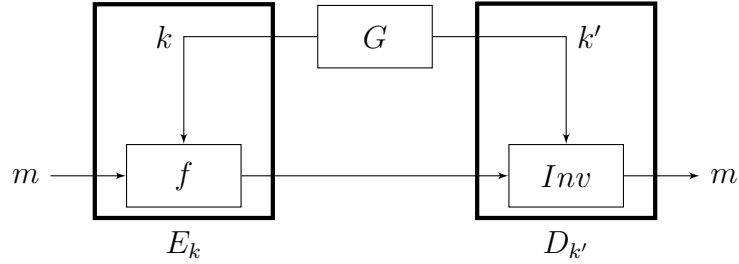


Figure 1.2: An outline of an asymmetric cryptosystem using a trap-door function.

The trouble with public key cryptography is that it cannot be shown that any attacker with arbitrarily large computational resources at their disposal will never be able to break the system's confidentiality in any way. This is because our key set  $K$  is finite, thus given enough time, it is always possible to find the private key by trying every single option (i.e. brute force).

This implies that in order to argue about the security of such cryptosystems, we must restrict ourselves to consider only attackers with very particular limitations.

In this thesis we consider security as defined by the Indistinguishability (IND) game.

Let our attacker be a pair of probabilistic algorithms  $\mathcal{A} = (A_1, A_2)$ .  $A_1$  takes a public key as input and outputs two messages and some internal state  $s$ ,  $A_2$  takes as input the public key, a ciphertext and  $A_1$ 's state  $s$ , then outputs a single bit.

IND

**Challenger**

**Attacker**

Challenger uses the key generator  $G$  to get keys  $(k, k')$

$$A_1(k) = (m_0, m_1, s)$$

$$\xleftarrow{(m_0, m_1)}$$

Choose a random bit  $b$

$$C = E_k(m_b)$$

$$\xrightarrow{C}$$

$$A_2(k, s, C) = b'$$

.....Attacker wins the game if  $b = b'$  .....

Clearly, if  $A_2$  simply randomly chooses the bit  $b'$ , then  $\mathcal{A}$  will win the game with a probability of  $\frac{1}{2}$ . We define the advantage the attacker has in playing the IND game by

$$\text{Adv}_{\mathcal{A}} = \left| \Pr[b = b'] - \frac{1}{2} \right|$$

**Definition 1.3.2.** (Decryption oracle) A decryption oracle is a black box machine that decrypts ciphertext in a single step.

**Definition 1.3.3.** (IND-CPA, IND-CCA1 and IND-CCA2) Let  $\mathcal{A} = (A_1, A_2)$  be an attacker playing the IND game for the public key cryptosystem  $S = (G, E, D)$ .

- If  $\mathcal{A}$  has no access to a decryption oracle, then it is running a *chosen plaintext attack*. If  $\mathcal{A}$ 's advantage in winning the IND game is negligible, then we say  $S$  is IND-CPA secure.
- If  $A_1$  has access to a decryption oracle, but  $A_2$  does not, then  $\mathcal{A}$  is running a *chosen ciphertext attack*. If  $\mathcal{A}$ 's advantage in winning the IND game is negligible, then we say  $S$  is IND-CCA1 secure.
- If  $A_1$  and  $A_2$  both have access to a decryption oracle, then  $\mathcal{A}$  is running an *adaptive chosen ciphertext attack*. If  $\mathcal{A}$ 's advantage in winning the IND game is negligible, then we say  $S$  is IND-CCA2 secure.

It is immediately clear, that if the encryption algorithm is deterministic, then  $\mathcal{A}$  will win the game every time. This is because, we assume that everything about the cryptosystem, except the private key, is known to  $\mathcal{A}$ , thus  $A_2$  can simply encrypt  $m_0$  and  $m_1$ , and compare the results to  $C$ .

## 1.4 Digital signatures

Signature protocols are used in cryptography, to ensure the *authenticity* of a given message. This means that our goal is to be able to ensure who the sender of a message was. Someone wishing to break such a protocol would have to be able to generate a fake, valid signature for their message, thus impersonating another party.

**Definition 1.4.1.** (Signature scheme) Let  $K$ ,  $M$  and  $C$  be finite sets. A deterministic signature scheme is an algorithm triple  $(G, S, V)$ , where

- $G$  is the probabilistic key generation algorithm. Its only input is randomness. It outputs  $(k, k') \in K$ , the public and private keys.
- $S$  is the deterministic signature algorithm, which takes the private key  $k'$  and message  $m \in M$  as input, and outputs the signature  $s \in C$ .
- $V$  is the deterministic verification algorithm, which takes the public key  $k$ , message  $m$  and signature  $s$  as input and outputs "Accept" or "Reject".

We say such a system is *sound* if for all  $(k, k') \in K$ , and all  $m \in M$

$$V_k(m, S_{k'}(m)) = \text{Accept}$$

**Example 1.4.1.** We see the signature scheme is very similar to the public key cryptosystem, thus we can make a similar use of a trapdoor function  $f$  as in Example 1.3.1.

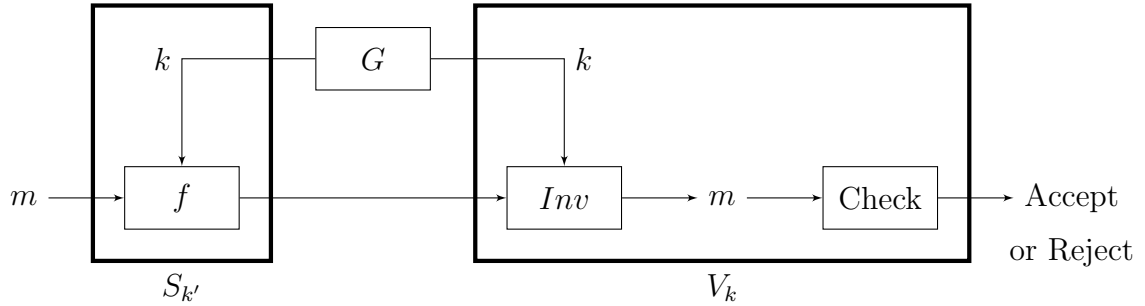


Figure 1.3: An outline of a signature scheme using a trap-door function.

Just as with cryptosystems, we can analogously define security of a signature scheme with respect to a game played by an attacker, known as forger in this context. A forger  $F$  is a probabilistic algorithm with a fixed initial state  $x_0$ . It takes some state  $x_i$  and a public key or signature as input, and outputs a message  $m$ , state  $x_{i+1}$  and either a request for a signature or a signature.

The forgery game we define as follows



## Forgery game

**Signer**

**Forger**

The signer uses the key generator  $G$  to get a random key pair  $(k, k')$

$$F(k, x_0) = (m_0, x_1, r_0)$$

..... For all rounds  $i \geq 0$  .....

$$\xleftarrow{(m_i, r_i)}$$

If  $r_i$  is a request, then  $s_i = S_{k'}(m_i)$

If  $r_i$  is a signature, the game stops.

$$\xrightarrow{s_i}$$

$$F(s_i, x_i) = (m_{i+1}, x_{i+1}, r_{i+1})$$

Round  $i + 1$  begins.

When the game stops, the signer checks if  $S_{k'}(m_i) = r_i$ , and if  $m_i \neq m_j$  for all  $j < i$ . If both are satisfied, the forger has won the game.

## 1.5 Two example cryptosystems

### 1.5.1 Diffie-Hellman key exchange protocol

The Diffie-Hellman key exchange protocol (DH), first published in 1976, marks the invention of public key cryptography. It is a protocol for two parties to generate a shared secret. Its security is based on the DLP of Example 1.2.1. An eavesdropper, Eve, cannot compute Alice and Bob's secret key  $K$  in an efficient manner, as they only have access to the information  $(p, g, A, B)$ , which implies that, to compute  $K$ , Eve would need to find one of the discrete logarithms  $dl(p, g, A)$  or  $dl(p, g, B)$ . Since Eve cannot compute these efficiently, we are content that Alice and Bob's secret is safe within some useful time frame, if we choose a large enough prime number  $p$ .

## Diffie-Hellman

**Alice**

**Bob**

Alice and Bob publicly choose a prime  $p$ , and  $g$  a generator of  $\mathbb{Z}_p^*$

choose a private integer  $a$

choose a private integer  $b$

$$A = g^a \mod p$$

$$B = g^b \mod p$$

$A$   
→

←  
 $B$

$$K = B^a = g^{a \cdot b} \mod p$$

$$K = A^b = g^{a \cdot b} \mod p$$

### 1.5.2 RSA

RSA is a public key encryption protocol. It is based on the RSA function of Example 1.2.2. The reason we use a trapdoor function for such a protocol is because now one of the communicating parties can use the function as a public key, and the trapdoor as a private key. The second party can input their message into the function, and feel confident there is no efficient way to compute their message knowing only the function output. Here  $G$  outputs keys in the space

## RSA

**Alice**

**Bob**

..... Set-up .....

choose primes  $p$  and  $q$

$$N = p \cdot q$$

choose  $e$  such that  $\gcd(e, \varphi(N)) = 1$

compute  $d$  such that  $e \cdot d = 1 \mod \varphi(N)$

..... Alice publishes the public key  $A = (N, e)$ ,  $(N, d)$  is the private key .....

Bob encrypts the message  $m$

$$M = m^e \mod N$$

$M$   
←

Alice decrypts the ciphertext  $M$

$$M^d = m^{e \cdot d} = m \mod N$$

$$K = \{((N, e), (N, d)) \mid N = p \cdot q \text{ where } p, q \text{ are primes and } e \cdot d = 1 \pmod{\varphi(N)}\}$$

And

$$E_{(N,e)} : \mathbb{Z}_N \rightarrow \mathbb{Z}_N \quad E_{(N,e)} : m \mapsto m^e \pmod{N} \quad (1.1)$$

$$D_{(N,d)} : \mathbb{Z}_N \rightarrow \mathbb{Z}_N \quad D_{(N,d)} : m \mapsto m^d \pmod{N} \quad (1.2)$$

This system is sound, since for all messages  $m \in \mathbb{Z}_N$

$$D_{(N,d)} \circ E_{(N,e)}(m) = m^{e \cdot d} = m^1 = m \pmod{N}$$

For this protocol the encryption function is  $RSA(N, e, m)$ . As we can see, if an eavesdropper is not privy to the private key, they would need to compute  $RSA^{-1}(N, e, M)$ . Thus Bob can be content that, if the primes  $p$  and  $q$  were chosen to be large enough, then the eavesdropper will not be able to compute the plaintext message within a useful time.

Note that, as defined here, RSA is a deterministic encryption protocol, thus it cannot be IND secure.



# Chapter 2

## Classical Elliptic Curve Cryptography

We now use the foundations presented in Chapter 1, to build protocols using the mathematics of elliptic curves. Such methods are known collectively as Elliptic Curve Cryptography (ECC).

### 2.1 The group of points on an Elliptic Curve

We define an elliptic curve over a field  $K$  as a smooth, algebraic plane projective curve of genus 1, with a distinguished  $K$ -rational point. This definition is discussed in detail in Appendix A, but for the purposes of our current discussion, it will suffice to note that all points but one on an elliptic curve over a field with characteristic not 2 or 3, satisfy the Weierstrass equation

$$Y^2 = X^3 + AX + B \quad (2.1)$$

where  $A, B \in K$ . The one point which cannot be derived as a solution to the equation above is called the *point at infinity*, and denoted here  $O$ . In this chapter we will speak exclusively of elliptic curves of this form for the sake of simplicity, however equivalent results hold for any elliptic curve.

**Definition 2.1.1.** (Addition of points on an EC) Let  $P_1 = (x_1, y_1)$ ,  $P_2 = (x_2, y_2)$  be finite<sup>1</sup>  $K$ -rational points on the elliptic curve  $E$  over the field  $K$ , satisfying (2.1). We define the addition of these as follows:

1. If  $x_1 = x_2$  and either  $y_1 \neq y_2$  or  $y_1 = y_2 = 0$ , then

$$P_1 + P_2 = O$$

2. Otherwise  $P_1 + P_2 = P_3 = (x_3, y_3)$  with

$$x_3 = m^2 - x_1 - x_2 \quad y_3 = m(x_1 - x_3) - y_1$$

where  $m = (y_2 - y_1)(x_2 - x_1)^{-1}$  if  $x_1 \neq x_2$ , and  $m = (3x_1^2 + A)(2y_1)^{-1}$  otherwise.

---

<sup>1</sup>Not the point at infinity.

Further, for any point  $P$  on  $E$ , we define

$$P + O = P$$

**Theorem 2.1.1.** *The set of points on an EC, along with addition as defined above, form an abelian group.*

*Proof.* By our definition, we see  $O$  is an identity element. Commutativity also follows directly from the definition.

The set of points being closed under addition, and, in particular, associativity are more involved proofs. The necessary background to read these can be found in Appendix A.  $\square$

For  $n \in \mathbb{N}_+$ , and  $P$  a point on an elliptic curve,  $nP$  denotes the point  $P$  added to itself  $n$  times.

**Definition 2.1.2.** (ECDLP) Let us define the set  $I$ :

$$I = \{(E(K), G) : E(K) \text{ is the set of } K\text{-rational points on an elliptic curve and } G \text{ is a generator of the group of points on the curve.}\}$$

Consider now the function  $\text{mult}(E(K), G, a) = aG \in E(K)$ , with  $(E(K), G) \in I$ . This is the elliptic curve point multiple function, and as defined it is bijective. Thus there is an inverse function  $\text{ecdl}(E(K), G, P) = (E(K), G, a)$  such that  $aG = P$ . This is the discrete logarithm function for elliptic curves.

It is believed that, not only is  $\text{mult}$  a tap-door function, but that further computing  $\text{ecdl}$  is at least as difficult as computing the discrete logarithm for integers.

The challenge of computing  $\text{ecdl}$  is known as the Elliptic Curve Discrete Logarithm Problem (ECDLP).

This problem, is the backbone of almost all elliptic curve based cryptosystems for classical computers.

## 2.2 ECDH key exchange protocol

ECDH uses, in principle, the same logic as the Diffie-Hellman protocol of section 1.5.1.

The domain parameters for this protocol make up the quintuple  $(K, E, q, h, G)$ , chosen as follows

- $K$  is a finite field.
- $E$  an elliptic curve defined over  $K$ .
- $G$  is a point of prime order  $q$ .
- $h$  is such that  $\#E(K) = h \cdot q$ , known as the cofactor.

