UNIVERSITY OF CALIFORNIA SAN DIEGO

# *p*-adic Integration on Modular Curves and Code-Based Cryptography

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Mathematics

by

Jun Bo Lau

Committee in charge:

Professor Kiran Kedlaya, Chair
Professor Russell Impagliazzo
Professor Jonathan Novak
Professor Cristian Popescu
Professor Claus Sorensen

2023

Copyright

Jun Bo Lau, 2023

All rights reserved.

The dissertation of Jun Bo Lau is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

________________________________________

________________________________________

________________________________________

________________________________________

________________________________________

Chair

University of California San Diego

2023

# EPIGRAPH

*A careful quotation*

*conveys brilliance.*

—Smarty Pants

TABLE OF CONTENTS

# ACKNOWLEDGEMENTS

Thanks to whoever deserves credit for Blacks Beach, Porters Pub, and every coffee shop in San Diego.

Thanks also to hottubs.

# VITA

| | |
|---|---|
| 2017 | M.Math University of Warwick, U.K. |
| 2017-2023 | Graduate Teaching Assistant, University of California, San Diego |
| 2023 | Ph. D. in Mathematics, University of California, San Diego |

# ABSTRACT

In this dissertation, we study two problems arising from arithmetic geometry.

Falting's theorem states that there are only finitely many rational points on curves of genus greater than 1. However, an explicit determination of all such points on a curve remains a hard problem. There are various approaches to computing rational points on higher genus curves and we use Coleman's theory of $p$-adic line integrals to study a particular class of curves with rich arithmetic origins, namely, the modular curves. In join work with Chen and Kedlaya, we implement a new algorithm that does not use the models of the modular curves and illustrate this method through the computation of several examples.

On the other hand, in anticipation of the development of powerful quantum computers in the next few decades, we study cryptosystems that rely on the hardness of certain number theoretical problems. In particular, we investigate BIKE, a cryptosystem presented as one of the candidates for the National Institute of Standards and Technology Post-Quantum Cryptography Standardization Process. We identified several factors that affect the security of the code-based cryptosystem as a potential quantum-attack-resistant candidate for real world applications through extensive simulations.

# Part I

# Coleman Integration on Modular Curves

# Chapter 1

# Preliminaries

All curves in this paper are smooth, projective and geometrically irreducible with good reduction at a prime $p$.

## 1.1 Introduction

Some of the oldest questions in number theory can be reformulated in modern terms: given a finite list of polynomials, what are the integer or rational solutions to this set of equations? In fact, these solutions can be viewed as integer or rational solutions of geometric objects – curves, surfaces or higher dimensional objects.

In this project, we focus on the case of curves. A remarkable result, formulated by Mordell in 1922 and proved by Faltings in 1983, states that for curves of higher genus, there are only finitely many rational points on them.

**Theorem 1.1.1.** *(Mordell's conjecture/ Faltings's theorem) Let $X/\mathbb{Q}$ be a curve of genus $g \geq 2$, then the set of rational points $X(\mathbb{Q})$ is finite.*

However, Faltings's proofs are not effective, i.e., there is no way of explicitly determining the complete set of rational points on the curve. Before Faltings, Chabauty developed a method in this direction with the condition that if the rank of the Jacobian of the curve is strictly less than the genus, then one could compute this set of points. In [Col85b, Col85a] Coleman defined $p$-adic line integrals and re-interpreted Chabauty's method to explicitly compute the set of rational points. These Coleman integrals provide an effective method to problems in arithmetic geometry, including but not limited to, torsion points on Jacobians of curves ( Manin-Mumford conjecture), $p$-adic heights on curves, $p$-adic polylogarithms, Mordell conjecture (rational points), etc. In [BD18, BD17], Balakrishnan and Dogra developed quadratic Chabauty as a computational tool to study the set of rational points as long as the curve satisfies a certain quadratic Chabauty bound, involving the rank of the Jacobian, genus and Néron-Severi rank of the Jacobian.

There are several approaches to numerically compute these Coleman integrals. Wetherell [Wet97] combined the certan properties of Coleman integrals and the arithmetic of the Jacobian to compute $\int_D \omega$, where $D$ is a divisor in the Picard group and $\omega$ is a holomorphic differential on the curve. The next approach relies on computing the Frobenius action in $p$-adic cohomology following Dwork's principle of analytic continuation along the Frobenius [BBK10, Tui16, Tui17, BT20]. However, both of these approaches have their shortcomings – Wetherell's method requires an explicit divisor in order to reduce the computation to a power series integration ("tiny integrals") and the second method requires an explicit equation of the curves as input.

We turn our attention to computing Coleman integrals on modular curves. The set of rational points on modular curves has special arithmetic meaning. For instance, the set of rational points $X_0(N)(\mathbb{Q})$ correspond to the torsion points of elliptic curves (Mazur's theorem). Another motivation to study modular curves comes from Serre's Uniformity Conjecture. Let $E$ be an elliptic curve defined over $K$. The group of $p$-torsion points $E[p](\bar{K})$ is isomorphic to $(\mathbb{Z}/p\mathbb{Z})^2$ and is acted upon by the absolute Galois group $\mathrm{Gal}(\bar{K}/K)$, giving rise to a representation $\rho_{p,E} : \mathrm{Gal}(\bar{K}/K) \to \mathrm{GL}_2(\mathbb{F}_p)$. In [Ser72], Serre proved the following:

**Theorem 1.1.2.** *Suppose that $E$ does not have complex multiplication. Then there exists a number $N(E)$ such that $\rho_{p,E}$ is surjective for all $p > N(E)$.*

In the same paper, he posed the following question:

**Conjecture 1.1.3.** (Serre's Uniformity Conjecture) Given a number field $K$, then there exist a constant $N_K > 0$ such that for any elliptic curve $E$ defined over $K$ without complex multiplication, the corresponding Galois representation $\rho_{p,E}$ is surjective for all primes $p > N_K$.

Since modular curves parametrise elliptic curves with torsion data, this can be formulated in terms of rational points on modular curves:

**Conjecture 1.1.4.** (Serre's Uniformity Conjecture) Let $H \leq \mathrm{GL}_2(\mathbb{F}_p)$ be a proper subgroup such that the determinant map $\det : H \to \mathbb{F}_p^\times$ is surjective, then there exist a constant $N_K > 0$ such that for any prime $p > N_K$, the associated modular curve $X_H(p)$ has $K$-rational points coming only from cusps and elliptic curves with complex multiplication.

If $\rho_{p,E}$ is not surjective, the image lies inside some maximal proper subgroup of $\mathrm{GL}_2(\mathbb{F}_p)$. Therefore, one could prove the conjecture by showing that for $p$ large enough, the image of $\rho_{p,E}$ does not lie in any maximal subgroup. The classification of maximal subgroups of $\mathrm{GL}_2(\mathbb{F}_p)$ is known, originally due to [Dic]:

**Theorem 1.1.5.** *Let $H \leq \mathrm{GL}_2(\mathbb{Z}/p\mathbb{Z})$ not containing $\mathrm{SL}_2(\mathbb{Z}/p\mathbb{Z})$. Up to conjugacy, $H$ is one of the following:*

- *(Borel) $H \subseteq B_0(p) = \{(\begin{smallmatrix} * & * \\ 0 & * \end{smallmatrix})\}$*
- *(Normaliser of split Cartan) $H \subseteq N_s^+(p) = \{\begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}, \begin{pmatrix} 0 & \alpha \\ \beta & 0 \end{pmatrix} : \alpha, \beta \in \mathbb{F}_p^\times\}$*
- *(Normaliser of non-split Cartan) $H \subseteq N_s^+(p) = \{(\begin{smallmatrix} \alpha & 0 \\ 0 & \alpha^p \end{smallmatrix}), (\begin{smallmatrix} 0 & \alpha \\ \alpha^p & 0 \end{smallmatrix}) : \alpha \in \mathbb{F}_{p^2}^\times\}$*
- *(Exceptional) The image of $H$ in $\mathrm{PGL}_2(\mathbb{F}_p)$ is isomorphic to $A_4, S_4$ or $A_5$.*

Most of the cases have been resolved [Maz78, BPR13, BP11, Ser72], except for the normaliser of non-split Cartan. There has been some progress using quadratic Chabauty to find the rational points of the modular curve corresponding to the nonsplit Cartan of level 13 [BDM$^{+}$19] and level 17 [BDM$^{+}$21].

Since most modular curves satisfy the quadratic Chabauty bound [Sik17], we provide a model-free algorithm to compute Coleman integrals on modular curves arising arising from Serre's Uniformity Conjecture.

## 1.2 Background

### 1.2.1 Modular forms

In this section, we give a brief introduction of modular forms, following [DS05].

Let $\mathbb{H} := \{\tau \in \mathbb{C} : Im(\tau) > 0\}$ be the upper half complex plane. The special linear group $\mathrm{SL}_2(\mathbb{Z})$ acts on $\mathbb{H}$ via fractional linear transformations:

$$\gamma \cdot \tau = \frac{a\tau + b}{c\tau + d}$$

where $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right), \tau \in \mathbb{H}$.

**Definition 1.2.1.** Let $f : \mathbb{H} \to \mathbb{C}$ be a function and $k \in \mathbb{Z}$.

- The *automorphy factor* is a function

  $$\begin{aligned} j : \mathrm{GL}_2^+(\mathbb{R}) \times \mathbb{H} &\to \mathbb{C} \\ (\gamma, z) &\mapsto cz + d \end{aligned}$$

  where $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$.

- The *weight-k slash operator* is defined as

  $$\begin{aligned} (\,\cdot\,)|_k(\,\cdot\,) : \mathrm{Hom}(\mathbb{H}, \mathbb{C}) \times \mathrm{GL}_2^+(\mathbb{R}) &\to \mathrm{Hom}(\mathbb{H}, \mathbb{C}) \\ (f(z), \gamma) &\mapsto (f|_k\gamma)(z) := (\det\gamma)^{k-1} j(\gamma, z)^{-k} f(\gamma \cdot z). \end{aligned}$$

The automorphy factory satisfies a cocycle relation $j(\gamma_1\gamma 2, z) = j(\gamma_1, \gamma_2 z)j(\gamma_2, z)$ which implies that $\mathrm{GL}_2^+(\mathbb{R})$ acts on $\mathrm{Hom}(\mathbb{H}, \mathbb{C})$ via $f|_k\gamma_1\gamma_2 = (f|_k\gamma_1)|_k\gamma_2$.

Consider the projection map $\pi : \mathrm{SL}_2(\mathbb{Z}) \to \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$. We define congruence subgroups in the following way.

**Example 1.2.2.** Here are some common examples of preimages of $\pi$:

- $\Gamma(N) = \pi^{-1}(\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)) = \{\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}_2(\mathbb{Z}) : \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \equiv \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) \pmod N\}$.

- $\Gamma_1(N) = \pi^{-1}(\{\left(\begin{smallmatrix} 1 & * \\ 0 & 1 \end{smallmatrix}\right)\}) = \{\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}_2(\mathbb{Z}) : \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \equiv \left(\begin{smallmatrix} 1 & * \\ 0 & 1 \end{smallmatrix}\right) \pmod N\}$.

- $\Gamma_0(N) = \pi^{-1}(\{\left(\begin{smallmatrix} * & * \\ 0 & * \end{smallmatrix}\right)\}) = \{\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}_2(\mathbb{Z}) : \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \equiv \left(\begin{smallmatrix} * & * \\ 0 & * \end{smallmatrix}\right) \pmod N\}$.

**Definition 1.2.3.** $\Gamma \leq \mathrm{SL}_2(\mathbb{Z})$ is a *congruence subgroup* if there exists an integer $N \geq 1$ such that $\Gamma(N) \leq \Gamma$. The minimal such $N$ is called the *level* of $\Gamma$.

It follows immediately that congruence subgroups of $\mathrm{SL}_2(\mathbb{Z})$ have finite index and correspond to subgroups of $\mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$. The above examples are all congruence subgroups of level $N$.

**Definition 1.2.4.** Let $\Gamma \leq \mathrm{SL}_2(\mathbb{Z})$ be a congruence subgroup of level $N$, $k \geq 0$ an integer. We say a function $f : \mathbb{H} \to \mathbb{C}$ is a *modular form of weight k with level* $\Gamma$ if

1. $f$ is holomorphic,

2. $f$ is weight-$k$ invariant under $\Gamma$, i.e., $f|_k\gamma = f$ for all $\gamma \in \Gamma$,

3. $f|_k\alpha$ is holomorphic at $\infty$ for all $\alpha \in \mathrm{SL}_2(\mathbb{Z})$, i.e., $(f|_k\alpha)(z)$ is bounded as $z \to i\infty$.

If, in addition, $f|_k\alpha$ vanishes at infinity for all $\alpha \in \mathrm{SL}_2(\mathbb{Z})$, we say that $f$ is a *cusp form*. We denote the set of weight-$k$ modular forms with respect to $\Gamma$ (resp. cusp forms) as $\mathscr{M}_k(\Gamma)$ (resp. $\mathscr{S}_k(\Gamma)$).

Suppose $f$ is a modular form of weight $k$ with level $\Gamma$. Since $\Gamma$ is a congruence subgroup, $\left(\begin{smallmatrix} 1 & h \\ 0 & 1 \end{smallmatrix}\right) \in \Gamma$ for some minimal integer $h \geq 1$, this integer is the *width* of the cusp $\infty$. Since a modular form satisfies $f|_k\gamma = f$ for $\gamma \in \Gamma$, we have $(f|_k\left(\begin{smallmatrix} 1 & h \\ 0 & 1 \end{smallmatrix}\right))(z) = f(z+h) = f(z)$, so $f(z)$ is $h\mathbb{Z}$-periodic and admits a Fourier expansion $f(\tau) = \sum_{n=0}^{\infty} a_n q_h^n$ where $q_h = \exp(2\pi i\tau/h)$. The third condition of modular forms implies that the Fourier expansion begins at index 0 and cusp forms satisfy $a_0 = 0$.

**Example 1.2.5.** Let $G_k(\tau) = \sum_{(c,d)\neq(0,0)} 1/(c\tau+d)^k$. This is a modular form of weight $k$ for $\mathrm{SL}_2(\mathbb{Z})$ called *Eisenstein series*.

The *j-invariant* is a modular form of weight 0, i.e., a modular function and an element of $\mathbb{C}(X(\mathrm{SL}_2(\mathbb{Z})))$, with $q$-expansion:

$$j : \mathbb{H} \to \mathbb{C}, j(\tau) = 1728\frac{(60G_4(\tau))^3}{(60G_4(\tau))^3 - 27(140G_6(\tau))^2} = \frac{1}{q} + 744 + 196884q + \dots.$$

It is a standard result that $\mathscr{M}_k(\Gamma) \supseteq \mathscr{S}_k(\Gamma)$ are finite dimensional complex vector spaces. Modular forms and modular curves are related by the fact that there is an isomorphism between the space of weight 2 cusp forms and the space of holomorphic differentials on the modular curve $X(\Gamma)$.

$$\mathscr{S}_2(\Gamma) \xrightarrow{\cong} H^0(X(\Gamma), \Omega^1)$$
$$f(\tau) \mapsto f(\tau)d\tau.$$

### 1.2.2 Modular curves

In this section, we define our object of study. Modular curves have rich structures as Riemann surfaces, algebraic curves and moduli spaces of elliptic curves (with some torsion information). We frequently use properties from various perspectives interchangeably.

### As Riemann surfaces

Let $\Gamma \leq \mathrm{SL}_2(\mathbb{Z})$ be a subgroup of finite index. $\mathbb{H}$ inherits the Euclidean topology from $\mathbb{C}$ and so $Y(\Gamma) := \Gamma\backslash\mathbb{H}$ carries the quotient topology that is Hausdorff. $Y(\Gamma)$ can be compactified by adjoining cusps, which are orbits of $\mathbb{P}^1(\mathbb{Q})$ under the action of $\Gamma$. The resulting quotient space $X(\Gamma) := \Gamma\backslash\mathbb{H}^*$ where $\mathbb{H}^* := \mathbb{H} \cup \mathbb{P}^1(\mathbb{Q})$ is called the modular curve associated to $\Gamma$. One could further show that by considering elliptic points and cusps, one can choose suitable charts, therefore giving $Y(\Gamma)$ and $X(\Gamma)$ the structure of Riemann surface.

