Cryptanalysis of factoring and the discrete logarithm problem and their ramifications on the smooth numbers and modular curves

Razvan Barbulescu

université de BORDEAUX

Cryptanalysis of factoring and the discrete logarithm problem
and their ramifications on the smooth numbers and modular curves

Razvan Barbulescu

Mémore présenté et soutenu le 8 juillet 2025 pour l'obtention de l'

Habilitation a diriger des recherches

(spécialité informatique)

Composition du jury :

Rapporteurs :	Jean-François Biasse, professeur à University of South Florida
	David Kohel, professeur à l'Université d'Aix-Marseille
	Fre Vercauteren, professeur à Katholieke Universiteit Leuven
Examinateurs :	Karim Belabas, professeur à l'Université de Bordeaux
	Cécile Dartyge, maîtresse de conférences à l'Université de Lorraine
	Ariane Mézard, professeure à Sorbonne Université

Contents

Chapter 1

Introduction

In this manuscript we present the connexions between a series of research works that we've conducted. In all of them we have the motivation to analyse the mathematical foundations of cryptography and hence to make possible secure and fast credit cards, internet communications, distant authentication to a machine, electronic signature of documents, etc. We have hence to analyse the hardness of a series of mathematical problems and, while doing so, we are reliant on various mathematical theorems. In some cases they exist in the literature and our works are new applications of these results. But in other cases we formulate and prove new mathematical statements, which have a mathematical relevance independent of the motivation we had to study them.

A landmark event in cryptography is the development of quantum computers, a new paradigm of computing which breaks a series of cryptosystems, called pre-quantum, while it leaves others, called post-quantum, at the same level of security with minor modifications. I launched the present research line in the years 2010-2015 when pre and post-quantum cryptography were both interesting. The preference for pre-quantum cryptography was based on the absence of post-quantum cryptography from the NIST recommendations and the estimation of the experts according to which pre-quantum cryptography was to be secure for at least twenty years, see for e.g. [Mos18] where the estimates of the period 2010-2015 are described.

In 2016, the NIST issued a report [CCJ+16] where it is estimated that post-quantum cryptography must be deployed alongside pre-quantum cryptography before 2036. Then, in the period 2016-2025 there was a lot of progress in the development of quantum computers. A measure of their size is the number of quantum bits (qubits) and the record evolved from 16 qubits in 2016 to 1433 qubits in 2024.

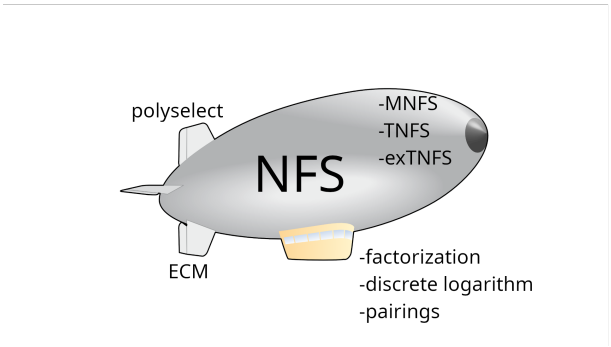


Figure 1.1: Illustration of the NFS algorithm, its variants, its cryptographic applications and its more fundamental ramifications.

In Figure 1 we try an analogy between pre-quantum cryptography, and especially an algorithm called NFS, and zeppelin airships: both of which had their time of glory and then fell in disgrace but left behind an important legacy. In Chapter 2 we present the number field sieve (NFS), an algorithm which has many steps, each of which having made the object of improvements of mathematical and algorithmic nature. In our works we presented new variants which change the asymptotic complexity and was part of the teams who obtained computation records. For example, in a record computation in a field with p^2 elements for a prime p the computation time was 160 times less than expected (see Table 2.2 for more details). In the illustration, the NFS algorithm is the main object of study which was improved and modified in many fashions (like a zeppelin airship, which was the object of variants (e.g. helium vs hydrogen) and technical improvements). In Chapter 3 we present the cryptographic applications of the NFS. It is the state-of-the art algorithm to attack the RSA and DSA cryptosystems, two of the most important solutions for TLS, a component of https. A newer variant of NFS, proposed before our works, extended the algorithm to attack a new type of cryptosystems known as pairing-based. All these cryptosystems are pre-quantum and (as the applications of the zeppelin airships are outdated) they will become disallowed after 2036 according to a NIST report [MPR+24]. In Chapters 4 and 5 we present the mathematical results which were formulated as ramifications of the NFS algorithm (in the same way the industry of zeppelin airships lead the foundations of fabric mechanics). Finally, in Chapter 6 we adopt the quantum paradigm and investigate a series of use cases taking profit among others of the machinery which was proposed for the pre-quantum cryptography.

1.1 Variants of the NFS algorithm

Let us introduce and summarize Chapter 2.

NFS is an algorithm from the Index Calculus family: in 1922 Kraitchik [Kra22] proposed an algorithm to factor integers. If one can find a non-trivial factor then one can completely factor by iterating the procedure, so we focus on the first task. In the same vein as in Lagrange's algorithm, one computes a random solution of the equation $x^2 \equiv y^2 \pmod N$, with uniform probability (in some variants under heuristics). If N is an odd number other than a prime power, then at least half of the solutions (x, y) are such that $x \equiv y$ modulo some prime factors of N and $x \equiv -y$ modulo the others. By computing $\gcd(x - y, N)$ thanks to the Euclid's algorithm one finds a non-trivial factor

of N. In a book for high-school students, Kraitchik proposed an algorithm where one enumerates the residues $x^2 \bmod N$ for $x = \lceil \sqrt{N} \rceil$, etc. Clearly, if the residue, seen as an element of $\mathbb{Z}$ is a square then one can immediately terminate the algorithm. However, the proportion of squares in an interval of length N is $1/\sqrt{N}$, so in the general case the residue is not an integer square. Kraitchik proposed to collect the values of those x for which $x^2 \bmod N$ are B-smooth, i.e. having all prime factors less than a bound B. The equations of the form

$$x^2 \equiv \prod_{p<B} p^{e(x,p)} \bmod N, \tag{1.1}$$

for some exponents $e(p)$, are called relations. When $\pi(B)$ relations are collected one solves a linear system and finds exponents $f(x) \in \{0,1\}$ such that

$$\forall p < B, \sum_{x \text{ relation}} f(x)e(x,p) \equiv 0 \bmod 2.$$

Then one has $x^2 \equiv y^2 \bmod N$ where

$$y = \prod_{p<B} p^{\frac{1}{2}\sum_{x \text{ relation}} f(x)e(x,p)},$$

The NFS algorithm considers two irreducible polynomials f and g in $\mathbb{Z}[x]$ which have a common root m modulo N. Let α_f (resp. α_g) be a complex root of f (resp. g). Without entering into details of algebraic number theory we note that under some assumptions on f and g which are easy to satisfy one collects polynomials $\phi \in \mathbb{Z}[x]$, called relations, such that

$$\begin{array}{rcl} \phi(\alpha_f) & = & \prod_{p(\alpha_f) \text{ small elements of } \mathbb{Q}(\alpha_f)} p^{e(\phi,p)} \\ \phi(\alpha_g) & = & \prod_{q(\alpha_g) \text{ small elements of } \mathbb{Q}(\alpha_g)} q^{e(\phi,q)}, \end{array} \tag{1.2}$$

for some integers $e(x,p)$ and $e(x,q)$.

Then one has an equation which is analogous to (1.1):

$$\prod_{p(\alpha_f) \text{ small elements of } \mathbb{Q}(\alpha_f)} p(m)^{e(\phi,p)} \equiv \phi(m) \equiv \prod_{q(\alpha_g) \text{ small elements of } \mathbb{Q}(\alpha_g)} q(m)^{e(\phi,q)} \bmod N \tag{1.3}$$

Our contribution concerns the modification of the algorithm. We proposed several modifications of the NFS:

- In a variant called the multiple number field sieve we replace the two polynomials f and g by a large number of polynomials; the idea was proposed by Coppersmith in a different context.
- In a variant called the tower number field sieve we proposed to replace the polynomials f and g, which in the classical variant belong to $\mathbb{Z}[x]$, with polynomials in $\mathbb{Z}(\iota)[x]$ for a complex number ι (e.g. $\mathbb{Z}[i][x]$ where $i^2 = -1$).
- In a practical improvement we proposed a new method to select the polynomials f and g which have smaller coefficients and degree and therefore speed up the relation collection. The method applies only when the NFS is applied to the finite fields with p^n elements when n has a given size relative to the bit size of p^n. In particular for cryptographic sizes the method applies the best for $n = 2$ and $n = 3$.
- In a variant called the extended number field sieve (exTNFS) we combined the TNFS and the Conjugation method of polynomial selection. This applies to a wide range of finite fields p^n, in particular $n = 4, 6, 8, 10$ and 12, which have important consequences in cryptography.

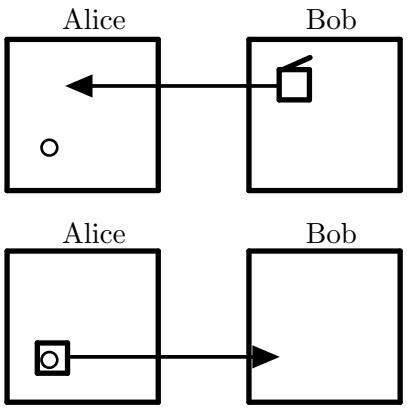


Figure 1.2: Illustration of a public key encryption method.

1.2 Cryptographic applications

Cryptography is divided as follows:

- Symmetric cryptography where two parties, called Alice and Bob, have a common secret and they use an encryption method to ensure the confidence of their communications. Other objectives include the so called hash functions.

- Asymmetric cryptography, or public-key cryptography, where Alice and Bob desire to establish a common secret key. This is called key encapsulation method (KEM) and can be done in two manners: 1) Alice selects a random key and sends it to Bob via a public key encryption method or 2) Alice and Bob communicate back and forth via the insecure channel until they reach a common key. Let us give more details:

    1. This can be done as in Figure 1.2 where Alice sends an encrypted message to Bob (see [DH76]). Bob makes public his public key and Alice uses it to encrypt her message. Finally, Bob uses his private key to decrypt. In the illustration the public key corresponds to an open padlock, encrypting corresponds to putting the message in a box and closing it with the padlock, while the private key corresponds to the key which allows to open the padlock again, which is kept (secret) by Bob.
    2. A second manner to establish a common secret key is that Alice and Bob communicate back and forth through the unsecure channel until they establish a key, this is called a key exchange method (see Figure 1.3 for an illustration). In the illustration, there exists a common source of identical balls of unknown weight which are publicly available, Alice and Bob have sources of identical balls whose weight are known only to their owners. In the first round of the key exchange method, Bob puts a common ball in a box, adds a ball of his own and sends the box to Alice. At the same time Alice puts a common ball in another box, adds a ball of her own and sends it to Bob. In the second round Alice and Bob add a ball of their own to the received box. The common secret is the weight of the obtained box, which is equal to the two resulted boxes.

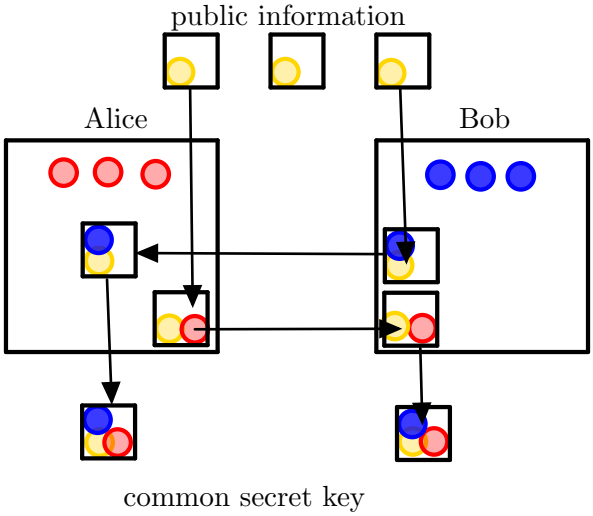


Figure 1.3: Illustration of a public key exchange method.

Other tasks of public key cryptography include the authentication and electronic signature.

Several realizations of the KEMs have been proposed but only three have been standardized by NIST until 2020 and deployed in TLS. The RSA cryptosystem is an asymmetric encryption method (see point 1. and Figure 1.2) and relies on the hardness of factoring integers; to have the largest security one uses integers $N = pq$, where p and q are two primes. Other methods based on the knapsack problem and error correcting codes were proposed in the same period (late 1970s) but they were much slower for the same security level and haven't been standardized before 2020. Since Shor's quantum algorithm has a variant for integer factorization, RSA is part of the pre-quantum cryptography. Recently NIST has selected asymmetric encryption methods based on lattices (ML-KEM) and on error correction codes, which are part of post-quantum cryptography.

The Diffie-Hellman (DH) key exchange method (see point 2. and Figure 1.3) can be used whenever one has a commutative group G acting on a set X; we denote the action of $a \in G$ on $x \in X$ by $[a]x$. The public information is an element $g \in X$. Alice draws a random secret $a \in G$ and Bob draws a random secret $b \in G$. In round 1 of the protocol, Alice computes $g_a := [a]g$ and Bob $g_b := [b]g$ and they make them public. In round 2 Bob computes $g_{ba} := [b]g_a$ and Alice computes $g_{ab} := [a]g_b$. Because G is commutative, one has $g_{ba} = g_{ab}$, so Alice and Bob have a common piece of information. There have been (partial) theorems on the equivalence between the problem of computing g_{ab} from g_a and g_b and the problem of computing g when given $[g]x$ and x. The latter is called the discrete logarithm problem in general, but sometimes one specified the group because its hardness is not stable by group isomorphism.

Here are three examples:

- $X = \mathbb{Z}/p\mathbb{Z}$ for a prime p (or more generally $\mathbb{F}_q^*$ for a prime power q). Note that X is a cyclic group, so the group $G = \mathbb{Z}/|X|\mathbb{Z}$ acts as follows: for any $a \in \mathbb{Z}/|X|\mathbb{Z}$ and $g \in X$,

$$[a]g = g^a \bmod p.$$

This is abreviated as DH for Diffie-Hellman. The problem of computing a when given g and g^a is called the finite field discrete logarithm problem, (finite field DLP or simply DLP) which is vulnerable to Shor's quantum algorithm and is therefore part of pre-quantum cryptography.

- Let $\mathbb{F}_q$ be a finite field and $a, b \in \mathbb{F}_q$. We call rational points of the elliptic curve $y^2 = x^3+ax+b$ (or more generally $y^2 = f(x)$) the set X of projective solutions in $\mathbb{F}_q$ of the equation. It is known that X is a commutative group and one sets as before $G = \mathbb{Z}/|X|\mathbb{Z}$. The action of $a \in \mathbb{Z}/|X|\mathbb{Z}$ on $g \in X$ is
$$[a]g = \underbrace{g + g + \cdots + g}_{a \text{ times}}.$$
This is abbreviated as "elliptic curve Diffie-Hellman" (ECDH). The construction generalizes to the Jacobian associated to hyperelliptic curves of genus 2, which are a competitor to elliptic curve cryptography. The problem of computing a when given g and $[a]g$ is called the elliptic curve discrete logarithm problem (ECDLP), an instance of the DLP so ECDH is part of pre-quantum cryptography.

- Let E be an elliptic curve and G the set of subgroups of E. It is known that for any subgroup $A \subset E$ there exists a unique elliptic curve E' and a rational map from E to E', called an isogeny, whose image is isomorphic to E/A. Then one sets X to be the set of isogenies starting from E, i.e. $\{E/A \mid A \text{ subgroup of } E\}$. The action of a subgroup A on an elliptic curve E/B is
$$[A](E/B) = E/(A + B).$$
One obtains the supersingular elliptic curve isogeny-based Dieffie-Hellman (SIDH), a competitive candidate for standardization as part of the post-quantum cryptography.

An important remark is that the electronic signature can be done with a modification of RSA, and hence based on a hardness assumption of the integer factorization, with a method called DSA which is based on the hardness of the DLP and on ECDSA which is based on the hardness of the ECDLP. Hence, before 2020 all the NIST-standardized electronic signatures were based on the factorization, the DLP or the ECDLP, being therefore vulnerable to Shor's algorithm. Alternative methods are now NIST-starndardized.

A more recent tool in public-key cryptography are bilinear maps, which allow to do tasks like zero-knowledge proofs (zknarks), identity-based encryption (IBE) and others. They are maps

$$\eta : E_1 \times E_2 \to \mathbb{F}_q^*,$$

where E_1 and E_2 are elliptic curves and $\mathbb{F}_q$ a finite field. Breaking the DLP in $\mathbb{F}_q^*$ or the ECDLP in one of the elliptic curves E_1 and E_2 compromises the security of the pairing.

The main contribution of my PhD thesis is to tackle the DLP (in finite fields). The only finite fields recommended by the NIST were $\mathbb{F}_p$ with p prime and $\mathbb{F}_{2^n}$ (and slightly slower $\mathbb{F}_{3^n}$) for an integer n. The latter case was less recommended (e.g. by ANSSI) and, to our knowledge, it had no industrial deployment. The improvements on the $\mathbb{F}_p$ case didn't have an impact on the key sizes. The situation was different on the case of small characteristic, i.e. the fields $\mathbb{F}_{2^n}^*$, where the quasi-polynomial algorithm [BGJT14] determined the elimination of these cryptosystems.