## ECDH

**Alice**

**Bob**

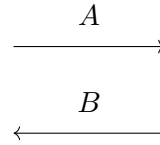
Alice and Bob publicly choose domain parameters  $(K, E, q, h, G)$

choose a private positive integer  $a$

choose a private positive integer  $b$

$$A = aG$$

$$B = bG$$



$$K = aB = a(bG)$$

$$K = bA = b(aG)$$

Notice that this protocol suffers from a key vulnerability: Alice or Bob have no way of knowing where the messages  $A$  and  $B$  came from, thus one could perform a man-in-the-middle attack.

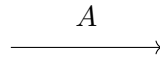
**Alice**

**Eve**

**Bob**

$$A = aG$$

$$B = bG$$



$$X = xG$$

$$K_a = axG \xleftarrow{X}$$

$$Y = yG$$



$$\xleftarrow{B} K_b = ayG$$

As we can see Eve has now agreed on keys  $K_a$  and  $K_b$  with Alice and Bob respectively, and can read all correspondence between them from this point on. Meanwhile Alice and Bob are none the wiser, as they are convinced their messages come directly from the other party.

To resolve this issue, Alice and Bob can sign all their messages, for example using the next protocol.

## 2.3 ECDSA

In this protocol, we assume Alice wants to sign their messages, so Bob can be certain these are indeed from Alice.

Alice must first choose some public domain parameters  $(K, E, q, h, G)$ , defined in the same way as in ECDH, and a hash function  $H$ . Then a message  $m$  can be signed, using the private key  $x \in \mathbb{Z}_+$ , as follows

#### ECDSA Signing

---

```

1: Choose a random number  $k \in \{1, \dots, q-1\}$ 
2:  $P \leftarrow kG$ 
3:  $r \leftarrow$  the first coordinate of  $P \pmod q$ 
4: if  $r = 0$  go to 1
5:  $e \leftarrow H(m)$ 
6:  $s \leftarrow (e + xr)k^{-1} \pmod q$ 
7: if  $s = 0$  go to 1
8: return  $(r, s)$ 

```

Bob can now check the signature of message  $m$ , using the public key  $T = xG$ .

#### ECDSA Verification

---

```

1: if  $r, s \notin \{1, \dots, q-1\}$  Reject
2:  $e \leftarrow H(m)$ 
3:  $u \leftarrow e \cdot s^{-1} \pmod q$ 
4:  $v \leftarrow r \cdot s^{-1} \pmod q$ 
5:  $P \leftarrow uG + vT$ 
6: if  $(r = \text{the first coordinate of } P \pmod q)$  Accept
7: else Reject

```

Returning to the man-in-the-middle attack for ECDH, Alice now signs the message  $A = aG$ , sending  $(A, (r, s))$  to Bob. Eve can no longer pretend to be Alice as that would require them to find a signature for the bogus message  $Y$ , or equivalently, winning a Forgery game against ECDSA. It has been proven that ECDSA is secure against an attacker playing the Forgery game[3], as long as it is using an arithmetically unbiased random number generator, and a hash function that is

- rarely zero
- zero resistant
- first and second preimage resistant and
- collision resistant

## 2.4 Benefits of ECC

ECC systems have a very clear advantage over other popular public key cryptosystems such as RSA: key size. Due to the difficulty of solving the ECDLP, ECC systems are equally secure as RSA instances using considerably shorter keys.



<b>Strength</b>	<b>Key size in bits</b>
Low	512
Medium	1024
High	2048
Very high	4096

Table 2.1: Key length to relative security.

<b>RSA key size in bits</b>	<b>ECC key size in bits</b>
1024	192
2048	224
3072	256
7680	384
15360	521

Table 2.2: Comparison of key security for RSA and ECC protocols such as ECIES.

At the time of writing, the recommended key sizes for RSA[10] are seen in Table 2.1. The equivalent necessary key length for ECC protocols are in Table 2.2.

As we can see, with an ECC key the size of a low-security RSA key, we can guarantee security equivalent to RSA with key almost 30 times larger. This feature makes ECC both less memory-intensive and faster. However, this does not take away from the complexity of implementing ECC protocols, brought by the comparatively more complex mathematics underlying ECC.



# Chapter 3

## Quantum machines

As mentioned in the introduction of this thesis, to this day there is no certainty in the future of quantum machines. Some say it is but a bubble, fated to burst any day now, others speculate there is real potential in quantum computers. What is clear is that there is no consensus. Regardless, the very philosophy of cryptography and cybersecurity at large is to be prepared for what is and for what might be. Thus here we take a tentative look at the future and ask, what might be if quantum computers become a viable addition to our computation methods?

We begin by presenting the core concepts underlying quantum computation, to see how it is different from the classical version we are used to, and how it is, in some instances, more powerful. To do this we return to our models of computation from Chapter 1, and expand on them. Later we present Shor’s factoring algorithm, a quantum algorithm with the potential to break many modern public key cryptosystems.

### 3.1 Quantum computation

In Chapter 1, we considered models of computation and computers in a purely theoretical manner, however, in order to understand the quantum computer, we must move away from the purely mathematical, and instead envision the computer as a physical system.

First let us review the classical model under this new lens. Every physical system has states it can take, observables and a time evolution. There are three places where we can observe a “state” in the Turing machine. There is the current internal state  $q \in \mathbf{Q}$ , the current symbols in the tape, and the current position of the the head on the tape, as seen in Figure 3.1.

We can thus say that the state of the system of the Turing machine is given by all these elements together. We can further say that, at every time interval a time evolution occurs, given by the transition function.

Using this informal idea as inspiration, we can now begin to describe the quantum Turing machine, using the model presented by Deutsch and Penrose in [8]. It is recommended the reader unfamiliar with quantum mechanics now read Appendix B before continuing.

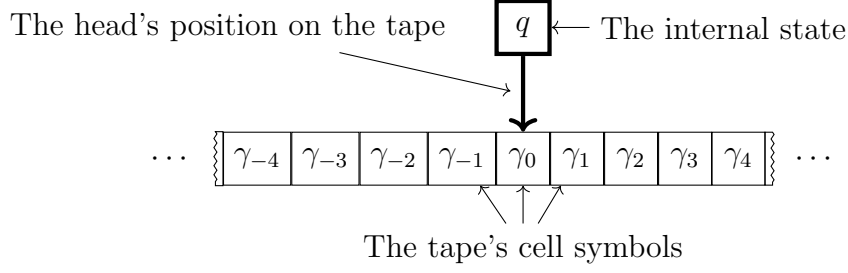


Figure 3.1: The states of the Turing machine.

**Definition 3.1.1.** (Quantum Turing machine) A quantum Turing machine  $\mathbf{Q}$ , running in discrete steps of duration  $T$ , consists of

- $N$  2-state observables  $\{\hat{q}_i\}_{i \in \mathbb{Z}_+}$ , the processor
- An infinite sequence of 2-state observables  $\{\hat{\gamma}_i\}_{i \in \mathbb{Z}}$ , forming the memory
- An observable  $\hat{x}$ , whose spectrum is  $\mathbb{Z}$ , which indicates the position of the head

All possible states of  $\mathbf{Q}$  are given by the space  $\mathcal{H}$ , spanned by the simultaneous eigenvectors of all the above observables,  $|x; \mathbf{q}; \boldsymbol{\gamma}\rangle$ .

The time evolution of  $\mathbf{Q}$  is given by a constant unitary operator  $U$  on  $\mathcal{H}$ , which gives us the state after  $n$  steps  $|\psi_{nT}\rangle$  by

$$|\psi_{nT}\rangle = U^n |\psi_0\rangle$$

where  $|\psi_0\rangle$  is the state of  $\mathbf{Q}$  at time 0, when computation begins, and satisfies

$$|\psi_0\rangle = \sum_m \lambda_m |0; 0; \boldsymbol{\gamma}\rangle \quad (3.1)$$

$$\sum_m |\lambda_m|^2 = 1 \quad (3.2)$$

where only a finite number of  $\lambda_m$  are non-zero.

We reserve the first observable in the processor  $\hat{q}_0$  to act as a flag for termination. This observable begins in state  $|0\rangle$ , and is set to state  $|1\rangle$  upon completion of the computation. This way we may observe this value throughout the computation, without affecting the rest, and observe the entire system only when  $\hat{q}_0$  is in state  $|1\rangle$ .

Having this model, we can quite naturally derive complexity classes analogous to the classical case. The quantum counterpart of  $\mathbf{P}$  we denote  $\mathbf{EQP}$ , and that of  $\mathbf{BPP}$  we denote  $\mathbf{BQP}$ .

Note that, while the quantum Turing machine is very useful for defining complexity classes, it is not the generally used model for describing quantum algorithms, this one is most often the quantum circuit, which we will introduce next.

### 3.1.1 Quantum circuits

The quantum circuit is, at least in theoretical terms, like any classical circuit. It is composed of *edges*, upon which a single unit of information flows, the qubit in our case, and *nodes*, or *gates*, that perform transformations on a small number of qubits. However, where in the classical case the edges and nodes of this graph have a clear physical counterpart (wires and physical logic gates), they do not in the quantum case. In fact, in the case of the quantum circuit, it could be that its physical implementation has no flow of qubits at all, as they may be implemented via static particles. However, the idea of seeing quantum algorithms as a sequence of simple operations remains useful for a more natural description of the processes, and easier analysis.

The usual basic gates used in a quantum circuit are given in Table 3.1. Note that, while most of these act on one or two qubits at a time, they may be used in a system with multiple qubits. In these cases when we say gate  $G$  is applied to the  $i^{\text{th}}$  qubit, we mean the operator  $I \otimes \dots \otimes I \otimes G \otimes I \otimes \dots \otimes I$  is applied to the entire system, where  $I$  is the identity operator on a single qubit.








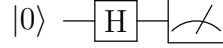
Name	Symbol	Operator definition
Pauli-X		$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y		$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z		$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard		$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Controlled not (CNOT)		$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Swap		$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Measurement		A measurement of a single qubit, after this the qubit will be in one of the basis states.

Table 3.1: Commonly used quantum gates and their definition.

**Theorem 3.1.1.** *All quantum circuits can be constructed using only unary and CNOT gates [9].*

**Example 3.1.1.** *The following circuit generates perfectly random bits.*

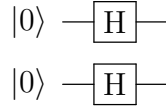


*The internal state transitions are as follows*

$$|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \mapsto |0\rangle \text{ or } |1\rangle \text{ each with probability } \frac{1}{2}$$

*This circuit has no classical analogue, and would allow for a perfect implementation of the symmetric encryption protocol One Time Pad.*

**Example 3.1.2.** *Consider applying the Hadamard gate to all qubits of an 2-qubit system*



*This is the Hadamard-Walsh gate, denoted  $W_2$ , for two qubits, and can be defined simply as  $W_2 = H \otimes H$ . This construction can be generalised to any number of qubits, and will become very useful for quantum parallelism.*

## 3.2 Quantum algorithms

In order to understand many quantum algorithms, and in particular the algorithms we will present in this text, we must first familiarize ourselves with two important sub-procedures: quantum parallelism and the quantum Fourier transform (QTF).

### 3.2.1 Quantum parallelism

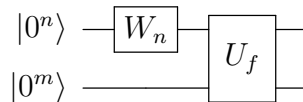
Despite the name, quantum parallelism does not in fact give us many different results to work with as classical parallelism would. Rather it aims to create a superposition of all input, output pairs for a function  $f$ . While this can be used as a single state within quantum computations, the moment we observe it it will collapse into a single input, output pair. Thus we have in practice computed only one observable output.

Let  $f : \mathbb{Z}_N \rightarrow \mathbb{Z}_M$  be a function, that acts on the  $N = 2^n$  numbers encoded by an  $n$ -qubit state, the *input register*, and returns numbers between 0 and  $M = 2^m$ , encoded by an  $m$ -qubit state, the *output register*. The transformation

$$U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$$

is then a linear, unitary operator, and for  $y = 0 \dots 0 = 0^m$  it maps the superposition  $\sum_x a_x |x, 0\rangle$  to the state  $\sum_x a_x |x, f(x)\rangle$ .

Consider now the following circuit



It describes applying the transformation  $U_f$  to the superposition of values obtained by applying the Hadamard-Walsh operator to the first  $n$  qubits, all initialised to  $|0\rangle$ . It gives us the final state

$$\frac{1}{\sqrt{N}} \sum_x |x\rangle \otimes |f(x)\rangle$$

This superposition of input, output pairs is the result of quantum parallelism for the function  $f$ .

### 3.2.2 The quantum Fourier transform

The quantum Fourier transform (QFT) is a variant of the discrete Fourier transform (DFT). We review the latter, more familiar transform first. Let  $f : \mathbb{Z}_N \rightarrow \mathbb{C}$  be a function, the DFT, denoted  $\mathcal{F}$ , is a transformation of such functions that produces the function  $F : \mathbb{Z}_N \rightarrow \mathbb{C}$  defined as

$$F(x) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} f(k) \exp\left(2\pi i \frac{kx}{N}\right)$$

The QFT works on the amplitudes of a state  $\sum_x a_x |x\rangle$ , by modelling them as a function of the basis states  $a_\bullet : \mathbb{Z}_N \rightarrow \mathbb{C}$ . Thus it transforms the states by

$$\sum_x a_x |x\rangle \mapsto \sum_x A_x |x\rangle$$

Where  $A_\bullet = \mathcal{F}(a_\bullet)$ <sup>1</sup>.

### 3.2.3 Shor's factoring algorithm

We have now all the necessary pieces to present Shor's factoring algorithm. Given an integer  $N$  to factorise, this algorithm does so in two parts. First it finds the order  $r$  of a random number  $a$  modulo  $N$ , using a quantum subroutine. Then, assuming the order is not odd or  $a^{\frac{r}{2}} \equiv -1 \pmod{N}$ , we use the fact that

$$(a^{\frac{r}{2}} + 1)(a^{\frac{r}{2}} - 1) \equiv 0 \pmod{N}$$

to derive a non-trivial factor of  $N$ : either  $\gcd(a^{\frac{r}{2}} + 1, N)$  or  $\gcd(a^{\frac{r}{2}} - 1, N)$ .