This approach allows us to use techniques from Riemann surfaces, e.g., genus/ramification theory, Riemann-Hurwitz formula, Riemann-Roch, etc. to study modular curves.

### As algebraic curves

For a finite index subgroup $\Gamma \leq \mathrm{SL}_2(\mathbb{Z})$. The associated modular curve $X(\Gamma)$ has the structure of a compact Riemann surface. Compact Riemann surfaces and complex algebraic curves are equivalent notions [For81]. Note that we are also considering modular curves where the determinant map on the subgroup $H \leq \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ is surjective. By Theorem 7.6.3 in [DS05], these algebraic curves are in fact defined over $\mathbb{Q}$. We have a Galois-theoretic correspondence between curves and their function fields:

**Theorem 1.2.6.** *(Curves-Fields Correspondence) For any field k, there is a bijection:*

$$\{C/k \textit{ smooth projective algebraic curves}\}/\cong \;\leftrightarrow\; \{K/k \textit{ function field extensions over } k\}/\sim$$

$$C \mapsto k(C)$$

*Furthermore, this is contravariant: a nonconstant morphism from algebraic curves $C$ to $C'$ over $k$ corresponds to a field morphism from $k(C')$ to $k(C)$.*

The above theorem allows us to work with simpler objects, i.e., we can replace curves and their morphisms by fields and field injections. In particular, the function field of the modular curve $X(\Gamma)$ consists of modular functions of weight 0 and level $\Gamma$.

### As moduli spaces of elliptic curves

For each $\tau \in \mathbb{H}$, one could associate it with a lattice $\Lambda_\tau := \mathbb{Z} + \tau \cdot \mathbb{Z} \subseteq \mathbb{C}$. The resulting quotient space $\mathbb{C}/\Lambda_\tau$ is a compact Riemann surface of genus 1, an elliptic curve. Conversely, for any elliptic curve, as a genus 1 compact Riemann surface, the homology group of the elliptic curve $H_1(E,\mathbb{Z})$ is generated by two loops, $\gamma_1, \gamma_2$. For an invariant differential $\omega$ of the elliptic curve, we can construct the lattice generated by the periods $\Lambda_E = (\int_{\gamma_1} \omega) \cdot \mathbb{Z} + (\int_{\gamma_2} \omega) \cdot \mathbb{Z}$. This can be renormalised so that $\Lambda_E = \mathbb{Z} + \tau \cdot \mathbb{Z}$ with $\tau = (\int_{\gamma_1} \omega)/(\int_{\gamma_2} \omega) \in \mathbb{H}$. In particular the points on $\mathbb{H}$ correspond to elliptic curves.

For $\Gamma \leq \mathrm{SL}_2(\mathbb{Z})$, the modular curve $X(\Gamma)(\bar{\mathbb{Q}})$ parametrise elliptic curves with some torsion data, i.e., a pair $(E,\phi)$ where $E$ is an elliptic curve defined over $\bar{\mathbb{Q}}$ and $\phi$ is an isomorphism of its $N$-torsion points $\phi : E[N] \to (\mathbb{Z}/N\mathbb{Z})^2$. Two points $(E_1,\phi_1),(E_2,\phi_2)$ are isomorphic if there is an isomorphism of elliptic curves $\psi : E_1 \to E_2$ and some matrix $M \in \Gamma$ such that the diagram commutes:

$$\begin{tikzcd}
E_1[N] \arrow[r, "\phi_1"] \arrow[d, "\psi"] & (\mathbb{Z}/N\mathbb{Z})^2 \arrow[d, "M"] \\
E_2[N] \arrow[r, "\phi_2"] & (\mathbb{Z}/N\mathbb{Z})^2.
\end{tikzcd}$$

Furthermore, there is an action of the absolute Galois group $Gal(\bar{\mathbb{Q}}/\mathbb{Q})$ on $(E,\phi)$ and we say that $(E,\phi)$ is a $\mathbb{Q}$-rational point if it is invariant under the action. We can view points on modular curves as elliptic curves with certain torsion structures which allows us to apply properties of elliptic curves to study the rational points on $X(\Gamma)$.

**Example 1.2.7.** Let $H \leq \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ be a subgroup such that

- $-I \in H$,
- the determinant map $\det : H \to (\mathbb{Z}/N\mathbb{Z})^\times$ is surjective.

Then for an integer $N \geq 1$, we have the congruence subgroup $\Gamma_H(N) = \{A \in \mathrm{SL}_2(\mathbb{Z}) : A \pmod{N} \in H\}$, which gives rise to the modular curves $X_H := X(\Gamma_H(N))$.

Following Example 1.2.2, the corresponding modular curves parametrise:

- $X(N) := X(\Gamma(N))$ consists of $(E,(P,Q))$ an elliptic curve and a pair of points generating the $N$-torsion subgroup of $E$.
- $X_1(N) := X(\Gamma_1(N))$ consists of $(E,Q)$ an elliptic curve and a point of order $N$.
- $X_0(N) := X(\Gamma_0(N))$ consists of $(E,C)$ an elliptic curve and a cyclic subgroup of order $N$.

### 1.2.3 Hecke operators

We begin with the definition of Hecke operators as operators on spaces of modular forms. These are used in conjunction with spectral theory to show that the inner product space of modular forms contains a basis of modular forms that are eigenvectors under the Hecke operators $\{T_p\}_p$. Hecke operators are defined on modular forms and modular curves. We use both the transcendental and algebraic/geometric definitions of Hecke operators in our algorithm.

**Definition 1.2.8.** Let $\Gamma_1, \Gamma_2$ be congruence subgroups of $SL_2(\mathbb{Z})$ and $\alpha \in GL_2^+(\mathbb{Q})$.

- We define the *double coset* $\Gamma_1 \alpha \Gamma_2$ as the set

$$\Gamma_1\alpha\Gamma_2 := \{\gamma_1\alpha\gamma_2 : \gamma_1 \in \Gamma_1, \gamma_2 \in \Gamma_2\}$$

- This gives rise to the *double coset operators*:

$$\begin{aligned}(\,\cdot\,)|_k[\Gamma_1\alpha\Gamma_2] : \mathscr{M}_k(\Gamma_1) &\to \mathscr{M}_k(\Gamma_2)\\ f(\tau) &\mapsto f|_k\Gamma_1\alpha\Gamma_2 := \sum_i f|_k\beta_i\end{aligned}$$

  where $\Gamma_1\alpha\Gamma_2 = \bigcup_i \Gamma_1\beta_i$ is a (finite) disjoint coset decomposition that does not depend on the choice of decomposition. This map restricts to an operator on the space of cusp forms $(\,\cdot\,)|_k[\Gamma_1\alpha\Gamma_2] : \mathscr{S}_k(\Gamma_1) \to \mathscr{S}_k(\Gamma_2)$.

We follow the approach [Ass20] to define Hecke operators.

**Definition 1.2.9.** Fix a congruence subgroup $\Gamma$ with $\bar{\Gamma} \leq SL_2(\mathbb{Z}/N\mathbb{Z})$. Let $\alpha \in M_2(\mathbb{Z})$ such that $\det(\alpha) \in \det(\bar{\Gamma})$ and $\alpha \pmod N \in \bar{\Gamma}$. We define the Hecke operator as

$$T_p = T_\alpha = (\,\cdot\,)|_k[\Gamma\alpha\Gamma]$$

**Example 1.2.10.** ([DS05] Prop. 5.2.1) The theory of Hecke operators can be made explicit for certain congruence subgroups. The Hecke operator $T_p = [\Gamma_1(N)\left(\begin{smallmatrix}1 & 0\\ 0 & p\end{smallmatrix}\right)\Gamma_1(N)]_k$ on $\mathscr{M}_k(\Gamma_1(N))$ has the following formulae:

$$T_p f = \begin{cases} \sum_{i=0}^{p-1} f|_k \begin{pmatrix}1 & j\\ 0 & p\end{pmatrix}, & \text{if } p|N,\\ \sum_{i=0}^{p-1} f|_k \begin{pmatrix}1 & j\\ 0 & p\end{pmatrix} + f|_k(\left(\begin{smallmatrix}m & n\\ N & p\end{smallmatrix}\right)\left(\begin{smallmatrix}p & 0\\ 0 & 1\end{smallmatrix}\right)), & \text{if } p \nmid N, \text{ where } mp - nN = 1.\end{cases}$$

There is also an algebraic/geometric interpretation of the double coset operator as a morphism of divisor groups. For $\Gamma_1, \Gamma_2$ congruence subgroups, $\alpha \in GL_2^+(\mathbb{Q})$, $\Gamma_3 := \alpha^{-1}\Gamma_1\alpha \cap \Gamma_2$ and $\Gamma_3' := \alpha\Gamma_3\alpha^{-1}$. Since points on the modular curve $X(\Gamma)$ have the form $\Gamma\tau$, we have a diagram at the level of groups which induces a diagram on modular curves:

$$\begin{tikzcd}
\Gamma_2 & \Gamma_3 \arrow[l, hook'] \arrow[r, "\cong"] & \Gamma_3' \arrow[r, hook] & \Gamma_1 \\
X_2 & X_3 \arrow[l, "\pi_2"'] \arrow[r, "\cong"] & X_3' \arrow[r, "\pi_1"] & X_1
\end{tikzcd}$$

Suppose $\Gamma_3/\Gamma_2 = \bigcup_j \Gamma_3\gamma_{2,j}$ and $\beta_j = \alpha\gamma_{2,j}$. Then the double coset operator induces a map on the divisor groups after $\mathbb{Z}$-linear extension:

$$\begin{aligned}
\mathrm{Div}(X_2) &\to \mathrm{Div}(X_1) \\
\Gamma_2\tau &\mapsto \sum_j \Gamma_1\beta_j\tau
\end{aligned}$$

Specialising to the case of Hecke operator, we obtain a similar diagram.

We could benefit from the moduli interpretation of modular curves for the case of Hecke operators by defining it as a correspondence. For $H \leq GL_2(\mathbb{Z}/N\mathbb{Z})$ and $p$ coprime to $N$, we obtain the modular curve $X_H$ and its fiber product $X_H(p) := X_0(p) \times_{X(1)} X_H$. There are two degeneracy maps $\alpha, \beta : X_H(p) \to X_H$ defining the Hecke operator at $p$ where one forgets the cyclic group of order $p$ and the other quotients out by the cyclic group of order $p$.

$$\begin{tikzcd}
& X_H(p) \arrow[dl, "\alpha"'] \arrow[dr, "\beta"] & \\
X_H \arrow[rr, dashed] & & X_H
\end{tikzcd}$$

By Picard functoriality, for a point $(E,\mathfrak{n}) \in X_H$ where the level structure $\mathfrak{n}$ is determined by $H$, we have an algebraic description of the Hecke operator at $p$:

$$T_p(E,\mathfrak{n}) := \alpha^*\beta_*(E,\mathfrak{n}) = \sum_{f:E\to E',deg(f)=p} (E', f(\mathfrak{n})).$$

### 1.2.4 Coleman integrals

Coleman's construction of $p$-adic line integrals share many similar properties as their complex-analytic analogue. Below we record some properties of Coleman integrals from [CdS88, Col85c] that will be used in our calculations.

**Theorem 1.2.11.** *Let $X/\mathbb{Q}_p$ be a smooth, projective, and geometrically irreducible curve with good reduction at p, let J be the Jacobian of X.Then there is a p-adic integral*

$$\int_P^Q \omega \in \overline{\mathbb{Q}}_p$$

*with $P, Q \in X(\overline{\mathbb{Q}}_p), \omega \in H^0(X, \Omega^1)$ satisfying:*

1. *The integral is $\overline{\mathbb{Q}}_p$ linear in $\omega$,*
2. *We have additivity of endpoints:*

$$\int_P^Q \omega = \int_P^R \omega + \int_R^Q \omega,$$

3. $$\int_P^Q \omega + \int_{P'}^{Q'} \omega = \int_P^{Q'} \omega + \int_{P'}^{Q} \omega$$

*Thus, we can define $\int_D \omega$, where $D \in Div_X^0(\overline{\mathbb{Q}}_p)$. Also, if D is principal, $\int_D \omega = 0$,*

4. *There is an open subgroup of $J(\mathbb{Q}_p)$ such that $\int_P^Q \omega$ can be computed in terms of power series in some uniformiser by formal term-by-term integration. In particular, $\int_P^P \omega = 0$,*

5. *The integral is compatible with the action of $Gal(\overline{\mathbb{Q}}_p/\mathbb{Q}_p)$. In particular, if $P, Q \in X(\mathbb{Q}_p)$ then $\int_P^Q \omega \in \mathbb{Q}_p$.*

6. *Let $P_0 \in X(\overline{\mathbb{Q}}_p)$ be fixed and $\omega \neq 0$. Then the set of $P \in X(\overline{\mathbb{Q}}_p)$ reducing to $X(\overline{\mathbb{F}}_p)$ such that $\int_{P_0}^P \omega = 0$ is finite,*

7. *If $U \subseteq X, V \subseteq Y$ are wide open subspaces of the rigid analytic spaces $X, Y$, $\omega$ a 1-form on $V$, and $\phi : U \to V$ a rigid analytic map, then we have the change of variables formula:*

$$\int_P^Q \phi^* \omega = \int_{\phi(P)}^{\phi(Q)} \omega,$$

8. *We have an analogue of the Fundamental Theorem of Calculus: $\int_P^Q df = f(Q) - f(P)$,*

**Definition 1.2.12.** The Coleman integral $\int_P^Q \omega$ is called a *tiny integral* if $P$ and $Q$ reduce to the same point in $X_{\mathbb{F}_p}(\overline{\mathbb{F}}_p)$, i.e., they lie in the same residue disc.

Explicitly, if $P$ and $Q$ are in the same residue disc, then the differential form can be expressed as a power series in terms of a uniformiser at $P$. The tiny integral can be computed by formally integrating the power series and evaluated at the endpoints:

$$\int_P^Q \omega = \int_{t(P)}^{t(Q)} \omega(t) = \int_{t(P)}^{t(Q)} \sum a_i t^i dt = \sum \frac{a_i}{i+1}(t(Q) - t(P))^{i+1}.$$

Coleman's construction is suitable for computations. In [BBK10], the authors demonstrated an algorithm to compute single Coleman integrals for hyperelliptic curves. Their method is based on Kedlaya's algorithm for computing the Frobenius action on the de Rham cohomology of hyperelliptic curves [Ked01] and this is generalized to arbitrary smooth curves [BT22, Tui16, Tui17]. Despite recent developments in this direction, the current implementations require nice affine plane models for the curves as inputs. Since modular curves tend to have large gonality, such models are not readily available and are often bottlenecks in existing algorithms.

# Chapter 2

# Coleman Integration on Modular Curves

In this section, we introduce an algorithm that computes single Coleman integrals on modular curves. The modular curves in consideration have congruence subgroups $\Gamma_H \leq SL_2(\mathbb{Z})$ where $H \leq GL_2(\mathbb{Z}/N\mathbb{Z})$ and

- $-I \in H$,
- $\det : H \to \mathbb{Z}/N\mathbb{Z}$ is surjective.

Furthermore, our method extends to the Atkin-Lehner quotients of modular curves with a slight modification, i.e., by choosing a different uniformiser.

Another innovation is that the algorithm does not make use of the affine models of the modular curves, which are often required as inputs in previous algorithms. Furthermore, we can compute Coleman integrals between any two points that are not necessarily on the same residue disc.