In the case of pairings, the fastest implementations are those where the target field $\mathbb{F}_q$ is such that $q = p^{12}$ for a prime p or $q = 2^n$ (and similarly 3^n) for an integer n, see e.g. [BLTMR+09],

	pairing	time (ms)	reference	comment
before	BN	0.85	[BGDM+10]	broken by NFS variants
attacks	binary field	1.87*	[BLTMR+09]	broken by the quasi-poly algo.
	BLS12	1.3		pre-quantum secure
after	BN	1.4	[GMT20]	pre-quantum secure
attacks	GMT8	1.5		pre-quantum secure

Table 1.1: Best record implementations of pairings before and after the attacks presented in this document. The case of binary pairings has a $*$ because they have inportant speed-ups on dedicated hardware, i.e. FPGA.

[BGDM+10] and [GMT20] and other works with similar results. Table 1.2 shows that better understanding of the NFS algorithm implied that the overall performances of pairing-based cryptography, rather than certain pairings, are slightly less attractive than previously believed. We go more into depth in Chapter 3.

1.3 Mathematical results of a more fundamental nature inspired by the NFS

1.3.1 Significance of the error term in the smoothness formula of binary forms

For any integer $n > 1$, $P^+(n)$ is the largest prime factor of n and by convention we set $P^+(1) = 1$. A y-friable integer (called y-smooth in a cryptographic context as in Section 2) is an integer such that $P^+(n) < y$. We call $\psi(x, y)$ the cardinality of the set

$$\Psi(x, y) = \{n \leq x \mid P^+(n) < y\}.$$

In a study of y-friable numbers, Dickman considered the ρ function defined as follows:

$$\begin{array}{ll} \rho(u) = 1 & 0 \leq u \leq u \\ u\rho'(u) + \rho(u-1) = 0 & u > 1. \end{array}$$

Thanks to a heuristic model, Dickman formulated the conjecture that

$$\psi(x, x^{1/u})/x \sim \rho(u), \tag{1.4}$$

when u is in an interval with respect to x which has to be specified. De Bruin proved the result for a certain range of values of y. In Chapter 4 we require the conjecture for smaller values of y: $\log y = (\log x)^c$ for a constant $0 < c < 1$. This theorem was proven a few decades later:

Theorem 1 (Canfield, Erdös, Pomerance 1983)**.** *Uniformly on the domain $x \geq 3$, $(\log\log x)^{5/3+\epsilon} \leq \log y \leq \log x$. one has $\psi(x, y)/x = \rho(u) = u^{-u+o(u)}$.*

An ideal is y-friable if its norm is y-friable and we denote $\psi_K(x, y)$ the cardinality of the set

$$\Psi_K(x, y) = \{\mathfrak{u} \text{ integer ideal in } K \mid \mathrm{N}(\mathfrak{u}) \leq x \text{ and } P^+(\mathrm{N}(\mathfrak{u})) < y\}. \tag{1.5}$$

Hildebrand and Tenenbaum proved that, uniformly on the domain of the CEP theorem

$$\frac{\psi_K(x, x^{1/u})}{\psi_K(x, \infty)} \sim \rho(u).$$

Saias extended Equation (1.4) and obtained an asymptotic development with a constant number I of terms. Finally, a theorem of Hanrot, Wu and Thenenbaum states that

$$\psi(x, x^{1/u})/x = \sum_{i<I} a_i \frac{\rho^{(i)}(u)}{(\log u)^i} + O_{I,\epsilon}\left(\frac{\rho^{(I+1)}(u)}{(\log y)^{I+1}}\right), \tag{1.6}$$

when $u \leq (\log y)^{3/5-\epsilon}$ and $a_1, \ldots, a_I$ are constants.

In cryptography, one requires results on finer sets of integers, the most notably being the images of polynomials and the cardinalities of elliptic curves.

Definition 2. *1. Let $f \in \mathbb{Z}[x]$ be a univariate polynomial of content 1 and call F its associated binary form $F(X,Y) = Y^{\deg f} f(X/Y)$. Let $\mathcal{K} \in \mathbb{R}^2$ be a compact set whose boundary is a continuous curve with piecewise continuous derivative. We set for any x and B*

$$\begin{array}{rcl} \Psi_F(x,B) & = & \{(X,Y) \in x\mathcal{K} \bigcap \mathbb{Z}^2 \mid P^+(F(X,Y)) < B\} \\ \Psi_F^{(1)}(x,B) & = & \{(X,Y) \in x\mathcal{K} \bigcap \mathbb{Z}^2 \mid \gcd(X,Y) = 1 \text{ and } P^+(F(X,Y)) < B\}. \end{array} \tag{1.7}$$

2. Let E be an elliptic curve with rational coefficients. For any x and B we set

$$\Psi_E(x,B) = \{p \leq x \text{ prime where } E \text{ has good reduction} \mid P^+(|E(\mathbb{F}_p)|) < B\}. \tag{1.8}$$

The cardinalities of these sets are denoted by $\psi_F^{(1)}(x,B)$ and $\psi_E(x,B)$.

We prove theorems on the asymptotic development of these quantities. The first order of approximation is the same as that of a random integer of the same size, which is not unexpected. The second order term, however is interesting because it allows to compare the polynomials (resp. the elliptic curves) to each other. We get more into depth in Chapter 4.

1.3.2 Progress on Mazur's program

Let E be an elliptic curve with coefficients in a number field k. For any integer N, we note $E[N] = \{P \in E(\overline{\mathbb{Q}}), [N]P = 0\}$. The field $\mathbb{Q}(E[N])$, generated by the x and y coordinates of the N-torsion points is called the N-torsion field and is a Galois extension of $\mathbb{Q}$. We call the representation mod N the embedding

$$\begin{array}{rccl} \rho_{E,N,\mathbb{Q}}: & \mathrm{Gal}(\mathbb{Q}(E[N])) & \to & \mathrm{Aut}(E[N]) \\ & \sigma & \mapsto & (x,y) \mapsto (\sigma(x), \sigma(y)) \end{array}$$

For any basis P and Q of $E[N]$ we have a non-canonical isomorphism $\mathrm{Aut}(E[N]) \simeq \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$:

$$\begin{array}{l} P^\sigma = aP + bQ \\ Q^\sigma = cP + dQ \end{array} \mapsto \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Mazur's program demands to find the set of all elliptic curves which have a given Galois representation. In Chapter 5 we explain that this is always possible to give a finite description of one of the following types:

- the list of curves if there are finitely many;

- formulas of rational fractions if the family is so-called of genus 0;

- an elliptic curve and a point whose multiples have the x coordinate equal to the j-invariants of the elliptic curves of the desired set.

See Chapter 5 for the particular cases of Mazur's program that we solved.

1.4 Use cases of quantum computing

1.4.1 A new paradigm of computation

Quantum computers are a technology of the future, but also a source of inspiration today for theoretic computer science. In a nutshell, a quantum computer of n quantum bits (qubits for short) stores information on a non-zero linear combination of elements of $\{0,1\}^n$:

$$|\psi\rangle = \sum_{i\in\{0,1\}^n} a_i|i\rangle.$$

To store the same information on a classical computer it requires the space of 2^n complex numbers at a precision equal to the error rate of the quantum computer. We don't have access directly to the full information of a quantum computer. Instead we can measure $|\psi\rangle$ and the output can be any of the vectors $i \in \{0,1\}^n$ following the law:

$$\text{Prob}\begin{pmatrix}\text{result of the}\\ \text{measure is } |i\rangle\end{pmatrix} = \frac{|a_i|^2}{\sum_{j\in\{0,1\}^n}|a_j|^2}. \tag{1.9}$$

A computational operation, also called quantum gate, is the application of any invertible linear map to the vector $|\psi\rangle$. To emulate its effect on a classical computer, before improvements, it requires to multiply a $2^n \times 2^n$ matrix with complex entries by a column vector. A universal set of gates formed by the four gates below:

$$H = \tfrac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \quad S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix} \quad \text{CNOT} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix} \quad T = \begin{pmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{pmatrix}. \tag{1.10}$$

Note that a universal set of gates is defined as a set which allows to approximate any gate up to a given error threshold.

The development of quantum computers is in its infancy: the largest quantum computers built until 2025 have between 100 and 1500 qubits. Error mitigation and error correction are needed to keep the errors below a certain threshold. The number of qubits required by an algorithm, called logical qubits, is multiplied by a factor between 5 and 1000 to obtain the number of qubits of a quantum computer which executes it, called physical qubits.

1.4.2 Two objectives of quantum computing

Quantum algorithms are a source of inspiration[1] for classical algorithms. For example the equivalence of the average case and worst case of the SVP was proven first by a quantum algorithm (see [Reg09]) and later by a classical one (see [Pei09]). The problem of computing short generators in ideals of multiquadratic number fields was first solved by a efficient quantum algorithm (see [BS16]) and later by an efficient classical one (see [BBdV+17]).

In [BB24] we proposed an extension of the DLP variant [EG24] of Regev's quantum algorithm to the Jacobian of hyperelliptic curves of large genera. We proved that the hardness of the DLP decreases when the genus increases. This mirrors a result of Adleman et al. [ADH94] in the classical paradigm.

In other cases the quantum algorithms allow to answer questions which are still open in the classical paradigm. For example, the literature of integer factorization and that of the DLP are mirrors of each other. Similarly, Shor's seminal article [Sho94] proposed quantum algorithms for the two problems in two sections which are apparently unrelated. A natural question is if the two problems reduce to each other or, in the negative case, how are they related. Kitaev [Kit96] answered the question in the quantum paradigm by the fact that the two problems are direct applications of the same problem, called the hidden subgroup problem.

Given a number field K, the unit group computation is an ingredient of the cryptanalysis of certain lattice-based cryptosystems. The cyclotomic fields have some of thee fastest arithmetic and are preferable for implementing these cryptosystems, so it is important to decide whether the unit group can be computed with fewer resources than in the general case. In [BP23] we proposed an algorithm which uses fewer qubits than the general purpose algorithm. This doesn't have any consequence on the time complexity in both the quantum and the classical paradigm.

The cryptosystems based on the elliptic curve discrete logarithm have a better speed when the elliptic curves are carefully selected. The NIST and Certicom recommendations contain a short list of elliptic curves, which are used by almost all users of ECDH and ECDSA. This corresponds to a negative answer to the question whether the ECDLP requires fewer resources if the tackled elliptic curve has small coefficients. In [BBP24] we proposed a variant of Regev's quantum algorithm to attack elliptic curves with small coefficients. This separates the two paradigms because there is no argument for a similar result on the classical computers.

1.4.3 Seeking the quantum advantage

Quantum computers having between 100 and 1000 qubits have been shipped to computing centers of public research organisms, e.g. Pasqal shipped a 100 qubit analogical computer to the GENCI French computing center and the digital computer Quandela is scheduled to be in exploitation before 2026. These computers are large enough to illustrate Shor's algorithm on toy examples of less than 10 qubits. It opens the question of a hybrid implementation of the NFS to factor integers of up to 70 decimal digits. For this, on a classical computer one uses the ECM algorithm whereas this is expected to be replaced some day by a quantum computer. In [Bar21] we ran an experiment to tune the parameters of the cado-nfs software for a hybrid computation.

The development of quantum computers is done in parallel with the implementation of algorithm, hence the applications show what are the most urgent features to be made possible. Ekerå's variant [Eke20] replaces Shor's algorithm, which executes a single run by numerous runs of less

[1] This can be compared with the geometric theorems which inspire results in algebraic geometry.

operations each. Similarly, Regev's algorithm has the same overall gate complexity as Shor's initial algorithm, but it is distributed in a large number of runs. In [BBP24] we proposed a certification technique and explicit details for an implementation of the algorithm on toy examples. We go more into depth in Chapter 6.

Chapter 2

New and old variants of the NFS algorithm

2.1 Number field sieve

The factorization and DLP in finite fields, central to public-key cryptology, have an asymptotic difficulty that depends on the number field sieve algorithm (NFS) or certain analogous algorithms. A notation adapted to the complexity of NFS is as follows. For any N, $c > 0$ and $0 \leq \alpha \leq 1$ we set:

$$L_N(\alpha, c) = \exp\left((c + o(1))(\log N)^{\alpha}(\log\log N)^{1-\alpha}\right).$$

Here N is the function's variable while α and c are constants which compare the complexities of various algorithms. When c is not important, we simply write $L(\alpha)$.

Every NFS variant which uses two polynomials can be modified to use multiple ones, as is it was mentioned in Section 2.5. In some cases this allows to reduce the constant c in the complexity $L_N(\alpha, c)$. For any variant its multiple field counterpart is denoted with an M prefix. All the variants which use two fields are particular cases of the extended number field sieve (exTNFS).

Let us recall the main stages of exTNFS when the target field is $\mathbb{F}_{p^n}$. As an illustration we use Figure 2.1.

1. Polynomial selection. Given a parameter η, chosen among the divisors of n, one selects a polynomial $h \in \mathbb{Z}[t]$ of degree η which is irreducible modulo p. We call R the maximal order of the number field of h and note that $R/pR \simeq \mathbb{F}_{p^\eta}$. Let ω be a root of h in $\mathbb{F}_{p^\eta}$. Then one selects two polynomials f and g in $\mathbb{Z}[t, x]$ so that $f(\omega, x)$ and $g(\omega, x)$, seen as elements of $\mathbb{F}_{p^\eta}[x]$, have a common factor $\varphi(x)$ which is irreducible of degree $\kappa := n/\eta$. In the particular case when $\gcd(\eta, \kappa) = 1$ we can take $f, g \in \mathbb{Z}[x]$.

2. Sieve. Given two parameters A and B, one collects all (up to sign) pairs $(a(t), b(t))$ of polynomials of degree $\leq \eta - 1$ in $\mathbb{Z}[t]$ or equivalently the tuples in the set $\{(a_0, \ldots, a_{\eta-1}, b_0, \ldots, b_{\eta-1}) \in [-A, A]^{2\eta} \mid a_0 \geq 0\}$, called sieving domain, so that N_f and N_g are B-smooth, where

$$N_f = \mathrm{Res}_t\left(\mathrm{Res}_x\left(\sum_{i=0}^{\eta-1} a_i t^i - x\sum_{i=0}^{\eta-1} b_i t^i, f(t, x)\right), h(t)\right)$$

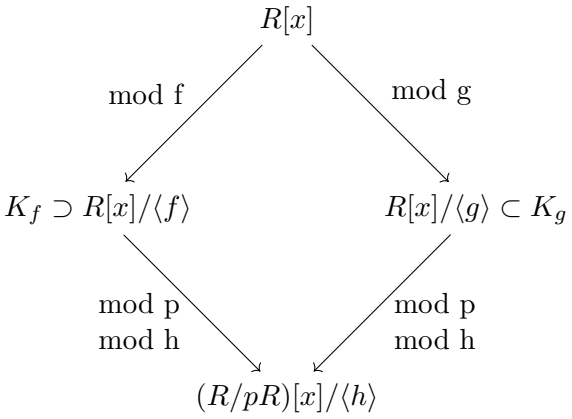


Figure 2.1: Commutative diagram of the exTNFS algorithm.

is the norm on the f side, and similarly for g instead of f. In order to emphasize the analogy with the simpler variants of NFS, we put $E = A^\eta$ which is a good approximation of the square root of the cardinality of the sieving domain.

3. Filtering. Unknowns which occur in a single relation are called singletons and are deleted together with the corresponding equation. Additionally, using elementary transformations of the matrix one can create new singletons. This leads to a smaller matrix and hence a faster resolution of the linear system.

4. Linear algebra step. One computes the right kernel of the sparse matrix obtained after the filtering using the Wiedemann algorithm or its block variants. The coordinates of the kernel vector are called virtual logarithms.

5. Individual logarithms. Given a generator g of $\mathbb{F}_{p^n}$ and an element h, compute the discrete logarithm $\log_g h$ using the virtual logarithms.

The history of NFS started with Pollard's seminar article [Pol93], which targeted factorization of a particular set of integers. It was rapidly adapted to factoring any integers and to solving the DLP in prime finite fields, i.e. $n = 1$. This simplifies considerably the algorithm: $h(t) = t$, i.e. $R = \mathbb{Z}$, the pairs of polynomials $(a(t), b(t))$ are here pairs of coprime integers (a, b) with $a > 0$ and the constraints on the polynomials f and g are simpler: $f, g \in \mathbb{Z}[x]$ irreducible having a common root modulo p, which has an elementary solution (see e.g. my thesis [Bar13]).

When the target field is $\mathbb{F}_{p^n}$ with $n > 1$ the analysis of the algorithm was done more than ten years later and it requires a careful choice of the number fields involved.