---

<sup>1</sup>For an efficient implementation of this as a quantum algorithm see [15]

Shor's factoring algorithm for  $N \in \mathbb{Z}$

---

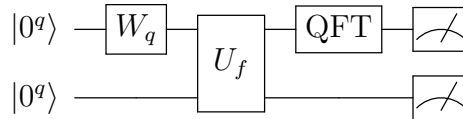
```

1 : Pick a random number  $1 < a < N$ 
2 : Compute  $n = \gcd(a, N)$  using the Euclidean algorithm.
3 : if  $n \neq 1$  then
4 :   return  $n$ 
5 :  $|y\rangle \leftarrow$  Quantum core
    -----
    1 : Choose  $Q = 2^q$  such that  $2N^2 \leq Q \leq 4N^2$ 
    2 : Set  $q$  input registers to  $|0\rangle$ 
        // These store qubit representations of the values in  $\mathbb{Z}_Q$ 
    3 : Set  $q$  output registers to  $|0\rangle$ 
    ..... End of setup .....
    4 : Pass the input register through a Hadamard-Walsh gate
    5 : Use quantum parallelism, for the function  $f(x) = a^x \mod N$ 
        // The machine is now in state  $\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x, f(x)\rangle$ 
    6 : Pass the input register through the QFT
    7 : Measure the machine's state to get the pure state  $|y, z\rangle$ 
    8 : return  $|y\rangle$ 
    -----
6 : Using extended fractions find approximations  $\frac{d}{r}$  of  $\frac{y}{Q}$  such that
    
$$s < N \text{ and } \left| \frac{y}{Q} - \frac{d}{s} \right| < \frac{1}{2Q}$$

7 : if  $a^r \not\equiv 1 \mod N$ 
8 :   Try multiples of  $r$ . If one of these  $kr$  works, then  $r \leftarrow kr$ 
9 :   If no multiple of  $r$  works, go back to step 5
10 : if  $r$  is odd or  $a^{\frac{r}{2}} \equiv -1 \mod N$ 
11 :   Go back to step 1
12 : Either  $\gcd(a^{\frac{r}{2}} + 1, N)$  or  $\gcd(a^{\frac{r}{2}} - 1, N)$  is a non-trivial factor  $m$  of  $N$ 
13 : return  $m$ 

```

The circuit for the quantum core is as follows



This algorithm computes non-trivial factors of integers in polynomial time with high probability, thus placing integer factorisation in the complexity class **BQP**.

If this algorithm could be implemented in a quantum computer with a large number of qubits at its disposal, then there would be an efficient way to compute the inverse of the *RSA* function of Example 1.2.2, thus breaking the *RSA* cryptosystem. There is also a version of this algorithm that efficiently computes discrete logarithms,



and thus breaks the Diffie-Hellman key exchange, the El Gamal cryptosystem, and elliptic curve methods such as ECDSA.

To put it simply, if a large quantum computer were to be developed, it would effectively spell the end of public key cryptography as we know it today.



# Chapter 4

## SIDH: a broken protocol

In Chapter 2, we saw how the hardness of the ECDLP is used in classical ECC protocols. Here we return to the theory of elliptic curves, to explore the supersingular isogeny problem, conjectured to be unlikely to be possible to solve efficiently on either classical or quantum computers. We present here one of the simpler protocols based on this problem, SIDH, and a recent attack on it.

### 4.1 Supersingular elliptic curves

Let  $K$  be a field, and  $F$  any field extension of  $K$ . An isogeny<sup>1</sup>  $\phi : E_1(F) \rightarrow E_2(F)$  of elliptic curves  $E_1/K$  and  $E_2/K$  is a rational map defined over  $F$ , satisfying  $\phi(O) = O$ . We will denote isogenies of degree  $d$  as  $d$ -isogenies.

As isogenies between curves  $E_1, E_2$  are also homomorphism of abelian groups, they once again form an abelian group. Further the endomorphisms defined over  $F$  of an elliptic curve form a ring denoted

$$\text{End}_F(E) = \{\phi : E \rightarrow E \mid \phi \text{ is an isogeny}\}$$

For an elliptic curve  $E/\mathbb{F}_p$ , we will denote the ring of isogenies  $E \rightarrow E$ , with coefficients also only in  $\mathbb{F}_p$  by  $\text{End}_p(E)$ .

Quite counter-intuitively, it can be shown that the size of the endomorphism ring of an elliptic curve  $E/K$ , where  $K$  is finite, is quite closely connected to the subgroup of torsion points  $E[p]$ , where  $p$  is the characteristic of  $K$ . This relationship naturally gives rise to two types of elliptic curve.

**Definition 4.1.1.** (Quaternion Algebra) A quaternion algebra over  $\mathbb{Q}$ , is an algebra of the form

$$\mathcal{Q} = \{a + b\alpha + c\beta + d\alpha\beta \mid a, b, c, d \in \mathbb{Q}\}$$

where  $\alpha^2, \beta^2 \in \mathbb{Q}$ ,  $\alpha^2 < 0$ ,  $\beta^2 < 0$  and  $\alpha\beta = -\beta\alpha$ .

**Theorem 4.1.1.** *Let  $E$  be an elliptic curve over a finite field  $K$  of prime characteristic  $p$ . Then either*

---

<sup>1</sup>An exact definition of isogeny, as well as a more complete discussion of the properties of isogenies can be found in Section A.4 of Appendix A

1.  $E[p^r] \simeq \mathbb{Z}/p^r\mathbb{Z}$  for all  $r \geq 1$ , and, if  $j(E) \in \bar{\mathbb{F}}_p$ , then  $\text{End}(E)$  is an order in an imaginary quadratic field;
2. or the following equivalent statements hold:
  - (a)  $E[p^r] = \{0\}$  for some  $r \geq 1$
  - (b)  $\text{End}(E)$  (over the closure  $\bar{K}$ ) is an order in a quaternion algebra (and is thus non-commutative).
  - (c) The trace of the Frobenius endomorphism  $\phi_{p^r}$  for some  $r \geq 1$  is congruent to 0 mod  $p$ . For  $p \geq 5$  this statement is equivalent to the trace of Frobenius being equal to 0.

*Proof.* Parts 1, 2(a), 2(b): [17, Theorem V.3.1]. Equivalence of 2(c): [18, Proposition III.8.6] and [18, Proposition V.2.3]  $\square$

**Definition 4.1.2.** (Ordinary and supersingular ECs) We say an elliptic curve  $E$  that satisfies statement 1 of Theorem 4.1.1 is ordinary. If it satisfies one of the equivalent statements in 2 then we say it is supersingular<sup>2</sup>.

Supersingular curves have the largest possible endomorphism rings, and cannot be isogenous to an ordinary curve [17]. It is important to note that in Theorem 4.1.1. we consider specifically those isogenies defined over  $\bar{K}$ , meaning that the coefficients appearing in  $\phi$  need not be elements of  $K$ . An example of such an isogeny is given in Example A.4.2, where the constant  $\sqrt{-1}$  need not be an element of  $K$ . For the purposes of our application of supersingular elliptic curves we need to make this distinction. It turns out that when the curve  $E$  is defined over a prime field, and we also restrict ourselves to isogenies with coefficients in the prime field, then the resulting ring of isogenies,  $\text{End}_p(E)$ , is never an order in a quaternion algebra. In this case, as seen in [4], we have that  $\text{End}_p(E)$  is an order in an imaginary quadratic field given by:

$$\text{End}_p(E) \otimes_{\mathbb{Z}} \mathbb{Q} \cong \mathbb{Q}(\sqrt{t^2 - 4p}) \quad (4.1)$$

where  $t$  is the trace of the Frobenius endomorphism, and  $p$  is the prime order of the field. A proof of this can be found in [20, Lecture 13 Theorem 13.18]<sup>3</sup>.

Supersingular curves are avoided for use in the ECC protocols of Chapter 2. This is because the ECDLP for these curves is equivalent to the DLP for integers, thus weakening security, if our assumption that the ECDLP is harder than the DLP is indeed correct. However, a different, and potentially more complex problem arises when considering these curves.

**Conjecture 4.1.1.** The *supersingular isogeny problem*, defined by:

“Let  $E_1$  and  $E_2$  be two isogenous supersingular elliptic curves. Compute an isogeny between them.”

cannot be solved efficiently by either a classical or a quantum computer.

<sup>2</sup>Supersingular elliptic curves should not be confused with singular curves. All supersingular elliptic curves are non-singular by the definition of an elliptic curve.

<sup>3</sup>Note that this source uses slightly different notation to what is seen here. An explanation of the notation is found in the same source, Lecture 12 Definition 12.2.

This conjecture can be restated in terms of graph theory. We can define a graph whose nodes are elliptic curves up to isomorphism over some field  $K^4$ , and whose edges are the isogenies between them. Such a graph is known as an *isogeny graph*. Clearly this construction will form two disconnected subgraphs, one of ordinary curves and the other of supersingular curves.

The conjecture can hence be equivalently thought as the problem of finding a walk between  $E$  and  $E'$  in the isogeny graph.

We may further restrict the construction of the isogeny graph to consider only isogenies of degree  $d$  between elliptic curves  $E/\mathbb{F}_q$ . This is known as a  $d$ -isogeny graph. It can be shown these graphs are  $(d+1)$ -regular Ramanujan graphs, meaning in our context that these graphs have the following rapid mixing result.

**Theorem 4.1.2.** *Let  $G$  be a  $d$ -isogeny graph with  $h$  vertices,  $S$  any subset of the vertices in  $G$ , and  $e$  any vertex (elliptic curve) in  $G$ . Then a random walk starting at  $e$  of length at least  $\frac{\log 2h/|S|^{1/2}}{\log(d+1)/2\sqrt{d}}$  will end in  $S$  with probability at least  $\frac{|S|}{2h}$*

*Proof.* [12, Lemma 2.1] □

This property, in intuitive terms, implies that as we take a random walk in the isogeny graph, evidence of where the walk began vanishes quickly. Or equivalently, as we compute compositions of random isogenies, evidence of the starting elliptic curve quickly vanishes.

**Theorem 4.1.3.** *Let  $E$  be an elliptic curve and  $\Phi \subseteq E$  a finite subgroup of  $E$ . Then there is a unique elliptic curve  $E'$  and a separable isogeny  $\phi : E \rightarrow E'$  such that*

$$\ker \phi = \Phi$$

*The elliptic curve  $E'$  is usually also denoted  $E/\Phi$ .*

*Proof.* [17, Proposition III.4.12] □

A sub-exponential time quantum algorithm for computing the isogeny of Theorem 4.1.3 are given in [6].

## 4.2 Supersingular Isogeny Diffie Hellman

We now present the Supersingular Isogeny Diffie-Hellman (SIDH) protocol by De Feo and Jao. This was believed to be a very promising key exchange protocol until recently, and had advanced to the fourth round of NIST's Post-Quantum Cryptography standardisation process as the instantiation Supersingular Isogeny Key Exchange (SIKE), before the attack we show in the next section was discovered.

---

<sup>4</sup>These are usually represented by their  $j$ -invariants.

### 4.2.1 Set-up

We begin by choosing the finite field  $\mathbb{F}_q = \mathbb{F}_{p^2}$ , where  $p$  is a prime number of the form  $p = a^{e_A} \cdot b^{e_B} \cdot f \pm 1$ , where  $a$  and  $b$  are small primes.

We fix the following public parameters:

- A supersingular curve  $E_0$ , defined over  $\mathbb{F}_{p^2}$
- The pair  $\{P_A, Q_A\}$ , generating  $E_0[a^{e_A}]$
- The pair  $\{P_B, Q_B\}$ , generating  $E_0[b^{e_B}]$

Together, our domain parameters, to be agreed before the start of the protocol, form the sextuple  $(\mathbb{F}_q, E_0, P_A, Q_A, P_B, Q_B)$ .

### 4.2.2 Key exchange protocol

SIDH

**Alice**

**Bob**

..... Let  $(\mathbb{F}_q, E_0, P_A, Q_A, P_B, Q_B)$  be the domain parameters.....

Choose random integers  $m_A, n_A$

Choose random integers  $m_B, n_B$

Compute the isogeny  $\phi_A$

Compute the isogeny  $\phi_B$

$\phi_A : E_0 \rightarrow E_A$

$\phi_B : E_0 \rightarrow E_B$

$\ker \phi_A = \langle m_A P_A + n_A Q_A \rangle$

$\ker \phi_B = \langle m_B P_B + n_B Q_B \rangle$

$\xrightarrow{(E_A, \phi_A(P_B), \phi_A(Q_B))}$

$\xleftarrow{(E_B, \phi_B(P_A) + \phi_B(Q_A))}$

Compute the isogeny  $\phi'_A$

Compute the isogeny  $\phi'_B$

$\phi'_A : E_B \rightarrow E_{AB}$

$\phi'_B : E_A \rightarrow E_{BA}$

$\ker \phi'_A = \langle m_A \phi_B(P_A) + n_A \phi_B(Q_A) \rangle$

$\ker \phi'_B = \langle m_B \phi_A(P_B) + n_B \phi_A(Q_B) \rangle$

.... Shared key:  $j(E_{AB}) = j(E_{BA}) = j(E_0 / \langle m_A P_A + n_A Q_A, m_B P_B + n_B Q_B \rangle)$ ....

Note, the above protocol relies on the commutativity of the following diagram.

## 4.3 The Castryck-Decru attack

In August 2022 Wouter Castryck and Thomas Decru published a devastating attack to the SIDH protocol shown in Section 4.2, that showed SIDH is not only not

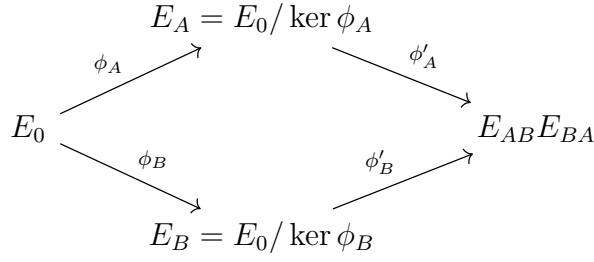


Figure 4.1: SIDH commutative diagram.

quantum-resistant, but breakable by a single core classical computer in heuristically polynomial time [5]. Here we present an overview of this attack. It is recommended that the reader unfamiliar with the foundational properties of elliptic curve isogenies, particularly the dual isogeny, read Section A.4 of Appendix A before continuing with this section.

### 4.3.1 Preliminaries

The attack uses the torsion points sent over the insecure channel during the key exchange to retrieve Bob’s secret isogeny  $\phi_B$ . Intuitively speaking, the secret isogeny is retrieved one piece at a time, checking at each stage that the correct piece has been found.