**TODO: review this part**

The general strategy works as follows:

1. Reduce the problem of computing arbitrary Coleman integrals into a sum of tiny integrals,

2. Find a basis of holomorphic 1-forms and a suitable uniformiser,

3. Formally integrate and evaluate at the end points.

## 2.1 Breaking the Coleman integrals into tiny integrals

Let $X/\mathbb{Q}$ be a modular curve associated to a congruence subgroup $\Gamma$, two points $Q, R \in X(\bar{\mathbb{Q}})$, $\{\omega_1, \dots, \omega_g\}$ a $\mathbb{Q}$-basis of $H^0(X, \Omega^1)$ where $g$ is the genus of the curve and $p$ a prime of good reduction on $X$.

The Hecke operator at $p$, $T_p$ acts on the weight 2 cusp forms, which corresponds to the holomorphic 1-forms:

$$T_p^* \begin{pmatrix} \omega_1 \\ \vdots \\ \omega_g \end{pmatrix} = A \begin{pmatrix} \omega_1 \\ \vdots \\ \omega_g \end{pmatrix}.$$

where $A$ is the Hecke matrix acting on the basis of cusp forms. Integrating between the points $Q$ and $R$ gives:

$$\begin{pmatrix} \int_R^Q T_p^* \omega_1 \\ \vdots \\ \int_R^Q T_p^* \omega_g \end{pmatrix} = A \begin{pmatrix} \int_R^Q \omega_1 \\ \vdots \\ \int_R^Q \omega_g \end{pmatrix}.$$

For any $\omega \in H^0(X, \Omega^1)$, using the definition of Hecke operator as a correspondence and the functoriality of Coleman integrals, we obtain the following equality:

$$\int_R^Q T_p^*(\omega) = \int_{T_p(R)}^{T_p(Q)} \omega = \sum_{i=0}^{p} \int_{R_i}^{Q_i} \omega,$$

where $T_p(Q) = \sum_{i=0}^{p} Q_i$ and $T_p(R) = \sum_{i=0}^{p} R_i$.

By considering $((p+1)\int_Q^R \omega - \int_Q^R T_p^* \omega)$, we have the following fundamental equation:

$$((p+1)I-A)\begin{pmatrix}\int_R^Q \omega_1\\ \vdots\\ \int_R^Q \omega_g\end{pmatrix} = \begin{pmatrix}\sum_{i=0}^p \int_{Q_i}^Q \omega_1 - \sum_{i=0}^p \int_{R_i}^R \omega_1\\ \vdots\\ \sum_{i=0}^p \int_{Q_i}^Q \omega_g - \sum_{i=0}^p \int_{R_i}^R \omega_g\end{pmatrix}. \tag{2.1}$$

The $Q_i$'s and $R_i$'s are by definition $p$-isogenous to $Q$ and $R$, therefore, the Eichler-Shimura relation ([DS05] Theorem 8.7.2) implies that they are in the same residue discs respectively. So the vector on the right hand side consists of sums of tiny integrals. On the left hand side, the matrix $((p+1)I-A)$ is invertible by the Ramanujan bound – the Hecke matrix $A$ has eigenvalues $\{a_p\}$ which satisfy $|a_p| \leq 2\sqrt{p}$.

From the above discussion, since any $\omega$ is a linear combination of the $\omega_j$'s, we can simultaneously compute the Coleman integrals $\int_Q^R \omega$ once we have evaluated the tiny integrals $\sum_{i=0}^p \int_{Q_i}^Q \omega$ and $\sum_{i=0}^p \int_{R_i}^R \omega$.

## 2.2 Computing a basis of cusp forms

The spaces of cusp forms for the congruence subgroups $\Gamma(N), \Gamma_1(N)$ and $\Gamma_0(N)$ are available in software packages [The22b] and [BCP97]. In [Zyw20], the author gave a method to compute the action of $SL_2(\mathbb{Z})$ and $GL_2(\mathbb{Z}/N\mathbb{Z})$ on $\mathscr{S}_k(\Gamma(N))$ , which uses the properties that [BN20]:

- There is an action of $SL_2(\mathbb{Z}/N\mathbb{Z})$ induced from $SL_2(\mathbb{Z})$ on the cusp forms,
- $\left(\begin{smallmatrix} 1 & 0 \\ 0 & d \end{smallmatrix}\right)$ acts on the coefficients of the $q$-expansion by $\zeta_N \mapsto \zeta_N^d$, where $\zeta_N$ is a $N$-th root of unity.

Furthermore, for $H \leq GL_2(\mathbb{Z}/N\mathbb{Z})$ satisfying the conditions above, $\mathscr{S}_2(\Gamma(N))^H$, the space of weight 2 cusp forms invariant under $H$ is isomorphic to $H^0(X_H, \Omega^1)$.

For congruence subgroups $\Gamma_0^+(N) := \Gamma_0(N)/w_N$ with an Atkin-Lehner involution, we modify Zywina's Magma implementation to compute our examples.

## 2.3 Hecke operators as double coset operators

Hecke operators act on both cusp forms and the divisor group of the modular curve. To compute them as a double coset operator, we need to compute the coset representatives $\Gamma_H\backslash\Gamma_H\alpha\Gamma_H$ for congruence subgroups $\Gamma_H$. A few key lemmas will give us a procedure to compute the coset representatives.

**Lemma 2.3.1.** *([DS05] Lemmata 5.1.1, 5.1.2) Let* $\Gamma, \Gamma_1, \Gamma_2$ *be congruence subgroups and* $\alpha \in GL_2^+(\mathbb{Q})$. *Then,*

1. $\alpha^{-1}\Gamma\alpha \cap SL_2(\mathbb{Z}) \leq SL_2(\mathbb{Z})$ *is a congruence subgroup.*

2. *There is a bijection:*

$$(\alpha^{-1}\Gamma_1\alpha\cap\Gamma_2)\backslash\Gamma_2 \leftrightarrow \Gamma_1\backslash\Gamma_1\alpha\Gamma_2$$
$$(\alpha^{-1}\Gamma_1\alpha\cap\Gamma_2)\gamma_2 \mapsto \Gamma_1\alpha\gamma_2$$

*More concretely,* $\{\gamma_{2,i}\}$ *is a set of coset representatives for* $(\alpha^{-1}\Gamma_1\alpha\cap\Gamma_2)\backslash\Gamma_2$ *if and only if* $\{\alpha\gamma_{2,i}\}$ *is a set of coset representatives of* $\Gamma_1\backslash\Gamma_1\alpha\Gamma_2$.

**Lemma 2.3.2.** *([Shi94] Lemma 3.29(5)) Let* $\alpha \in M_2(\mathbb{Z})$ *be such that* $\det(\alpha) = p$ *and* $\alpha \pmod N \in H$. *If* $\Gamma_H\alpha\Gamma_H = \bigcup_i \Gamma_H\alpha_i$ *is a disjoint union, then* $SL_2(\mathbb{Z})\alpha SL_2(\mathbb{Z}) = \bigcup_i SL_2(\mathbb{Z})\alpha_i$ *is a disjoint union.*

The procedure for computing the Hecke operator as a double coset operator is as follows:

1. Find $\alpha \in M_2(\mathbb{Z})$ satisfying $\det(\alpha) = p$, $\alpha \pmod N \in H$,

2. Find the coset representatives $\{\alpha_i\}$ in $(\alpha^{-1}SL_2(\mathbb{Z})\alpha \cap SL_2(\mathbb{Z}))\backslash SL_2(\mathbb{Z})$. Usually, $\alpha$ will be chosen such that $(\alpha^{-1}SL_2(\mathbb{Z})\alpha \cap SL_2(\mathbb{Z}))$ has a clear description. By Lemma 2.3.1, $SL_2(\mathbb{Z})\backslash SL_2(\mathbb{Z})\alpha SL_2(\mathbb{Z})$ has coset representatives $\{\alpha\alpha_i\}$,

3. By Lemma 2.3.2, for each $\alpha\alpha_i$, find $\beta_i \in SL_2(\mathbb{Z})$ such that $\beta_i\alpha\alpha_i \in \Gamma_H$. Then $\{\beta_i\alpha\alpha_i\}$ will be the desired coset representatives for $\Gamma_H\backslash\Gamma_H\alpha\Gamma_H$.

## 2.4 Tiny integrals via complex number approximation

We present a method to compute tiny integrals by comparing Taylor coefficients of a system of equations and recovering them as algebraic number approximations from complex solutions.

**Algorithm 2.4.1.** *Computing* $\sum_{i=0}^{p}\int_{Q_i}^{Q}\omega$

***Input:***

- $\tau_0 \in \mathbb{H}$ *such that* $\Gamma\tau_0$ *corresponds to a rational point* $Q$ *on* $X$*, and* $q_0 := e^{2\pi i\tau_0/h}$ *where* $h$ *is the width of the cusp.*
- *A good prime* $p$ *which does not divide* $j(Q)$ *or* $j(Q)-1728$.
- *A cusp form* $f \in \mathscr{S}_2(\Gamma)$ *given by its* $q$*-expansion where* $q = e^{2\pi i\tau/h}$*. We denote the corresponding* $1-$*form by* $\omega$.

***Output:***

- *The sum of tiny Coleman integrals* $\sum_{i=0}^{p}\int_{Q_i}^{Q}\omega \in \mathbb{Q}_p$*, where* $T_p(Q) = \sum_{i=0}^{p} Q_i$.

***Steps:***

1. *(Writing* $\omega$ *as a power series in terms of an uniformiser* $u$*) Fix a precision* $n$*. Find* $x_i \in \mathbb{Q}$ *such that*

$$\omega = (\sum_{i=0}^{n} x_i(u)^n + \mathscr{O}(u^{n+1}))d(u). \tag{2.2}$$

*These* $x_i$*'s can be found using the following steps:*

*a. Write* $u$ *and* $\omega_i$ *as power series expansions of* $q-q_0$ *by differentiating their* $q$*-expansions* *and evaluating at $q_0$:*

$$u = \sum_{i=1}^{C_1} a_i (q-q_0)^i + O((q-q_0)^{C_1+1}),$$
$$\omega = \sum_{i=0}^{C_2} b_i (q-q_0)^i + O((q-q_0)^{C_2+1}) dq,$$
$$d(u) = (\sum_{i=1}^{C_1} i a_i (q-q_0)^{i-1} + O((q-q_0)^{C_1})) dq,$$

*where $C_1, C_2$ are some fixed precision determined by $n$ and the norm of $q_0$. The coefficients $a_i$, $b_i$'s are in $\mathbb{C}$.*

b. *Replace $\omega$, $u$ and $d(u)$ by their power series expansions in $q-q_0$ as in equation (2.2). Comparing the coefficients of $(q-q_0)^k$ on both sides gives us the following linear system:*

$$\begin{pmatrix} a_1 & 0 & 0 & \dots & 0 \\ 2a_2 & a_1^2 & 0 & \dots & 0 \\ 3a_3 & 3a_1a_2 & a_1^3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (n+1)a_{n+1} & \sum_{i=1}^n a_i(n+1-i)a_{n+1-i} & * & \dots & a_1^{n+1} \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

c. *Solve this system of equations and get complex approximations of $x_i$'s. These $x_i$'s can be recovered as elements in $\mathbb{Q}$ using `algdep` from `PARI/GP`. This is likely to succeed given sufficient complex precision.*

2. *Calculate $u(Q_i)$ as algebraic numbers. In practice, we use the $j$-invariant function as an uniformiser. We calculate $j(Q_i)$ transcendentally by evaluating the $q$-expansion of the $j$-function on $\beta_i(\tau_0)$ and then obtain the algebraic approximation. On the other hand, the roots of the modular polynomial $\Phi_p(x, j(Q)) = 0$ are the $j$-invariants of elliptic curves that are*

*p-isogeneous to Q. This gives another (algebraic) method to compute* $j(Q_i)$.

3. *Compute the sum of tiny integrals* $\sum_{i=0}^{p} \int_{Q}^{Q_i} \omega \approx \sum_{i=0}^{p} \int_{0}^{u(Q_i)} (\sum_{j=0}^{n} x_j u^j du)$ *with its p-adic expansion.*

# Chapter 3

# Computations and examples

In the previous chapter, we outlined an algorithm to compute single Coleman integrals on modular curves between any two known points. The modular curves in consideration come from Serre's Uniformity Conjecture, i.e., they are of the form $X = X_H(N) := \mathbb{H}^+/\Gamma_H(N)$ and they satisfy:

- $\Gamma_H(N) \leq \mathrm{SL}_2(\mathbb{Z})$ where $H \leq \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$,
- $-I \in H$,
- $\det : H \to \mathbb{Z}/N\mathbb{Z}$ is surjective.

Moreover, we also consider quotients of modular curves by the action of Atkin-Lehner involutions. We will demonstrate three classes of examples, namely, $X_0(N)$, $X_0^+(N)$, and $X_{ns}^+(N)$, while gathering the necessary ingredients such as known rational points, basis of differentials and the action of the Hecke operators to perform Coleman integration.

## 3.1 $X_0(N)$

Let $N$ be positive integer. The modular curve $X = X_0(N)$ is defined to be the quotient of the upper half plane by the congruence subgroup $\Gamma_0(N) = \{\left(\begin{smallmatrix} * & * \\ 0 & * \end{smallmatrix}\right) \pmod{N}\} \leq \mathrm{SL}_2(\mathbb{Z})$. As a moduli space, the $\mathbb{Q}$-rational points of $X$ correspond elliptic curves $E$ defined over $\mathbb{Q}$ such that $E$ admits a $\mathbb{Q}$-rational isogeny of degree $N$ to another elliptic curve $E'$. Given a point $Q$ on $X$, to find the coset representative on the upper half plane, one first finds the ratio of periods $\tau' \in \mathbb{H}$ of the elliptic curve $Q$, which corresponds to $\mathrm{SL}_2(\mathbb{Z})\tau' \in \mathrm{SL}_2(\mathbb{Z})\backslash\mathbb{H}$ satisfying $j(\tau') = j(E)$. This can be done by finding an elliptic curve $E_\tau/\mathbb{C}$ with $j$-invariant $j_E$ via the universal elliptic curve:

$$y^2 + xy = x^3 - \frac{36}{j_E - 1728}x - \frac{1}{j_E - 1728}$$

Then, one iterates through the cosets of $\mathrm{SL}_2(\mathbb{Z})/\Gamma_0(N)$ to find $\gamma$ such that its $j$-invariant satisfies:

$$j(\gamma\tau') = j(N\gamma\tau') = j(E)$$

As a result, the point $Q$ corresponds to $\Gamma_0(N)\gamma\tau' \in \Gamma_0(N)\backslash\mathbb{H}^+$. One could find a basis of weight 2 cusp forms $\mathscr{S}_2(\Gamma_0(N))$ and the action of Hecke operators on the basis of cusp forms using well known methods that are implemented in SAGEMATH[Ste07, The22b]. Let $\omega \in H^0(X, \Omega^1)$, $Q \in X(\mathbb{Q})$. We follow Algorithm 2.4.1 for computing $\sum_{i=0}^{p} \int_{Q_i}^{Q} \omega$. We choose $j(\tau) - j(Q)$ as our uniformiser.

### 3.1.1 Example: $X_0(37)$

- **Curve data:** We consider the modular curve $X = X_0(37)$. $X$ is a hyperelliptic curve. Comparing relations between $q$-expansions of rational functions $x, y \in \mathbb{C}(X)$, we obtain a plane model $y^2 = -x^6 - 9x^4 - 11x^2 + 37$ [MSD74]. There are four $\mathbb{Q}$-rational points $Q = (1,-4), R = (-1,-4), S = (1,4), T = (-1,4)$, where $Q,R$ are noncuspidal rational points and $S,T$ are cuspidal rational points.