2.2 The tower number field sieve (TNFS)

To solve the DLP in $\mathbb{F}_{p^n}$, Schirokauer proposed what can be seen in the exTNFS framework as the case $\eta = n$. In [BGK15], we adapted the algorithm proposed by Schirokauer 15 years earlier. The objective is to use the polynomial selection method adapted to $\mathbb{F}_p$ for any field $\mathbb{F}_{p^n}$ with $n > 1$. To do this, we consider a polynomial $h \in \mathbb{Z}[x]$ of degree n such that p is inert in its number field; we

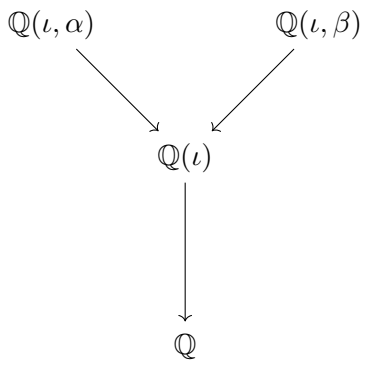


Figure 2.2: The number fields involved in the TNFS algorithm.

denote by ι (resp. ω) a root of h in $\mathbb{Q}[x]/\langle h \rangle$) (resp. $\mathbb{F}_p[x]/\langle h \rangle$). We then choose two polynomials f and $g \in \mathbb{Z}[x]$ which have a common root modulo p. Figure 2.2 illustrates the number fields involved in the TNFS algorithm. In the sieve step one enumerates pairs $(a(\iota), b(\iota))$ in $\mathbb{Z}[\iota]$.

Our main contribution was to present Schirokauer's variant in a modern language and to do the analysis, so that the available range of exponents n is computed.

The time cost of the algorithm is determined by the size of the norms $N_f = \mathrm{N}_{\mathbb{Q}(\iota,\alpha)/\mathbb{Q}}(\phi(\alpha))$ and $N_g = \mathrm{N}_{\mathbb{Q}(\iota,\beta)/\mathbb{Q}}(\phi(\beta))$. A precise analysis shows that

$$N_f = C(n)|h|^{n(\deg f - 1)}|f|^n \max(|a|, |b|)^{n \deg f}, \tag{2.1}$$

where $C(n)$ is a factor whose logarithm is negligible compared to the logarithm of the other factors, so its contribution is hidden in the $o(1)$ term of the algorithm's complexity. Since h can be equal to any lift to $\mathbb{Z}[x]$ of irreducible polynomials of degree n of $\mathbb{F}_p[x]$, we use the heuristic that a proportion of $1/n$ of the 3^n possible polynomials with coefficients in $\{-1, 0, 1\}$ are irreducible; we therefore use the heuristic that $|h| = 1$.

In order to keep a set of polynomials ϕ of the same cardinality as in the classical case, we must take $\| \phi \| = E^{1/n}$ where E is the bound on the norm of the polynomials $\phi \in \mathbb{Z}[x]$ used in the classical version. Note that $|f|$ depends only on p: there exists a constant c depending on the method of polynomial selection method such that $|f| = p^c$. We thus obtain

$$\log_2(N_f) = (1 + o(1)) \log_2((p^n)^c E^{\deg f}).$$

Thus we obtain the same binary size of N_f as when the field of the discrete logarithm is $\mathbb{F}_P$ for P a prime of the same bit size as p^n.[1]

2.3 Practical improvements and record computations

Joux, Lercier, Smart and Vercauteren [JLSV06] solved the same problem of extending the NFS from the case $n = 1$ to the case $n > 1$ in a different fashion than Schirokauer. Inside the exTNFS

[1] This algorithm was recognized as one of the top 3 papers at the Asiacrypt 2015 conference, and we were invited to submit a more comprehensive version to the Journal of Cryptology.

framework, one takes $\eta = 1$, i.e. $R = \mathbb{Z}$ in Figure 2.1 and selects polynomials f and g, which inside $\mathbb{F}_p[x]$ have a common divisor φ which is irreducible of degree n.

The construction of such polynomials has been the subject of much work. Some methods, e.g., conjugation [BGGM15], only adapt to certain exponents n for a given bit size of p^n. For example, the conjugation method allows to obtain a complexity of $L(1/3, \sqrt[3]{48/9})$ when $p = L_{p^n}(2/3, \sqrt[3]{12})$, which is better than the complexity obtained with the best polynomial selection method for $p = L_{p^n}(\ell_p)$ with $1/3 < \ell_p < 2/3$.

We sought to understand the complexity of the discrete logarithm problem beyond its asymptotic complexity. Indeed, hidden factors are important when establishing key sizes for use in cryptographic standards and can open new avenues of research that lead to changes in asymptotic complexity.

2.3.1 New methods of polynomial selection

The polynomial selection step takes a time that is negligible from an asymptotic point of view and, in computational records, takes between 0% and 10% of the total time, but which determines the speed of the rest of the calculations. Polynomial selection solves the following problem: given a prime p and three integers n, d_f, d_g such that $\min(d_f, d_g) \geq n$, find two polynomials f and g with coefficients such that their reductions modulo p have a common factor $\phi \in \mathbb{F}_p[x]$ that is irreducible of degree n.

Let's abbreviate $d_f = \deg f$ and $d_g = \deg g$. When $d_f = d_g = n$, the polynomial selection has an optimal solution (JLSV_1 method of Joux et al. [BGGM15, Sec 3.1]):

- f is a polynomial of degree n that is irreducible modulo p;
- $g = f + p$.

A variation proposed by the same authors aims to obtain two polynomials f and g such that $\max(\| f \|, \| g \|) < cp$ for a constant c. This is possible thanks to rational reconstruction, an application of the Lenstra-Lenstra-Lovasz (LLL) theorem: given a prime p and an integer $a \in [0, p-1]$, there exist two integers $u, v \in [1, \sqrt[4]{2}\sqrt{p}]$. such that

$$a \equiv \frac{u}{v} \pmod{p}.$$

We start by choosing two polynomials f_1 and f_2 such that $\deg f_1 = n$ and $\deg f_2 \leq n-1$. Then we choose an integer $a \in [2^{1/4}\sqrt{p}, p-1]$ such that $f := f_1 + af_2$ is irreducible modulo p. We calculate u and v such that $a \equiv u/v \pmod{p}$ and set $g := vf_1 + uf_2$. Note that f and g are equal modulo p up to a constant factor. The proof that $f \neq g$ follows from the fact that $(a, 1)$ is not a valid rational reconstruction of a because $a > 2^{1/4}\sqrt{p}$.

In [BGGM15], we used rational reconstruction to obtain a new method. As in the JLSV_1 method, we begin by choosing two polynomials f_1 and f_2 such that $f_1 + \sqrt{e}f_2$ is irreducible modulo p, where $\sqrt{e}$ denotes an integer lift of a solution to the equation $x^2 - e \equiv 0 \pmod{p}$ for the smallest prime $e = 2, 3, 5, \ldots$ such satisfying these conditions. Then, we apply rational reconstruction to $\sqrt{e}$:

$$\sqrt{e} \equiv \frac{u}{v} \pmod{p}.$$

Finally, we set $g = vf_1 + uf_2$ and $f = f_1^2 - ef_2^2$. Note that the polynomial g has the same degree and norm as in the JLSV_1 method. The polynomial f is such that $\deg f = 2n$ and $\| f \| = O(\log p)$.

$p = L_Q(\ell_p)$	$1/3 < \ell_p < 2/3$	best $\ell_p = 2/3$	$2/3 < \ell_p \leq 1$
TNFS [Sch00, BGK15]	none	none	64
NFS-JLSV [JLSV06]	128	64	64
NFS-(Conj and GJL) [BGGM22]	96	48	64
exTNFS [KB16]	48	48	64

Table 2.1: The complexity of each algorithms in the medium and large prime cases. Each cell indicates c if the complexity is $L_Q(1/3, (c/9)^{\frac{1}{3}})$. When $\ell_p > 1/3$ we assume that n is smooth.

field	decimal digits	ref.	comment	date
$\mathbb{F}_{p^2}$	160	[BGGM14]	first record of more than 100 digits	June 2
$\mathbb{F}_{p^2}$	180	160 times faster than [BGGM15] contrary to previous estimations and records	August 2014 the DLP in $\mathbb{F}_p$ (Bouvier et al. 2014)	
$\mathbb{F}_{p^3}$	156	(Joux et al. 2006)	beat a record of 120 digits	2015
$\mathbb{F}_{p^4}$	120	more than 100 digits.	first record for $n = 4$ of	2015

Table 2.2: DLP records in $\mathbb{F}_{p^n}$ for small $n > 1$

A detailed analysis is required to compare the conjugation to JLSV$_1$. When $p = L_{p^n}(2/3, 12^{1/3})$ we obtain a complexity of $L_{p^n}(1/3, \sqrt[3]{48/9})$, which is the lowest complexity over all non-small cases studied.

We conclude this section with a summary of the complexity of the variants of NFS in Table 2.1 and the records we obtained in Table 2.2.

2.4 The use of Galois isomorphisms

In the FFS algorithm, analogous to NFS, each pair (a, b) such that $F(a, b)$ and $G(a, b)$ are B-smooth allows to obtain $n - 1$ other pairs without computation. This reduces the cost of the sieve by a factor n. Furthermore, there are relations of the form $\log p = c \log \mathfrak{p}$ for any ideal $\mathfrak{p}$ in the factor base, where c is a constant. This reduces the linear system by a factor n and the cost of the linear algebra step by a factor n^2 (Wiedemann's algorithm has quadratic time complexity).

In [BGGM22], we describe a method for constructing pairs of polynomials that save a factor of n during the sieve step in logarithm computation in the field $\mathbb{F}_{p^n}$. This has resulted in a factor of n in computational records we performed for $n = 2$ and $n = 3$ (see Table 2.2), as well as in Laurent Grémy's record for $n = 6$.

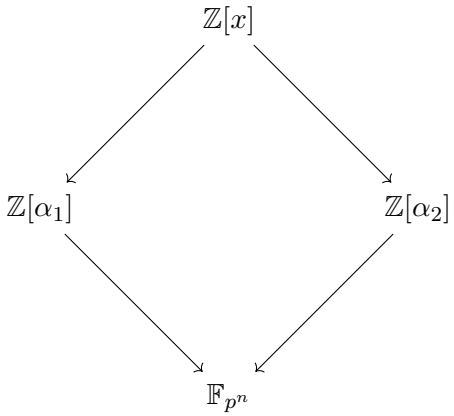


Figure 2.3: Commutative diagram of the NFS algorithm.

2.5 The Multiple number field sieve (MNFS)

In this section we denote by f_1 and f_2 the two polynomials f and g used in NFS. If α_1 and, respectively, α_2 are roots of f_1 and f_2 in their number fields, then any polynomial ϕ in the set considered such that $\mathrm{N}_{\mathbb{Q}(\alpha_1)}(\phi(\alpha_1))$ and $\mathrm{N}_{\mathbb{Q}(\alpha_2)}(\phi(\alpha_2))$ are B-smooth. In its classical version, the number field sieve obtains multiplicative relations using the diagram in Figure 2.3. Recall that the algorithm consists of enumerating the polynomials $\phi(x) \in \mathbb{Z}[x]$ of degree and coefficients bounded in absolute value by some given parameters. If $\mathbb{Z}[\alpha_1]$ and $\mathbb{Z}[\alpha_2]$ are factorial rings (UFDs), one writes $\phi(\alpha_1) = \prod_i p_i(\alpha_1)^{u_i}$, $\phi(\alpha_2) = \prod_j p_j(\alpha_1)^{v_j}$ and one obtains

$$\prod_i \overline{p_i(\alpha_1)}^{u_i} = \overline{\phi(\alpha_1)} = \overline{\phi(\alpha_2)} = \prod_j \overline{p_j(\alpha_1)}^{v_j}.$$

In the general case, the computation of the equations is different, but the input data are the same: a polynomial $\phi(x)$ for which the two norms are B-smooth.

In [BP14], we showed that in the case of medium characteristic, we can obtain more polynomials of the same degree and coefficient size by setting $f_i = \mu_i f_1 + \nu_i f_2$ with μ_i and ν_i rational numbers. We thus obtain a new commutative diagram: Given a polynomial ϕ, any pair (f_i, f_j) such that $\mathrm{N}_{\mathbb{Q}(\alpha_i)}(\phi(\alpha_i))$ and $\mathrm{N}_{\mathbb{Q}(\alpha_j)}(\phi(\alpha_j))$ are B-smooth gives an equation. This increases the probability of success by a factor of $V(V-1)/2$ compared to the classical variant with the same parameters. The complexity analysis requires modifying the smoothness bound B and we obtain a complexity of the form $L(1/3, c)$ for a value of c lower than that of the classical variant. Note that in the case of factorization, Don Coppersmith proposed a variant where, contrary to our variant, the pairs of polynomials used are of the form (f_1, f_i) with $i = 2, 3, \ldots$. Indeed, in that case, the polynomials f_1 and f_2 do not have the same degree, so that the linear combinations have norms greater than the best of the two polynomials.

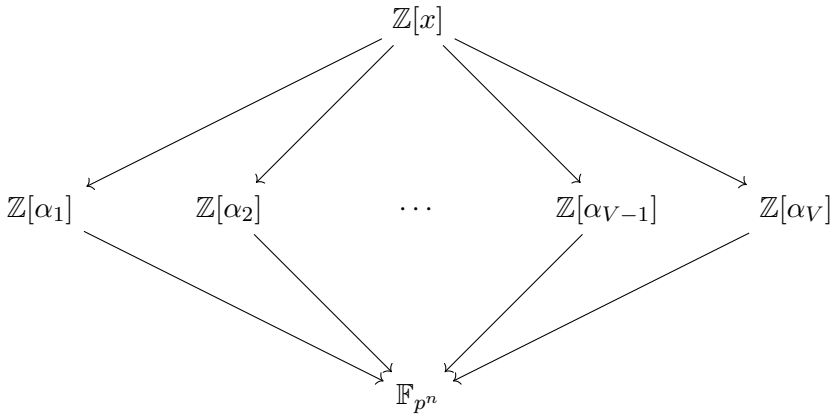


Figure 2.4: Commutative diagram of the MNFS algorithm.

Chapter 3

Updating key sizes of pairings

Pairings are a mathematical tool which has been known to cryptographers for a long time and which switched sides during its history. If in the early 90's it was on the attacker's side, it is now used to create secure cryptologic protocols.

Let E be an elliptic curve defiined over a fiinite fiield $\mathbb{F}_q$, r an integer number, P a point of order r and μ an r-th root of unity in the algebraic closure $\mathbb{F}_q$. Note that $\langle P\rangle \simeq \mathbb{Z}/r\mathbb{Z}$. The Weil pairing (restricted to the subgroup generated by P) is defined in an abstract way by the map

$$\begin{array}{rccc} & e : \langle P\rangle \times \langle P\rangle & \to & \langle \mu\rangle \\ \forall (a,b) \in \mathbb{Z}/r\mathbb{Z} & ([a]P, [b]P) & \mapsto & \mu^{ab}. \end{array} \tag{3.1}$$

Two properties of the Weil pairing are direct:

- bilinearity: for all $a, b, a', b' \in \mathbb{Z}/r\mathbb{Z}$ one has

$$\begin{array}{rcl} e([a+a']P, [b]P) & = & e([a]P, [b]P) \cdot e([a']P, [b]P) \\ e([a]P, [b+b']P) & = & e([a]P, [b]P) \cdot e([a]P, [b']P); \end{array}$$

- non-degenerance: for any $a \in \mathbb{Z}/r\mathbb{Z}\backslash\{0\}$, there exists $b \in \mathbb{Z}/r\mathbb{Z}$ so that

$$e([a]P, [b]P) \neq 1$$

  and similarly with the roles of a and b inverted.

Weil's pairings are unique up to the choice of the generator μ (see III.8 in [Sil09]).

3.1 Families of pairings

Freeman, Scott and Teske [FST10] made a taxonomy of known pairing-friendly families of elliptic curves. Given a bit size and an embedding degree k, most of them are constructed in two steps:

- one selects a prime power q of prescribed bit size and an integer t so that any elliptic curve over $\mathbb{F}_q$ of trace t has embedding degree k and its cardinality has a large prime factor r;
- one uses the CM method [AM93], which, given a prime power q and an integer t, allows to construct elliptic curves over $\mathbb{F}_q$ of trace t.

The CM method has complexity $O(D^{1+\epsilon})$ where D is the unique integer so that $(4q - t^2)/D$ is a perfect square. This imposes that we fix D in advance: it will be either small or will have common factors with q. By definition $\#E(\mathbb{F}_q) = q + 1 - t$ so we ask the existence of a prime r so that $q + 1 - t \equiv 0 \pmod{r}$. Finally, the property that k is the embedding degree of the curve is equivalent to $\Phi_k(q) \equiv 0 \pmod{r}$. We summarize the conditions on the output of the first step as follows:

$$\begin{array}{ll} \text{CM-1.} & \Phi_k(t-1) \equiv 0 \pmod{r} \\ \text{CM-2.} & q + 1 - t \equiv 0 \pmod{r} \\ \text{CM-3.} & \exists y, 4q = Dy^2 + t^2. \end{array}$$

Several methods of solving the CM system have been proposed:

- Supersingular curves. One takes $t = 0$ and $D = q$ so that the CM-3 condition is satisfied for any values of k and r. Although this value of D is large, this is a particular value where the CM method is fast. Note however (see [Bar16, Prop 1]) that this solution is possible only for embedding degree 2, which is too small for cryptography.

- Pinch-Cocks [CP01]. One starts by replacing Equation CM-2 with

$$\text{CM-2'} \quad Dy^2 + (t-2)^2 \equiv 0 \pmod{r},$$

  so we obtain an equivalent system CM-1, CM-2' and CM-3. One selects $r \equiv 1 \pmod{k}$ and $(\frac{-D}{r}) = 1$. This implies that CM-1 is satisfied by setting t equal to a root of the cyclotomic polynomial and one can solve CM-2' for y. Finally one sets $q = (Dy^2 + t^2)/4$. The drawback is that one doesn't control the relative size of the parameters.