**Theorem 4.3.1.** *Let  $E/K$  be an elliptic curve and  $n$  a positive integer, not divisible by  $\text{char}(K)$ . Then there is a pairing, called the  $e_n$ -Weil pairing*

$$e_n : E[n] \times E[n] \rightarrow \mu_n$$

where  $\mu_n$  are the  $n^{\text{th}}$  roots of unit, that satisfies the properties

1.  $e_n$  is bilinear.
2.  $e_n$  is non-degenerate in both variables.
3.  $e_n(P, P) = 1$  for all  $P \in E[n]$ .
4.  $e_n(P, Q) = e_n(Q, P)^{-1}$  for all  $P, Q \in E[n]$ .

*Proof.* [23, Section 11.2] □

**Definition 4.3.1.** (Anti-isometry) An anti-isometry with respect to a bilinear pairing of groups  $f : A \times A \rightarrow G$ , is a mapping  $\psi : A \rightarrow A$  such that

$$f(\psi(P), \psi(Q)) = f(P, Q)^{-1}$$

For our purposes, we will consider exclusively anti-isometries with respect to the  $e_n$ -Weil pairings, thus will refer to these simply as “anti-isometries”.

**Definition 4.3.2.** (Isogeny factorisation configuration) Let  $E_1$  and  $E_2$  be two elliptic curves defined over  $K$  and  $N \geq 2$  an integer. An isogeny factorisation configuration of order  $N$  from  $E_1$  to  $E_2$  is a triplet  $(f, H_1, H_2)$  where

- $f : E_1 \rightarrow E_2$  is an isogeny
- $H_1, H_2 \leq \ker f$  are two subgroup schemes with  $\#H_1 + \#H_2 = N$  and  $\#H_1 \cdot \#H_2 = \deg f$

If further  $H_1 \cap H_2 = \{0\}$  then we say  $(f, H_1, H_2)$  is an *isogeny diamond configuration*.

**Theorem 4.3.2.** (*Kani's reducibility criterion*) Let  $\mathbf{f} = (f, H_1, H_2)$  be an isogeny diamond configuration of order  $N$  from  $E_1$  to  $E_2$ . We set  $n = N/d$  and  $k_i = \#H_i/d$  for  $i = 1, 2$ , where  $d = \gcd(\#H_1, \#H_2)$ . Then  $f$  factors uniquely over  $[d]$ , that is, for some isogeny  $f'$ ,  $f = f' \circ [d]$  and there is a unique reducible anti-isometry  $\psi : E_1[N] \rightarrow E_2[N]$  such that

$$\psi(k_1 P_1 + k_2 P_2) = f'(P_2 - P_1) \quad \forall P_i \in [n]^{-1}(H_i)$$

Further, every reducible anti-isometry is of this form

*Proof.* [13, Theorem 2.6] □

**Example 4.3.1.** Consider the elliptic curve  $E_0/\mathbb{F}_{p^2}$ , where  $p$  is a prime of the form  $a^{e_A} \cdot b^{e_B} \cdot f \pm 1$ , with  $a$  and  $b$  primes and  $a^{e_A} > b^{e_B}$ . Assume the points  $P_A, Q_A$  generate  $E[a^{e_A}]$ . Let  $\phi : E_0 \rightarrow E_B$  be a  $b^{e_B}$ -isogeny, and  $\gamma : E_0 \rightarrow C$  a  $c$ -isogeny, where  $c = a^{e_A} - b^{e_B}$ .

Now, consider the following isogeny

$$\psi = [-1] \circ \phi \circ \hat{\gamma} : C \rightarrow E_B$$

Here  $\ker \psi$  is a cyclic group of order  $c \cdot b^{e_B}$ , thus having two unique cyclic subgroups  $H_1$  and  $H_2$  of respective orders  $c$  and  $b^{e_B}$ , with  $H_1 \cap H_2 = \{0\}$ .

We see that

$$\#H_1 + \#H_2 = a^{e_A} \quad \text{and} \quad \#H_1 \cdot \#H_2 = \deg \psi$$

Thus, clearly,  $(\psi, H_1, H_2)$  is an isogeny diamond configuration.

In this attack, to check we have found correct pieces of our secret isogeny  $\phi_B$  we must compute a chain of  $(a, a)$ -isogenies<sup>5</sup>. By using auxiliary  $c$ -isogenies, as in Example 4.3.1, Kani's theorem above ensures that the last step in the chain splits in the case of a correct guess.

### 4.3.2 The attack

Now we are ready to put all the pieces together, and show how a single core computer can break SIDH.

Our attacker, Eve, has access to the following information:

- The public domain parameters  $(\mathbb{F}_{p^2}, E_0, P_A, Q_A, P_B, Q_B)$ , as defined above.
- The triplet  $(E_A, \phi_A(P_B), \phi_A(Q_B))$ .

---

<sup>5</sup>These are isogenies on products of two elliptic curves  $\alpha \times \beta : E_1 \times E_2 \rightarrow E_3 \times E_4$ , where  $\deg \alpha = \deg \beta = a$ .



- The triplet  $(E_B, \phi_B(P_A), \phi_B(Q_A))$ .

We further assume Eve has a  $b^\beta$ -isogeny  $\tau : E_0 \rightarrow E_{\text{start}}$  for some  $\beta \geq 0$ . Here  $E_{\text{start}}$  is chosen to be either  $y^2 = x^3 + x$  or  $y^2 = x^3 + 6x^2 + x$ , which are the two commonly chosen base curves in SIDH.

Now Eve begins the attack by first performing set-up steps, which compute the isogeny  $\gamma_{\text{start}}$  that will be used in every iterative step later. This computation relies on the fact that both  $E_{\text{start}}$  curves have an endomorphism  $2i$  satisfying  $(2i)^2 = [-4]$ . For the curve  $E_{\text{start}} : y^2 = x^3 + x$  this endomorphism is the composition of  $[2]$  with  $i : (x, y) \mapsto (-x, \sqrt{-1}y)$ . For  $E_{\text{start}} : y^2 = x^3 + 6x^2 + x$  the isogeny  $2i$  is formed by composing the 2-isogeny  $\varphi$  to  $x^3 + x$  with the  $i$  isogeny above and with the dual  $\hat{\varphi}$ .

In Castryck and Decru's prepublication, this attack is presented for the specific case that the small primes  $a$  and  $b$  are set to  $a = 2$  and  $b = 3$ . First we present this case.

#### Setup

---

- 1 : Factorise  $c = 2^{e_A} - 3^{e_B}$
- 2 : **if** all factors of  $c$  are congruent to 1 mod 4 **then**
- 3 : Find integers  $u, v$  such that  $c = u^2 + 4v^2 = (u + 2iv)(u - 2iv)$
- 4 :  $\gamma_{\text{start}} \leftarrow [u] + [v] \circ 2i$

Once set-up is complete, Eve runs the following iterative algorithm

#### Iteration

---

- // Here we assume  $E_0 = E_{\text{start}}$ , but all steps are easily altered by appropriately composing with  $\tau$
- 1 : Choose  $\beta_1 \geq 1$  minimal such that there is an  $\alpha_1 \geq 0$  for which  
 $c_1 = 2^{e_A - \alpha_1} - 3^{e_B - \beta_1} > 0$   
 And  $c_1$  has only prime factors congruent to 1 mod 4.
  - 2 : Write  $\phi_B = \phi_1 \circ \kappa_1$ , with  $\kappa_1$  a  $\beta_1$ -isogeny.
  - 3 :  $E_1 \leftarrow \kappa_1(E_0)$ ,  $P_1 \leftarrow \kappa_1(a^{\alpha_1} P_A)$ ,  $Q_1 \leftarrow \kappa_1(a^{\alpha_1} Q_A)$
  - 4 : Let  $\tilde{\kappa}_1 : E_{\text{start}} \rightarrow C_1$  be the isogeny with kernel  $\gamma_{\text{start}}(\ker \kappa_1)$ .
  - 5 :  $\gamma_1 \leftarrow \frac{\tilde{\kappa}_1 \circ \gamma_{\text{start}} \circ \hat{\kappa}_1}{3^{\beta_1}}$
  - 6 :  $P_1 \leftarrow \gamma_1(P_A)$ ,  $Q_1 \leftarrow \gamma_1(Q_A)$
  - 7 : Check if  $C_1 \times E_B / \langle (P_{c_1}, a^{\alpha_1} P_A), (Q_{c_1}, a^{\alpha_1} Q_A) \rangle$  is a product of elliptic curves.
  - 8 : **if no then** try another  $\kappa_1$
  - 9 : **else**
  - 10 : Choose  $\beta_2 > \beta_1$  minimal such that there is an  $\alpha_2 \geq 0$  for which  
 $c_2 = 2^{e_A - \alpha_2} - 3^{e_B - \beta_2} > 0$   
 And  $c_2$  has only prime factors congruent to 1 mod 4.
  - 11 : Write  $\phi_1 = \phi_2 \circ \kappa_2$ , with  $\kappa_2$  a  $\beta_2$ -isogeny.
  - 12 : Continue analogously to steps 3 to 12.
  - 13 : **return**  $\phi_B = \kappa_r \circ \dots \circ \kappa_3 \circ \kappa_2 \circ \kappa_1$

### Some considerations on the generalisation of the Castryck-Decru attack

When we attempt to generalise the above attack, for any two small primes  $a$  and  $b$ , we now have to consider the following question:

**Question 4.3.1.** When can the integer  $c = a^{e_A} - b^{e_B}$  be written in the form

$$u^2 + 4v^2 \tag{4.2}$$

where  $u$  and  $v$  are integers?

This question is answered in [7] for the case where  $a = 2$  and  $b$  is odd. In this case alone  $c$  is odd, and Corollary 2.6 in [7] gives us that it can be written in the form 4.2 if and only if all prime factors of  $c$  are congruent to 1 mod 4.

In cases where  $a = b = 2$  then we have that

$$\begin{aligned} c &= 2^{e_A} - 2^{e_B} \\ &= 2^{e_B}(2^{e_A - e_B} - 1) \end{aligned}$$

The term  $2^{e_B}$  is either a square or can be written in the form  $2(2^{e_B - 1}) = 2^{e_B - 1} + 2^{e_B - 1}$ , where  $2^{e_B - 1}$  is a square and also a multiple of 4. Thus, using Lemma 2.1 from [7], we have that  $c$  can be written in the form 4.2 if  $(2^{e_A - e_B} - 1)$  has only prime factors congruent to 1 mod 4.

Other cases seem to be not as well studied. However, we note that in the case where  $E_{\text{start}} : y^2 = x^3 + x$ , then we can compute a representation of  $c$  of the form  $u^2 + v^2 = (u + vi)(u - vi)$ , which is possible as long as  $c$  has no prime factor congruent to 3 mod 4 with odd exponent. Then we may set  $\gamma_{\text{start}} = [u] + [v] \circ i$ .

This latter form has the benefit of being more general, and thus more values of  $c$  can be written in this form, and thus we have more of these simple  $\gamma_{\text{start}}$  at our disposal.

It is to be noted that this still leaves many values of  $c$ , where this trick is not possible. In these cases, finding a  $\gamma_{\text{start}}$  isogeny with the required properties of some other form becomes decidedly non-trivial.

# Chapter 5

## CSIDH

We have now seen an example of a simple isogeny-based protocol, and how these can be liable to attack, when too much information is given about the isogenies used. In this chapter we present an, as of yet, unbroken key exchange protocol, using the same foundations as SIDH, and only slightly more complex mechanics. This is CSIDH<sup>1</sup>, first proposed by Castryck et al. in [4].

### 5.1 The class-group action

We know from Section 4.1, that supersingular elliptic curves  $E/\mathbb{F}_p$ , where  $p$  is a prime number, are orders in an imaginary quadratic field of the form  $\mathbb{Q}(\sqrt{t^2 - 4p})$ . We consider here ideals of such orders, and their action on elliptic curves.

**Definition 5.1.1.** (Norm) Let  $K$  be a quadratic number field and  $\mathcal{O} \subseteq K$  an order. The order of an  $\mathcal{O}$ -ideal  $\mathfrak{a}$  is given by

$$N(\mathfrak{a}) = |\mathcal{O}/\mathfrak{a}|$$

We may define two operations on two-sided ideals (which are the type we are considering here). These are addition

$$\mathfrak{a} + \mathfrak{b} = \{a + b \mid a \in \mathfrak{a}, b \in \mathfrak{b}\}$$

and multiplication

$$\mathfrak{a}\mathfrak{b} = \{a_1b_1 + a_2b_2 + \dots + a_nb_n \mid a_i \in \mathfrak{a}, b_i \in \mathfrak{b} \text{ for } i = 1, \dots, n, \text{ and } n \in \mathbb{N}_+\}$$

**Definition 5.1.2.** (Fractional ideal) We say an  $\mathcal{O}$ -submodule of  $K$  is a fractional ideal of  $\mathcal{O}$ , if it is of the form  $\alpha\mathfrak{a}$ , where  $\alpha \in K^*$  and  $\mathfrak{a}$  is an  $\mathcal{O}$ -ideal. Further we say a fractional ideal  $\mathfrak{a}$  is *invertible* if there exists another fractional ideal  $\mathfrak{b}$  such that  $\mathfrak{a}\mathfrak{b} = \mathcal{O}$ .

Invertible fractional ideals form, by their construction, an abelian group under ideal multiplication, which we will denote  $I(\mathcal{O})$ .

---

<sup>1</sup>Pronounced as “sea side” would be pronounced in English.

**Definition 5.1.3.** (Ideal-class group) The group of invertible fractional ideals quite clearly contains the group of principal fractional ideals  $P(\mathcal{O})$ . Thus we may define the ideal-class group by

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

Every ideal class  $[\mathfrak{a}] \in cl(\mathcal{O})$  has an integral ideal representative.

**Definition 5.1.4.** (Free and transitive action) The action of a finite commutative group  $G$  on a set  $X$  is said to be

- free, if the statement  $g \cdot x = x$  for some  $x \in X$  implies that  $g$  is the identity element in  $G$ .
- transitive, if for all  $x, x' \in X$  there is a  $g \in G$  such that  $x = g \cdot x'$ .