- **Rational points:** Since the $j$-function is a modular function on $X_0(37)$ and that $X_0(37)$ is hyperelliptic, we could express $j$-function as a rational function of $x$ and $y$ to compute that

$$j(Q) = -9317 = -7 \cdot 11^3,$$
$$j(R) = -162677523113838677 = -7 \cdot 137^3 \cdot 2083^3.$$

  The points $Q,R$ corresponds to the elliptic curve $E_Q, E_R$ with $j$-invariants $j(Q), j(R)$ containing a cyclic subgroup of order 37 (or equivalently, with a degree 37-isogeny). This information could be verified in LMFDB [LMF22]. Following the method in Section 3.1, we obtain the upper half plane representatives of $Q,R$ as follows:

$$\tau_Q \approx 0.5 + 0.17047019819380 \cdot i \in \mathbb{H},$$
$$\tau_R \approx 0.5 + 0.39635999889406 \cdot i \in \mathbb{H}.$$

- **Basis of differential forms:** One could compute that $\mathscr{S}_2(\Gamma_0(N))$ has $\mathbb{C}$-dimension 2 and a basis of the space of weight 2 cusp forms using SageMath. Furthermore, the action of Hecke operators on the basis of cusp forms is available on SageMath. Linear algebra leads to an eigenbasis $\{f_0, f_1\}$ of the $\mathbb{C}$-vector space $\mathscr{S}_2(\Gamma_0(37))$ with the following $q$-expansions:

$$\begin{aligned} f_0 &= q + q^3 - 2q^4 + O(q^6), \\ f_1 &= q - 2q^2 - 3q^3 + 2q^4 - 2q^5 + O(q^6). \end{aligned}$$

- **Hecke action:** We choose $p = 3$, and $T_3(f_0) = f_0, T_3(f_1) = -3f_1$. Therefore the Hecke operator matrix $T_3$ is $\left(\begin{smallmatrix} 1 & 0 \\ 0 & -3 \end{smallmatrix}\right)$.

- **Algorithm 2.4.1 and results:** Let $\omega_0, \omega_1$ be 1-forms that corresponds to cusp forms $-\frac{1}{2}f_0$, $-\frac{1}{2}f_1$ respectively in order to obtain $\omega_0 = \frac{dx}{y}$ and $\omega_1 = \frac{xdx}{y}$. This way, we can get a direct comparison with MAGMA's hyperelliptic curve implementation. Now, we proceed with computing the Coleman integrals on $\omega_0$, $\omega_1$.

  We explain how to calculate $\sum_{i=0}^{p} \int_{Q_i}^{Q} \omega_1$ using Algorithm 2.4.1. By comparing complex coefficients and using `algdep` to algebraically approximate complex numbers, we first obtain rational coefficients $x_i$ in the expansion of $\omega_1$ about $j = j(Q)$:

$$\begin{aligned} \omega_1 =& (-9317) + \frac{717409}{2 \cdot 37 \cdot 47}(j - j(Q)) + \frac{253086749261192}{37^2 \cdot 47^3}(j - j(Q))^2 \\ &+ \frac{17680454407703835 1043955}{37^3 \cdot 47^5}(j - j(Q))^3 + O((j - j(Q))^4)\ \ d(j - j(Q)). \end{aligned}$$

  After that, the $j$-invariants $j(Q_i)$ of $Q_i$'s for $i = 0, \ldots, 3$ can be realised as the roots of the modular polynomial $\Phi_3(j(Q), X) = 0$. In the last step, we substitute the roots into a sum of local power series:

$$\begin{aligned} \sum_{i=0}^{3} \int_{Q_i}^{Q} \omega_1 = \sum_{i=0}^{3} \int_{j(Q_i) - j(Q)}^{0} & (-9317) + \frac{717409}{2 \cdot 37 \cdot 47} t + \frac{253086749261192}{37^2 \cdot 47^3} t^2 \\ & + \frac{176804544077038351043955}{37^3 \cdot 47^5} t^3 + \cdots dt \end{aligned}$$

Our results are listed in the table below. One can verify the results by comparing with the hyperelliptic model of this curve or with the MAGMA's hyperelliptic curves implementation of [BT22].

| | |
|---|---|
| $\sum_{i=0}^{3}\int_{Q_i}^{Q}\omega_0$ | $O(3^{14})$ |
| $\sum_{i=0}^{3}\int_{Q_i}^{Q}\omega_1$ | $3^2+3^3+3^9+3^{10}+2\cdot 3^{11}+3^{12}+2\cdot 3^{13}+O(3^{14})$ |
| $\sum_{i=0}^{3}\int_{R_i}^{R}\omega_0$ | $O(3^{14})$ |
| $\sum_{i=0}^{3}\int_{R_i}^{R}\omega_1$ | $3^2+3^3+3^9+3^{10}+2\cdot 3^{11}+3^{12}+2\cdot 3^{13}+O(3^{14})$ |

**Table 3.1**: Coleman Integrations on $X_0(37)$

## 3.2 $X_0^+(N)$

Consider the modular curve $X_0(N)$ from the previous example. There is an involution acting on the points of $X_0(N)$, called the Atkin-Lehner involution $w_N := \frac{1}{\sqrt{N}}\left(\begin{smallmatrix} 0 & -1 \\ N & 0 \end{smallmatrix}\right)$. One could verify that $w_N^2$ acts as the identity on the $\Gamma_0(N)$-orbits of $\mathbb{H}$. Let $\Gamma_0^+(N) := \Gamma_0(N) \cup w_N\Gamma_0(N)$. The compactification of the quotient of the upper half plane by $\Gamma_0^+(N)$ gives rise to the modular curve $X := X_0^+(N)$.

**Proposition 3.2.1.** Suppose $\Gamma_0(N)\tau \in X_0(N)$ corresponds to the elliptic curve with torsion data $(E_1, \phi : E_1 \to E_2)$, then $w_N(\Gamma_0(N)\tau)$ corresponds to $(E_2, \hat{\phi} : E_2 \to E_1)$, where $\hat{\phi}$ is the dual isogeny.

*Proof.* $\Gamma_0(N)\tau$ corresponds $(E_\tau, \langle \frac{1}{N}, \tau \rangle)$ up to isomorphism. As $w_N \cdot \tau = \frac{-1}{N\tau}$, $w_N \cdot \Gamma_0(N)\tau$ corresponds to $[E_{\frac{1}{N\tau}}, \langle \frac{1}{N}, \frac{1}{N\tau} \rangle]$. Note that the relation between complex tori over $\Gamma_0(N)$ and elliptic curves with a cyclic subgroup of order $N$ are captured by the following isomorphism $E_\tau / \langle \frac{1}{N}, \tau \rangle \cong \mathbb{C} / \langle \frac{1}{N}, \tau \rangle$. It is clear that $\langle \frac{1}{N}, \tau \rangle = \tau \langle 1, \frac{1}{N\tau} \rangle$, hence $E_{\frac{1}{N\tau}}$ is isomorphic to $E_\tau / \langle \frac{1}{N}, \tau \rangle$. It remains to check that the dual isogeny of $\phi : E \to E_\tau / \langle \frac{1}{N}, \tau \rangle$ is indeed the isogeny induced by $E_{\frac{1}{N\tau}}$. This can be checked by first computing the dual isogeny and comparing kernels. □

The above proposition provides a moduli interpretation for $X_0^+(N)$, i.e., the $\mathbb{Q}$-points correspond to unordered pairs of elliptic curves $(\phi_1 : E_1 \to E_2, \phi_2 : E_2 \to E_1)$ such that $\phi_1$ is an isogeny of degree $N$, and $\phi_2$ is the dual isogeny, with the additional requirement that they are $\mathrm{Gal}(\bar{\mathbb{Q}} \backslash \mathbb{Q})$-invariant. Note that by complex multiplication theory, it is possible that the elliptic curves $E_1, E_2$ or the isogenies $\phi_1, \phi_2$ may not be defined over $\mathbb{Q}$ but over a quadratic extension of $\mathbb{Q}$, and in that case the elliptic curves or isogenies are fixed by nontrivial Galois element of the quadratic extension.

The expected rational points on $X$ correspond to elliptic curves with complex multiplication. Following [Mer18, Sta75], we have a list of discriminants of imaginary quadratic number fields with class number one:

$$\mathscr{D} = \{-3, -4, -7, -8, -11, -12, -16, -19, -27, -28, -43, -67, -163\}.$$

Let $E$ be a CM elliptic curve such that its endomorphism ring $\mathscr{O}_E$ has discriminant $\Delta_E \in \mathscr{D}$. Elliptic curves $E$ such that $N$ splits or ramifies in $\mathscr{O}_E$ give rise to rational points on $X$ [Gal99]. Iterating through the class number one discriminants, we have list of candidates of expected rational points coming from CM elliptic curves. We denote the one of the rational points by $Q$ and find the upper half plane representative via the following steps.

The endomorphism ring $\mathscr{O}_E$ is an order in an imaginary quadratic field and therefore has a generator $\tau_E$ and we factor the ideal $(N)$ into a product of principal ideals $\mathfrak{m}\bar{\mathfrak{m}}$ in $\mathscr{O}_E$. Write $\mathfrak{m} = (\alpha)$. Since $\alpha \in \mathscr{O}_E$, there exists integers such that $\alpha = c\tau_E + d$. Euclidean algoritm gives two integers $a, b$ such that $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}_2(\mathbb{Z})$. In this case, the upper half representative is $\tau_Q = \gamma \cdot \tau_E$.

To compute the basis of cusp forms $\mathscr{S}_2(\Gamma_0^+(N))$, one observes that by the definition of $X_0^+(N)$, $\mathscr{S}_2(\Gamma_0^+(N)) = \{f \in \mathscr{S}_2(\Gamma_0(N)) : f|_2 w_N = f\}$.

The Hecke operator action can be understood through its definition as a double coset operator:

**Lemma 3.2.2.** *Let $\alpha \in \mathrm{GL}_2^+(\mathbb{Q})$. The coset representatives of $(\alpha^{-1}\Gamma_0^+(N)\alpha \cap \Gamma_0^+(N))\backslash\Gamma_0^+(N)$ is the same as that of $(\alpha^{-1}\Gamma_0(N)\alpha \cap \Gamma_0(N))\backslash\Gamma_0(N)$.*

*Proof.* Observe that

$$\begin{aligned} \alpha^{-1}\Gamma_0^+(N)\alpha \cap \Gamma_0^+(N) &= \alpha^{-1}(\Gamma_0(N) \cup w_N\Gamma_0(N))\alpha \cap (\Gamma_0(N) \cup w_N\Gamma_0(N)) \\ &= (\alpha^{-1}\Gamma_0(N)\alpha \cap \Gamma_0(N)) \cup (\alpha^{-1}(w_N\Gamma_0(N))\alpha \cap w_N\Gamma_0(N)) \end{aligned}$$

Now, by Lemma 2.3.1, one has an explicit description of the double coset representatives of $\Gamma_0(N)\alpha\Gamma_{(}N)$ and one could show that the two sets of coset representatives above are equal. □

In particular, the above Lemma implies that, for a prime $p$, the Hecke operator $T_p$ on $X_0^+(N)$ and $X_0(N)$, as a double coset operator, has the same coset representatives:

$$(\,\cdot\,)|_k[\Gamma_0^+(N)\alpha\Gamma_0^+(N)] = (\,\cdot\,)|_k[\Gamma_0(N)\alpha\Gamma_0(N)] : f \mapsto \sum_i f|_k\beta_i = \sum_{i=0}^{p-1} f|_k\left(\begin{smallmatrix}1 & i\\ 0 & p\end{smallmatrix}\right) + f|_k\left(\begin{smallmatrix}p & 0\\ 0 & 1\end{smallmatrix}\right)$$

For the uniformiser, we require a combination of modular functions that is invariant under the Atkin-Lehner involution $w_N$. Since $j(w_N\cdot\tau) = j(-1/N\tau) = j(N\tau)$, we can choose $j + j_N$ as our uniformiser, where $j_N(\tau) := j(N\tau)$. For a given point $Q = (E_1 \leftrightarrow E_2)$ and the points $Q_i$ in the same residue disc, the endpoints of the sum of tiny integrals are $j(Q_i) + j(NQ_i)$ where $j(Q_i)$ and $j(NQ_i)$ can be computed as roots of the modular polynomials $\Phi_p(x, j(E_1)) = 0$ and $\Phi_p(x, j(E_2)) = 0$.

### 3.2.1 Example: $X_0^+(67)$

- **Curve data:** We consider the modular curve $X = X_0^+(67)$. $X$ is a hyperelliptic curve. Again, by comparing relations between $q$-expansions of rational functions $x, y \in \mathbb{C}(X)$, we obtain a plane model $y^2 = x^6 + 2x^5 + x^4 - 2x^3 + 2x^2 - 4x + 1$. A quick box search yields two rational points $R = (0, -1), S = (1, 1)$ on $X$.

- **Uniformisers:** We use $j + j_N$ as the uniformiser since it is a modular function invariant under the Atkin-Lehner involution.

  **TODO: messy!check this part**

- **Rational points:** For the rational points $R, S$, their upper half plane representatives can be found as follows. $R$ is the pair $\{\phi_1 : E_1 \to E_1, \hat{\phi}_1 : E_1 \to E_1\}$, with $j(E_1) = -2^{18}3^3 5^3$. $E_1/\mathbb{Q}$

has CM by the ring of integers $\mathscr{O}_{K_1}$ where $K_1 = \mathbb{Q}(\sqrt{-43})$. 67 splits in $\mathscr{O}_{K_1}$ implies that such pair of isogenies exists. Similarly, $S$ is the pair $\{\phi_2 : E_2 \to E_2, \hat{\phi}_2 : E_2 \to E_2\}$, with $j(E_2) = 2^6 5^3$. $E_2/\mathbb{Q}$ has CM by the ring of integers $\mathscr{O}_{K_2}$ with $K_2 (= \mathbb{Q}(\sqrt{-2}))$, 67 splits in $\mathscr{O}_{K_2}$ as well. Note that both $R$ and $S$ are not fixed by the Atkin-Lehner involution, since that corresponds to the case when 67 is ramified.

We have $j(R) = 2^6 5^3, D(R) = -8$, hence $\tau_R = \sqrt{-2}$. Following the steps described in the previous section, we have $(67) = (7+3\sqrt{-2})(7-3\sqrt{-2})$ and the Euclidean algorithm gives

$$7+3\sqrt{-2} = 7+3\cdot\sqrt{-2} \implies \hat{\gamma} = \left(\begin{smallmatrix} 1 & 2 \\ 3 & 7 \end{smallmatrix}\right)$$

Therefore,

$$\begin{aligned} \hat{\tau}_R = \hat{\gamma}\tau_R = \frac{\sqrt{-2}+2}{3\sqrt{-2}+7} \\ \approx 0.298507462686567 + 0.0211076651100462 \cdot i. \end{aligned}$$

Similarly, we have $j(S) = 2^4 3^3 5^3, D(S) = -12, \tau_S = \sqrt{-3}$. $(67) = (8+\sqrt{-3})(8-\sqrt{-3})$ and the Euclidean algorithm gives

$$8+\sqrt{-3} = 8+1\cdot\sqrt{-3} \implies \hat{\gamma} = \left(\begin{smallmatrix} -1 & -9 \\ 1 & 8 \end{smallmatrix}\right)$$

Therefore,

$$\hat{\tau}_S = \hat{\gamma}\tau_S = -\frac{\sqrt{-3}+9}{\sqrt{-3}+8}$$
$$\approx 1.11940298507463 - 0.0258515045905802 \cdot i.$$

- **Basis of differential forms:** $\mathscr{S}_2(\Gamma_0(67))$ has dimension 5. One could compute the action of $w_{67}$ on the space and find a 2-dimensional subspace spanned by cusp forms invariant under the Atkin-Lehner involution using SAGEMATH, to get a basis of $H^0(X,\Omega^1)$:

$$\omega_0 = f_0\, dq/q \ = 2q - 3q^2 - 3q^3 + 3q^4 - 6q^5 + O(q^6)\ \ dq/q$$
$$\omega_1 = f_1\, dq/q \ = -q^2 + q^3 + 3q^4 + O(q^6)\ \ dq/q$$

- **Hecke action:** Let $p = 13$ be a good prime. The Hecke matrix on this subspace is $T_{13} = \left(\begin{smallmatrix} -7/2 & 15/2 \\ 3/2 & -7/2 \end{smallmatrix}\right)$.