- Dupont-Enge-Morain [DEM05]. Once again we start by replacing Equation CM-2 by Equation CM-2'. Then we see Equations CM-1 and CM-2' as a system which has to be solved with $y, t \in \mathbb{F}_r$:

$$\begin{cases} \Phi_k(t-1) = 0 \\ Dy^2 + (t-2)^2 = 0. \end{cases}$$

  The key ingredient of the resolution is the resultant. Note however that this method has the drawback that the number of curves constructed is small (see the LogJam attack [ABD+15] for a way to take profit of this fact).

- Sparse families (e.g. MNT [MNT01]). The following construction is possible for all integers k so that $\varphi(k) = 2$, i.e. $k = 3$, 4 and 6, but for simplicity we present only the case $k = 3$. We set $r = \Phi_k(t-1)$ so that Equation CM-1 is satisfied. Next we set $q = r + t - 1$, which satisfies CM-2. The method was generalized by Freeman when $\varphi(k) = 4$ but cannot be generalized further (see [Bar16, Prop 2]). The drawback is hence that k is limited to a very small set.

- Complete families (e.g [BN05]). Once again we replace Equation CM-2 by CM-2'. Then we set r equal to a polynomial $r(x)$ whose number field contains $Q(\sqrt{-D}, \zeta_k)$ for a kth root of unity ζ_k. This translates into

  1. Φ_k is totally split modulo $r(x)$;
  2. $x^2 + D$ is totally split modulo $r(x)$.

  Next we take t to be a polynomial $t(x)$ so that $\Phi_k(t(x)) \equiv 0 \pmod{r(x)}$. Since Equation CM-2' factors we can set $y(x) = t(x) \cdot \frac{t(x)}{\sqrt{-D}}$ where $\frac{1}{\sqrt{-D}}$ is a polynomial $z(x)$ in $\mathbb{Q}[x]$ so that

JLSV	NFS-Conj	Joux-Pierrot	exTNFS-Conj	SexTNFS-JP
[JLSV06]	[BGGM15]	[JP13]	[KB16]	[KB16]
128	96	64	48	32

Table 3.1: Complexity of several variants of the NFS which apply to the fields $\mathbb{F}_{p^k}$ when p is special and middle-sized. It is expressed by the constant c so that the complexity is $L(1/3, (c/3)^{1/3})$.

$Dz^2 - 1 \equiv \pmod{r(x)}$. Finally we set $q(x) = \frac{1}{4}(Dy(x)^2 + t(x)^2)$. The advantage of this method is that pairing-friendly curves can be generated on the fly by evaluating r and q at integer values x. The drawback is that the primes q are suited for the SNFS.

- Other families like Menezes-Köblitz are less competitive than the previous ones.

3.2 The variants of NFS suited for primes of special form: STNFS and SexTNFS

In the context of the NFS algorithm, an integer N is special if there exists a polynomial $P \in \mathbb{Z}[x]$ of small degree and coefficients of absolute value less than $\log N$ such that $N = P(u)$ for some $u \in \mathbb{Z}$. Historically, the special numbers were important because they include Fermat's numbers and the integers of low Hamming or NAF weight. In cryptography, they play an important role because the finite fields whose characteristic is special have a very fast arithmetic (see e.g; the mpfq documentation [GT07]). When the target field is prime, the direct solution is to take $f(x) = P(x)$ and $g(x) = x - u$.

Joux and Pierrot [JP13] proposed a method to tackle the fields $\mathbb{F}_{p^k}$ with $k > 1$. One selects a polynomial $S(x) \in \mathbb{Z}[x]$ of degree k such that $g(x) := S(x) - u$ is irreducible modulo p. Then one sets $f(x) = P(S(x))$.

In a precise estimation of the norm size (see [KB16]) we noted that for large values of k the degree of f is very large. When k is composite, i.e. $k = \kappa\eta$ for two integers κ and η, we proposed an alternative called SexTNFS: to combine the Joux-Pierrot polynomial selection with the exTNFS algorithm. To fix ideas we assume that $\gcd(\kappa, \eta) = 1$ but SexTNFS can be implemented for any composite k (see [KJ17]). One selects a polynomial $h(t) \in \mathbb{Z}[x]$ of degree η which is irreducible modulo p. One applies the Joux-Pierrot method to the field $\mathbb{F}_{p^\kappa}$, obtains a polynomial $S(x)$ and sets $f(x) = P(S(x))$ and $g(x) = S(x) - u$.

In Table 3.1 we recall the complexity of several variants of the NFS when k is composite and p is middle-sized i.e. $k^{1/2} < \log p < k^2$.

3.3 NFS applied to the pairing-friendly curves

The SexTNFS variant of NFS (see 3.2) was proposed after the security estimation of pairings done in [Len01].

In [BD19] we proposed a cost model, which is common to all variants of NFS:

$$\text{cost} = \frac{2B}{\mathcal{A}\log B}\rho\left(\frac{\log_2 N_f}{\log_2 B}\right)^{-1}\rho\left(\frac{\log_2 N_g}{\log_2 B}\right)^{-1} + c\frac{B^2}{\mathcal{A}^2(\log_2 B)^4}, \tag{3.2}$$

Family	η	h	g	ω	$\mathcal{A}$
BN, BLS12	6	Φ_7	x^2-u+t	7	6
KSS16	16	Φ_{17}	$x-u$	17	16
KSS18	18	Φ_{19}	$x-u$	19	18

Table 3.2: Values of η and f, g, h to tackle the underlying DLP instances of the BN, BLS12, KSS16 and KSS18 pairings at the 128 bit security level.

where $c = 2^7(\log_2 e)^2$.

We compared all the values of the parameters and found that the best algorithms are as in Table 3.3. In particular, BN-254 which was previously evaluated at 128 bits of security has 100 bits of security. In [BEMG20] we made the cost estimation for over 200 families of pairings.

Chapter 4

Analytic number theory applied to binary forms and elliptic curves

4.1 Smooth numbers

In Sections 4.2 and 4.3 we present results on the smoothness probabilities $\psi_F^{(1)}(x,B)/\psi_F^{(1)}(x,\infty)$ and $\psi_E(x,B)/\psi_E(x,\infty)$.

4.2 Rigorous analysis of polynomial selection stage of NFS

A commonly used heuristic in cryptography states that the B-smoothness probability of the images of a given binary form is the same as the B-smoothness probability of a random integer of the same size. Note that if the binary form F factors as $F = F_1F_2$, for any (X,Y) and for any parameter B, $F(X,Y)$ is B-smooth if and only if $F_1(X,Y)$ and $F_2(X,Y)$ are both B-smooth. This is the case of the NFS for factorization and discrete logarithms.

Consider the smoothness probability of the images of quadratic polynomials of discriminant $D < 0$ such that D is a fundamental discriminant. Let K be the number field of F. Note that ψ_K (see Equation 1.5) coincides with ψ_F (see Equation 1.7). To take into account the difference between ψ_F and $\psi_F^{(1)}$, we recall that a principal ideal of K is one which has no prime inert factor and we define

$$\Psi_K^{(1)}(x,B) = \{\mathfrak{u} \text{ principal integer ideal in } K \mid \mathrm{N}(\mathfrak{u}) \leq x \text{ and } P^+(\mathrm{N}(\mathfrak{u})) < B\}$$

and $\psi_K^{(1)}$ its cardinality. In [BL17] we proved

$$\psi_K^{(1)}(x,x^{1/u})/\psi_K^{(1)}(x,\infty) \sim \rho(u).$$

The second issue we tackle is the polynomial selection. A heuristic tool used in practice is Murphy's α which assigns a real value to all irreducible polynomials f. It is the sum of a series:

$$\alpha(f) = \sum_{p \text{ prime}} \alpha_p(f),$$

where $\alpha_p(f)$ is an expression accounting for the roots of f modulo the powers of p; the more roots the smaller value of α_p. Several questions are important for this function:

Q1 Does the series defining α converge for every irreducible polynomial?

Q2 How many terms have to be added to obtain the sum of the series at a given precision?

Q3 Is it true, for any pair of irreducible polynomials f_1 and f_2, if $\alpha(f_1) < \alpha(f_2)$ then, uniformly for x and B in a domain to be specified, $\psi_{F_1}(x, B) > \psi_{F_2}(x, B)$?

Q4 If one considers all the polynomials of a given degree and with coefficients up to a given bound, what is the probability to decrease the value of α by more than a given constant if one continues the search indefinitely? Equivalently, what is the cost of the polynomial selection if the objective is to optimize α up to a given additive constant?

In the following we present how the four questions were answered in [BL17].

Definition 3. *Let p be a prime. The average p-adic valuation of the values $F(n_1, n_2)$ with coprime $(n_1, n_2) \in x\,\mathcal{K} \bigcap \mathbb{Z}^2$ is given by*

$$\mathrm{cont}_p(f, \mathcal{K}) := \lim_{x\to\infty} \frac{\sum_{(n_1,n_2)\in x\,\mathcal{K}\bigcap\mathbb{Z}^2, \gcd(n_1,n_2)=1} \mathrm{val}_p\, F(n_1, n_2)}{|\{(n_1, n_2) \in x\,\mathcal{K} \bigcap \mathbb{Z}^2 \mid \gcd(n_1, n_2, p) = 1\}|}$$

with the convention $\mathrm{val}_p(0) = 0$.

$$\alpha_p(f, \mathcal{K}) = (\log p)\left(\frac{1}{p-1} - \mathrm{cont}_p(f, \mathcal{K})\right).$$

$$\alpha(f, \mathcal{K}, z) = \sum_{p\le z} \alpha_p(f, \mathcal{K}).$$

We call $n_p(f)$ the number of degree-1 prime ideals above p in the number field of f. Then, for all $p \nmid \mathrm{Disc}(f)$ (see [BL17, Prop 2.3]), one has:

$$\alpha_p(f, \mathcal{K}) = \frac{\log p}{p-1}\left(1 - n_p(f)\frac{p}{p+1}\right).$$

For any number field K we set

$$R_K(t) := \sum_{\substack{\mathfrak{p} \text{ prime ideal of degree } 1 \\ \mathrm{N}(\mathfrak{p}) \le t}} \log \mathrm{N}(\mathfrak{p}).$$

Up to a series which converges like $\sum_p \frac{1}{p^2}$, the series defining α converges like $(R_{\mathbb{Q}}(t) - R_K(t))/t$. Lagarias and Odlyzko (1977) considered the largest real zero $0 < \beta(K) < 1$ of ζ_K if it exists and $1/2$ otherwise. They recall a bound on $\beta(K) < 1$ as an expression of the degree d_K and the discriminant $\mathrm{Disc}(K)$ of K. They proved that

$$\left|\frac{R_K(t)}{t} - 1 + \frac{1/\beta(K)}{t^{1-\beta(K)}}\right| \le \exp\left(-cd_K^{-1/2}(\log t)^{1/2}\right).$$

Hence $(R_K(t) - R_{\mathbb{Q}}(t))/t$ converges in a sub-exponential manner. This answers question Q1. It also settles question Q2 but the convergence speed is too slow to be used in cryptography.

The same authors proved a stronger result for any number field K where the Riemann Hypothesis holds for ζ_K:

$$\left|\frac{R_K(t)}{t} - 1\right| \leq a_K \frac{(\log t)^2}{t^{1/2}},$$

for an effectively computable constant a_K. Now, when K is the number field of a polynomial f, when GRH holds for $\zeta_{\mathbb{Q}}$ and ζ_K, the rest of $\alpha(f)$ has an explicit bound $O(\frac{1}{\sqrt{t}})$. We noted that this answers question Q2 in a usable manner although it has the drawback that it is conditional under the GRH.

To study question Q3 we restrict to the case when f defines an irreducible quadratic field and has a fundamental discrimimant. We need an asymptotic development of $\psi_f^{(1)}(x, B)/\psi_f^{(1)}(x, \infty)$, which depends on f. Since at the first approximation the expression is $\rho(u)$, which is independent on f, we computed the development at the second order.

Theorem 4 (Th 1.1 in [BL17]). *Let $F(X_1, X_2) \in \mathbb{Z}[X_1, X_2]$ be an irreducible quadratic form such that* Disc(F) *is negative and fundamental. Let $\mathcal{K}_F$ be the compact defined by*

$$\mathcal{K}_F = \{(x_1, x_2) \in \mathbb{R}^2 \mid |F(x_1, x_2)| \leq 1\}.$$

Then, there exists $\kappa > 0$ such that, for any $\epsilon > 0$ and uniformly in the domain

$$x \geq 3, \quad \exp\left((\log\log x)^{5/3+\epsilon}\right) \leq y \leq x^2(\log x)^{-\kappa},$$

we have

$$\frac{\psi_F^{(1)}(x\,\mathcal{K}_F, y)}{\psi_F^{(1)}(x\,\mathcal{K}_F, \infty)} = \frac{\psi(x^2 e^{\alpha(f)}, y)}{\psi(x^2 e^{\alpha(f)}, \infty)}\left(1 + O\left(\frac{\log^2(u+1)}{\log^2 y}\right)\right),$$

where $\alpha(f)$ is Murphy's α (see Definition 3).

The proof is relatively direct. Let K be the number field of f, ζ_K its zeta function and

$$\forall s \in \mathbb{C}, \quad \mathcal{F}_K(s) = \frac{\zeta_K(s)}{\zeta_{\mathbb{Q}}(d_K s)}.$$

We set

$$\gamma_0(K) = \frac{\mathrm{Res}_{s=1}\zeta_K}{\zeta_{\mathbb{Q}}(d_K)} \quad \text{and} \quad \gamma_1(K) = \frac{\frac{\partial \mathcal{F}_K(s)}{\partial s}}{\mathcal{F}_K(s)}\gamma_0(K).$$

We make explicit the constants a_0 and a_1 in Equation (1.6) and obtained (see Eq (17) in [BL17]):

$$\psi_F^{(1)}(x, y)/(6x/\pi^2) = \rho(u) + \frac{\gamma_1(K)}{\gamma_0(K)}\frac{\rho'(u)}{\log y} + O\left(\rho(u)\left(\frac{\log(u+1)}{\log y}\right)^2\right). \tag{4.1}$$

Given a constant c one also has

$$\psi(xe^c, y)/(xe^c) = \rho(u) + c\frac{\rho'(u)}{\log y} + O\left(\rho(u)\left(\frac{\log(u+1)}{\log y}\right)^2\right). \tag{4.2}$$

One balances Equations (4.1) and (4.2) by setting $c = \alpha(f)$ as in Definition (3)

$$\alpha(f) = \frac{\frac{\partial \mathcal{F}_K(s)}{\partial s}}{\mathcal{F}_K(s)} - (\gamma - 1). \tag{4.3}$$

In [Lam15], Lamzouri defined $\gamma_K := c_1(K)/c_0(K)$ where

$$\gamma_K(s) = \frac{\alpha_K}{s-1} + \sum_{i=1}^{\infty} \gamma_i(K)(s-1)^i.$$

When injecting this equation in the definition of $\mathcal{F}_K$ one finds that

$$\gamma_K = \alpha(f) + c,$$

for an explicit value of the constant c.

We conclude that Q4 was answered by Lamzouri:

Theorem 5 (Th 1.3 in [Lam15]). *Let x be large and let $\mathcal{F}(x)$ be the set of all fundamental discriminants D such that $|D| \leq x$. There exist two explicitely computable constants A, C such that uniformly in the range $1 \ll \tau \ll \log\log x - 2\log\log\log x - C$, we have*

$$\frac{1}{|\mathcal{F}(x)|}\left|\{D \in \mathcal{F}(x) \quad | \quad |\gamma_{\mathbb{Q}(\sqrt{D})}| > \tau\}\right| \leq \exp\left(-\frac{e^{\tau - A}}{\tau}\right).$$

This offers a precise estimation of the number of polynomials to by enumerated during the polynomial selection phase of NFS.

4.3 Rigorous analysis of curve comparison for ECM

Chebyshev studied the so-called "prime races". For $a = 1$ and $a = 3$ one sets

$$\pi_a(x) = |\{p \text{ prime} \leq x \mid p \equiv a \pmod 4\}|. \tag{4.4}$$

Indeed, Dirichlet proved that

$$\pi_1(x) \sim \pi_3(x) \sim \frac{x}{2\log x}$$

but an asymptotic development at a higher order could a priori mean that one is always larger than the other.

Chebyshev's races can be naturally generalized to elliptic curves. For any elliptic curve E with rational coefficients one sets

$$\pi_E(x) = |\{p \text{ prime} \leq x \mid |E(\mathbb{F}_p)| \text{ is prime}\}|. \tag{4.5}$$

If E_1 and E_2 are two elliptic curves one can compare the sets $\pi_{E_1}(x)$ and $\pi_{E_2}(x)$. Köblitz [Kob88] conjectured that the ratio $\pi_{E_2}(x)/\pi_{E_2}(x)$ is bounded between two explicit constants and one can also study the difference (see [Coj17] for a review).

Pomerance and Sorenson [PS95] considered an extension of the primes races. Let a be an integer, let $y = y(x)$ be a positive function of x and set

$$\psi_a(x, y) = \{p \text{ prime} \leq x \mid P^+(p - a) < y\} \tag{4.6}$$

Pomerance gave equivalents for $\psi_1(x, y)$ and $\psi_{-1}(x, y)$ for a function y such that x/y is bounded. Jie Wu et al. [LWX20] extended his results to smaller values of y under the Elliott-Halberstam conjecture.