**Theorem 5.1.1.** *Let  $\mathcal{O}$  be an order in an imaginary quadratic field and  $\phi \in \mathcal{O}$ . Let  $Ell_p(\mathcal{O}, \phi)$  be the set of elliptic curves  $E/\mathbb{F}_p$  with  $End_p(E)$  being the order  $\mathcal{O}$ , and with Frobenius endomorphism  $\phi$ . If  $Ell_p(\mathcal{O}, \phi)$  is non-empty, then the ideal-class group  $cl(\mathcal{O})$  acts freely and transitively on this set, via the map*

$$\begin{aligned} cl(\mathcal{O}) \times Ell_p(\mathcal{O}, \phi) &\rightarrow Ell_p(\mathcal{O}, \phi) \\ ([\mathfrak{a}], E) &\mapsto E/\mathfrak{a} \end{aligned}$$

where  $\mathfrak{a}$  is the integral representative of the class.

We will denote the curve  $E/\mathfrak{a}$  more simply as  $\mathfrak{a}E$ .

*Proof.* [16, Theorem 4.5] □

Note that in the above theorem, the curve  $\mathfrak{a}E$  is the same curve given by Theorem 4.1.3.

We now return to the  $d$ -isogeny graphs we introduced in Section 4.1. The following result by Kohel, Delfs and Galbraith, gives us a case where the endomorphism ring of elliptic curves in a connected  $d$ -isogeny subgraph are the same order in the same imaginary quadratic field.

**Theorem 5.1.2.** *Let  $p \geq 5$  be a prime number and let  $V$  be a connected component of the  $d$ -isogeny graph of elliptic curves over the field  $\mathbb{F}_p$ . We assume  $p \equiv 11 \pmod{12}$  or that  $V$  contains no curve with  $j = 0$  or 1728. Let  $t$  be the shared trace of the Frobenius endomorphism of all curves in  $V$ , and let  $K = \mathbb{Q}(\sqrt{t^2 - 4p})$ . Assume that  $d \nmid t^2 - 4p$ .*

*Then all elliptic curves in  $V$  have the same  $\mathbb{F}_p$ -rational endomorphism ring  $\mathcal{O} \subseteq K$ , which is locally maximal at  $d$ . Moreover, if  $t^2 - 4p$  is a (non-zero) square modulo  $d$ , then  $V$  has a cycle whose length equals the order of  $[\mathfrak{l}]$  in  $cl(\mathcal{O})$ , where  $\mathfrak{l}$  is a prime ideal dividing  $d\mathcal{O}$ . If not, then  $V$  consists of a single vertex and no edges.*

*Proof.* [4, Theorem 4] □

For the purposes of the CSIDH protocol, we will consider only curves in the Montgomery form

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

We represent these curves by their Montgomery coefficient  $A$ . This representative is unique up to isomorphism, as is shown below.

**Theorem 5.1.3.** *Let  $p \geq 5$  be a prime congruent to 3 mod 8, and let  $E/\mathbb{F}_p$  be a supersingular elliptic curve, further let  $\phi_p$  be the Frobenius endomorphism. Then  $\text{End}_p(E) = \mathbb{Z}[\phi_p]$  if and only if there exists an  $A \in \mathbb{F}_p$  such that  $E$  is isomorphic to  $E_A : y^2 = x^3 + Ax^2 + x$  over  $\mathbb{F}_p$ . If such an  $A$  exists, then it is unique.*

*Proof.* [4, Proposition 8] □

We may describe the following problem, given by the ideal class group action defined above

*“Given two supersingular elliptic curves  $E, E'$  defined over the same field  $\mathbb{F}_p$ , and with the same endomorphism ring  $\mathcal{O}$ , compute an ideal  $\mathfrak{a}$  of  $\mathcal{O}$ , such that  $\mathfrak{a}E = E'$ . We further require the ideal be represented in such a way that its action can be efficiently evaluated.”*

We call this problem the “CSIDH key recovery problem”. It is clear to see it is analogous to the ECDLP of Chapter 2, and the DLP of Chapter 1.

**Conjecture 5.1.1.** There are no efficient classical or quantum algorithms solving the CSIDH key recovery problem.

## 5.2 CSIDH

We now use the theorems of the previous section to form the key exchange protocol CSIDH. We begin by defining our domain parameters as follows

- Fix a large prime  $p = 4 \cdot l_1 \dots l_n - 1$ , where  $l_i$  are small distinct odd primes.
- Fix the base curve  $E_0 : y^2 = x^3 + x$ , over  $\mathbb{F}_p$  with endomorphism ring  $\mathcal{O}$ .

Note that  $E_0$  is supersingular and  $p > 5$ , thus the trace of the Frobenius endomorphism of  $E_0$  is 0. This implies, by Theorem 5.1.2, that  $\text{End}_p(E_0)$  is an order in the imaginary quadratic field  $\mathbb{Q}(\sqrt{-p})$ . By Theorem 5.1.3, we further see that  $\text{End}_p(E_0) = \mathbb{Z}[\phi_p]$ .

Theorem 5.1.2 further ensures that the ideals  $l_i\mathcal{O}$  split as

$$l_i\mathcal{O} = \mathfrak{l}_i \bar{\mathfrak{l}}_i$$

where  $\mathfrak{l}_i = (l_i, \phi_p - 1)$  and  $\bar{\mathfrak{l}}_i = (l_i, \phi_p + 1)$ .

## CSIDH

**Alice**

**Bob**

..... Let  $(p, E_0)$  be the domain parameters. ....

Choose random integers  $(a_1, \dots, a_n)$

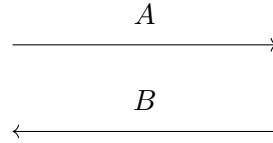
Choose random integers  $(b_1, \dots, b_n)$

Define the ideal class  $[\mathbf{a}] = [\mathfrak{l}_1^{a_1} \dots \mathfrak{l}_n^{a_n}]$

Define the ideal class  $[\mathbf{b}] = [\mathfrak{l}_1^{b_1} \dots \mathfrak{l}_n^{b_n}]$

Compute the Mont. coefficient  $A \in \mathbb{F}_p$   
of the curve  $E_A = \mathbf{a}E_0$

Compute the Mont. coefficient  $B \in \mathbb{F}_p$   
of the curve  $E_B = \mathbf{b}E_0$



Compute the curve  $E_{AB}$   
 $E_{AB} = \mathbf{a}E_B$

Compute the curve  $E_{BA}$   
 $E_{BA} = \mathbf{a}E_B$

The shared key at the end of the protocol is the Montgomery coefficient  $S$  of the curve  $E_{AB} \cong E_{BA} \cong [\mathbf{a}][\mathbf{b}]E_0$  in the form  $E_{AB} : y^2 = x^3 + Sx^2 + x$ .

Theorem 5.1.3 ensures that our protocol is sound, as ideal multiplication is commutative, and  $[\mathbf{a}][\mathbf{b}]E_0$  must be  $\mathbb{F}_p$ -isomorphic to a unique curve in Montgomery form.

## 5.3 Discussion

Without the use of torsion point images, CSIDH is impervious to the Castryck-Decru attack shown in the previous chapter. At time of writing, no significantly devastating attack has been found on CSIDH, thus making it a tentative addition to our library of key exchange protocols.

This protocol is computationally heavier than Diffie-Hellman, however, where Diffie-Hellman becomes insecure in the face of Shor's algorithm of Section 3.2.3, the current state-of-the art classical and quantum algorithms cannot yet feasibly break CSIDH, as Castryck et al. discuss in [4]. This makes CSIDH into a potential replacement for Diffie-Hellman, if and when quantum computation becomes powerful enough to pose a danger to our current classical methods.

As of now, post-quantum cryptography is still very young, and more investigation is certainly needed into these protocols, to ascertain whether they are in fact useful and secure alternatives.

# Appendix A

## Geometric principles of Elliptic Curves

This Appendix reviews the algebraic geometrical principles underlying the theory of elliptic curves discussed in this text. The discussion here is quite brief, and is only designed to provide a reminder or introduction to these concepts. For the reader that wishes to learn further details, we recommend they read [17] or [11] for a higher level overview of algebraic geometry.

### A.1 The projective plane

**Definition A.1.1.** (Projective plane over  $K$ ) Let  $K$  be a field, and  $\sim$  an equivalence relation between elements of  $K^3 \setminus \{(0, 0, 0)\}$  defined as

$$(x_0, y_0, z_0) \sim (x_1, y_1, z_1) \text{ iff } \exists \lambda \in K \setminus \{0\} : (x_0, y_0, z_0) = \lambda(x_1, y_1, z_1)$$

The projective plane over  $K$ , denoted  $K\mathbf{P}^2$ , has the set of all equivalence classes given by  $\sim$  as a point set.

We call points on the projective plane  $K\mathbf{P}^2$  of the form  $(x : y : 1)$  *finite points*, and those of the form  $(x : y : 0)$  *points at infinity*. The finite projective points form the affine plane  $K\mathbf{A}^2$ , embedded in the projective plane.

**Definition A.1.2.** (Algebraic plane projective curve) Let  $K$  be a field. An algebraic plane projective curve  $C/K$  is a homogeneous polynomial  $C(x, y, z)$  with coefficients in  $K$ . For any field extension  $F$  of  $K$ , the set

$$C(F) = \{(x : y : z) \in F\mathbf{P}^2 \mid C(x, y, z) = 0\}$$

are the  $F$ -rational points of  $C$ .

If all the partial derivatives with respect to  $x, y$  and  $z$  vanish at a point  $P \in C(F)$ , we say  $P$  is a *singular point* and that  $C(F)$  is *singular*. If there is no such point then we say  $C(F)$  is *non-singular* or *smooth*.

**Example A.1.1.** Consider the polynomial  $C(x, y, z) = x^2 + y^2 - 2^2 z^2$  with real coefficients. This is an algebraic projective curve over  $\mathbb{R}$ . The  $\mathbb{R}$ -rational points of this curve are

$$C(\mathbb{R}) = \{(x : y : z) \in \mathbb{RP}^2 \mid C(x, y, z) = 0\}$$

We can intuitively visualize this set in three dimensional real space, as the plot in Figure A.1. In this plot every line on the surface (all of which pass through the origin) corresponds to a point in the projective plane. The finite  $\mathbb{R}$ -rational points

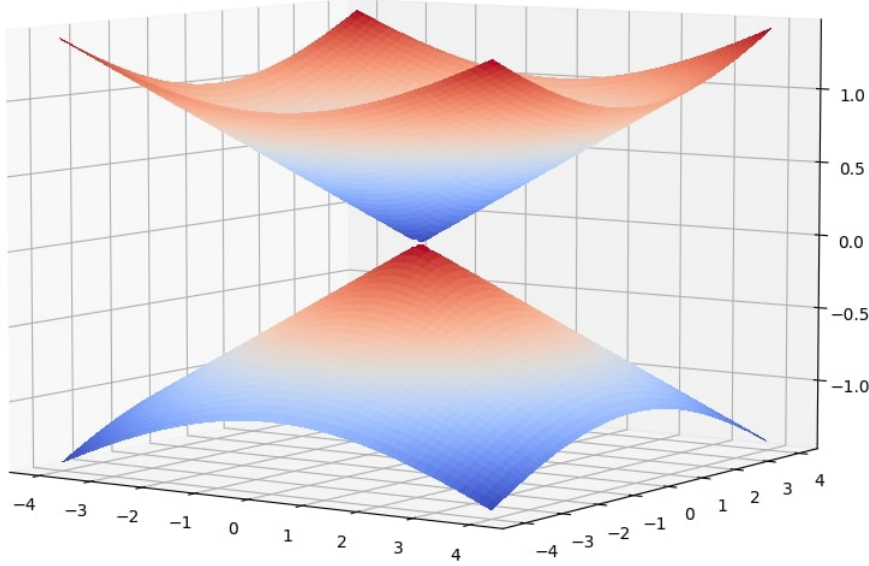


Figure A.1: Intuitive plot of  $C(\mathbb{R})$ .

of this curve form a circle of radius 2 in the affine plane. There are no points at infinity in  $C(\mathbb{R})$ .

**Example A.1.2.** Consider the polynomial  $E(x, y, z) = y^2 z - x^3 - Axz^2 - Bz^3$  with coefficients in the field  $\mathbb{F}_7$ . This is an algebraic projective curve over  $\mathbb{F}_7$ . The  $\mathbb{F}_7$ -rational points of this curve are

$$E(\mathbb{F}_7) = \{(x : y : z) \in \mathbb{F}_7 \mathbf{P}^2 \mid y^2 z = x^3 + Axz^2 + Bz^3\}$$

This curve has a singular point at infinity, regardless of the choice of coefficients. This point is  $(0 : 1 : 0)$ .

An important algebraic property of a curve, which we will require for the definition of an elliptic curve, is its *genus*. This is a numerical value that is derived from the Riemann-Roch theorem. Explaining this concept in its entirety is beyond the scope of this text, we direct the reader to [18, Section II.5] for an in-depth discussion.

**Example A.1.3.** Curve  $C/\mathbb{R}$  from example A.1.1 is of genus 0. Curve  $E/\mathbb{F}_7$  from example A.1.2 is of genus 1.



## A.2 Elliptic curves

With the definitions of the space we are working in laid out, we are now ready to introduce the elliptic curve in the projective plane  $K\mathbf{P}^2$ , and the polynomials that define them.

### A.2.1 The Weierstrass equation

**Definition A.2.1.** Let  $K$  be a field. An elliptic curve over  $K$  is a smooth, algebraic plane projective curve of genus 1, with a distinguished  $K$ -rational point.

It can be shown that the  $K$ -rational points of any elliptic curve satisfy the equation

$$y^2z + a_1xyz + a_3yz^2 = x^3 + a_2x^2z + a_4xz^2 + a_6z^3 \quad (\text{A.1})$$

where  $a_1, a_2, a_3, a_4, a_6 \in K$  [17]. From this equation, we immediately see that any elliptic curve has a single point at infinity:  $(0 : 1 : 0)$ . This point we shall denote  $O$ . If we restrict ourselves to only the finite points, on the affine plane, we find they satisfy the *generalized Weierstrass equation*

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (\text{A.2})$$

**Definition A.2.2.** (Discriminant) For an elliptic curve  $E$  with generalised Weierstrass equation A.2, we define the discriminant

$$\Delta = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6$$

Where  $b_2 = a_1^2 + 4a_2$ ,  $b_4 = 2a_4 + a_1a_3$ ,  $b_6 = a_3^2 + 4a_6$  and  $b_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$ .

**Theorem A.2.1.** Let  $E$  be an elliptic curve with generalised Weierstrass equation A.2.  $E$  is non-singular if and only if the discriminant  $\Delta \neq 0$ .