- **Algorithm 2.4.1 and results:** Step 1 of Algorithm 2.4.1 gives a power series expansion of the differential forms for the uniformiser $j := j + j_N$ (for simplicity, we use this notation). For example, $\omega_0$ at $j = j(R)$ has the following power series expansion:

$$\begin{aligned}\omega_0 = &\frac{-1}{2^7\cdot 5^2\cdot 7^2} + \frac{3047}{2^{15}\cdot 5^5\cdot 7^6}(j - j(R)) + \frac{-38946227}{2^{24}\cdot 5^8\cdot 7^{10}}(j - j(R))^2 \\ &+ \frac{33888900627}{2^{32}\cdot 5^{10}\cdot 7^{14}} + \frac{-110823337943341}{2^{42}\cdot 5^{13}\cdot 7^{17}}(j - j(R))^3 + O((j - j(R))^4)\ \ d(j - j(R)).\end{aligned}$$

The endpoints $j(Q_i) + j(NQ_i)$ appearing in the sum of tiny integrals can be computed as mentioned in the previous section. Finally, we compute the Coleman integrals and our results can be verified with the MAGMA implementation by [BT22] since $X$ is hyperelliptic.

| | |
|---|---|
| $\sum_{i=0}^{3}\int_{R_i}^{R}\omega_0$ | $2\cdot 13+13^2+3\cdot 13^3+7\cdot 13^4+11\cdot 13^5+8\cdot 13^6+8\cdot 13^7+7\cdot 13^8+13^9+O(13^{10})$ |
| $\sum_{i=0}^{3}\int_{R_i}^{R}\omega_1$ | $11\cdot 13+8\cdot 13^2+6\cdot 13^3+8\cdot 13^4+3\cdot 13^5+6\cdot 13^6+6\cdot 13^7+7\cdot 13^8+11\cdot 13^9+O(13^{10})$ |
| $\sum_{i=0}^{3}\int_{S_i}^{S}\omega_0$ | $10\cdot 13+8\cdot 13^2+2\cdot 13^5+5\cdot 13^6+10\cdot 13^7+2\cdot 13^8+2\cdot 13^9+O(13^{10})$ |
| $\sum_{i=0}^{3}\int_{S_i}^{S}\omega_1$ | $3\cdot 13+7\cdot 13^2+2\cdot 13^3+10\cdot 13^4+8\cdot 13^5+5\cdot 13^6+8\cdot 13^8+10\cdot 13^9+O(13^{10})$ |

**Table 3.2**: Coleman Integrations on $X_0^+(67))$

## 3.3 $X_{ns}^{+}(p)$

For a prime $p$, we start with the definitions of the nonsplit Cartan subgroup $C_{ns}$ and its normaliser $C_{ns}^{+}$. Let $\{1,\alpha\}$ be a $\mathbb{F}_p$- basis of $\mathbb{F}_{p^2}$. Suppose that $\alpha$ satisfies a minimal polynomial $X^2 - tX + n \in \mathbb{F}_p[X]$. For any $\beta = x + y\alpha \in \mathbb{F}_{p^2}^{\times}$, there is a multiplication-by-$\beta$ map with respect to the basis $\{1,\alpha\}$:

$$\begin{aligned} i_\alpha : \mathbb{F}_{p^2}^{\times} &\to \mathrm{GL}_2(\mathbb{F}_p) \\ \beta &\mapsto \left(\begin{smallmatrix} x & -ny \\ y & x+ty \end{smallmatrix}\right) \end{aligned}$$

Given this choice of basis, we define the nonsplit Cartan subgroup $C_{ns}(p) \leq GL_2(\mathbb{F}_p)$ as the image of $i_\alpha$. The normaliser of the nonsplit Cartan subgroup $C_{ns}^{+}(p)$ is the subgroup generated by $C_{ns}(p)$ and the conjugation map ( under $i_\alpha$) coming from $\mathrm{Gal}(\mathbb{F}_{p^2}/\mathbb{F}_p)$. If we have the freedom to choose the basis, then $\alpha$ can be picked to be the squareroot of a quadratic nonresidue $\varepsilon$ in $\mathbb{F}_{p^2}$ satisfying $X^2 - \varepsilon^2 = 0$. Then, we have:

$$C_{ns}^{+}(p) = \langle \left(\begin{smallmatrix} x & \varepsilon^2 y \\ y & x \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right) : (x,y) \in \mathbb{F}_p^2 \backslash (0,0) \rangle.$$

If $\langle \beta \rangle = \mathbb{F}_{p^2}^{\times}$, then we can write down the generators of $C_{ns}^{+}(p)$

**Example 3.3.1.** Let $p = 13, \varepsilon = \sqrt{7}, \mathbb{F}_{p^2}^{\times} = \langle 1 + \varepsilon \rangle$. Then

$$C_{ns}^{+}(13) = \langle \left(\begin{smallmatrix} 1 & 7\cdot 1 \\ 1 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right) \rangle.$$

The modular curve corresponding to the normaliser of nonsplit Cartan subgroup $C_{ns}^{+}(p)$ is defined as the compactification of the quotient of the upper half plane by the lift of $C_{ns}^{+}(p)$ to a subgroup $\Gamma_{ns}^{+}(p) \leq \mathrm{SL}_2(\mathbb{Z})$.

Finding a basis of $\mathscr{S}_2(\Gamma_{ns}^+(p))$ can be done following Zywina's MAGMA implementation as in Section 2.2. For the purpose of exposition, suppose $\mathscr{S}_2(\Gamma_{ns}^+(p)) = \{f_1, \dots, f_g\}$.

To find the upper half plane representatives of the expected rational points, we follow a similar procedure for $X_0(N)$. First, in the list of class number one discriminants $\mathscr{D}$, the expected points correspond to the discriminants $\Delta$ such that $p$ is inert in the corresponding order $\mathscr{O}_\Delta$ [Maz77]. Once we have the list of expected points $\{P_1, \dots P_r\}$, one could use the same method of inverting the $j$-invariant function to find $\mathrm{SL}_2(\mathbb{Z})$-orbits $\{\tau_1, \dots, \tau_r\}$. The cosets of $\Gamma_{ns}^+(p)\backslash \mathrm{SL}_2(\mathbb{Z})$ allow us to find the correct upper half plane representatives corresponding to $\{P_1, \dots P_r\}$. The problem now reduces to computing $\Gamma_{ns}^+(p)\backslash \mathrm{SL}_2(\mathbb{Z})$. There is bijection:

$$\mathrm{SL}_2(\mathbb{Z})/\Gamma_{ns}^+(p) \to \mathrm{SL}_2((\mathbb{Z}/p\mathbb{Z})/C_{ns}^+(p) \cap \mathrm{SL}_2(\mathbb{Z}/p\mathbb{Z})$$
$$\Gamma_{ns}^+(p)\gamma \mapsto (C_{ns}^+(p) \cap \mathrm{SL}_2(\mathbb{Z}/p\mathbb{Z}))\bar{\gamma}.$$

Therefore, once we obtained coset representatives $\{\gamma_i\}$ of $\mathrm{SL}_2((\mathbb{Z}/p\mathbb{Z})/C_{ns}^+(p) \cap \mathrm{SL}_2(\mathbb{Z}/p\mathbb{Z})$, we can verify if $\gamma_i\tau$ is a $\mathbb{Q}$-rational point on $X$ for $\tau \in \{\tau_1, \dots, \tau_r\}$ by considering the canonical embedding, i.e., we can check if $(f_1(\gamma_i \cdot \tau) : \dots : f_g(\gamma_i \cdot \tau)) \in \mathbb{P}^{g-1}$ has rational coordinates.

For the Hecke operators, recall that they act on the cusp forms and on the divisor group of points and we need to distinguish both cases.

On the cusp forms, there are two major steps: find the double coset representatives and then decompose these representatives into products on simpler matrices, for which there are algorithms to compute the slash-$k$ operators [Zyw20, DS05]. The Hecke operator at the prime $\ell$ acts as a double coset operator:

$$[\Gamma_{ns}^{+}(p)\alpha\Gamma_{ns}^{+}(p)]_2 f = \sum f|_2\alpha_i,$$

where $\{\alpha_i\}_{i=0,\dots,p}$ are the double coset representatives of $\Gamma_{ns}^{+}(N)\backslash\Gamma_{ns}^{+}(p)\alpha\Gamma_{ns}^{+}(p)$. It turns out that the representatives have the form $\alpha_i = \varepsilon\varepsilon'\left(\begin{smallmatrix}1 & 0\\ 0 & p\end{smallmatrix}\right)\beta$ or $\varepsilon\varepsilon'\beta\left(\begin{smallmatrix}p & 0\\ 0 & 1\end{smallmatrix}\right)$, where $\varepsilon,\varepsilon' \in \mathrm{SL}_2(\mathbb{Z})$ depends on $\alpha$ and $\beta$ comes from the standard cosets of $\Gamma^0(p)\backslash\mathrm{SL}_2(\mathbb{Z})$. The motivation for this decomposition is that Zywina's algorithm [Zyw20] can compute the slash-$k$ operator on determinant 1 matrices and the two matrices on the right can be resolved using techniques from [DS05]

In the first case, $f|_2\alpha_i = f|_2\varepsilon\varepsilon'\left(\begin{smallmatrix}1 & 0\\ 0 & p\end{smallmatrix}\right)\beta$ is given by Zywina's algorithm and explicit formulas found in Chapter 5, Section 2 of [DS05]. For the second case, one uses the fact that $\left(\begin{smallmatrix}1 & 0\\ 0 & p\end{smallmatrix}\right)\left(\begin{smallmatrix}mp & n\\ N & 1\end{smallmatrix}\right) = \left(\begin{smallmatrix}m & n\\ N & p\end{smallmatrix}\right)\left(\begin{smallmatrix}p & 0\\ 0 & 1\end{smallmatrix}\right)$ where $mp - nN = 1$. So the last coset $\alpha_p$ is of the form $\varepsilon\varepsilon\beta\left(\begin{smallmatrix}p & 0\\ 0 & 1\end{smallmatrix}\right)$. Again, Zywina's algorithm allows us to compute the slask-$k$ operator for the first three matrices of determinant 1 while $\left(\begin{smallmatrix}p & 0\\ 0 & 1\end{smallmatrix}\right)$ acts by shifting the indices by multiples of $p$ via the slask-$k$ operator.

Since we already have a basis $\{f_1,\dots,f_g\}$ of weight 2 cusp forms on $\Gamma_{ns}^{+}(p)$ by Zywina's algorithm, writing $[\Gamma_{ns}^{+}(p)\alpha\Gamma_{ns}^{+}(p)]_2 f_i$ as a linear combination of the basis elements of $\mathscr{S}_2(\Gamma(N),\mathbb{Q}(\zeta_N))$ would give us the Hecke matrix.

The Hecke operator on points can be computed in two ways as before. Firstly, if we have the double coset representatives, we can evaluate the points. Secondly, we could find the roots of the modular polynomial. Each approach has its (dis)advantages: we can evaluate cusp forms on explicit representatives but this will require a closer analysis of the group structure of $C_{ns}^{+}(N)$ and high enough complex precision; the modular polynomials give us the $j$-invariants of $p$-isogeneous points but they have large coefficients.

### 3.3.1 Example: $X_{ns}^{+}(13)$

We consider the cursed curve $X = X_{ns}^{+}(13)$ [BDM$^{+}$19]. Let $C_{ns}^{+}(13)$ be defined by choosing the quadratic non-residue to be 7 as in the previous example, and let $\Gamma_{ns}^{+}(13)$ be the lift of $C_{ns}^{+}(13)$ in $\mathrm{SL}_2(\mathbb{Z})$.

- **Basis of differential forms:** Using Zywina's Magma implementation [Zyw20] , we obtain a basis of cusp forms as follows:

$$\begin{aligned} f_0 =&(3\zeta_{13}^{11}+\zeta_{13}^{9}+3\zeta_{13}^{8}+\zeta_{13}^{7}+\zeta_{13}^{6}+3\zeta_{13}^{5}+\zeta_{13}^{4}+3\zeta_{13}^{2}+1)q\\ &+(-\zeta_{13}^{10}-2\zeta_{13}^{9}-\zeta_{13}^{7}-\zeta_{13}^{6}-2\zeta_{13}^{4}-\zeta_{13}^{3}-2)q^2+O(q^3)\\ f_1 =&(4\zeta_{13}^{11}+2\zeta_{13}^{9}+5\zeta_{13}^{8}+5\zeta_{13}^{5}+2\zeta_{13}^{4}+4\zeta_{13}^{2})q\\ &+(-3\zeta_{13}^{11}-5\zeta_{13}^{10}-4\zeta_{13}^{9}-4\zeta_{13}^{8}-4\zeta_{13}^{7}-4\zeta_{13}^{6}-4\zeta_{13}^{5}-4\zeta_{13}^{4}-5\zeta_{13}^{3}-3\zeta_{13}^{2}-2)q^2+O(q^3)\\ f_2 =&(\zeta_{13}^{10}-2\zeta_{13}^{7}-2\zeta_{13}^{6}+\zeta_{13}^{3})q\\ &+(-\zeta_{13}^{11}-2\zeta_{13}^{10}-2\zeta_{13}^{8}-2\zeta_{13}^{5}-2\zeta_{13}^{3}-\zeta_{13}^{2}+2)q^2+O(q^3), \end{aligned}$$

where $\zeta_{13}$ is a 13-th primitive root of unity and $q = e^{\frac{2\pi i\tau}{13}}$.

- **Curve data:** The method of canonical embedding [Gal96] gives us the following model:

$$\begin{aligned} &X^4-\frac{7}{12}X^3Y-\frac{37}{30}X^2Y^2+\frac{37}{30}XY^3-\frac{3}{10}Y^4-\frac{61}{60}X^3Z+\frac{41}{15}X^2YZ\\ &-\frac{103}{60}XY^2Z+\frac{19}{60}Y^3Z-\frac{23}{6}X^2Z^2+\frac{87}{20}XYZ^2-\frac{14}{15}Y^2Z^2-\frac{199}{60}XZ^3\\ &+\frac{97}{60}YZ^3-\frac{11}{15}Z^4=0, \end{aligned}$$

where $X$, $Y$ and $Z$ corresponds to $f_0$, $f_1$ and $f_2$ respectively. The rational points can be found by a box search:

$$\{(\frac{3}{5}:2:1),(-2:2:1),(-2:\frac{-9}{2}:1),(-2:\frac{-7}{3}:1),(\frac{7}{3}:2:1),(\frac{5}{4}:2:1),(11:\frac{43}{2}:1)\}$$

.

- **Uniformisers:** $\mathscr{S}_2(\Gamma_{ns}^{+}(13)) \subseteq \mathscr{S}_2(\Gamma(N),\mathbb{Q}(\zeta_N))$ so the $j$-function is still a modular function for the normaliser of nonsplit Cartan and therefore can be used as an uniformiser.