In [BJ24] we extend the races to the smooth values of elliptic curves. Indeed, let E be an elliptic curve, $y = y(x)$ a positive function and set

$$\psi_E(x, y) = \left|\{p \leq x \text{ prime where } E \text{ has good reduction } \mid P^+(|E(\mathbb{F}_p)|) < y\}\right|. \tag{4.7}$$

Then we compare $\pi_{E_1}(x, y)$ and $\pi_{E_2}(x, y)$ for a pair of elliptic curves E_1 and E_2. Consider the following questions.

Q1 Let E_1 and E_2 be two elliptic curves with rational coefficients. Can one define a domain of pairs (x, y) so that, uniformly on the domain,

$$\psi_{E_1}(x, y) > \psi_{E_2}(x, y)?$$

In this case we say that E_1 is more ECM-friendly than E_2.

Q2 To some extent, the prime race above is related to Chebotarev's density theorem applied to $\mathrm{Gal}(\mathbb{Q}(i))$ where i is a primitive 4-th root of unity. For any elliptic curve E and any integer N the field $\mathbb{Q}(E[N]))$ is Galois which embeds in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ (see 5.1). Given the Galois image of two elliptic curves, can one prove that one is more ECM-friendly than the other?

Conjecture 6 (Hyp B in [Wan18] for $\mathbb{Q}$, extension of Prop. 2.2 in [Pol16] for quadratic K)**.** *Let K be either $\mathbb{Q}$ or an imaginary quadratic field of class number one. For any $a, c \in K$,*

$$\Pi_K(x; c, a) = \{p \in \mathcal{O}_K, \text{ prime}, \| p \| \leq x, p \equiv a \pmod{c}\},$$

where $\| \cdot \|$ is the algebraic norm. Let $\pi_K(x; a, c)$ be the cardinality of $\Pi_K(x; c, a)$ and let q be restricted to the set of primes. Then for any fixed $a \in \mathbb{Z}$ and $A > 0$ we have

$$\sum_{\substack{\| q \| \leq x^{1-\delta} \\ (q,a)=1}} \left|\pi_K(x; q, a) - \frac{\pi(x)}{\varphi(q)}\right| \ll_A \frac{x}{(\log x)^A},$$

where the constant implied by $\ll$ is uniform on $x \geq x_0$. Here $\varphi(q) = |(\mathcal{O}_K/q\mathcal{O}_K)^*|$.

We are now ready to state the main result.

Theorem 7. *Let $E/\mathbb{Q}$ be a CM elliptic curve by an order in an imaginary quadratic field K with $h_K = 1$. Set $\alpha(E) = L'(1.\chi)/L(1, \chi)$ where χ is the Dirichlet character attached to the Kronecker symbol of K. Then $\alpha(E)$ measures how ECM-friendly E is. Precisely, let $0 < \delta < 1$ and $x > y$ be real numbers such that*

$$y < x^{1-\delta}.$$

If $E_1/\mathbb{Q}$ and $E_2/\mathbb{Q}$ are elliptic curves satisfying the above hypothesis then $\alpha(E_1) < \alpha(E_2)$ implies that E_1 is more ECM-friendly than E_2 with respect to (x, y), asymptotically as $x \to \infty$.

The quantity $\alpha(E)$ is better known under the following form.

Proposition 8. *Let E be an elliptic curve with CM and let $\alpha(E)$ be the constant in Theorem 7. For all rational primes ℓ we set*

$$\alpha_\ell(E) = \log \ell \left(\frac{1}{\ell - 1} - \mathbb{E}_p(\mathrm{val}_\ell(|E(\mathbb{F}_p)|))\right),$$

where $\mathbb{E}_p$ denotes the average value in the sense of Chebotarev density over random primes p. Then we have

$$\alpha(E) = \sum_{\ell} \alpha_\ell(E).$$

The theorem is proved by relating the quotient $\psi_E(x,y)/\psi_E(x,\infty)$ to an analoguous quotient involving classical smooth number counting functions.

In view of our main results, it is natural to ask about the order of magnitude of $\psi_E(x,y)$.

Our second main result addresses this question.

Theorem 9. *Let K be a number field. Let (x,y,z) be three positive integers such that $u := \frac{\log x}{\log y}$ and $v := \frac{\log y}{\log z}$ are as in the domain*

$$\Delta: \quad u \leq \frac{\log x}{\log_2 x} \text{ and } v \leq \frac{\log_2 x}{\log_3 x}.$$

Then we have

$$\psi_v(x,y)/x \quad \geq \quad \rho(v)\rho(u)(1+o(1)),$$

uniformly on Δ.

Chapter 5

ECM-friendly curves

5.1 Review of the literature for p-adic Galois images

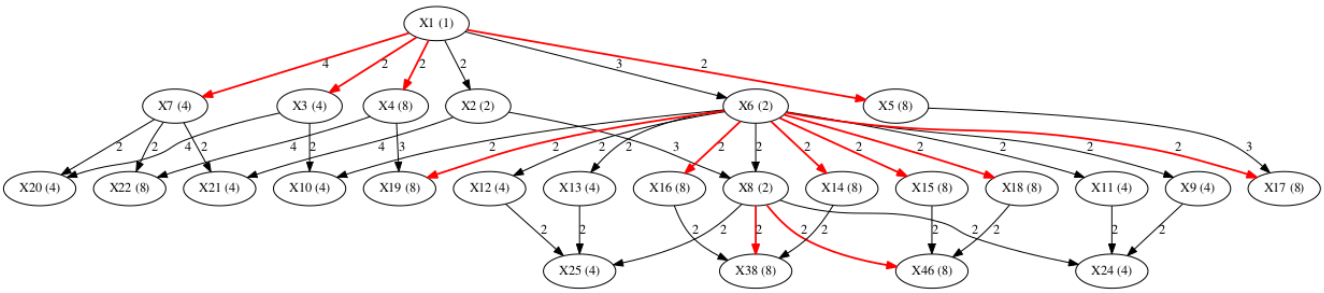
Theorem 10 (Shimura 1971)**.** *Let $\Gamma \subset \Gamma(1)$ be a congruence group of level N and let $H = \Gamma/\Gamma(N)$. Assume that $-I \in H$ and $\det H = (\mathbb{Z}/N\mathbb{Z})^*$. Then the modular curve of Γ has a (singular) model defined by a polynomial $X_H(t, j) \in \mathbb{Z}[t, j]$ such that, up to conjugacy,*

$$j(E) \subset H \Leftrightarrow \exists t \in \mathbb{Q}, X_H(t, j(E)) = 0.$$

computing Galois representation = testing whether j is on a list of plane curves

Lemma 11. *Let $G \subset \mathrm{GL}_2(\mathbb{Z})$ be such that $\Gamma(\ell^k) \subset G$ with $\ell^k \notin \{2, 3, 4\}$. Then every maximal subgroup H of G contains $\Gamma(\ell^{k+1})$.*

Example 12 ($\ell = 2$)**.**



Definition 13 (Mazur's program B)**.** *Given a number field K, determine the list of integers N and subgroups $H \subset GL_2(\mathbb{Z}/N\mathbb{Z})$ which can occur as Galois image. For each H give the finite list of j invariants if there are finitely many or parametrize the family if it is infinite.*

The strategy to accomplish the program is as follows:

Q1 Solve the case when N is a prime power

- prove Serre's uniformity conjecture : the ℓ-adic Galois image is $\mathrm{GL}_2(\mathbb{Z}_\ell)$ for all elliptic curves and all $\ell > 37$.

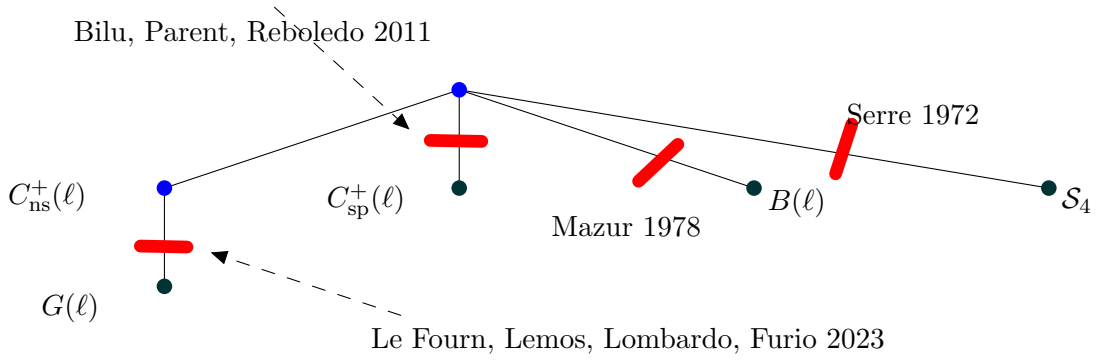


Figure 5.1: Results on Serre's uniformity.

- for each $\ell \leq 37$ compute the subgroup tree of $\mathrm{GL}_2(\mathbb{Z}_\ell)$ until all the leaves have genus ≥ 2. Find their complete list of rational points.

Q2 Compute the intersection of X_{Γ_1} and X_{Γ_2} for subgroups $\Gamma_1 \in \mathrm{GL}_2(\mathbb{Z}_{\ell_1})$ and $\Gamma_2 \in \mathrm{GL}_2(\mathbb{Z}_{\ell_2})$ with $\ell_1 \neq \ell_2$.

Q3 Compute subfamilies which have an extra entanglement, i.e. image mod N_2N_2 not equal to the CRT of the images mod N_1 and N_2.

Important progress has been made on the conjecture:

- Serre 1972 : stated the problem and eliminated the case of exceptional subgroups S_4 and A_5
- Mazur 1978 : $X_0(\ell)$ has only cusps and CM points for $\ell \notin \{2, 3, 5, 7, 11, 13, 17, 19, 37, 43, 67, 163\}$
- Bilu 1995 and Bilu-Parent 2008 : when the Runge method applies, $\log|j| < p^\alpha$ with $\alpha < 1$. This includes the cases $X_0(N)$, and $X_{\mathrm{sp}}^{+}(\ell)$.
- Gaudron and Rémond 2011 : if elliptic curve of j-invariant j has an isogeny of order ℓ then $h(j) > c\ell$ for an explicit constant c
- Bilu, Parent, Rebolledo 2013 : since $cp < \log|j| < p^\alpha$ it is enough to test on computer the $p \leq 10^8$.

Theorem 14 (Cox and Parry 1984)**.** *There exists an effectiv bound $N < N(g)$ where N is the level and g the genus of a congruence group.*

Theorem 15 (Sutherland & Zywina 2017)**.** *For $\ell = 2, 3, 5, 7, 11, 13$ there are $1201, 47, 23, 15, 2$ respectively 11 sub-groups of $\mathrm{GL}_2(\mathbb{Z}_\ell)$ which occur as $\rho_{E,\ell^\infty}(\mathrm{Gal}_{\mathbb{Q}})$ for infinitely many j-invariants of $\mathbb{Q}$; for $\ell > 13$ the image is surjective.*

Theorem 16 (Rouse & Zureick-Brown 2015)**.** - *There exist precisely 1208 possible images in $\mathrm{GL}_2(\mathbb{Z}_2)$ for the non-CM curves defined over $\mathbb{Q}$ and all have a level $N \leq 2^5$.*

- *Among these, 6 are obtained by a single value of j, one is obtained twice and the others occur infinitely many times.*

Theorem 17 (Rouse, Sutherland, Zureick-Brown 2021)**.** *Complete list of rational points on the curves of genus $g \geq 2$ of prime power level, except for 5 modular curves of unknown status.*

In [BS22] we obtained that there are precisely 1525 Galois images in $\prod_\ell \mathrm{GL}_2(\mathbb{Z}_\ell)$ which are obtained for infinitely many curves.[1]

Theorem 18 (Daniels, Gonzalez-Jimenez 2023)**.** *Intersection of curves X_{γ_1} and X_{Γ_2} whose levels are prime powers and genus is 0 or 1. Complete list of rational points except for 55 curves of genus $g \geq 2$ of unknown status.*

Theorem 19 (Jones, McMurdy, Lozano-Robledo, Daniels, Morrow 2020-2023)**.** *Except for a finite explicit list of modular curves, all the entanglement cases are as follows:*

1 there are 3 families of genus 0 of non-abelian entanglement

2 all the other entanglement cases are abelian and of type : Weil, discriminant, CM or fake CM, except for a finite explicit set of curves of genus $g \geq 2$.

The general case is out of reach for several reasons:

1 Serre's conjecture requires upper bounds and Runge doesnt apply to Cartan non-split

2 the 1207 subgroups of 2-adic images are too many to make sense

3 there is no algorithm to find all rational points on a modular curve

5.2 Complete list of ECM-friendly Montgomery curves

Theorem 20 (B. Shinde 2019)**.** *An elliptic curve has an equation $By^2 = x^3 + Ax^2 + x$ if and only if it has a cyclic isogeny of order 4.*

Corollary 21. *Montgomery curves have doubling of the form $(x_{2P} : z_{2P}) = g(f(x_P : z_P))$.*

Proof. Let E be a Montgomery curve and let $\langle P \rangle$ be the kernel of its isogeny of order 3. Let $f : E \to E$" be the isogeny of kernel $\langle 2P \rangle$. Let $\hat{f} : E' \to E$ be the dual isogeny.
Since $|\ker f| = |\ker \hat{f}| = 2$, the map

$$P \mapsto \hat{f}(f(P))$$

has kernel of order 4 contained inside $E[2]$, so it is the doubling. □

Theorem 22 (Lemos)**.** *Serre's uniformity conjecture is proven for the elliptic curves which have a cyclic isogeny of order r with $r \in \{2, 3, 5, 7, 13\}$.*

Lemma 23. *Let E_1 be an elliptic curve admitting a cyclic 4-isogeny of kernel $\langle P_1 \rangle$.*

[1] This phrase was missing from the reviewed version of this thesis.

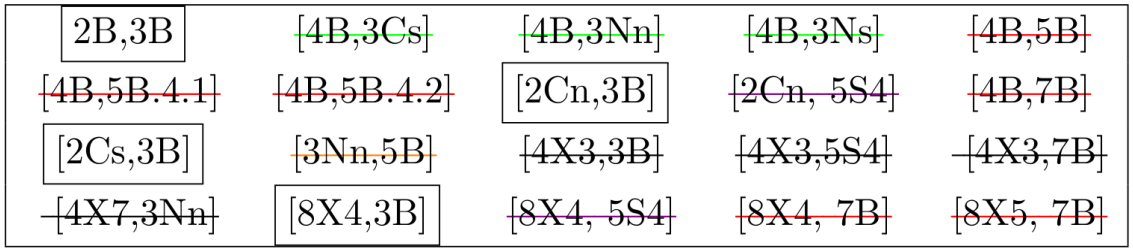


Figure 5.2: The families of unknown status are empty when intersected with $X_0(4)$.

1 The isogeny graph of E_1 contains three more curves E_2, E_3 and E_4.

2 Let A_i, $i = 1, 2, 3, 4$ be the A-parameters of the four curves. Then we have $A_2 = -A_1$, $A_4 = -A_3$ and $(A_1 - 1)(A_3 - 1) = -11$.

has an isogeny graph formed of at least four curves.

Proof. 1. Let P_1 be a 4-torsion point on $E_1(\overline{\mathbb{Q}})$. Let $\phi : E_1 \to E_2$ be the isogeny of kernel $\langle 2P \rangle$ and $\widehat{\phi} : E_2 \to E_1$ its dual isogeny. Let Q_2 be a 2-torsion point of E_2 such that $\ker \widehat{\phi} = \langle Q_2 \rangle$. Set $P_2 = \phi(P_1)$ and note that $2P_2 = \phi(2P_1) = 0$ and $P_2 \neq \mathcal{O}$ because $\ker \phi \neq \mathbb{Z}/4\mathbb{Z}$. Hence $E_2[2](\mathbb{Q}) \simeq \mathbb{Z}/2 \times \mathbb{Z}/2$.

Now E_2 admits three 2-isogenies by its 2-torsion points, one of which is $\widehat{\phi}$ whereas the other others arrive in two new curves E_3 and E_4.

2. An alternative proof of the first point is obtained by direct computations with curves in the field $\mathbb{Q}(A)$ where A is seen as a formal variable. Again, by direct computations one computes the j-invariants and the relations which relate the A-parameters of the four curves. □

Lemma 24 (Bilu Parent 2011)**.** *Assume that X_G is defined over $\mathbb{Q}$, and assume that the absolute Galois group $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ acts non-transitively on the cusps of X_G. Then for any $P \in Y_G(\mathbb{Z})$ we have*

$$\log |j(P)| \leq 30|G|N^2 \log N.$$

A direct verification with LMFDB shows that, when r is as in Theorem 22, the cusps of $X_0(r)$ are fixed by $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ and therefore the above lemma applies:

Corollary 25. *Let $r \in \{2, 3, 5, 7, 13\}$. Let G be a congruence group whose level is not divisible by r and which is contained in a split or nonsplit Cartan. Then any point on $X_0(r) \bigcap X_G$ has the j-coordinate such that*

$$\log_2 |j| \leq 30r^3(r-1)^2 \log_2 r.$$

Proposition 26. *There are no rational non-cusp non-CM points on $X_{G_2} \bigcap X_{G_p}$ when G_2 is a subgroup of the Borel mod 2, called 2B, and G_p a subgroup of $C_{\mathrm{sp}}(7)^+$ or $C_{\mathrm{ns}}(11)^+$.*

Proof. Note first that Bilu and Parent's bound applies. We use this bound for $r = 2$ and obtain: $\log_2 |j| \leq 240$.