*Proof.* First we show that  $O$  is a non-singular point. We know that the homogeneous polynomial defining the curve on the projective plane  $E$  is of the form

$$P(x, y, z) = y^2z + a_1xyz + a_3yz^2 - x^3 - a_2x^2z - a_4xz^2 - a_6z^3$$

We see that

$$\frac{\partial P}{\partial z}(x, y, z) = y^2 + a_1xy + 2a_3yz - a_2x^2 - 2a_4xz - 3a_6z^2$$

Thus at the point  $O = (0 : 1 : 0)$  we have that  $\frac{\partial P}{\partial z}(O) = 1 \neq 0$ .

Now we consider the finite points.

The only change of variables fixing the point at infinity and preserving the Weierstrass equation is

$$\begin{aligned} x &= \alpha^2x' + \beta \\ y &= \alpha^3y' + \alpha^2\gamma x' + \delta \end{aligned}$$

with  $\alpha, \beta, \gamma, \delta \in \bar{K}$ , and  $\alpha \neq 0$ . Through this substitution we find that

$$\begin{aligned}\alpha^2 b'_2 &= b_2 + 12\beta \\ \alpha^4 b'_4 &= b_4 + \beta b_2 + 6\beta^2 \\ \alpha^6 b'_6 &= b_6 + 2\beta b_4 + \beta^2 b_2 + 4\beta^3 \\ \alpha^8 b'_8 &= b_8 + 3\beta b_6 + 3\beta^2 b_4 + \beta^3 b_2 + 3\beta^4\end{aligned}$$

Thus we find that

$$\alpha^{12} \Delta' = \Delta$$

Suppose now that  $E$  had a finite singular point  $P = (x_0, y_0)$ . The above logic implies that the change in variables given by

$$\begin{aligned}x &= x' + x_0 \\ y &= y' + y_0\end{aligned}$$

leaves the value of the discriminant unchanged. Thus we can assume, without loss of generality, that the singular point is  $(0, 0)$ .

Using the generalised Weierstrass equation of  $E$ , given by:

$$E : p(x, y) = y^2 + a_1 xy + a_3 y - x^3 - a_2 x^2 - a_4 x - a_6 = 0$$

we find that

$$a_6 = p(0, 0) = 0 \quad a_4 = \frac{\partial p}{\partial x}(0, 0) = 0 \quad a_3 = \frac{\partial p}{\partial y}(0, 0) = 0$$

Thus we find that we may simplify the polynomial  $p$  to

$$p(x, y) = y^2 + a_1 xy - x^3 - a_2 x^2$$

This equation has discriminant  $\Delta = 0$ .

To show that  $\Delta \neq 0$  when  $E$  is non-singular, we assume for simplicity that  $\text{char}(K) \neq 2$ . Then we can apply the change of variables  $y = 2y' + a_1 x + a_3$ , which gives us the new equation

$$E : p'(x, y) = y^2 - 4x^3 - b_2 x^2 - 2b_4 x - b_6 = 0$$

If  $E$  had a finite singular point  $P = (x_0, y_0)$ , then it would have to satisfy

$$\frac{\partial p'}{\partial x}(x_0, y_0) = 12x_0^2 + 2b_2 x_0 + 2b_4 = 0 = 2y_0 = \frac{\partial p'}{\partial y}(x_0, y_0)$$

This implies that any singular point is of the form  $(x_0, 0)$ , where  $x_0$  is a double root of the polynomial  $4x^3 + b_2 x^2 + 2b_4 x + b_6$ . This polynomial has discriminant  $16\Delta$ , where  $\Delta$  is the discriminant of  $E$ . Thus we have that if  $\Delta = 0$ , then there exists a double root of the polynomial  $4x^3 + b_2 x^2 + 2b_4 x + b_6$ , and hence a singular point of  $E$ .  $\square$

**Definition A.2.3.** (j-Invariant) For an elliptic curve  $E$  with generalised Weierstrass equation A.2, we define its j-invariant as

$$j = \frac{(b_2^2 - 24b_4)^3}{\Delta}$$

Where  $b_2 = a_1^2 + 4a_2$ ,  $b_4 = 2a_4 + a_1a_3$ , and  $\Delta$  is the discriminant.

We can simplify equation A.2, in fields whose characteristic is not 2 or 3. First if  $ch(K) \neq 2$ , then 2 has a multiplicative inverse. It can be easily seen that we can re-arrange A.2 as

$$(y + 2^{-1}(a_1x + a_3))^2 = x^3 + (a_2 + 4^{-1}a_1^2)x^2 + (a_4 + 2^{-1}a_1a_3)x + (4^{-1}a_3^2 + a_6)$$

which can be expressed as

$$Y^2 = x^3 + a'_2x^2 + a'_4x + a'_6$$

Now, if  $ch(K) \neq 3$ , we can perform a further transformation by letting  $X = x + 3^{-1}a'_2$  and we arrive at the *Weierstrass equation* for  $E$

$$Y^2 = X^3 + AX + B \tag{A.3}$$

In this case the discriminant can also be simplified to  $\Delta = 4A^3 + 27B^2$ , and the j-invariant to  $j = 1728 \frac{(4A)^3}{\Delta}$ .

Note that this is not the only useful “simpler” form for the equation of an elliptic curve. If  $ch(K) \neq 2$ , we may also use *Montgomery form*

$$BY^2 = X^3 + AX^2 + X \tag{A.4}$$

Elliptic curves in Montgomery form may be represented by their Montgomery coefficients  $(A, B)$ , or simply by the coefficient  $A$  when  $B = 1$ .

**Theorem A.2.2.** *Let  $E_1$  and  $E_2$  be two elliptic curves, defined over the field  $\bar{K}$ . Then  $E_1$  and  $E_2$  are isomorphic if and only if they have the same j-invariant.*

*Proof.* We use the same change of variables introduced in the proof of Theorem A.2.1

$$\begin{aligned} x &= \alpha^2x' + \beta \\ y &= \alpha^3y' + \alpha^2\gamma x' + \delta \end{aligned}$$

with  $\alpha, \beta, \gamma, \delta \in \bar{K}$ , and  $\alpha \neq 0$ . This substitution gives us

$$j' = \frac{(b_2'^2 - 24b_4')^3}{\Delta'} = \alpha^{12} \frac{(b_2^2 - 24b_4)^3}{\Delta} = j$$

Thus we see that isomorphic elliptic curves have the same j-invariant.

We now show that elliptic curves with the same j-invariant must be isomorphic. We show this for the case where  $char(K) \neq 2, 3$  for simplicity, but the result can be extended to any field. If

$$\begin{aligned} E_1 : y^2 &= x^3 + Ax + B \\ E_2 : y'^2 &= x'^3 + A'x' + B' \end{aligned}$$

Then

$$j = 1728 \frac{(4A)^3}{4A^3 + 27B^2} = 1728 \frac{(4A')^3}{4A'^3 + 27B'^2} = j'$$

and thus we find that  $A^3B'^2 = A'^3B^2$ .

If  $A = 0$ , then  $j = 0$  and we have an isomorphism given by the change of variables  $(x, y) = (\alpha^2x', \alpha^3y')$ , where  $\alpha = (B/B')^{1/6}$ .

If  $B = 0$ , then  $j = 1728$  and we have an isomorphism given by the change of variables  $(x, y) = (\alpha^2x', \alpha^3y')$ , where  $\alpha = (A/A')^{1/4}$ .

If  $AB = 0$ , then  $j \neq 0, 1728$  and we have an isomorphism given by the change of variables  $(x, y) = (\alpha^2x', \alpha^3y')$ , where  $\alpha = (B/B')^{1/6} = (A/A')^{1/4}$ .

□

**Example A.2.1.** Consider the algebraic plane projective curve  $E/\mathbb{R}$ , whose finite  $\mathbb{R}$ -rational points are defined by the equation

$$y^2 = x^3 + 2x + 1$$

The plot of this curve on the affine real plane can be seen in Figure A.2. The

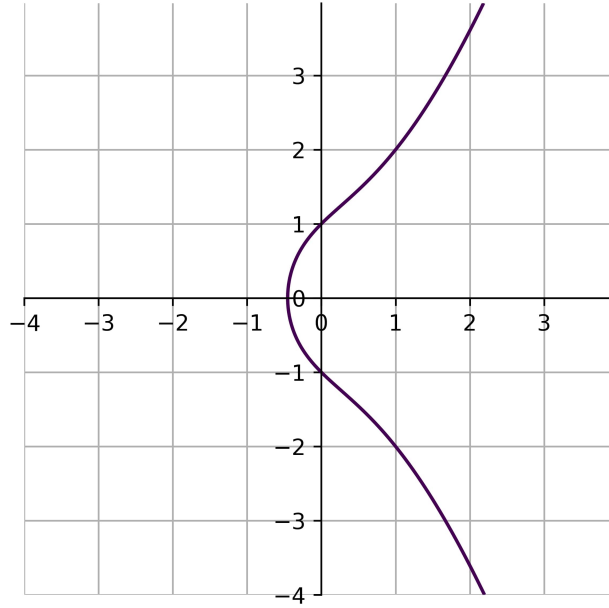


Figure A.2: Plot of  $y^2 = x^3 + 2x + 1$  in  $\mathbb{R}\mathbf{A}^2$ .

discriminant of this curve is  $\Delta = 4 \cdot 2^3 + 27 \cdot 1^2 = 59 \neq 0$ . Thus this is a non-singular curve (i.e. it is smooth). Equation A.3 is in the form of a Weierstrass equation, thus we conclude  $E$  is an elliptic curve.

## A.3 Tangent and secant lines

In Chapter 2 we saw that the  $K$ -rational points of an elliptic curve form an abelian group with respect to point addition. Here we define the addition of points in a more general manner and provide necessary knowledge for the omitted parts of the proof of Theorem 2.1.1.

**Definition A.3.1.** (Tangent) Let all points of the projective curve  $C$  be given implicitly by  $f(x, y, z) = 0$ . Let  $P = (x_0 : y_0 : z_0)$  be one such point, then the tangent to  $C$  at  $P$  is given by

$$\frac{\partial f}{\partial x}(P)(x - x_0) + \frac{\partial f}{\partial y}(P)((y - y_0) \frac{\partial f}{\partial z}(P)(z - z_0) = 0$$

If all partial derivatives are 0 at  $P$ , we say  $C$  has no tangent at  $P$ .<sup>1</sup>

**Definition A.3.2.** (Secant) Let all points of the projective curve  $C$  be given implicitly by  $f(x, y, z) = 0$ . Let  $P = (x_0 : y_0 : z_0)$  and  $Q = (x_1 : y_1 : z_1)$  be two such points. The line  $\overline{PQ}$  is the secant line to  $C$ , through  $P$  and  $Q$ .

While in Chapter 2, we saw an algebraic definition of point addition, specifically for elliptic curves in fields of characteristic not 2, or 3. we can use the intersection of lines with the curve to geometrically define the same process.

**Definition A.3.3.** (Addition of points on an EC) Let  $P, Q$  be  $K$ -rational points on the elliptic curve  $E$  over the field  $K$ . We define the addition of these as follows:

Let  $L$  be the line through a point  $P$  and  $O$ . If  $P = O$  this is a tangent line, otherwise it is a secant line. This line will intersect  $E$  at a third point  $Q$ , counting multiplicities. We define  $-P = Q$ .

For any three points of intersection of a line  $L$  with  $E$ ,  $P, Q, R$ , we define  $P + Q = -R$ .

This definition has allowed us to generalise this operation to all elliptic curves, irrespective of the field we are working with. This definition also allows us to have a more visual intuition of the operation, as exemplified in Figure A.3.

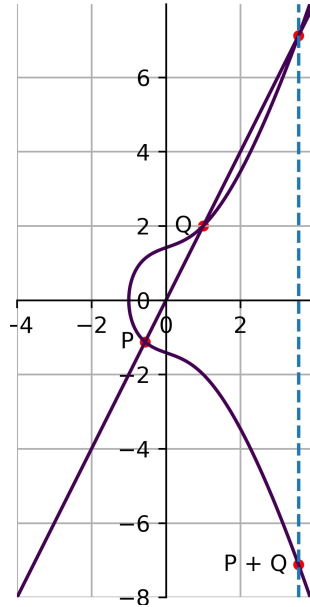


Figure A.3: Point addition example on the real affine plane.

<sup>1</sup>This should never be the case in a non-singular curve.

While our definition appears to be sound, it remains unclear whether it is true that every line intersecting an elliptic curve will indeed intersect it at three points, counting multiplicities, as is implied. This fact directly follows from Bézout's Theorem.

**Theorem A.3.1.** (*Bézout's Theorem*) Let  $f(x, y, z) = 0$  and  $g(x, y, z) = 0$  give two polynomial curves  $F$  and  $G$  in  $K\mathbf{P}^2$ . Let  $E$  be an algebraically closed field containing  $K$ . If  $F$  and  $G$  are not multiples of a common non-constant polynomial, then they intersect in  $E$  at  $\deg(f) \cdot \deg(g)$  points, counting multiplicities.

*Proof.* [2] □

**Corollary A.3.1.1.** Let  $K$  be algebraically closed,  $L$  a line and  $E$  an elliptic curve in  $K\mathbf{P}^2$ .  $L$  intersects  $E$  at 3 points, counting multiplicities.

Thus we find that the set of  $K$ -rational points of an elliptic curve will always be closed under addition.

From these same geometric definitions it becomes slightly less tedious to prove the associativity of this operation. One proof utilising these concepts can be found in [23].

## A.4 Isogenies

**Definition A.4.1.** (Isogeny) Let  $E_1$  and  $E_2$  be elliptic curves over a field  $K$ , and  $F$  any field extension of  $K$ . An isogeny  $\phi : E_1(F) \rightarrow E_2(F)$  is a rational map of the form

$$\phi(x, y) = (f(x, y), g(x, y))$$

where  $f$  and  $g$  are rational functions, and  $\phi(O) = O$ .

Such maps are always group homomorphisms between the groups of  $F$ -rational points  $E_1(F)$  and  $E_2(F)$  [17, Theorem III.4.8].

If there exists an isogeny between the curves  $E_1, E_2$ , such that  $\phi(E_1) \neq \{O\}$ , we say they are *isogenous*.

We will denote isogenies from  $E_1(F)$  to  $E_1(F)$  more simply as  $\phi : E_1 \rightarrow E_2$ .

**Example A.4.1.** Let  $E/K$  be an arbitrary elliptic curve, and  $m \in \mathbb{Z}$ . Then we may define the isogeny

$$[m] : E \rightarrow E \quad [m](P) = mP \quad \forall P \in E(\bar{K})$$

In particular if  $m = 0$ , we define  $[0](P) = O \quad \forall P \in E(\bar{K})$ .