- **Rational points:** Among the class number one discriminants $\Delta$ in $\mathscr{D}$, we find $\Delta$ such that 13 is inert in the corresponding order $\mathscr{O}_\Delta$. The set $\{-7,-8,-11,-19,-28,-67,-163\}$ contains discriminants that give rise to 7 expected rational points on $X$. We pick $Q$ to be the point that corresponds to discriminant $-7$, and $R$ to be the point that corresponds to discriminant $-11$. Following the notations in previous section, we have $\tau_7 = \frac{1}{2}+\frac{1}{2}\sqrt{-7}$ and $\tau_{11} = \frac{1}{2}+\frac{1}{2}\sqrt{-11}$. We then compute the coset representatives of $\mathrm{SL}_2(\mathbb{Z})/\Gamma_{ns}^{+}(13)$,

$$\{g_0,\dots,g_{77}\} = \{T^i,\, (T^2)ST^i,\, (T^3)ST^i,\, (T^4)ST^i,\, (T^5)ST^i,\, (T^{12})ST^i \text{ for } i=0,\dots,12\},$$

where $T = \left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)$, $S = \left(\begin{smallmatrix}0&-1\\1&0\end{smallmatrix}\right)$ are the two generators of $\mathrm{SL}_2(\mathbb{Z})$. By evaluating $f_0$, $f_1$, $f_2$ at $g_i(\tau_7)$ and $g_i(\tau_{11})$ for $i=0,\dots,77$, we obtain the correct $\Gamma_{ns}^{+}(13)$-orbit representatives for $Q$ and $R$, $\tau_Q = \frac{4+2\sqrt{-7}}{3+\sqrt{-7}}$, $\tau_R = \frac{13+\sqrt{-11}}{2}$. As in the previous section, the correct representative for $Q$ can be found by evaluating $\frac{f_0(g_i(\tau_7))}{f_2(g_i(\tau_7))}$ and $\frac{f_1(g_i(\tau_7))}{f_2(g_i(\tau_7))}$ for different coset representatives $g_i$ so that the ratios are rational numbers. Applying the same method to all the 7 discriminants, we get their corresponding rational points as computed from the model above.

- **Hecke action on forms:** We choose $p$ to be 11. Let $\alpha = \left(\begin{smallmatrix}-13&44\\42&-143\end{smallmatrix}\right)\left(\begin{smallmatrix}1&0\\0&11\end{smallmatrix}\right)$ be the element $\alpha \in M_2(\mathbb{Z})$ with $\det(\alpha)=11$, $\alpha \pmod{13} \in C_{ns}^{+}(13)$. To find the double coset representatives we start with finding the coset representatives for $\mathscr{S} := (\alpha^{-1}\,\mathrm{SL}_2(\mathbb{Z})\alpha \cap \mathrm{SL}_2(\mathbb{Z}))\backslash\,\mathrm{SL}_2(\mathbb{Z}) =$

$\Gamma^0(11)\backslash \mathrm{SL}_2(\mathbb{Z})$. For each $\beta \in \mathscr{S}$, we found a corresponding $\gamma \in \Gamma^0(11)$ such that the representative $\beta' = \gamma\beta \in \Gamma^+_{ns}(13)$. We define the set of coset representatives to be $\mathscr{S}' := (\alpha^{-1}\Gamma^+_{ns}(13)\alpha \cap \Gamma^+_{ns}(13))\backslash\Gamma^+_{ns}(13)$ and the set of corresponding $\gamma$'s to be $\Gamma$:

$$
\begin{aligned}
\mathscr{S} &= \{\left(\begin{smallmatrix}1 & i\\ 0 & 1\end{smallmatrix}\right), i = 0,1,\ldots,10\} \cup \{\left(\begin{smallmatrix}66 & 5\\ 13 & 1\end{smallmatrix}\right)\},\\
\Gamma &= \{\left(\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & 0\\ -2 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & 11\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & -55\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & 22\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & -44\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & 33\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & -33\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & 44\\ 0 & 1\end{smallmatrix}\right),\\
&\quad \left(\begin{smallmatrix}1 & -22\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}-1 & -55\\ 0 & -1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & -44\\ 0 & 1\end{smallmatrix}\right)\},\\
\mathscr{S}' &= \{\left(\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & 1\\ -2 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & 13\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & -52\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & 26\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & -39\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & 39\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & -26\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}1 & 52\\ 0 & 1\end{smallmatrix}\right),\\
&\quad \left(\begin{smallmatrix}1 & -13\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}-1 & -65\\ 0 & -1\end{smallmatrix}\right), \left(\begin{smallmatrix}-506 & -39\\ 13 & 1\end{smallmatrix}\right)\}.
\end{aligned}
$$

From the bijection

$$
\begin{aligned}
\Gamma^+_{ns}(13)\backslash\Gamma^+_{ns}(13)\alpha\Gamma^+_{ns}(13) &\to (\alpha^{-1}\Gamma^+_{ns}(13)\alpha \cap \Gamma^+_{ns}(13))\backslash\Gamma^+_{ns}(13)\\
\Gamma^+_{ns}(13)\delta &\mapsto (\alpha^{-1}\Gamma^+_{ns}(13)\alpha \cap \Gamma^+_{ns}(13))\alpha^{-1}\delta
\end{aligned}
$$

we can get the double coset representatives of $\Gamma^+_{ns}(13)\backslash\Gamma^+_{ns}(13)\alpha\Gamma^+_{ns}(13)$:

$$
\begin{aligned}
\mathscr{S}_\alpha &= \{\left(\begin{smallmatrix}-13 & 4\\ 42 & -13\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 11\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}-13 & 4\\ 42 & -13\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 11\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ -2 & 1\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 1\\ 0 & 1\end{smallmatrix}\right), \ldots,\\
&\quad \left(\begin{smallmatrix}-13 & 4\\ 42 & -13\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 11\end{smallmatrix}\right)\left(\begin{smallmatrix}-1 & -55\\ 0 & -1\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 10\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}-13 & 4\\ 42 & -13\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 11\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & -44\\ 0 & 1\end{smallmatrix}\right)\left(\begin{smallmatrix}66 & 5\\ 13 & 1\end{smallmatrix}\right)\}\\
&= \{\left(\begin{smallmatrix}-13 & 4\\ 42 & -13\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 11\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right), \left(\begin{smallmatrix}-13 & 4\\ 42 & -13\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ -22 & 1\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 11\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 1\\ 0 & 1\end{smallmatrix}\right), \ldots,\\
&\quad \left(\begin{smallmatrix}-13 & 4\\ 42 & -13\end{smallmatrix}\right)\left(\begin{smallmatrix}-1 & -5\\ 0 & -1\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 11\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 10\\ 0 & 1\end{smallmatrix}\right),\\
&\quad \left(\begin{smallmatrix}-13 & 4\\ 42 & -13\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & -4\\ 0 & 1\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & 0\\ 0 & 11\end{smallmatrix}\right)\left(\begin{smallmatrix}66 & 5\\ 13 & 1\end{smallmatrix}\right) = \left(\begin{smallmatrix}-13 & 4\\ 42 & -13\end{smallmatrix}\right)\left(\begin{smallmatrix}1 & -4\\ 0 & 1\end{smallmatrix}\right)\left(\begin{smallmatrix}6 & 5\\ 13 & 11\end{smallmatrix}\right)\left(\begin{smallmatrix}11 & 0\\ 0 & 1\end{smallmatrix}\right)\}.
\end{aligned}
$$

Following the discussion in the previous section, the Hecke matrix is $A = \left(\begin{smallmatrix}0 & -1 & 2\\ 4 & -4 & 3\\ -1 & 1 & 4\end{smallmatrix}\right)$ in our fundamental equation

$$((p+1)I-A)(\int_Q^R \omega_i) = (\sum_j \int_Q^{Q_j} \omega_i - \sum_j \int_R^{R_j} \omega_i).$$

- **Algorithm 2.4.1 and results:** In Step 1 of Algorithm 2.4.1, linear algebra over $\mathbb{C}$ gives a power series expansion of the differential form $\omega_0$ at $j = j(Q)$:

$$\omega_0 = \frac{1}{3^4 \cdot 5^2 \cdot 13} + \frac{23}{3^{10} \cdot 5^5 \cdot 13}(j - j(Q)) + \frac{4}{3^{13} \cdot 5^7 \cdot 13}(j - j(Q))^2$$
$$+ \frac{437174}{3^{22} \cdot 5^{10} \cdot 13^3}(j - j(Q))^3 + \frac{138504533}{3^{28} \cdot 5^{13} \cdot 13^4}(j - j(Q))^4 + O((j - j(Q))^5) \;\; d(j - j(Q)).$$

The Hecke images can be found by computing the roots of the modular polynomial equation $\Phi_{11}(j(Q),x) = 0$. Next, we compute the integrals as in Step 3. We record our results in the following table.

| | |
|---|---|
| $\sum_{i=0}^{11} \int_{Q_i}^{Q} \omega_0$ | $10 \cdot 11^{-1} + 9 + 9 \cdot 11 + 6 \cdot 11^2 + 7 \cdot 11^3 + 9 \cdot 11^4 + O(11^5)$ |
| $\sum_{i=0}^{11} \int_{Q_i}^{Q} \omega_1$ | $8 \cdot 11^{-1} + 7 + 7 \cdot 11 + 2 \cdot 11^2 + 6 \cdot 11^3 + 6 \cdot 11^4 + O(11^5)$ |
| $\sum_{i=0}^{11} \int_{Q_i}^{Q} \omega_2$ | $10 \cdot 11^{-1} + 8 + 8 \cdot 11 + 11^2 + 9 \cdot 11^4 + O(11^5)$ |
| $\sum_{i=0}^{11} \int_{R_i}^{R} \omega_0$ | $7 \cdot 11^{-1} + 2 + 3 \cdot 11 + 9 \cdot 11^2 + 3 \cdot 11^3 + 5 \cdot 11^4 + O(11^5)$ |
| $\sum_{i=0}^{11} \int_{R_i}^{R} \omega_1$ | $6 + 6 \cdot 11 + 11^3 + 5 \cdot 11^4 + O(11^5)$ |
| $\sum_{i=0}^{11} \int_{R_i}^{R} \omega_2$ | $7 \cdot 11^{-1} + 4 + 11 + 10 \cdot 11^2 + 10 \cdot 11^3 + 5 \cdot 11^4 + O(11^5)$ |

**Table 3.3**: Coleman Integrations on $X_{ns}^+(13))$

## 3.4 Conclusion and future work

# Part II

# Decoding Failures of BIKE

# Chapter 4

# Preliminaries

## 4.1 Introduction

Most cryptosystems implemented today rely on certain hard problems in number theory, such as factorisation or the discrete log problem. These problems fall into the general category of Hidden Subgroup Problems. Recently, there has been significant research on quantum computers and quantum algorithms which make use of quantum phenomena to solve some of these problems that are deemed difficult on classical computers([Sho99, Joz01]).

While building a large-scale quantum computer is an engineering challenge, some scientists predict that within the twenty to fifty years, sufficiently powerful quantum computers will be built to break most if not all current public key cryptography infrastructure. Taking into account the amount of time to implement quantum resistant cryptosystems in public, the National Institute of Standards and Technology (NIST) initiated a process in 2016 to standardise post-quantum digital signature algorithms (DSA), public-key encryption (PKE), and key-encapsulation mechanisms (KEM). Initially, there were 82 submissions. As of April 2023, there 4 algorithms are selected for standardisation while there are three code-based candidates that are still going through evaluation.

|  | PKE/KEM | DSA |
|---|---|---|
| Selected |  |  |
| Lattice | 1 | 2 |
| Hash | 0 | 1 |
| Candidates |  |  |
| Code | 3 | 0 |

**Table 4.1**: NIST Post-Quantum Standardisation Process - Round 4

There is also an on-ramp call for new DSA's in order to diversify the signature portfolio to include signature schemes that are not based on lattices.

In this document, we focus on code-based cryptography, more specifically, one of the 4th round candidates in NIST's standardisation process, BIt-flipping Key Encapsulation (BIKE) [ABB$^+$21]. In 1978, McEliece introduced the use of error-correcting codes in cryptography [McE78]. Originally, error-correcting codes are used in telecommunications in which one party transmits a message through a noisy channel and the recipient recovers the original message from a noisy codeword. In McEliece's proposal, one would use a structured code and hide a message by adding as many errors as the decoder can remove so that the codewords are indistinguishable from random codes. So far, there are no major classical or quantum attacks on the McEliece system but the downside is that it suffers from having large key sizes which make implementations costly.

BIKE is an instance of a more general scheme, called Quasi-Circulant Moderate Density Parity Check (QC-MDPC) codes [MTSB13]. QC-MDPC codes have much smaller key sizes compared to the McEliece cryptosystem and have not suffered from major attacks. One difference between QC-MDPC codes and McEliece's variants is that QC-MDPC codes use decoders which depend on probabilitistic properties, not algebraic ones. Therefore, one expects decoding failures to occur. Furthermore, decoding failures also reveal information about the secret key. An attack by [GJS16] exploits these failures by collecting a set of failure-causing inputs and recover the secret key. With this in mind, one needs to consider the use of ephemeral versus statis keys in applications and also verify certain security conditions, called indistinguishability under chosen cipher attack

(IND-CCA). NIST has considered BIKE as one of the promising candidates and has expressed concerns about its IND-CCA security and decoding failure analysis.

By design, it is not feasible to directly compute the average Decoding Failure Rate (DFR) for BIKE at cryptographic security levels. It is possible to measure DFR's via extrapolation methods to estimate the DFR for larger parameters from smaller ones [SV19a, DGK20b]. But one needs to consider a phenomenon known as the *error floor* region of DFR curves to avoid an underestimate of DFR for larger code sizes. It is known that for LDPC and MDPC codes, the logarithm of the DFR drops significantly faster than linearly, and then linearly as the signal-to-noise ratio is increased [DGK20c, Ric03]. Thus a typical DFR curve contains a concave *waterfall* region followed by a near-linear *error floor* region. One must accurately predict the error floor of a DFR curve to accurately predict the DFR for cryptographically relevant code sizes.

For LDPC codes, the error floor regions have been studied extesively via their Tanner graph representations. Recent work [Vas21a, Vas21b] has considered several factors affecting the DFR of QC-MPDC codes: choice of decoder [Til18, SV19a], classes of weak keys, and sets of problematic error patterns.

Our approach to this problem is to study a scaled-down version of BIKE, and identify various properties of QC-MDPC codes and their decoding failures through extensive experiements.

## 4.2 Background on code-based cryptography

In this chapter, we recall the main definitions from coding theory and what will be needed to construct QC-MDPC codes in cryptography. Throughout this chapter, $q$ is a prime power. In our case, we will study codes when $q = 2$, over the binary field $\mathbb{F}_2$.

### 4.2.1 Coding theory

We start with the basic definitions from coding theory.

**Definition 4.2.1.** A linear code $\mathscr{C} \subseteq \mathbb{F}_q^n$ is a $k$-dimensional linear subspace of the vector space $\mathbb{F}_q^n$. Such a code is called a $[n,k]$-code. The elements of $\mathscr{C}$ are called codewords.

**Definition 4.2.2.** The rate of a $[n,k]$-code $\mathscr{C}$ is the ratio $R = \frac{k}{n}$.

In the context of communications or cryptography, this means that for every $n$-bit of information, there are $k$ symbols of useful information and $(n-k)$ symbols of redundant information. Usually, the redundant information is used to detect or correct errors.

We introduce some tools from linear algebra.

**Definition 4.2.3.** Let $\mathscr{C}$ be a $[n,k]$-code. A generator matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ is a $k \times n$ matrix whose rows form a basis of $\mathscr{C}$. So $\mathscr{C} = \{vG : v \in v \in \mathbb{F}_q^k\}$.

**Definition 4.2.4.** Given a $[n,k]$-code $\mathscr{C}$, the dual code $\mathscr{C}^\perp$ is defined to be

$$\mathscr{C}^\perp := \{v \in \mathbb{F}_q^n | \forall w \in \mathscr{C} w \cdot v = 0\}$$

where $w \cdot v$ is the usual dot product. The generator matrix of $\mathscr{C}^\perp$ is called the parity check matrix.

**Remark 4.2.5.** Let $\mathscr{C}$ a linear code, $\mathbf{G}$ its generator matrix and $\mathbf{H}$ its parity check matrix. Then,

- $\mathbf{G}, \mathbf{H}$ satisfy $GH^\top = 0$.