The curve $X_{\mathrm{ns}}(11)^+$ has genus 1 and a Weierstrass model is known. One can directly compute its integral points and check that they correspond to cusp and CM points.

The curve $X_{\mathrm{ns}}(9)^+$ has equation $j = J(t)$ where J is a rational function whose numerator and denominator have degree 27. An exhaustive search of the values of t required to consider

$$\log_2 |t| \approx \frac{240}{27},$$

or equivalently approximately 1000 values.

The situation is similar for $X_{\mathrm{ns}}(7)^+$, $X_{\mathrm{sp}}(7)^+$ and respectively $X_{\mathrm{ns}}(5)^+$ which are given by equations of the form $j = J(t)$ with J of degree 21, 28 and 10 respectively.

□

Chapter 6

Shor-like algorithms for discrete logarithm and unit group

Shor's algorithm can be viewed as an application of the hidden subgroup problem (HSP). Assume that G is an abelian group whose cardinality is an n-bit integer and f is a function defined on G, not necessarily a morphism whose period subset is $H \subset G$ and which can be computed by an algorithm which is given. The HSP consists in computing H.

We assume that $G \equiv \oplus_{i=1}^{m} \langle g_i \rangle$ for an absolute constant m and some given generators $g_1, \ldots, g_m$. The elements of G are represented and enumerated as tuples $(z_1, \ldots, z_m) \in (\mathbb{Z}/|G|\mathbb{Z})^m$. We write $yz = \sum_{i=1}^{m} y_i z_i \in \mathbb{Z}/|G|\mathbb{Z}$. To fix ideas, we assume that the orders of the generators are known, which is the case when solving the DLP. The case of unknown orders is very similar but we don't discuss it here.

The HSP solver has a quantum procedure and a classical post-treatment. In the quantum procedure one computes the superposition

$$\psi_1 = \sum_{z \in G} |f(z)\rangle |z\rangle.$$

One measures the register $f(z)$ and obtains, for some fixed z_0,

$$\psi_2 = \sum_{z \in z_0 + H} |z\rangle.$$

The quantum Fourier transform, applied to ψ_2, gives

$$\begin{aligned} \psi_3 &= \sum_{y \in G} \sum_{z \in z_0 + H} e^{2\pi i \frac{yz}{|G|}} |z\rangle |y\rangle \\ &= \sum_{y \in H^\perp} |y\rangle, \end{aligned}$$

where $H^\perp = \{y \in G \mid \forall z \in H, \quad yz = 0\}$. The measurement of ψ_3 yields a vector of $H^\perp$ which is randomly drawn with uniform probability.

After repeating the quantum procedure a constant number of times one can extract a basis of $H^\perp$ and further a basis of H.

6.1 Review of the literature: the CHSP solver

Consider now the case $G = \mathbb{Z}^m$ or $\mathbb{R}^m$ and $f : G \to X$, where X is a finite set. The periods of f are all the real vectors ℓ such that

$$\forall x, \quad f(x + \ell) = f(x).$$

The notion which corresponds in the case of a lattice L to an orthogonal set is the dual lattice:

$$L^* = \{y \in \mathbb{R}^m \mid \forall z \in L \quad yz \in \mathbb{Z}\}.$$

Quantum computers perform calculations at a given given error rate and hence one has a fixed error bound on the coordinates of the output y. If an HSP solver is applied directly one has to compute a basis of L^* from a list of $m + o(1)$ approximations of vectors of the lattice. This is related to the LWE problem and is not known to be polynomial.

A more general and more difficult problem than HSP is as follows.

Definition 27. *Let f be a function defined on $\mathbb{Z}^m$ or $\mathbb{R}^m$. Assume that its set of periods L has full rank and an upper bound on $\mathrm{Vol}(L)$ is given. The continuous hidden subgroup problem (CHSP) consists in computing L when given an algorithm to compute f.*

A CHSP solver of polynomial time complexity exists but only for the functions f subject to a series of technical conditions [dBDF20]. The modifications with respect to Shor's algorithm are as follows:

- the initial state is not a uniform superposition but one with coefficients in Gaussian distribution;
- the precision of the parameter z can be tuned
- the post-treatment is done with vectors of a much larger precision than in the HSP solver.

More precisely, the Buchmann-Pohst algorithm, which is based on the LLL algorithm, takes as input an approximation of precision $O(m^4)$ of $m + O(1)$ generators of a lattice L of dimension m and an upper bound on $\mathrm{Vol}(L)$ and outputs a basis of L. The space complexity of the CHSP solver is that of storing m coordinates at the precision required by the Buchmann-Pohst algorithm: a total of $O(m^5)$ qubits (see [BP23, Cor 37]).

6.2 An improvement based on cyclotomic units

The cyclotomic fields have a subgroup of finite index: the cyclotomic units. Recall that the Minkowski embedding of a number field of signature (n_1, n_2) is a tuple of field morphisms $\sigma : K \to \mathbb{R}^{n_1} \times \mathbb{C}^{n_2}$. The unit lattice L of K is the image of $\mathcal{O}_K^*$ by $\log |\sigma_i|$ with $i \in [1, n_1 + n_2]$. We call M the sub-lattice of cyclotomic units, which has full rank (see [Was12, Th 8.3] for a closed formula for its index).

In [BP23] we modified the CHSP solver when tackling cyclotomic fields. One runs the quantum procedure at a lower precision, so that the number of qubits is reduced, and modifies the classical post-treatment to begin by a step to increase the precision before executing the Buchmann-Pohst algorithm.

Recall that for a lattice L, a positive real $\delta > 0$ and a vector $y \in \mathbb{R}^m$, the bounded distance decoding problem, denoted $\mathrm{BDD}(y, L, \delta)$ consists in computing the closest point of L with respect to y, under the guarantee that the distance is bounded by δ.

The naive algorithm to solve the BDD was analysed by Babai and its good speed is compensated by the fact that it can only correct very small distances.

Lemma 28 ([Bab86] Eq. (4.3)). *Let L be a lattice, B the basis of a basis of L and $\widetilde{y}$ a vector in $\mathbb{R}^{\dim L}$. Algorithm 1 solves $\mathrm{BDD}(\widetilde{y}, L, \delta)$ in classical polynomial time when $\delta < 1/(2\| B \|_\infty)$.*

Algorithm 1 Babai's BDD solver.

Require: $\delta > 0$, B the matrix defining the lattice $L \subset \mathbb{R}^m$ and $\widetilde{y} \in \mathbb{R}^m$ such that $d(\widetilde{y}, L) < 1/(2\| B \|_\infty)$

Ensure: $\mathrm{CVP}(\widetilde{y}, L)$ and its coordinates in basis B

compute $\widetilde{z} := B^{-1}\widetilde{y}$;

round $z = (z_1, \ldots, z_n) := (\lfloor \widetilde{z_1} \rceil, \ldots, \lfloor \widetilde{z_n} \rceil)$

return $y := Bz \in L$ and $z \in \mathbb{Z}^m$.

Figure 6.1 illustrates our modification of the CHSP solver. One runs the quantum procedure at precision $\log_2 \| B_{M^*} \|$ and obtains a vector close to L^* at this precision. One applies Babai's algorithm with respect to M^*, so that the precision becomes arbitrarily large in polynomial time. This precision allows to do the classical post-treatment. Note that the value of $\| B_{M^*} \|$ is given in the literature (see [CDPR16, Th 3.1]), $\log_2 \| B_{M^*} \| = O(\log m)$:

Theorem 29 (Th 43 in [BP23]). *Algorithm 1 in [BP23] computes a basis of the unit group of $\mathbb{Q}(\zeta_m)$ is poly(m) time and uses $O(m^2 \log m)$ qubits.*

6.3 Review of the literature: Regev's algorithm

Let $\mathbb{G}$ be an abelian group for which we use the multiplicative notation and let $n = \lceil \log_2 |\mathbb{G}| \rceil$. Let $g_1 = g$ and $g_2, \ldots, g_d \in \mathbb{G}$. Contrary to Shor's algorithm, which evaluates g^z in superposition for all n-bit positive integers, Regev's algorithm evaluates $\prod_{i=1}^d g_i^{z_i}$ in superposition over all the d-tupes of $\lceil n/d \rceil$-bit positive integers $(z_1, \ldots, z_d)$.

When N is an integer to be factored and $\mathbb{G} = (\mathbb{Z}/N\mathbb{Z})^*$. Consider the two lattices

$$\begin{array}{rcl} \mathcal{L} & = & \{z \in \mathbb{Z}^d : (\prod_{i=1}^d g_i^{z_i})^2 \equiv 1 \pmod N\} \\ \mathcal{L}_0 & = & \{z \in \mathbb{Z}^d : \prod_{i=1}^d g_i^{z_i} \equiv \pm 1 \pmod N\}. \end{array} \tag{6.1}$$

Any vector of $z \in \mathcal{L} \backslash \mathcal{L}_0$ allows to find a non-trivial factor of N: one sets $X = \prod_{i=1}^d g_i^{z_i}$ and computes $\gcd(X - 1, N)$. This is a non-trivial divisor of N under the condition that N is an odd integer which is not a prime power.

Based on number theoretic heuristics $\mathcal{L} \neq \mathcal{L}_0$ (see e.g. the subgroup obstruction in [Pil24]). To find a vector $z \in \mathcal{L} \backslash \mathcal{L}_0$ one can use a CHSP solver: $\mathcal{L}$ is the period lattice of $f(z) = \prod_{i=1}^d g_i^{z_i}$, once a basis $\{b_1, \ldots, b_d\}$ is found, one of the b_i's is outside $\mathcal{L}_0$.

However, an important remark is that one can find a vector of $\mathcal{L} \backslash \mathcal{L}_0$ without computing a basis of $\mathcal{L}$. A post-treatment similar to the Buchmann-Pohst procedure requires a precision $n/d + \log_2 T$

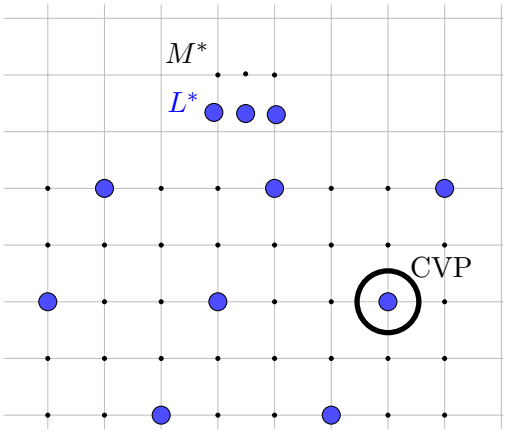


Figure 6.1: Illustration of Lemma 28. The black lattice of small dots is known and its basis is short so that one can solve CVP for it. One has an oracle producing points $\widetilde{y}$ in the little disk around the bold blue dots. Then, if one brings $\widetilde{y}$ to the small dot y, then y will automatically be in a bold dot.

to find a family of vectors which span all the vectors of $v \in \mathcal{L}$ with $|v| \leq T$. By Minkowski's theorem (or also by the LLL theorem) $\mathcal{L}$ has always a vector of $O(d + n/d)$ bits. If $\mathcal{L}\backslash\mathcal{L}_0$ has a vector of $O(d+n/d)$ bits then Regev's algorithm requires a precision of $O(\max(d, n/d))$ bits in the quantum procedure. Since the CHSP solver's precision is super-linear, Regev has an advantage and in the following we keep the presentation of [Reg25] which is independent of the CHSP solver.

Consider now the gate complexity of Regev's algorithm. The dominating step in the HSP solver, CHSP solver and Regev's algorithm is the computation of the multi-scalar product $\prod_{i=1}^{d} g_i^{z_i}$ for n/d-bit integers z_i. Assume for a moment that one has a lookup table with indices $\varepsilon \in \{0,1\}^d$:

$$\mathrm{tab}(\varepsilon_1, \ldots, \varepsilon_d) = \prod_{i=1}^{d} g_i^{\varepsilon_i}.$$

Its cost is $O(\frac{n}{d}M)$ where M is the cost of a multiplication in the group.

Since lookup tables have 2^d entries they require this number of qubits, which is super-polynomial. Instead Regev's algorithm computes on-the-fly the entries of the table. Regev analyzed the cost of this computation, which is negligible with respect to M.

6.3.1 Variants of Regev's algorithm for the DLP

Ekerå and Gärtner [EG24] extended the algorithm to the multiplicative group $\mathbb{G} = (\mathbb{Z}/p\mathbb{Z})^*$ for a prime p. This case and that of the factorization uses $g_i = p_i$, the i-th prime. Their small bit size is crucial in keeping small the cost of the on-the-fly computation.

6.3.2 The DLP with pre-computations

They also introduced a variant of "DLP with pre-computations": An attacker knows the group $\mathbb{G}$, selects the elements g_1, g_2, …, g_{d-1} used in Regev's algorithm and computes their discrete

logarithm using Shor's algorithm. Finally, when the challenge $x = g_d$ is given, the attacker must solve the DLP for x. This problem is relevant in practice because the implementations are based on a short list of groups $\mathbb{G}$ which are recommended by the NIST. An attacker has a long period of time to do the pre-computations whereas she has to solve the DLP with pre-computations in a very short time, e.g. when authentication to a server by ssh (see the LogJam attack [ABD+15] for more on this problem).

The lattices $\mathcal{L}$ and $\mathcal{L}_0$ from Equation (6.1) are replaced here as follows:

$$\begin{array}{rcl} \mathcal{L} & = & \mathcal{L}(\mathbb{G}, S) := \left\{ (z_1, \ldots, z_d) \in \mathbb{Z}^d \;\middle|\; \prod_{i=1}^{d} g_i^{z_i} = 1_{\mathbb{G}} \right\}, \\ \mathcal{L}_0 & = & \{(z_1, \ldots, z_d) \in \mathcal{L} \mid z_d \equiv 0 \pmod{r}\}. \end{array} \tag{6.2}$$

Given that g is a generator, for any vector $(z_1, \ldots, z_d)$ of $\mathcal{L}$, the following holds:

$$\sum_{i=1}^{d} z_i \log_g g_i \equiv 0 \pmod{r}.$$

For any $(z_1, \ldots, z_d) \in \mathcal{L} \setminus \mathcal{L}_0$ we have:

$$\log_g x \equiv -z_d^{-1} \left(\sum_{i=1}^{d-1} z_i \log_g g_i \right) \pmod{r}. \tag{6.3}$$

Consequently, the solution to the discrete logarithm problem is found.

In [BBP24] we developed an algorithm to certify that the heuristic in Regev's algorithm is true for a given group $\mathbb{G}$ and $g_i \in \mathbb{G}$ with $1 \leq i \leq d-1$. Indeed, note that the complexity of the SVP is exponential in the dimension of the lattice, which is $d = \sqrt{n}$, so it is sub-exponential in the bit size n. Since the elliptic curve parameters are set to withstand an exponential attack, it is possible to do this computation on a classical computer.

Proposition 30. *Consider a full-rank lattice $\mathcal{L}$ in $\mathbb{Z}^d$ with volume $\mathrm{Vol}(\mathcal{L}) = q$, where q is an n-bit integer. Consider a full rank sub-lattice $\mathcal{L}_0$ such that $\lambda_1(\mathcal{L}_0) \geq \sqrt{d} q^{1/d}$. Then $\log_2(\lambda_{\mathcal{L}_0}(\mathcal{L})) \leq \frac{n+1}{d} + \frac{1}{2}\log_2 d$. In particular, if $d \leq \sqrt{n}$ and $\mathcal{L}$ is certified, then Regev's algorithm with the parameter $T = \exp(n/d)$ is successful in solving DLP with pre-computations.*

We conclude that it is possible to certify the correctness of Regev's algorithm before the execution. This is important in a practical implementation to ensure that there are no "theoretical bugs".

6.3.3 The multidimensional DLP

In 1993, Brands [Bra93] created the first cryptographic application based on an extension of DLP, called multidimensional DLP. The problem constantly receives attention in cryptography.

Definition 31. *Let $\mathbb{G}$ be a commutative group whose order is an n-bit integer and let $g_1, \ldots, g_d \in \mathbb{G}$ be given. The multidimensional discrete logarithm problem consists in finding, if they exist, the $\lceil n/d \rceil$-bit integers z_i, $1 \leq i \leq d$ such that*

$$[z_1]g_1 + \cdots + [z_d]g_d = 0 \text{ and } z_d \neq 0.$$

A major question on the multidimensional DLP is whether its complexity is the same, as an expression of n, as the one of the DLP as an expression of $|\mathbb{G}|$:

$$\text{gate complexity(multidimensional DLP)} = O(nM(n))?$$

Let h be the naive height over $\mathbb{Q}$ (resp. $\mathbb{F}_2(t)$) and its extension to the elements of an elliptic curve E with coefficients in $\mathbb{Q}$ (resp. $\mathbb{F}_2(t)$). Let $2 \leq d \leq \sqrt{n}$ be a parameter, and let $g_1, \ldots, g_d$ be elements of $\mathbb{G}$ such that g_1 generates $\mathbb{G}$. Define

$$m := \max_{\varepsilon \in \{0,1\}^{d-2}} h(\sum_{i=2}^{d-1} [\varepsilon_i] g_i) \tag{6.4}$$

and assume that any such sum can be computed using $O(M(m)(\log_2 m)^2)$ gates.