As multiplication by any  $m \in \mathbb{Z}$  is well-defined for any elliptic curve, it follows that the set of isogenies  $\{[m] : E \rightarrow E \mid m \in \mathbb{Z}\}$  exists for all elliptic curves  $E$ .

**Example A.4.2.** Consider the elliptic curve  $E : y^2 = x^3 + x$  defined over a finite field  $K$ . Assuming there is an  $i \in K$ , such that  $i^2 = -1$ , then  $E$  has an isogeny we will call  $i : E \rightarrow E$ , defined by  $(x, y) \mapsto (-x, iy)$ .

**Lemma A.4.1.** The isogeny  $[0] : E_1 \rightarrow E_2$  defined by  $[0](P) = O \quad \forall P \in E_1(\bar{K})$  is the only constant isogeny from  $E_1$  to  $E_2$  for any two such elliptic curves.

*Proof.* Suppose  $\phi : E_1 \rightarrow E_2$ , with  $\phi \neq [0]$ , is constant. Then there is a  $Q \in E_2(\bar{K})$  such that  $\phi(P) = Q$  for all  $P \in E_1(\bar{K})$ . Now, since  $\phi$  is a group homomorphism it follows that

$$Q = \phi(mP) = m\phi(P) = mQ$$

for all  $m \in \mathbb{Z}$ . This is only possible if  $Q = O$ , thus we must have that  $\phi = [0]$ .  $\square$

If an isogeny  $\phi$  is non-constant we say it is *finite*.

**Definition A.4.2.** (Ideal) Let  $E/K$  be an elliptic curve (or any other projective curve), then the ideal of  $E$  is given by

$$I(E) = \{f \in \bar{K}[X] : f(P) = 0 \ \forall P \in E(\bar{K})\}$$

**Definition A.4.3.** (Function Field) Let  $E/K$  be an elliptic curve (or any other projective curve), and  $E(F)$  the  $F$ -rational points of  $E$  for some field extension  $F$  of  $K$ . The *projective coordinate ring* of  $E/K$  is defined by

$$K[E] = \frac{K[X]}{I(E) \cap K[X]}$$

This is an integral domain, and its quotient field, denoted  $K(E)$ , is the function field of  $E$ .

**Lemma A.4.2.** Let  $E_1/K$  and  $E_2/K$  be elliptic curves and  $\phi : E_1 \rightarrow E_2$  a finite isogeny. Composition by  $\phi$  induces an injection of function fields fixing  $K$

$$\phi^* : K(E_2) \rightarrow K(E_1)$$

Here we have that  $K(E_1)$  is a field extension of  $\phi^*K(E_2)$ .

*Proof.* [17, Theorem II.2.4(a)]  $\square$

**Definition A.4.4.** (Degree) We define the degree of a finite isogeny of elliptic curves over the field  $K$ ,  $\phi : E_1 \rightarrow E_2$ , as

$$\deg \phi = [K(E_1) : \phi^*K(E_2)]$$

By convention we set  $\deg[0] = 0$ . We also denote an isogeny of degree  $d$  as a  $d$ -isogeny.

Further we say that the isogeny  $\phi$  is *separable* (*inseparable*, *purely inseparable*) if the field extension  $K(E_1)/\phi^*K(E_2)$  has the respective property.

**Example A.4.3.** The isogeny  $[m] : E \rightarrow E$  is an  $m^2$ -isogeny for all elliptic curves  $E$  and all  $m \in \mathbb{Z}$ . This is simple to prove by induction.

**Lemma A.4.3.** If  $\phi$  is a separable isogeny, then we have that

$$\# \ker \phi = \deg \phi$$

*Proof.* [17, Theorem III.4.10(c)]  $\square$

### A.4.1 The group of isogenies

Given that isogenies of elliptic curves are group homomorphisms, it follows that they themselves form groups. Let  $K$  be a field, and  $F$  any field extension of  $K$ . We denote the group of isogenies defined over  $F^2$ , between curves  $E_1/K$  and  $E_2/K$ , by  $\text{Hom}_F(E_1, E_2)$ . If  $F = \bar{K}$ , we use the simpler notation  $\text{Hom}(E_1, E_2)$ . The addition rule of this group is defined by

$$(\phi + \psi)(P) = \phi(P) + \psi(P) \quad \forall \phi, \psi \in \text{Hom}_F(E_1, E_2), P \in E_1(K)$$

We denote  $\text{Hom}_F(E, E)$ , the isogenies defined over  $F$  from a curve to itself, by  $\text{End}_F(E)$ . Equivalently to above, the set of isogenies defined over  $\bar{K}$  from a curve to itself are simply denoted  $\text{End}(E)$ .

**Theorem A.4.4.** *Let  $E$  be an elliptic curve over the field  $K$ . Then the set of endomorphisms of  $E$ ,  $\text{End}(E)$  is a ring of characteristic 0, under the addition rule*

$$(\phi + \psi)(P) = \phi(P) + \psi(P) \quad \forall \phi, \psi \in \text{End}(E), P \in E(K)$$

*and multiplication rule*

$$(\phi \cdot \psi)(P) = \phi(\psi(P)) \quad \forall \phi, \psi \in \text{End}(E), P \in E(K)$$

*Proof.*  $(\text{End}(E), +)$  being an abelian group follows directly from the fact that  $(E(K), +)$  is an abelian group.

Associativity of  $\cdot$  follows directly from the associativity of endomorphism composition.

Clearly  $[1] : E \rightarrow E$  is the multiplicative identity.

Let  $\alpha, \beta, \gamma \in \text{End}(E)$ , then we have for all  $P \in E(K)$

$$\begin{aligned} (\alpha \cdot (\beta + \gamma))(P) &= \alpha((\beta + \gamma)(P)) \quad \text{multiplication rule} \\ &= \alpha(\beta(P) + \gamma(P)) \quad \text{addition rule} \\ &= \alpha(\beta(P)) + \alpha(\gamma(P)) \quad \text{group homomorphism property} \end{aligned}$$

Thus, we see that  $\cdot$  is right-distributive over  $+$ . Equivalently we can show it is left-distributive.

The characteristic of  $\text{End}(E)$  follows from the fact that  $[0]$ , the additive identity, is the only constant endomorphism over  $E$ . Hence, if  $\exists m \in \mathbb{Z}$  such that

$$[m][1] = [1] + \dots + [1] = [0]$$

Then we must have that  $[m] = [0]$ . □

Notice that the set of isogenies shown in Example A.4.1 is a subring of  $\text{End}(E)$  for all elliptic curves  $E$ . In fact, if  $E$  is defined over a field  $K$ , with  $\text{char}(K) = 0$ , then usually this is the entirety of  $\text{End}(E)$ , in other words  $\text{End}(E) \simeq \mathbb{Z}$ . Those curves whose endomorphism ring is strictly larger than  $\mathbb{Z}$  are said to have *complex multiplication*.

---

<sup>2</sup>That is isogenies whose coefficients are in  $F$ .



**Example A.4.4.** Let  $E$  be an elliptic curve over the finite field  $K$ , of prime characteristic  $p$ .

The Frobenius endomorphism is often, in curves with complex multiplication, an endomorphism that lies outside of  $\mathbb{Z}$ . It is defined for  $q = p^r$ ,  $r \in \mathbb{Z}_+$

$$\phi_q : E \rightarrow E \quad \phi_q(x, y) = (x^q, y^q)$$

The Frobenius endomorphism of the example above is one of the most crucial endomorphisms in the theory of elliptic curves. It can be shown, for example, that the Frobenius endomorphism of an elliptic curve  $E/\mathbb{F}_q$ , where  $q$  is a prime power, satisfies the characteristic equation

$$\phi_q^2 - t\phi_q + q = 0 \tag{A.5}$$

for the unique  $t = q + 1 - \#E(\mathbb{F}_q)$ . This value  $t$  is the *trace* of the Frobenius endomorphism.

**Theorem A.4.5.** Two elliptic curves  $E_1/\mathbb{F}_q$  and  $E_2/\mathbb{F}_q$  are isogenous if and only if the traces of their respective Frobenius endomorphisms are equal.

*Proof.* [21, Theorem 3.1] □

**Definition A.4.5.** ( $m$ -torsion points) Let  $E$  be an elliptic curve, and  $m \in \mathbb{Z}$ ,  $m \neq 0$ . The set of  $m$ -torsion points of  $E$  is defined by

$$E[m] = \{P \in E \mid [m](P) = mP = O\}$$

**Theorem A.4.6.** Let  $E$  be an elliptic curve and  $\Phi \subseteq E$  a finite subgroup of  $E$ . Then there is a unique elliptic curve  $E'$  and a separable isogeny  $\phi : E \rightarrow E'$  such that

$$\ker \phi = \Phi$$

*Proof.* [17, Theorem III.4.12] □

**Corollary A.4.6.1.** Let  $\phi : E_1 \rightarrow E_2$  be an isogeny. Then the ideal  $\ker \phi$  uniquely determines  $\phi$ , up to isomorphism.

## A.4.2 The dual isogeny

**Theorem A.4.7.** Let  $\phi : E_1 \rightarrow E_2$  be a finite  $d$ -isogeny. Then there exists a unique isogeny

$$\hat{\phi} : E_2 \rightarrow E_1$$

such that  $\phi \circ \hat{\phi} = [d]$ .

*Proof.* [17, Theorem III.6.1(a)] □

The isogeny  $\hat{\phi}$  is called the *dual isogeny* to  $\phi$ .

**Example A.4.5.** Let  $E/\mathbb{F}_q$ , where  $q$  is a prime power, be an elliptic curve. The dual of the Frobenius endomorphism  $\phi_q : E \rightarrow E$

**Theorem A.4.8.** *Let  $\phi : E_1 \rightarrow E_2$  be an isogeny. Then*

1. *If  $\deg \phi = d$ , then*

$$\phi \circ \hat{\phi} = \hat{\phi} \circ \phi = [d]$$

2. *Let  $\alpha : E_2 \rightarrow E_3$  be another isogeny, then*

$$\widehat{\alpha \circ \phi} = \hat{\phi} \circ \hat{\alpha}$$

3.  $\deg \hat{\phi} = \deg \phi$

*Proof.* If  $\phi$  is constant then clearly all statements are correct. Thus we prove them only in the case where all isogenies are finite, and in particular  $\deg \phi = d \neq 0$ .

1. By definition we have that

$$\phi \circ \hat{\phi} \circ \phi = \phi \circ [d]$$

Further we have that  $\phi \circ [d] = [d] \circ \phi$ . Thus, since  $\phi$  is finite, we must have that  $\phi \circ \hat{\phi} = [d]$

2. Let  $\deg \alpha = a$ , then

$$(\hat{\phi} \circ \hat{\alpha}) \circ (\alpha \circ \phi) = \hat{\phi} \circ [a] \circ \phi = [a] \circ \hat{\phi} \circ \phi = [ad]$$

Now, since the dual isogeny is unique, it follows that  $\widehat{\alpha \circ \phi} = \hat{\phi} \circ \hat{\alpha}$ .

3. We know that  $\deg \phi = d$ , and that  $\deg [d] = d^2$ , thus

$$[d^2] = [\deg(\hat{\phi} \circ \phi)] = [(\deg \hat{\phi})(\deg \phi)] = [d \deg \hat{\phi}]$$

Thus, we must have that  $\deg \hat{\phi} = d$ .

□

# Appendix B

## A brief discussion of quantum physics

The field of quantum mechanics aims to find a model to describe the behaviour of physical systems in time, and in particular that of those of atomic and subatomic scale.

Here we introduce the mathematical formulation of the model for quantum mechanical systems, in particular aimed at how we can use it to describe the system of a quantum computer, and the information within it. For the reader that wishes to expand on this introduction, we recommend [15] or [9].

### B.1 Foundational principles of quantum mechanics

In physics, all systems are modelled in terms of three concepts: states, observables and time evolution. The same is done in quantum theory. Here we represent all possible states a system may take as vectors<sup>1</sup> in the projective space  $\mathcal{H}\mathbf{P}^n$ , where  $\mathcal{H}$  is a complex Hilbert space.

**Definition B.1.1.** (Complex Hilbert space) A finite-dimensional complex Hilbert space  $\mathcal{H}$ , is a complete<sup>2</sup> vector space with an inner product  $\langle \cdot | \cdot \rangle : \mathcal{H} \times \mathcal{H} \rightarrow \mathbb{C}$ , satisfying

1.  $\langle x | y \rangle = \langle y | x \rangle^*$
2.  $\langle x | x \rangle \geq 0$  and  $\langle x | x \rangle = 0$  if and only if  $x = 0$
3.  $\langle x | \alpha y + \beta z \rangle = \alpha \langle x | y \rangle + \beta \langle x | z \rangle$

For any vectors  $x, y, z \in \mathcal{H}$ , and scalars  $\alpha, \beta \in \mathbb{C}$ .

---

<sup>1</sup>Note that in physics the term “vector” refers to certain specific quantities, and does not have the same general connotation it has in mathematics. Here “vector” always refers to the mathematical object. In physics literature the elements of these Hilbert spaces are usually called “kets”.

<sup>2</sup>In the sense that every Cauchy sequence has a limit.

Here we will denote vectors in Hilbert spaces using Dirac's bra-ket notation:  $|\psi\rangle$ . All vectors  $|\psi\rangle, |\phi\rangle \in \mathcal{H}$ , such that  $|\psi\rangle = e^{i\theta} |\phi\rangle$ , belong to the same equivalence class, and thus represent the same state in the projective space. We can, hence, assume all states to be represented by unit vectors.

We model physical quantities known as *observables*, as self-adjoint linear operators acting on the Hilbert space.

**Definition B.1.2.** (Self-adjoint operator) A linear operator  $\hat{A}$  on a finite-dimensional complex Hilbert space  $\mathcal{H}$ , is self-adjoint, if it satisfies the following

$$\langle \hat{A}x | y \rangle = \langle x | \hat{A}y \rangle$$

for all  $x, y \in \mathcal{H}$ .