- We can write $\mathbf{G}, \mathbf{H}$ in systematic form:

$$\mathbf{G} = [\mathbf{I}_k|P], \ , \mathbf{H} = [\mathbf{I}_{n-k}|P]$$

- One matrix can be determined by the other using linear algebra. In particular, we can define $\mathscr{C}$ in terms of its paritcy check matrix:

$$\mathscr{C} = \{c \in \mathbb{F}_q^n | Hc^\top = 0\}.$$

**Definition 4.2.6.** Let $\mathscr{C}$ be a $[n,k]$-code and $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ its parity check matrix. The syndrome of $x \in \mathbb{F}_q^n$ is the vector $Hx^\top \in \mathbb{F}_q^{n-k}$.

**Definition 4.2.7.** For an element $v = (v_0, v_1, \ldots, v_{n-1}) \in \mathbb{F}_q^n$, the support of $v$ is the set of indices with nonzero entries,

$$Supp(v) := \{i \in \{0,1,\ldots,n-1\} : v_i \neq 0\}$$

.

The Hamming weight of a vector is the number of nonzero entries: $|v| := |Supp(v)|$.

**Definition 4.2.8.** $\mathbb{F}_q^n$ is a metric space with the metric defined to be $d(x,y) := |x-y|$. This is called the Hamming distance between $x, y$.

**Definition 4.2.9.** The minimum distance of a linear code $\mathscr{C}$ is defined to be the minimum distance between two distinct codewords:

$$d(\mathscr{C}) := \min_{c_0,c_1 \in \mathscr{C}} d(c_0,c_1) = min_{c\in\mathscr{C}} d(c,0) = min_{c\in\mathscr{C}} |c|$$

.

**Definition 4.2.10.** Given $v, w \in \mathbb{F}_q^n$, the Schur product of $v, w$ is the componentwise product:

$$v \star w := (v_0 \cdot w_0, v_1 \cdot w_1, \ldots, v_{n-1} \cdot w_{n-1}).$$

### 4.2.2 Code-based cryptography

In this section, we define QC-MDPC codes and their representations as graphs.

**Definition 4.2.11.** A $[n,k]$ Low Density Parity Check (LDPC) code is a linear code for which its parity check matrix is sparse, meaning that the Hamming weight of reach row is in $O(1)$.

A $[n,k]$ Moderate Density Parity Check (MDPC) code is a linear code for which its parity check matrix is moderately sparse, meaning that the Hamming weight of reach row is in $O(\sqrt{n})$.

We introduce graph theory as a tool to study these codes.

**Definition 4.2.12.** Given a $[n,k]$-code $\mathscr{C}$ and its parity check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$. The Tanner graph of $\mathscr{C}$ (or $\mathbf{H}$) is the bipartite graph defined by the biadjacency matrix $\mathbf{H}$. The $n$ columns correspond to $n$ nodes, called variable nodes, and the $(n-k)$ rows correspond to $(n-k)$ nodes called check nodes.

Tanner graph is a powerful tool that is used to analyse decoders in LDPC codes since they have sparse parity check matrices. Therefore, it is easy to capture certain graph theoretic information, for example, number of cycles, girth of the graph, etc. On the other hand, MDPC codes have denser parity check matrices and the cost of analysis will increase exponentially in this situation.

LDPC and MDPC codes have their advantages and disadvantages in cryptography. For example, LDPC codes have better decoding performance than their MDPC counterparts but certain low weight codewords can lead to an attack on the McEliece cryptosystem using LDPC codes. MDPC codes can be made more secure but at the expense of more complicated decoder behaviours and large key sizes. Influenced by the development of NTRU [HPS98], lattice- and code-based cryptosystems adapted the use of quotient polynomial rings of the form $\mathbb{F}_2[x]/(x^n-1)$ in order to reduce key sizes. This leads to our next definition.

**Definition 4.2.13.** A circulant matrix is a matrix such that each row is a cyclic shift to the right of its previous row.

Let $r$ be a positive integer. If $\mathbf{H} \in \mathbb{F}_q^{r\times r}$ is a circulant matrix, then $\mathbf{H}$ can be defined by its first row $(h_0\ h_{r-1}\ \dots\ h_1)$:

$$\mathbf{H} = \begin{pmatrix} h_0 & h_{r-1} & \dots & h_2 & h_1 \\ h_1 & h_0 & \dots & h_3 & h_2 \\ \vdots & & \ddots & & \vdots \\ h_{r-2} & h_{r-3} & \dots & h_0 & h_{r-1} \\ h_{r-1} & h_{r-2} & \dots & h_1 & h_0 \end{pmatrix}$$

Observe that a circulant matrix can be defined as the cyclic shift of each column by one entry down of its previous column as well.

We say a matrix is a Quasi-Circulant (QC) matrix if it is a block sum of circulant matrices.

**Definition 4.2.14.** Fix positive integers $n_0, r, d$. A QC-MDPC code is a code with parity check matrix:

$$\mathbf{H} = [\mathbf{H}_0\ \dots\ \mathbf{H}_{n_0-1}]$$

where each block $\mathbf{H}_i$ is a $r\times r$ MDPC matrix, with each row having Hamming weight $d$ such that $n_0 d = O(\sqrt{n_0 r})$. One could check that this code is a $[n_0 r, n_0 r - r]$-code, with rate $R = 1 - 1/n_0$.

We call $r$ the block size of the code.

In code-based cryptography, there are two hard problems, in which most cryptosystems are based on:

1. (Syndrome Decoding Problem) Given a parity check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, a syndrome $s \in \mathbb{F}_q^{n-k}$ and a fixed weight $t \in \mathbb{Z}_{>0}$, find $e \in \mathbb{F}_q^n$ such that $|e| = t$ and $He^\top = s$.
2. (Codeword Finding Problem) Given a parity check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, a fixed weight $w \in \mathbb{Z}_{>0}$, find $e \in \mathbb{F}_q^n$ such that $|e| = w$ and $He^\top = 0$.

It is known that these problems are NP-hard [BMvT78]. The best known attacks are based on Prange's information set decoding algorithm and its improvements [Pra62, Ste89, MMT11, BJMM12].

### 4.2.3 KEM's and Niederreiter cryptosystems

BIKE is a Key Encapsulation Mechanism that is based on a Niederreiter cryptosystem. In this section, we provide definitions to the objects involved.

Modern cryptography uses hybrid cryptosystems: there is a public key cryptosystem combined with a symmetric key cryptosystem. The Key Encapsulation Mechanism allows the involved parties to exchange a common key, thus establishing a secure channel for future communications. The Diffie-Hellman key exchange is a well-known example of this instance.

**Definition 4.2.15.** A Key Encapsulation Mechanism (KEM) is a triple of probabilistic polynomial-time algoritms $(KeyGen, Encaps, Decaps)$ with:

- A key generation method:

$$KeyGen : \{0,1\}^\lambda \to K_{pub} \times K_{priv}$$

where $\lambda$ is the security parameter, $K_{pub}$ and $K_{priv}$ are the spaces of public and private keys respectively.

- An encapsulation method:

$$Encaps : K_{pub} \to M \times C$$

where $M$ and $C$ are the spaces of message and ciphertext. The encapsulation method has a public key as the input and outputs a pair of message and ciphertext.

- A decapsulation method:

$$Decaps : K_{priv} \times C \to M$$

The decapsulation method takes a private key and ciphertext pair as inputs then outputs the message or a failure, which is denoted by $\perp$.

The McEliece cryptosystem works as follows. One generates a code such that an efficient decoder is known then scrambles the generator matrix by changing the basis or permuting the coordinates. The resulting scrambled matrix is the public generator matrix of the code. Furthermore, the decoder must be able to detect and correct up to a certain number $t$ of errors. In the NIST PQC submission, the McEliece candidate uses Goppa codes which is known to have a good decoder [McE78, The22a, Pat75]. BIKE and Classic McEliece use the Niederreiter cryptosystem [NIE86], in which parity check matrices are used instead of the generator matrices. In this sense, it is dual to McEliece cryptosystems and share the same advantages (fast encryption and decryption) and disadvantages (large key sizes). The block size of Niederreiter cryptosystems is smaller than McEliece's.

# Chapter 5

# BIKE

## 5.1 Parameters and design

In this section, we focus on BIKE and its security. Details about the choice of parameters and designs can be found on the BIKE website [ABB$^+$21].

BIKE was originally designed for ephemeral use, i.e., in settings where a KEM key pair is generated for every key exchange instance. This provides a countermeasure against the GJS attack [**?**] where an attacker can use decoding failures to recover the private key of the scheme. Since the second round of the NIST PQC process, BIKE proporsed parameter sets that can security in the static-key setting.

The parameters of BIKE are as follows:

- $r =$ the block size of a parity check matrix,
- $n = 2r$; the length of the code,
- $d =$ the column weight of a parity check matrix,
- $w = 2d$; the row weight of a parity check matrix,

- $t =$ the error weight of an error pattern,
- $\lambda =$ the security parameter, in bits.

The design principles of BIKE are as follows:

- $r$ is a prime such that $x^r - 1 \in \mathbb{F}_2[x]$ has only two irreducible factors,
- $w = 2d = O(\sqrt{n})$,
- $\lambda \approx t - \frac{1}{2}\log_2(r) \approx w - \frac{1}{2}\log_2(r)$.

| $\lambda$ | $r$ | $w$ | $t$ |
|---|---|---|---|
| 128 | 12,323 | 142 | 134 |
| 192 | 24,659 | 206 | 199 |
| 256 | 40,973 | 274 | 264 |

**Table 5.1**: BIKE parameters

Following the Niederreiter framework, BIKE is defined in the following manner:

- ( Key generation) Randomly sample two rows $h_0, h_1 \in \mathbb{F}_2^r$ of fixed weight $d \in \mathbb{Z}_{>0}$ which form the (private) $r \times 2r$ QC-MDPC matrix $\mathbf{H} = [\mathbf{H}_0|\mathbf{H_1}]$. The public key is $\mathbf{H}' = [\mathbf{I}|\mathbf{H}_0^{-1}\mathbf{H}_1]$ in systematic form.
- ( Encapsulation) The message $m$ will be encoded as an error vector $e \in \mathbb{F}_2^{2r}$ of weight at most $t$. The ciphertext will be the syndrome $s = He^\top$.
- ( Decapsulation) Solve the syndrome decoding problem on inputs $H, s$ using the Black-Grey-Flip (BGF) decoder [DGK20c].

**TODO: include BGF decoder here**

## 5.2 Decoding failures, weak keys and near-codewords

We are interested in the decoding failure rate (DFR) for BIKE using the BGF decoder. In particular, we want to understand the factors causing the error floor phenomenon. In this subsection, we provide definitions of decoding failures and the factors causing them.

**Definition 5.2.1.** For a syndrome decoding instance $(H,s)$, where $s = He^\top$ for some unknown $e \in \mathbb{F}_2^{2r}$, we say a decoding failure occured when the output $e' := Decode(H,s)$ is such that $He'^\top \neq s$.

In [Vas21a], the author identifies three classes of weak keys for BIKE. These are parity check matrices which have higher than usual DFR's compared to random parity check matrices of the same parameters.

- (Type I) Parity check matrices whose rows in one of the circulant blocks contain consecutive nonzero bits [DGK20a].
- (Type II) Parity check matrices such that there are many nonzero bits at regular intervals in each row of one of the circulant blocks.
- (Type III) Parity check matrices such that there are many intersections between the columns of the two circulant blocks.

There are also some sets of vectors that are likely to cause decoding failures than random vectors.

**Definition 5.2.2.** Given a parity check matrix $\mathbf{H}$ of a linear code, we say an error vector $e \in \mathbb{F}_2^{2r}$ is a $(u,v)$-near codeword if $|e| = u$ and $|\mathbf{H}e^\perp| = v$.

When $u, v$ are small, these near codewords can be likely to cause decoding failures [Ric03]. Based on the structure of BIKE, Vasseur defines three sets with small $u, v$ as follows: Given a parity check matrix $\mathbf{H} = [\mathbf{H}_0|\mathbf{H}_1]$ with first rows defined by $\mathbf{h}_0, \mathbf{h}_1$ respectively,

- $\mathscr{C} := row(\mathbf{G})$ where $\mathbf{G} = [\mathbf{H}_1^\perp | \mathbf{H}_0^\perp]$ is the generator matrix. This is the set of rows of $\mathbf{G}$ consisting of $(2d, 0)$-near codewords.

- $\mathscr{N} := row([\mathbf{H}_0|\mathbf{0}]) \cup row([\mathbf{0}|\mathbf{H}_1])$. This is the set of rows of the individual circulant matrices in $\mathbf{H}$ consisting of $(d, d)$-near codewords.

- $2\mathscr{N} := \mathscr{N} + \mathscr{N}$. This is the set formed by sum of two elements from $\mathscr{N}$ consisting of $(2d - \varepsilon_0, 2d - \varepsilon_1)$-near codewords for some small $\varepsilon_0, \varepsilon_1 \geq 0$.

Furthermore, Vasseur identified and studied the proximity of error vectors to any of the three problematic sets $\mathscr{S} \in \{\mathscr{C}, \mathscr{N}, 2\mathscr{N}\}$ and how they affect the average DFR. These are the defined by:

$$\mathscr{A}_{t,\ell}(\mathscr{S}) := \{v \in \mathbb{F}_2^{2r} : |v \star c| = \ell \text{ for some } c \in \mathscr{S}\}$$

# Chapter 6

# Experiments

## 6.1 Methods

It is difficult to directly measure cryptographic parameters ($< 2^{-128}$). The common method is to compute DFR's for smaller code sizes then extrapolate in order to estimate the DFR for larger parameters [SV19b, DGK20b, DGK20a]. In this work, we analyse the decoding behaviour for BIKE parameters at the $\lambda = 20$ bits of security.

The parameters were selected according to BIKE design principles and the error weight $t$ is reduced to prevent almost likely decoding failures. Initial selected parameters are as follows: $(r, 2d, t, \lambda) = (523, 30, 18, 20)$. Later we include $389 \leq r \leq 827$ for prime $r$ such that $x^r - 1$ has only two irreducible factors modulo 2.

We use the Black-Grey-Flip (BGF) decoder in all experiments. Within the decoder, we used the original threshold selection function, defined as the maximum of two values: $\frac{d+1}{2}$, and the output of an adaptive threshold function, defined in section 2.5.1 of the BIKE v1.0 specification [ABB$^+$17]. The affine threshold functions in the current version of BIKE are derived from this original threshold rule.

We implement the weak key rejection algorithm defined in [Vas21a, Algorithm 15.3] in order to filter out the weak keys that impedes decoding. The definition of weak key depends on a parameter $T$. For cryptographic size parameters ($\lambda \geq 128$), Vasseur sets $T = 10$ for BIKE parameters. We instead use $T = 3$ for the weak key threshold, the smallest value of $T$ for which finding non-weak keys is feasible. This is justified by the following empirical observation: If we set $T = 4$, the decoding failure rate increases enormously; for example, an experiment with $(r,T) = (587,4)$ observed a DFR on the order of $2^{-8}$, compared to around $2^{-20}$ for $(r,T) = (587,3)$.

We use the Boston University Shared Computing Cluster [Bos], a heterogeneous Linux-based computing cluster with approximately 21000 cores, to run SAGEMATH and Rust implementations of the BGF decoder [ABB$^+$21, DGK20b] in all experiments. The experiments yielded a graph with both the waterfall and error floor regions for our parameter set in addition to many explicit examples of decoding failures that can be used for future analysis. All raw data and the decoder used for this paper are available at [ABH$^+$].

We first compute an average DFR for all suitable block lengths $r$ as follows. For $r$ in Table 6.1, we sample a random key $H$, rejecting any weak keys of types I,II,III as defined before, a random vector $e \in \mathbb{F}_2^{2r}$ of weight $t$, compute $s = He^T$, run BGF decoder on input $(H,s)$, and record the total number of failures. This procedure is run $N$ times where $10^8 \leq 10^9$ to ensure there are enough decoding failures at each $r$ for robust statistical analysis. The error vectors tested in the DFR experiment all had weight 18. The results of this experiment are displayed in Table 6.1 and plotted with best fit curves in Figure 6.1.