Theorem 32 (Th 1 in [BBP24])**.** *Let m be the parameter defined in Equation (6.4). If $m \leq n$, then Regev's algorithm solves the multidimensional DLP with*

$$\textit{gate complexity} = O\left(\frac{n}{d}(M(n) + \min(dM(n), M(m)\log_2 m))\right).$$

In particular, if the generators g_i, $1 \leq i \leq d$, are random elements of $\mathbb{G}$ the multidimensional DLP can be solved with $O(nM(n))$ gates.

6.4 Regev's algorithm on hyperelliptic curves of high genus

As it was seen in Section 1 any abelian group can be used in cryptography, the capital question being the hardness of its DLP. In 1989 Könlitz [Kob89] proposed the Jacobian of hyperelliptic curves, which are a competitive alternative to elliptic curves and a candidate for standardization.

Recall that a hyperelliptic curve H of genus g over a field k is an algebraic curve of equation $y^2 = f(x) + h(x)y$ where $f, h \in k[x]$, $\deg h \leq g$ and $\deg f = 2g+1$ or $2g+2$. In the following $\operatorname{char} k \neq 2$ so we can and will assume $h = 0$. The set of rational points of H, denoted $H(k)$, contains the pairs $(x, y) \in k^2$ solutions to the curve equation, together with the point at infinity ∞. The divisor group of H, $\operatorname{div}(H)$, is the free abelian group of $H(k)$. The degree of a divisor is the sum of its coefficients. To any function φ in the function field of H one associates an element of $\operatorname{div} H$: $\operatorname{div}\varphi = \sum_{P \in H(k)} n_P \cdot P$ where n_P is the multiplicity of f at P, with positive sign if it is a zero and with negative sign if it is a pole. The divisors associated to functions are called principal. The Jacobian of H, denoted $\operatorname{Jac}(H)$, is the quotient subgroup of degree-0 divisors by the subgroup of principal divisors.

Every element of $\operatorname{Jac}(H)$ has a unique representation of the form $\sum_{i=1}^{r} P_i - r(\infty)$ with $r \leq g$ and $P_i = (x_i, y_i)$ rational points other than ∞ with distinct x coordinates. This is the divisor representation of an element of $\operatorname{Jac}(H)$. The Mumford representation of an element of $\operatorname{Jac}(H)$ is a pair $(a, b) \in k[x]^2$ such that $b^2 - f \equiv 0 \pmod{a}$; it is reduced if $\deg b < \deg a$. Cantor [Can87] presented an algorithm for the addition law in Mumford representation, whose complexity is quasi-linear.

The conversion from the divisor to the Mumford representation is done as follows: $a(x) = \prod_{i=1}^{r}(x - x_i)$ and b is the unique polynomial such that $b(x_i) = y_i$ for $1 \leq i \leq r$. See Figure 6.2 for a summary of this conversion.

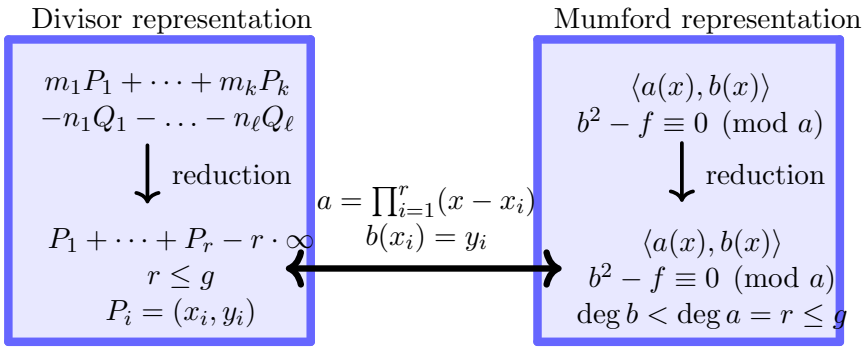


Figure 6.2: The divisor and Mumford representations of the elements of the Jacobian of a hyperelliptic curve.

6.4.1 The hardness of the DLP on the Jacobian of hyperelliptic curves

On a classical computer an attacker can apply Pollard's rho algorithm, whose complexity is $\tilde{O}(\sqrt{|\mathbb{G}|})$, or the Index calculus, whose complexity is sub-exponential when g is larger than a value depending on the bit size of $|\mathbb{G}|$. When g is constant the complexity of Index Calculus is smaller than that of Pollard's rho when $g = 2$ and $g = 1$, which corresponds to the elliptic curves. Hence, the hyperelliptic curves with cryptographic importance are those of genus two. The case of high genus is important above all for theoretical reasons and possibly to offer examples where quantum algorithms can be implemented with small qubit and gates resources.

Going back to Regev's algorithm, let $\mathbb{G} = \mathrm{Jac}(H)$ for an elliptic curve of genus g. The DLP variant of Regev's algorithm (see [EG24]) requires to set a parameter d and to select $g_1, \ldots, g_{d-2}$ such that the computation of $\sum_{i=1}^{d} \varepsilon_i g_i$ is negligible compared to the cost of an addition of general elements of $\mathrm{Jac}(H)$.

In [BB24] we set $d = \min(g, \lfloor\sqrt{n}\rfloor)$ and $g_i = (P_i) - (\infty)$ for $1 \leq i \leq d$. Note that, in particular, the heuristic implies that the Jacobian variety is generated by divisors of the form $(P) - (\infty)$. This has been proven to hold by two different methods in [Vol00, Main Theorem] and [Eng02, Corollary 4.2] under the condition $q \geq (8g-2)^2$ which is always satisfied in cryptographic applications. More generally, this condition is satisfied as soon as $g < \sqrt{n}$ and $n \geq 30$. Alternatively, when $g \geq \sqrt{n}$ one can replace q with $q^{2\log_2 n}$ so as to satisfy the condition; the runtime of Regev's algorithm is then multiplied only by a factor $\log_2 n$ leaving unchanged the complexity $\tilde{O}(n^{3/2})$.

Heuristic 33. *There exists a constant K such that the following holds. Let H be a hyperelliptic curve over $\mathbb{F}_q$ of genus g such that $|\mathrm{Jac}(H)|$ is an n-bit integer. Let $d = \min(g, \sqrt{n})$ and $b_1, \ldots, b_d$ be elements of $\mathrm{Jac}(H)$ of the form $(P) - (\infty)$ with $P \in H(\mathbb{F}_q)$ drawn uniformly at random. Then, almost surely, $b_1, \ldots, b_d$ span $\mathrm{Jac}(H)$ and the lattice L in Equation 6.2 has a basis whose vectors have norm at most $T = \exp(Kn/d)$.*

Given $\varepsilon \in \{0,1\}^d$, the divisor representation of $\sum_{i=1}^d \varepsilon_i g_i$ is $\sum_{\varepsilon_i=1} P_i - |\{i \in [1,r] \mid \varepsilon_i = 1\}| \cdot \infty$. This is computed for free. Since Cantor's algorithm requires Mumford's representation, the actual cost is that of the conversion.

Given a list of $r \leq g$ points $P_i = (x_i, y_i)$ one computes the polynomial $a(x) = \prod_{i=1}^r (x - x_i)$ with $O(g \log g)$ operations thanks to a multiplication tree. The fast interpolation algorithm ([vzGG03, Cor 10.2]) allows to obtain $b(x)$ such that $b(x_i) = y_i$ for all $1 \leq i \leq r$; this has a cost of $O(M(g) \log g)$ where $M(n)$ is the cost of a multiplication of n-bit integers.

Since $g \leq \sqrt{n}$, the cost of computing on the fly the entries of the table is $n^{\frac{1}{2}+o(1)}$ which is negligible with respect to the cost of a composition in the Jacobian which has cost $n^{1+o(1)}$. The proof made in the general case in [Reg25] applies and one obtains the following result.

Theorem 34 (under Heuristic 33)**.** *Let H be a hyperelliptic curve of genus g defined over $\mathbb{F}_q$, such that the cardinality of its Jacobian variety, $|\mathrm{Jac}(H)|$, is an n-bit integer. Then there exists an explicit probabilistic quantum algorithm which succeeds with probability $1 - o(1)$ and has complexity $\tilde{O}\left((d + \frac{n}{d})n\right)$ where $d = \min(g, \sqrt{n})$.*

The above algorithm merely gains a factor two when $g = 2$ because $d = \min(g, \sqrt{n})$. On a series of examples we found points g_i with small coordinates. Take for example, the Buhler–Koblitz curve "4GLV127-BK" which has been considered for cryptographic use [BK98, BCM14]. It is defined as $C : y^2 = x^5 + 17$ over $\mathbb{F}_q$ where $q = 2^{64}(2^{63} - 27443) + 1$; its Jacobian variety $\mathrm{Jac}(C)$ has a 254-bit prime group order.

Using Magma [BCP97] we searched for "twist curves" of the form $C_\delta : y^2 = x^5 + 17\delta$ whose Jacobian variety has large rank over $\mathbb{Q}$. For instance, $\delta = 3576896$ is a tenth power in $\mathbb{F}_q$ and the variety $\mathrm{Jac}_{\mathbb{Q}}(C_\delta)$ has rank six. A basis in projective Mumford representation is as follows:

$$\begin{aligned}
b_1 &= (x - 8, -7800, 1),\\
b_2 &= (x + 36, -584, 1),\\
b_3 &= (x - 332, -2008392, 1),\\
b_4 &= (x^2 - 77x - 3228, 911x + 21452, 2),\\
b_5 &= (x^2 - \tfrac{165}{49}x - \tfrac{30636}{49}, \tfrac{109269}{343}x + \tfrac{37988}{343}, 2),\\
b_6 &= (x^2 + \tfrac{48552}{529}x + \tfrac{4131648}{529}, -\tfrac{29522176}{36501}x + \tfrac{40433336}{12167}, 2).
\end{aligned}$$

This allows us to attack the discrete logarithm problem of $\mathrm{Jac}_{\mathbb{F}_q}(C)$ by transporting it to $\mathrm{Jac}_{\mathbb{F}_q}(C_\delta)$ where our generalized Regev's attack exploits these small, independent elements. This overall attack is eight times faster than Shor's algorithm.

6.5 Regev's algorithm on elliptic curves

Recall that the discrete logarithm of an element $x \in G$, denoted $\log_g x$, is the smallest non-negative integer z such that $x = [z]g$. The discrete logarithm problem consists in computing $\log_g x$ whereas the DLP with pre-computations is the same problem when the attacker has previously computed $\{\log_g x', x' \in X'\}$ for a set of $X' \subset \mathbb{G}$ of her choice. In the variants of the DLP treated in this article we restrict to the case where q is prime (see the Pohlig-Hellman [PH78] reduction).

Lemma 35. *The curves **Curve25519**, **BLS 12**, **BN**, **secp256k1** and **K-233** have twists of rank between 3 and 4 and have generators of small height. Hence Algorithm 2 can be run with parameter d between 5 and 6.*

Algorithm 2 Regev's algorithm for elliptic curves

Require: An elliptic curve E over $\mathbb{Q}$ (resp. $\mathbb{F}_2(t)$), a finite field $\mathbb{F}$ and two points P, Q on $E(\mathbb{F})$.
Ensure: $\log_P Q$

1: (classical computer) Apply Algorithm 3 to find $D \in \mathbb{Q}$ such that $\operatorname{rk} E_D \geq d-2$.
2: (classical computer) Compute a set of generators $P_1, \ldots, P_{d-2}$ of $E_D(\mathbb{Q})$
3: (quantum computer) Identify the points $P_1, \ldots, P_d$ with their images in $E_D(\mathbb{F})$. Set $P_{d-1} = \phi(Q)$ and $P_0 = \phi(P)$, where $\phi : E(\mathbb{F}) \to E_D(\mathbb{F})$ is the isomorphism $(x, y) \mapsto (x, y/\sqrt{D})$, and apply Regev's algorithm for $P_0, P_1, \ldots, P_{d-1}$.
4: **return** $\log_{P_0} P_{d-1}$

Algorithm 3 Rubin-Silverberg [RS02, Sec 9]

Require: an elliptic curve E with rational coefficients and an integer r
Ensure: an integer D such that the Mordell-Weil rank of E_D is larger or equal to r

(optional) Make a list of integers D which are likely to have large $\operatorname{rk}(E_D)$.
repeat
  D=next square-free integer (optionally from the list established in previous step)
  r_D = analytic rank of E_D
until $r_D \geq r$

Theorem 36 (under hypothesis made precise in [BBP24])**.** *Let $\epsilon > 0$ be an absolute constant and let $\mathbb{F}$ be a finite field of n-bit size. Let E be a given elliptic curve defined over $\mathbb{Q}$ (resp. $\mathbb{F}_2(t)$) such that $h(E) \leq (\log_2 n)^{1-\epsilon}$.*

Set $r = r(n) = \lfloor (\log n)^{1/2-\epsilon} \rfloor$, let D be the rational number of the smallest height (resp. the rational function of the smallest height) such that the quadratic D-twist E_D of E has rank r. Let $P_1, \ldots, P_r$ be a set of generators of $E_D(\mathbb{Q})$ (resp. $E_D(\mathbb{F}_2(t))$ with Neron-Tate heights.

Then Algorithm 2 with parameter $d := r + 2$ is successful in solving the DLP in $E(\mathbb{F})$ and has gate complexity $O(\frac{n}{d} M(n))$. This represents a speedup of a factor of $(\log n)^{\frac{1}{2}-\epsilon}$ with respect to Shor's algorithm.

Chapter 7

Perspectives

Many works presented in this document have been motivated by the cryptographic applications of algorithmic number theory. As a ramification, we formulated and proved theorems which are not restricted to the initial motivation. We focused mainly on the classical algorithms because they could be implemented, tested and improved in practice.

In this section, let us present a short list of questions that can be addressed in connexion to the quantum algorithms. Shor's algorithm has many variants and makes the object of numerous implementation improvements. We address further improvements in Section 7.1. An alternative to Shor's algorithm is Regev's algorithm (see Chapter 6). In Section 7.2 we extend a conjecture of Lang which is related to the algorithm.

A computational problem is said to have a quantum advantage if a quantum algorithm has a better complexity than the state-of-the-art classical algorithm and, additionally, its implementation on a quantum computer is faster than its resolution on a classical computer. The notion is a practical rather than a formal one, so that the first half a dozen announcements of quantum advantage have been contested. This raises the question if a quantum advantage can be found in a problem in algorithmic number theory. Indeed, they were the object of many works and the best classical implementations are often open source, e.g. in Pari/GP. In Sections 7.3 and 7.4 we study two candidate problems for quantum advantage: the square-free factorization and the resolution of Pell's equation.

7.1 An implementation of Shor's and Regev's algorithm

Recall that, when factoring an integer N, the gate complexity of Shor's algorithm is dominated by that of computing $a^x \bmod N$ for a constant a, an input x. The well-known square-and-multiply algorithm, which uses the base two digits of x does less than $2\log_2 x$ operations in $\mathbb{Z}/N\mathbb{Z}$. When a basis B is used instead, the algorithm does $2\log_B x$ operations in $\mathbb{Z}/N\mathbb{Z}$, having hence a speed-up $\log_2 B$. The additional cost is that it requires a lookup table of B entries. See [BGB+18] for a practical realization of the table, also called quantum read-only memory (QROM). This technique is called windowing.

Ekerå [Eke23] proposed and implemented a modification Shor's algorithm with multiple runs, each run using fewer gates then the original algorithm. This is important in practice because the quantum computers have a very short time of coherence.

An open question is to adapt windowing to Regev's algorithm (see Chapter 6) and to investigate

whether it has a multiple-run variant.

In the case of an implementation of Regev's algorithm for ECDLP, note that the entries of the lookup table are points. An open question is to test the practicality of storing points of the elliptic curve and to take advantage of their small height.

In the quantum implementation of the add-and-multiply algorithm, if the basis a is constant, one can precompute the values a^{2^i} with $i = 1, 2, \ldots$. However, if the basis is part of the input then one must uncompute the square in order not to use a large number of ancilla qubits. This has been studied in the classical paradigm under the name of point-halving [Knu99], but its efficiency on a quantum computer is to be tested. In particular, the efficiency of normal bases for the binary fields hasn't been tested.

7.2 Heuristics of the Mordell-Weil lattices

Moredell's proof of the finiteness of the rank of any elliptic curve is effective and therefore it can be implemented. Lang [Lan78, Ch IV.2] presents a method which finds a basis of points of small height when given a set of generators. In particular, the infinite descent (or 2-descent) can be implemented on the Jacobian of a hyperelliptic curve H, denoted $\mathrm{Jac}(H)$. The 2-division polynomial, which is used in the case of the elliptic curves, is replaced here by the 2-division ideal. Indeed, given an element $Q \in \mathrm{Jac}(H)$, to compute an element $P \in \mathrm{Jac}(H)$ such that $2P = Q$ one solves a polynomial system of fixed degree with respect to the genus of H.