**Example B.1.1.** *Spin is an observable. For a single spin- $\frac{1}{2}$  particle, we have three spin operators  $(S_x, S_y, S_z)$ , given by*

$$S_x = \frac{\hbar}{2} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad S_y = \frac{\hbar}{2} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad S_z = \frac{\hbar}{2} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Where  $\hbar$  is the reduced Planck constant. The matrices multiplying  $\frac{\hbar}{2}$  are known as the Pauli matrices. The total spin  $\hat{\mathbf{S}}$  is given by the operator

$$\hat{\mathbf{S}} = \frac{\hbar}{2} \boldsymbol{\sigma}$$

Where  $\boldsymbol{\sigma}$  is a vector whose components are the Pauli matrices. The eigenvectors of the spin operator  $S_z$ , from example B.1.1, are the unit vectors

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

With eigenvalues  $\frac{\hbar}{2}$  and  $-\frac{\hbar}{2}$  respectively. These eigenvectors span the Hilbert space  $\mathbb{C}^2$ .

**Lemma B.1.1.** *A self-adjoint operator  $\hat{A}$  on a complex Hilbert space has real eigenvalues.*

*Proof.* Let  $\lambda$  be an eigenvalue of  $\hat{A}$ , then we have  $\hat{A} |x\rangle = \lambda |x\rangle$  for a non-zero vector  $|x\rangle$ , and

$$\lambda^* \langle x | x \rangle = \langle \lambda x | x \rangle = \langle \hat{A}x | x \rangle = \langle x | \hat{A}x \rangle = \langle x | \lambda x \rangle = \lambda \langle x | x \rangle$$

$|x\rangle$  is not 0, thus  $\lambda^* = \lambda$ , and we have that  $\lambda$  must be real.  $\square$

**Lemma B.1.2.** *A self-adjoint operator  $\hat{A}$  on a complex Hilbert space has mutually orthogonal eigenvectors.*

*Proof.* Let  $|x\rangle, |y\rangle$  be two eigenvectors of  $\hat{A}$  corresponding to distinct eigenvalues  $\lambda$  and  $\mu$ , respectively. Then

$$\mu \langle y | x \rangle = \langle \hat{A}y | x \rangle = \langle y | \hat{A}x \rangle = \lambda \langle y | x \rangle$$

Since  $\lambda$  and  $\mu$  are real, we must have that  $\langle y | x \rangle = 0$ .  $\square$

The eigenvalues of an observable correspond to the possible values that observable may take. As the eigenvectors span the space, they also define a basis of the Hilbert space. Thus we can represent all states as linear combinations of the eigenvectors of an observable. Suppose we have an observable  $\hat{A}$ , with eigenvalues  $\{\lambda_1, \dots, \lambda_n\}$  and corresponding eigenvectors  $\{|x_1\rangle, \dots, |x_n\rangle\}$  spanning the space, then when the system is in state

$$|\psi\rangle = a_1 |x_1\rangle + \dots + a_n |x_n\rangle \quad a_1, \dots, a_n \in \mathbb{C}$$

upon measuring  $\hat{A}$ , we will observe value  $\lambda_i$  with probability  $|\langle x_i | \psi \rangle|^2 = |a_i|^2$ . This is known as the *Born rule*.

States of a dynamic quantum mechanical system will change over time, and we will denote the state of the system at time  $t$  by  $|\psi_t\rangle$ . The time evolution of a quantum state is described by Schrödinger's equation

$$i\hbar \frac{d|\psi_t\rangle}{dt} = H(t) |\psi_t\rangle \quad (\text{B.1})$$

Where  $H(t)$  is an observable, known as the Hamiltonian, and describes the total energy of the system. This differential equation allows us to derive the evolution operator that transforms the initial state of a system,  $|\psi_0\rangle$ , into the state at time  $t$ .

$$|\psi_t\rangle = e^{\frac{-iH(t) \cdot t}{\hbar}} |\psi_0\rangle \equiv U(t) |\psi_0\rangle \quad (\text{B.2})$$

What is important to note about the operator  $U(t)$ , is that it is unitary, and thus it preserves inner products and norms, and is invertible. This means that, given the state at any point in time, and the time evolution operator, we can compute the state of the system at all previous and future points in time.

## B.2 Qubit systems

A *qubit* is the quantum analogue of the classical bit, and it is how information is encoded in quantum computing.

Qubits are states in the projective space  $\mathcal{HP}^1$ , where  $\mathcal{H}$  is a complex Hilbert space spanned by two, arbitrary, basis vectors, which we will denote  $|0\rangle$  and  $|1\rangle$ . These two vectors are the *basis states*, and any linear combination of the two

$$\alpha_0 |0\rangle + \alpha_1 |1\rangle$$

where neither  $\alpha_0$  or  $\alpha_1$  are 0, is known as a *superposition* of the basis states. The scalars  $\alpha_0, \alpha_1$  are known as the *amplitudes* of the basis states.

**Definition B.2.1.** (Tensor product) Let  $V$  and  $W$  be two finite-dimensional complex Hilbert spaces, with bases  $A = \{|a_0\rangle, \dots, |a_n\rangle\}$  and  $B = \{|b_0\rangle, \dots, |b_m\rangle\}$  respectively. The tensor product  $V \otimes W$  of these two spaces is the space spanned by the the  $n \cdot m$  elements of the form  $|a_i\rangle \otimes |b_j\rangle$ . Where  $\otimes : V \times W \rightarrow V \otimes W$  is the tensor product operator, and satisfies

$$1. (|v\rangle + |v'\rangle) \otimes |w\rangle = |v\rangle \otimes |w\rangle + |v'\rangle \otimes |w\rangle$$

2.  $|v\rangle \otimes (|w\rangle + |w'\rangle) = |v\rangle \otimes |w\rangle + |v\rangle \otimes |w'\rangle$
3.  $(a|v\rangle) \otimes |w\rangle = |v\rangle \otimes (a|w\rangle) = a(|v\rangle \otimes |w\rangle)$

For all vectors  $|v\rangle, |v'\rangle \in V$ ,  $|w\rangle, |w'\rangle \in W$ , and all scalars  $a \in \mathbb{C}$ .

We will also denote the tensor product of two vectors  $|x\rangle \otimes |y\rangle$  more simply by  $|xy\rangle$  or  $|x, y\rangle$ .

By taking the tensor product of two single qubit spaces, we have generate a new space with  $2^2$  elements, this being the space whose points are pairs of qubits, spanned by  $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ . We can continue iteratively in this manner to create spaces with states given by an arbitrary number of qubits.

Most of the elements of a tensor product space  $V_1 \otimes \dots \otimes V_n$  cannot be written in the form

$$|\psi\rangle = |v_1\rangle \otimes \dots \otimes |v_n\rangle$$

where  $|v_i\rangle \in V_i$ . In quantum mechanical systems, those states that can be written as a tensor product of elements of the underlying spaces are known as *separable*, the rest are *entangled*.

**Example B.2.1.** Consider the state  $|\psi\rangle = \frac{1}{2}(|000\rangle + |001\rangle + |110\rangle + |111\rangle)$ . This state may seem entangled with respect to the tensor product of three single qubit spaces, as there is no way to write this state in the form

$$(a_0|0\rangle + b_0|1\rangle) \otimes (a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle)$$

However, if we consider it in terms of the tensor product of one two-qubit space and a single qubit space we have

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

And is thus we see it is separable.

## B.3 Measuring a quantum system

**Definition B.3.1.** (Projector) Let  $\mathcal{H}$  be a finite-dimensional Hilbert space. Let  $\mathcal{H} = H_1 \oplus H_2$  be a direct sum decomposition of the space. This means all  $|v\rangle \in \mathcal{H}$  can be written in the form

$$|v\rangle = |v_1\rangle + |v_2\rangle \quad |v_1\rangle \in H_1, \quad |v_2\rangle \in H_2$$

An operator  $P_i : \mathcal{H} \rightarrow H_i$  that maps  $|v\rangle$  to  $|v_i\rangle$  is a projector operator onto the subspace  $H_i$ .

Consider the state space of a two qubit system, it is spanned by the set  $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ , and can be decomposed into the direct sum of a variety of smaller subspaces. For example the four spaces spanned by  $|00\rangle, |01\rangle, |10\rangle$  and  $|11\rangle$  respectively, or the two spaces spanned by  $\{|00\rangle, |01\rangle\}$  and  $\{|10\rangle, |11\rangle\}$  respectively.

Depending on the chosen decomposition, and on the chosen basis for each space, we will have different projection operators.

When a quantum state is measured, we measure with respect to a given basis and space decomposition. A state  $|\psi\rangle$  in the space  $\mathcal{H} = H_1 \oplus \dots \oplus H_k$  can be written in the form

$$|\psi\rangle = \sum_{i=1}^k \alpha_i |v_i\rangle \quad |v_i\rangle \in H_i$$

and, after being measured, with respect to this particular decomposition, it will collapse into the pure state

$$\frac{P_i |\psi\rangle}{|P_i |\psi\rangle|} = \frac{|v_i\rangle}{||v_i\rangle|}$$

with probability  $|P_i |\psi\rangle|^2 = ||v_i\rangle|^2 = |\alpha_i|^2$ .

In general in this thesis, when we refer to measuring a qubit system, it will be with respect to the standard basis, and a direct sum decomposition into single qubit spaces, unless explicitly stated otherwise.

Notice that since after measurement the state of the system is changed, thus the act of measuring itself is a transformation of the system, and thus it becomes critical to keep track of when and how the system is being measured.





# Bibliography

- [1] Sanjeev Arora and Boaz Barak. *Computational complexity : a modern approach*. eng. Cambridge ; Cambridge University Press, 2009. ISBN: 978-0-521-42426-4.
- [2] Robert. Bix. *Conics and cubics : a concrete introduction to algebraic curves*. eng. 2nd ed. Undergraduate texts in mathematics. New York: Springer, 2006. Chap. 11. ISBN: 9780387392738.
- [3] D. Brown. “On the Provable Security of ECDSA”. In: *Advances in Elliptic Curve Cryptography*. Ed. by Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. Vol. 317. London Mathematical Society Lecture Note Series. Cambridge University Press, 2005. Chap. 2, pp. 21–40. ISBN: 978-0-521-60415-4.
- [4] W Castryck et al. “CSIDH: An Efficient Post-Quantum Commutative Group Action”. eng. In: *Advances in Cryptology – ASIACRYPT 2018*. Lecture Notes in Computer Science 11274 (2018), pp. 395–427. ISSN: 0302-9743.
- [5] Wouter Castryck and Thomas Decru. *An efficient key recovery attack on SIDH (preliminary version)*. Cryptology ePrint Archive, Paper 2022/975. <https://eprint.iacr.org/2022/975>. 2022. URL: <https://eprint.iacr.org/2022/975>.
- [6] Andrew Childs, David Jao, and Vladimir Soukharev. “Constructing elliptic curve isogenies in quantum subexponential time”. eng. In: *Journal of mathematical cryptography* 8.1 (2014), pp. 1–29. ISSN: 1862-2976.
- [7] Bumkyu Cho. “Integers of the form  $x^2+ny^2$ ”. eng. In: *Monatshefte für Mathematik* 174.2 (2014), pp. 195–204. ISSN: 0026-9255.
- [8] David Deutsch and Roger Penrose. “Quantum theory, the Church–Turing principle and the universal quantum computer”. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117. DOI: 10.1098/rspa.1985.0070. URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1985.0070>.
- [9] Mika Hirvensalo. *Quantum computing*. eng. 2nd. ed. Natural computing series. Berlin: Springer, 2004. ISBN: 3-540-40704-9.
- [10] IBM. *Size considerations for public and private keys*. Accessed: 18.05.2023. Sept. 28, 2022. URL: <https://www.ibm.com/docs/en/zos/2.5.0?topic=certificates-size-considerations-public-private-keys>.

- [11] Shigeru. Iitaka. *Algebraic geometry : an introduction to birational geometry of algebraic varieties*. eng. Graduate texts in mathematics ; 76. Berlin: Springer, 1982. ISBN: 0-387-90546-4.
- [12] David Jao, Stephen D. Miller, and Ramarathnam Venkatesan. “Expander graphs based on GRH with an application to elliptic curve cryptography”. eng. In: *Journal of number theory* 129.6 (2009), pp. 1491–1504. ISSN: 0022-314X.
- [13] Ernst Kani. “The number of curves of genus two with elliptic differentials”. eng. In: *Journal für die reine und angewandte Mathematik* 1997.485 (1997), pp. 93–122. ISSN: 0075-4102.
- [14] Oak Ridge National Laboratory and U.S. Department of Energy. *Frontier - direction of discovery*. Accessed: 18.05.2023. URL: <https://www.olcf.ornl.gov/frontier>.
- [15] Eleanor Rieffel and Wolfgang Polak. *Quantum Computing - A Gentle Introduction*. The MIT Press, 2014. ISBN: 978-0-262-01506-6.
- [16] René Schoof. “Nonsingular plane cubic curves over finite fields”. In: *Journal of Combinatorial Theory, Series A* 46.2 (1987), pp. 183–211. ISSN: 0097-3165. DOI: 10.1016/0097-3165(87)90003-3. URL: <https://www.sciencedirect.com/science/article/pii/0097316587900033>.
- [17] Joseph H. Silverman. *The arithmetic of elliptic curves*. eng. Graduate texts in mathematics ; 106. New York: Springer, 1986. ISBN: 0-387-96203-4.
- [18] Joseph H. Silverman. *The arithmetic of elliptic curves*. eng. 2nd ed. Graduate texts in mathematics ; 106. New York: Springer, 2009. ISBN: 978-0-387-09493-9.
- [19] Simon Singh. *The code book : the secret history of codes and code-breaking cryptography*. eng. London: Fourth Estate, 1999. ISBN: 1-85702-889-9.
- [20] Andrew Sutherland. *18.783 - Elliptic Curves - Lectures*. Accessed: 20.05.2023. 2022. URL: <https://math.mit.edu/classes/18.783/2022/lectures.html>.
- [21] John Tate. “Endomorphisms of abelian varieties over finite fields”. eng. In: *Inventiones mathematicae* 2.2 (1966), pp. 134–144. ISSN: 0020-9910.
- [22] Allan Turing. “On computable numbers, with an application to the Entscheidungsproblem (Proc. Lond. Math. Soc., series 2 vol. 42 (1937), pp. 230–265) — A correction (ibid. vol. 43 (1937), p. 544–546)”. eng. In: *Mathematical Logic*. Elsevier B.V, 2001, pp. 9–56. ISBN: 9780444504234.
- [23] Lawrence C. Washington. *Elliptic curves : number theory and cryptography*. eng. Discrete mathematics and its applications. Boca Raton: Chapman & Hall/CRC, 2003. ISBN: 1-58488-365-0.