To this point we haven't used the properties of the Mordell-Weil height. A set of generators of $\mathrm{Jac}(H)$ constitute a generating set of a lattice of dimension $\mathrm{rk}\,\mathrm{Jac}(H)$. In the case of elliptic curves, Lang [Lan83] made conjectures on the height of the smallest points on an elliptic curve which correspond to the Mordell-Weil lattice behaving as a random lattice. Due to the better algorithms of lattice reduction and lattice enumeration which have been developed for the lattice-based cryptography we can formulate precise generalizations of Lang's conjecture for high genus curves.

7.3 Seeking the quantum advantage via the square-free factorization

The problem of square-free factorization is as follows: given an integer of the form $N = P^r Q$ with known integer r and unknown integers P and Q, find P and Q. The problem has been formulated in algorithmic number theory [AM94] and later applied in cryptography [BDHG99]. The main classical attacks are either modifications of the ECM algorithm (see [Per01]) or based on lattice techniques (see [BDHG99]). No implementation has proven an advantage of factoring numbers of the form $N = P^2 Q$ with respect to general integers, e.g. RSA moduli.

The quantum algorithms for square-free factorization are based on a seemingly simple remark made in [LPDS12]: the hidden period of the Jacobi symbol $x \mapsto (\frac{x}{N})$ is essentially Q. Due to the fact that Shor's algorithm is probabilistic and must overcome numerical errors, the values which are "almost periods" are an obstacle. This is solved by using several Jacobi symbols: one computes, for some constant k the period of the function

$$\begin{array}{rccl} f: & \mathbb{Z}/N\mathbb{Z} & \to & \{\pm 1\}^k \\ & x & \mapsto & \left((\frac{x}{N}), (\frac{x+1}{N}), \ldots, (\frac{x+k}{N})\right). \end{array}$$

In a recent work [KMRVVK24] the quantum circuit to compute the Jacobi symbol has been improved. Our project is to study this circuit and to apply it to other similar problems or to further improve it.

7.4 Seeking the quantum advantage via Pell's equation

The solution set of Pell's equation, $x^2 - Dy^2 = 1$, corresponds to $\mathcal{O}^*$ or $(\mathcal{O}^*)^2$ where $\mathcal{O}^*$ is the unit group of $\mathbb{Q}(\sqrt{D})$. The state of the art to compute it is Buchmann's algorithm (see [BJJ10] for more recent improvements).

In the quantum paradigm, Hallgren [Hal07] proposed a polynomial-time algorithm. Contrary to Shor's algorithm, it hasn't been implemented and all the improvements of Shor's algorithm have to be adapted to this case. In particular, the number of qubits hasn't been optimized.

Bibliography

[ABD+15] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, et al. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 5–17, 2015.

[ADH94] Leonard M Adleman, Jonathan DeMarrais, and Ming-Deh Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In *Algorithmic Number Theory – ANTS-I*, pages 28–40. Springer, 1994.

[AM93] A Oliver L Atkin and François Morain. Elliptic curves and primality proving. *Mathematics of computation*, 61(203):29–68, 1993.

[AM94] Leonard M Adleman and Kevin S McCurley. Open problems in number theoretic complexity, II. In *International Algorithmic Number Theory Symposium*, pages 291–322. Springer, 1994.

[Bab86] László Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[Bar13] Razvan Barbulescu. *Algorithms of discrete logarithm in finite fields*. Theses, Université de Lorraine, December 2013.

[Bar16] Razvan Barbulescu. A brief history of pairings. In *International Workshop on the Arithmetic of Finite Fields WAIFI 2016*, volume 10064 of *Arithmetic of Finite Fields – WAIFI 2016*, Gand, Belgium, July 2016. Université de Gand, Springer.

[Bar21] Razvan Barbulescu. (Non)practicabilité de l'algorithme classique-quantique de factorisation des entiers. December 2021.

[BB24] Razvan Barbulescu and Gaetan Bisson. Regev's attack on hyperelliptic cryptosystems. working paper or preprint, December 2024.

[BBdV+17] Jens Bauch, Daniel J Bernstein, Henry de Valence, Tanja Lange, and Christine Van Vredendaal. Short generators without quantum computers: the case of multiquadratics. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 27–59. Springer, 2017.

[BBP24] Razvan Barbulescu, Mugurel Barcau, and Vicenţiu Paşol. A comprehensive analysis of Regev's quantum algorithm. working paper or preprint, December 2024.

[BCM14] Joppe Bos, Craig Costello, and Andrea Miele. Elliptic and hyperelliptic curves: a practical security analysis. In *Public-Key Cryptography — PKC 2014*, volume 8383 of *Lecture Notes in Computer Science*, pages 203–220, 2014.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system: the user language. *Journal of Symbolic Computation*, 24(3–4):235–265, 1997.

[BD19] Razvan Barbulescu and Sylvain Duquesne. Updating key size estimations for pairings. *Journal of Cryptology*, 32(4):1298–1336, 2019.

[BDHG99] Dan Boneh, Glenn Durfee, and Nick Howgrave-Graham. Factoring $n = p^r q$ for large r. In *Annual International Cryptology Conference*, pages 326–337. Springer, 1999.

[BEMG20] Razvan Barbulescu, Nadia El Mrabet, and Loubna Ghammam. A taxonomy of pairings, their security, their complexity. Available online at https://hal.science/hal-02129868, August 2020.

[BGB+18] Ryan Babbush, Craig Gidney, Dominic W Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. Encoding electronic spectra in quantum circuits with linear T complexity. *Physical Review X*, 8(4):041015, 2018.

[BGDM+10] Jean-Luc Beuchat, Jorge E González-Díaz, Shigeo Mitsunari, Eiji Okamoto, Francisco Rodríguez-Henríquez, and Tadanori Teruya. High-speed software implementation of the optimal ate pairing over Barreto–Naehrig curves. In *Pairing-Based Cryptography-Pairing 2010: 4th International Conference, Yamanaka Hot Spring, Japan, December 2010. Proceedings 4*, pages 21–39. Springer, 2010.

[BGGM14] Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. Discrete logarithms in GF(p̃2) — 160 digits, 2014. Announcement to the nmbrthry list: https://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind1406&L=NMBRTHRY&P=R658&1=NMBRTHRY&9=A&J=on&d=No+Match%3BMatch%3BMatches&z=4.

[BGGM15] Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. Improving NFS for the Discrete Logarithm Problem in Non-prime Finite Fields. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Proceedings, Part {I}*, volume 9056, pages 129–155, Sofia, Bulgaria, April 2015. Springer.

[BGGM22] Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. Improvements to the number field sieve for non-prime finite fields. https://inria.hal.science/hal-01052449, August 2022.

[BGJT14] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 1–16, Copenhagen, Denmark, May 2014. Springer.

[BGK15] Razvan Barbulescu, Pierrick Gaudry, and Thorsten Kleinjung. The Tower Number Field Sieve. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015*, volume 9453 of *Advances in cryptology–Asiacrypt 2015*, pages 31–58, Auckland, New Zealand, November 2015. International Association of Cryptologic Research, Springer.

[BJ24] Razvan Barbulescu and Florent Jouve. ECM And The Elliott-Halberstam Conjecture For Quadratic Fields. *Acta Arithmetica*, 2024.

[BJJ10] Jean-François Biasse and Michael J Jacobson Jr. Practical improvements to class group and regulator computation of real quadratic fields. In *International Algorithmic Number Theory Symposium*, pages 50–65. Springer, 2010.

[BK98] Joe Buhler and Neal Koblitz. Lattice basis reduction, Jacobi sums and hyperelliptic cryptosystems. *Bulletin of the Australian Mathematical Society*, 58:147–154, 1998.

[BL17] Razvan Barbulescu and Armand Lachand. Some mathematical remarks on the polynomial selection in NFS. *Mathematics of Computation*, 86:397–418, 2017.

[BLTMR+09] Jean-Luc Beuchat, Emmanuel López-Trejo, Luis Martínez-Ramos, Shigeo Mitsunari, and Francisco Rodríguez-Henríquez. Multi-core implementation of the Tate pairing over supersingular elliptic curves. In *International Conference on Cryptology and Network Security*, pages 413–432. Springer, 2009.

[BN05] Paulo SLM Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *International workshop on selected areas in cryptography*, pages 319–331. Springer, 2005.

[BP14] Razvan Barbulescu and Cécile Pierrot. The Multiple Number Field Sieve for Medium and High Characteristic Finite Fields. *LMS Journal of Computation and Mathematics*, 17:230–246, 2014.

[BP23] Razvan Barbulescu and Adrien Poulalion. The special case of cyclotomic fields in quantum algorithms for unit groups. In Nadia El Mrabet, Luca de Feo, and Sylvain Duquesne, editors, *AFRICACRYPT 2023*, volume 14064 of *Progress in Cryptology – AFRICACRYPT 2023*, page 229, Soussa, Tunisia, July 2023. Ministry of Communication Technologies of Tunisia and in partnership with the International association of cryptologic research (IACR), Springer.

[Bra93] Stefan A Brands. An efficient off-line electronic cash system based on the representation problem, 1993.

[BS16] Jean-François Biasse and Fang Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 893–902. SIAM, 2016.

[BS22] Razvan Barbulescu and Sudarshan Shinde. A classification of ECM-friendly families using modular curves. *Mathematics of Computation*, (91):1405–1436, 2022.

[Can87] David G Cantor. Computing in the Jacobian of a hyperelliptic curve. *Mathematics of computation*, 48(177):95–101, 1987.

[CCJ+16] Lily Chen, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray A Perlner, and Daniel Smith-Tone. *Report on post-quantum cryptography*, volume 12. US Department of Commerce, National Institute of Standards and Technology . . . , 2016.

[CDPR16] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 559–585. Springer, 2016.

[Coj17] Alina Carmen Cojocaru. Primes, elliptic curves and cyclic groups: a synopsis. In *Revue Roumain de Mathématiques Pures et Appliquées, Invited contributions to the Eighth Congress of Romanian Mathematicians (Iasi, 2015)*, volume 62, 2017.

[CP01] Clifford Cocks and RGE Pinch. Identity-based cryptosystems based on the weil pairing, 2001. Unpublished manuscript.

[dBDF20] Koen de Boer, Léo Ducas, and Serge Fehr. On the quantum complexity of the continuous hidden subgroup problem. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 341–370. Springer, 2020.

[DEM05] Régis Dupont, Andreas Enge, and François Morain. Building curves with arbitrary small MOV degree over finite prime fields. *Journal of Cryptology*, 18:79–89, 2005.

[DH76] Whitfield Diffie and Martin Hellman. New direction in cryptography. *IEEE Trans. Inform. Theory*, 22:472–492, 1976.

[EG24] Martin Ekerå and Joel Gärtner. Extending Regev's factoring algorithm to compute discrete logarithms. In *International Conference on Post-Quantum Cryptography–PQC 2024*, pages 211–242. Springer, 2024.

[Eke20] Martin Ekerå. On post-processing in the quantum algorithm for computing short discrete logarithms. *Designs, Codes and Cryptography*, 88(11):2313–2335, 2020.

[Eke23] Martin Ekerå. On the success probability of the quantum algorithm for the short dlp. *arXiv preprint arXiv:2309.01754*, 2023.

[Eng02] Andreas Enge. Computing discrete logarithms in high-genus hyperelliptic jacobians in provably subexponential time. *Mathematics of Computation*, 71(238):729–742, 2002.

[FST10] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23:224–280, 2010.

[GMT20] Aurore Guillevic, Simon Masson, and Emmanuel Thomé. Cocks–Pinch curves of embedding degrees five to eight and optimal ate pairing computation. *Designs, Codes and Cryptography*, 88(6):1047–1081, 2020.

[GT07] Pierrick Gaudry and Emmanuel Thomé. The mpfq library and implementing curve-based key exchanges. In *SPEED: software performance enhancement for encryption and decryption*, pages 49–64, 2007.

[Hal07] Sean Hallgren. Polynomial-time quantum algorithms for pell's equation and the principal ideal problem. *Journal of the ACM (JACM)*, 54(1):1–19, 2007.

[JLSV06] Antoine Joux, Reynald Lercier, Nigel Smart, and Frederik Vercauteren. The number field sieve in the medium prime case. In *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture notes in computer science*, pages 326–344. Springer, 2006.

[JP13] Antoine Joux and Cécile Pierrot. The special number field sieve in: Application to pairing-friendly constructions. In *International Conference on Pairing-Based Cryptography – Pairing 2013*, pages 45–61. Springer, 2013.

[KB16] Taechan Kim and Razvan Barbulescu. Extended Tower Number Field Sieve. In Jonathan Katz Matthew Robshaw, editor, *CRYPTO 2016*, volume 9814 of *Advances in cryptology – CRYPTO 2016–Part I*, pages 543–571, Santa Barbara, United States, August 2016. International association of cryptologic research, Springer.

[Kit96] A Kitaev. Quantum measurements and the abelian stabilizer problem. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 3, page 1, 1996.

[KJ17] Taechan Kim and Jinhyuck Jeong. Extended tower number field sieve with application to finite fields of arbitrary composite extension degree. In *Public-Key Cryptography–PKC 2017*, pages 388–408. Springer, 2017.

[KMRVVK24] Gregory D Kahanamoku-Meyer, Seyoon Ragavan, Vinod Vaikuntanathan, and Katherine Van Kirk. The Jacobi factoring circuit: Quantum factoring with near-linear gates and sublinear space and depth. *arXiv preprint arXiv:2412.12558*, 2024.

[Knu99] Erik Woodward Knudsen. Elliptic scalar multiplication using point halving. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 135–149. Springer, 1999.

[Kob88] Neal Koblitz. Primality of the number of points on an elliptic curve over a finite field. *Pacific journal of mathematics*, 131(1):157–165, 1988.

[Kob89] Neal Koblitz. Hyperelliptic cryptosystems. *Journal of Cryptology*, 1:139–150, 1989.

[Kra22] M Kraitchik. Théorie des nombres, vol. 1. *Gauthier-Villars, Paris*, 1922.

[Lam15] Youness Lamzouri. The distribution of Euler–Kronecker constants of quadratic fields. *Journal of Mathematical Analysis and Applications*, 432(2):632–653, 2015.

[Lan78] Serge Lang. *Elliptic curves: Diophantine analysis*, volume 231 of *Grundlehren dermathematischen Wissenschaften*. 1978.

[Lan83] Serge Lang. Conjectured Diophantine estimates on elliptic curves. *Arithmetic and Geometry: Papers Dedicated to IR Shafarevich on the Occasion of His Sixtieth Birthday Volume I Arithmetic*, pages 155–171, 1983.

[Len01] Arjen K Lenstra. Unbelievable security matching aes security using public key systems. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 67–86. Springer, 2001.

[LPDS12] Jun Li, Xinhua Peng, Jiangfeng Du, and Dieter Suter. An efficient exact quantum algorithm for the integer square-free decomposition problem. *Scientific reports*, 2(1):260, 2012.

[LWX20] Jianya Liu, Jie Wu, and Ping Xi. Primes in arithmetic progressions with friable indices. *Science China Mathematics*, 63(1):23–38, 2020.

[MNT01] Atsuko Miyaji, Masaki Nakabayashi, and Shunzou Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 84(5):1234–1243, 2001.

[Mos18] Michele Mosca. Cybersecurity in an era with quantum computers: Will we be ready? *IEEE Security & Privacy*, 16(5):38–41, 2018.

[MPR+24] Dustin Moody, Ray Perlner, Andrew Regenscheid, Angela Robinson, and David Cooper. Transition to post-quantum cryptography standards. Technical report, National Institute of Standards and Technology, 2024.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 333–342, 2009.

[Per01] René Peralta. Elliptic curve factorization using a "partially oblivious" function. In *Cryptography and Computational Number Theory*, pages 123–128. Springer, 2001.

[PH78] Stephen Pohlig and Martin Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (corresp.). *IEEE Transactions on information Theory*, 24(1):106–110, 1978.

[Pil24] Cédric Pilatte. Unconditional correctness of recent quantum algorithms for factoring and computing discrete logarithms. *arXiv preprint arXiv:2404.16450*, 2024.

[Pol93] John M Pollard. The lattice sieve. In *The development of the number field sieve*, pages 43–49. Springer, 1993.

[Pol16] Paul Pollack. A Titchmarsh divisor problem for elliptic curves. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 160, pages 167–189. Cambridge University Press, 2016.

[PS95] Carl Pomerance and Jonathan Sorenson. Counting the integers factorable via cyclotomic methods. *Journal of Algorithms*, 19(2):250–265, 1995.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[Reg25] Oded Regev. An efficient quantum factoring algorithm. *Journal of the ACM*, 2025.

[RS02] Karl Rubin and Alice Silverberg. Ranks of elliptic curves. *Bulletin of the American Mathematical Society*, 39(4):455–474, 2002.

[Sch00] Oliver Schirokauer. Using number fields to compute logarithms in finite fields. *Mathematics of Computation*, 69(231):1267–1283, 2000.

[Sho94] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science–SFCS'94*, pages 124–134. Ieee, 1994.

[Sil09] Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer, 2009.

[Vol00] José Felipe Voloch. Jacobians of curves over finite fields. *Rocky Montains Journal of Mathematics*, 30(2), 2000.

[vzGG03] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge University Press, 2003.

[Wan18] Zhiwei Wang. Autour des plus grands facteurs premiers d'entiers consécutifs voisins d'un entier criblé. *Quarterly Journal of Mathematics*, 69(3):995–1013, 2018.

[Was12] Lawrence C Washington. *Introduction to cyclotomic fields*, volume 83. Springer Science & Business Media, 2012.