**Remark 6.1.1.** Initially, the experiments in SAGEMATH used different numbers of trials ranging from $10^3$ to $10^8$ for different block sizes $r$. This is to ensure that we have enough data to compute the average DFR. Smaller block sizes have more decoding failures whereas larger block sizes have much fewer, calling the need for this adjustment. A future Rust implementation significantly optimised this process. Comparing the results from both implementations, there were quantitative differences but no qualitative ones.

## 6.2 Experimental results

As we have defined earlier, a decoding failure is an instance where, on input $(H,s)$, where $s$ is of the form $s = He^T$, the syndrome decoder output $e'$ is such that $He'^T \neq s$ or $e' \neq e$. The experiment was also designed to record any decoding instances where $He'^T = s$ and $e' \neq e$, but none were discovered.

**TODO: Updated data available**

| $r$ | Decoding failures | Decoding trials | $\log_2(\text{DFR})$ |
|---|---|---|---|
| 389 | 939 | $10^3$ | $-0.09$ |
| 419 | 680 | $10^3$ | $-0.56$ |
| 421 | 652 | $10^3$ | $-0.62$ |
| 443 | 3289 | $10^4$ | $-1.60$ |
| 461 | 1172 | $10^4$ | $-3.09$ |
| 467 | 850 | $10^4$ | $-3.56$ |
| 491 | 1524 | $10^5$ | $-6.04$ |
| 509 | 380 | $10^5$ | $-8.04$ |
| 523 | 946 | $10^6$ | $-10.05$ |
| 541 | 164 | $10^6$ | $-12.57$ |
| 547 | 70 | $10^6$ | $-13.80$ |
| 557 | 177 | $10^7$ | $-15.79$ |
| 563 | 108 | $10^7$ | $-16.50$ |
| 587 | 128 | $10^8$ | $-19.58$ |
| 613 | 61 | $10^8$ | $-20.64$ |
| 619 | 60 | $10^8$ | $-20.67$ |
| 653 | 37 | $10^8$ | $-21.37$ |
| 659 | 35 | $10^8$ | $-21.45$ |
| 661 | 37 | $10^8$ | $-21.37$ |
| 677 | 24 | $10^8$ | $-21.99$ |
| 701 | 20 | $10^8$ | $-22.25$ |
| 757 | 8 | $10^8$ | $-23.58$ |
| 827 | 7 | $10^8$ | $-23.77$ |

**Table 6.1**: Decoding failure rates for $r$-values such that $389 \leq r \leq 827$, $r$ is prime, and $x^r - 1$ has only two irreducible factors modulo 2. The data was computed using the parameters and methods described above.

For the security level $\lambda = 20$, we manage to reproduce the error floor region as predicted in [Ric03].

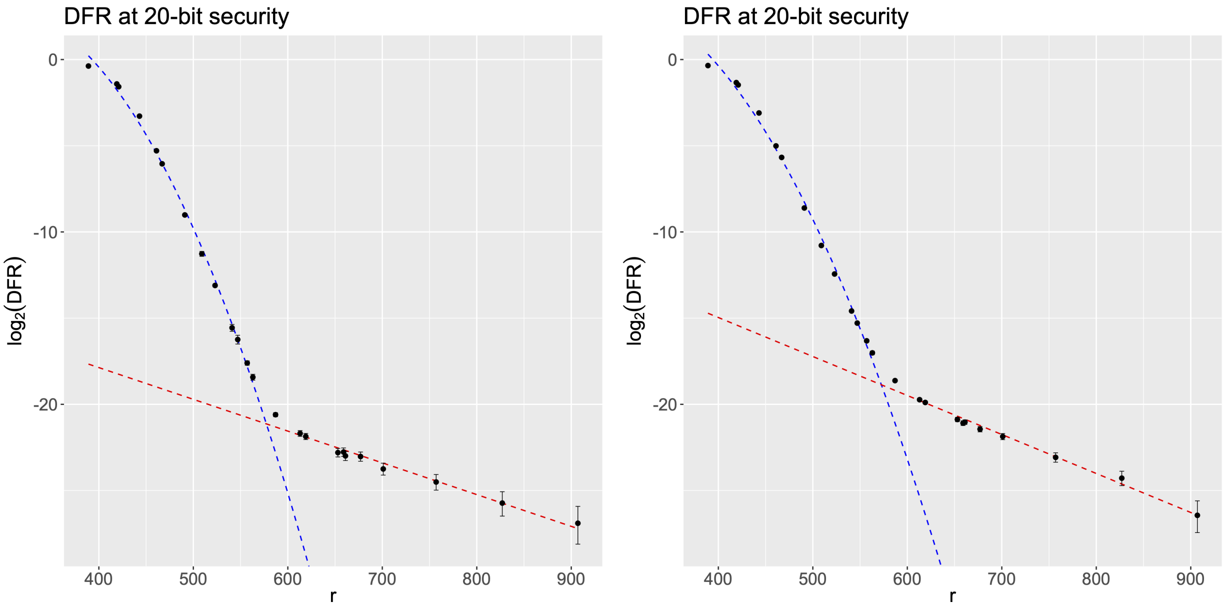


**Figure 6.1**: Semi-log plot of decoding failure rates for non-weak keys ($T = 3$, left) and for unfiltered random keys (right), with a 95% confidence interval for each $r$. There is a quadratic best fit (blue) in the waterfall region and a linear best fit (red) in the error floor region ($r \geq 587$).

## 6.3 DFR on $\mathscr{A}_{t,\ell}(\mathscr{S})$

Recall that Vasseur identified three problematic sets of $(u,v)$-near codewords and proposed a study on the effect of proximity of error vectors to these sets by defining the set:

$$\mathscr{A}_{t,\ell}(\mathscr{S}) := \{v \in \mathbb{F}_2^{2r} : |v \star c| = \ell \text{ for some } c \in \mathscr{S}\}$$

.

While $\ell$ measures the number of common intersections of an error vector with an element of $\mathscr{S}$, we can define another quantity that measures the distance of an error vector from $\mathscr{S}$:

$$\delta(e) = |c| + t - 2\ell$$

where $c \in \mathscr{S}$ is the vector with $|e \star c| = \ell$.

For $\ell$ high ( equivalently, $\delta$ low), decoding failures are extremely prevalent ( see Figure 6.2).

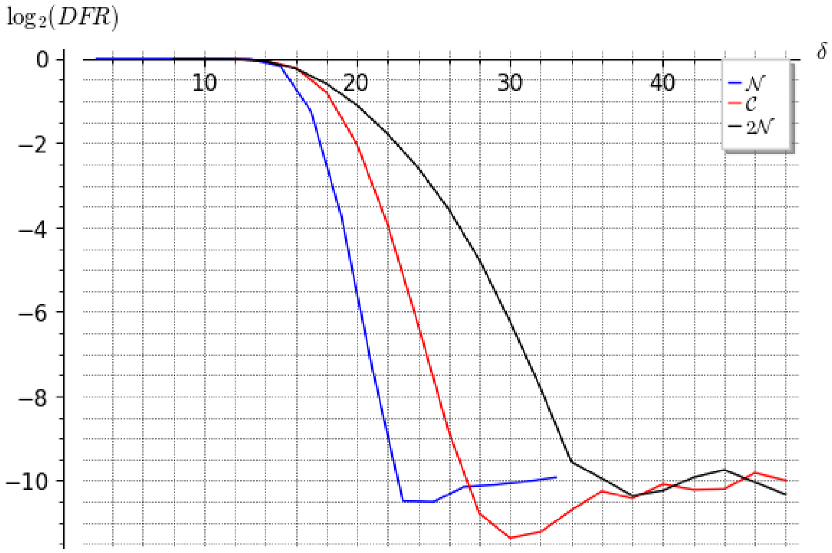


**Figure 6.2**: 20-bit security DFR versus $\delta$ for near-codeword sets $\mathscr{C}, \mathscr{N}, 2\mathscr{N}$ for $r = 523$

We study the relationship between $\mathscr{A}_{t,\ell}(\mathscr{S})$ for some $\ell$, $\mathscr{S} \in \{\mathscr{C}, \mathscr{N}, 2\mathscr{N}\}$ and decoding failures. Our data shows that it is highly unlikely for a decoding failure vector to have a high intersection with an element in $\mathscr{S}$. For a vector $v$, we define the maximum overlap of $v$ for a fixed $\mathscr{S}$ by computing the largest $\ell$ such that $v \in \mathscr{A}_{t,\ell}(\mathscr{S})$. Using the experimental data from $r = 587, N = 10^9$ we recorded **TODO: (verify this data)** 128 total decoding failures and stored the 128 random error vectors that led to decoding failure. The relationship between these decoding failure vectors and the sets $\mathscr{S}$ is shown below:

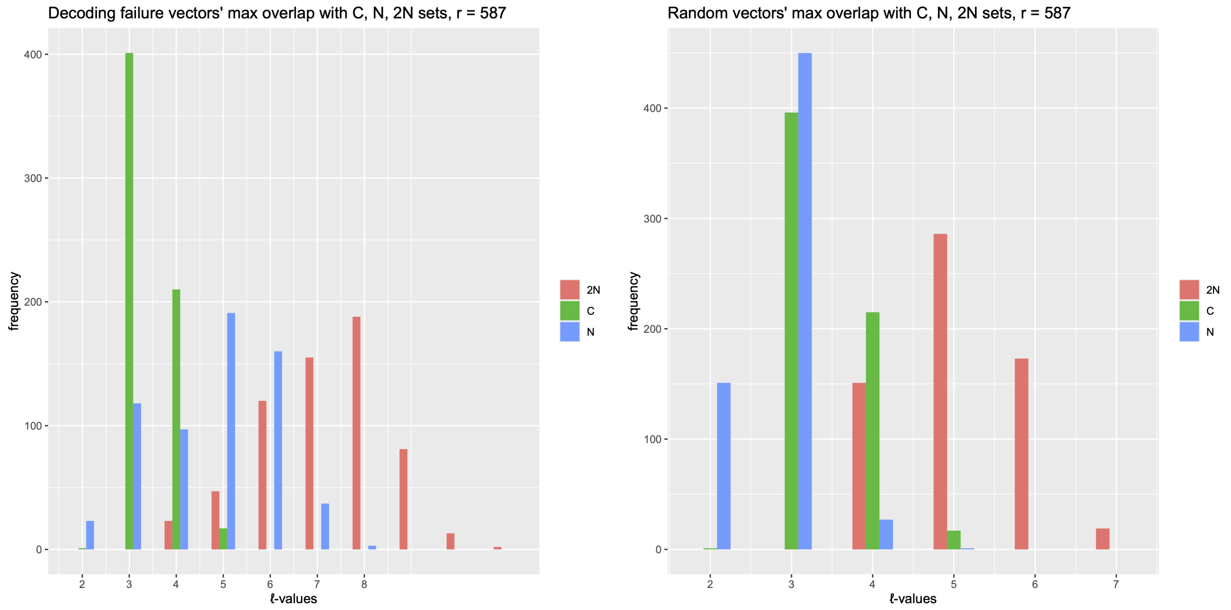


**Figure 6.3**: Distribution of maximum overlaps of decoding failure vectors (left) or random vectors (right) with the sets $\mathscr{C}$, $\mathscr{N}$, and $2\mathscr{N}$ for $r = 587$, using a weak key threshold of $T = 3$ to generate keys.

The cases where $\ell > 10$ are rare. It is expected that these problematic sets contribute to the decoding failures occuring in error floor region and our data suggests that some proportion of decoding failures can be explained by their proximity to $\mathscr{N}$ or $2\mathscr{N}$. However, it is also the case that a significant of the errors causing decoding failures do not have more overlap with $\mathscr{S}$ than typical random vectors. Further analysis is needed to determine what proportion of decoding failures are explained by this proximity.

## 6.4 Syndrome weight as an indicator

From the previous analysis of $\mathscr{A}_{t,\ell}(\mathscr{S})$, the syndrome weights of error vectors causing decoding failures in those sets are low. It is natural to investigate to if the syndrome weight can serve as a predictor for decoding failures.

For the experiment, for each suitable $r$, we generate $10^3$ instances of non-weak parity check matrices $\mathbf{H}$, random error vectors $e$, and then we compute the average weight of their syndromes $s = He^T$. For decoding failure error vectors, we extract the information from our previous DFR simulations to get the nonweak parity check matrices and the corresponding vector causing decoding failures, and then we compute the average weight of their syndromes. Figure 6.4 below gives a summary of our results:

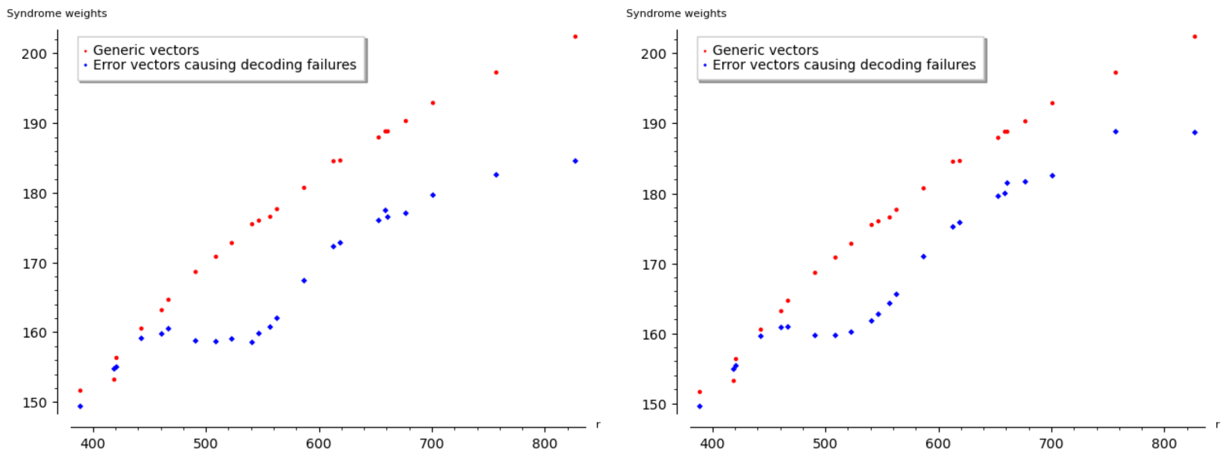
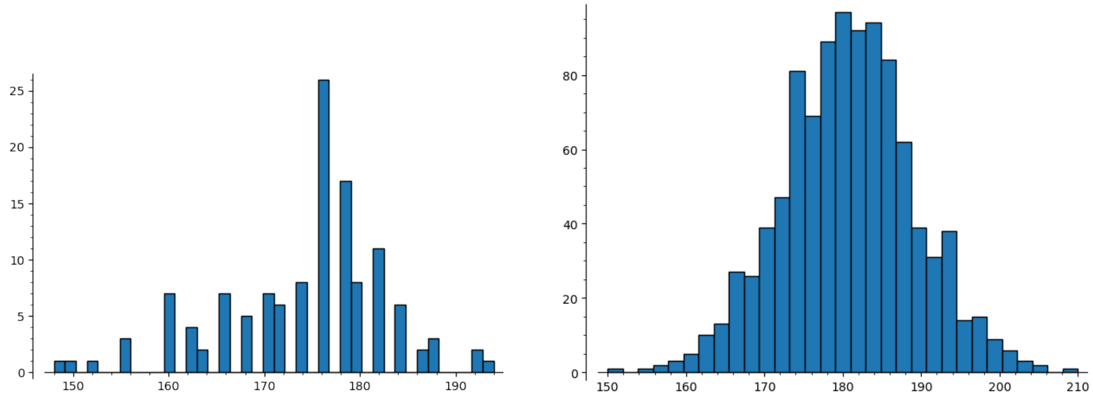


**Figure 6.4**: Distribution of syndrome weights for random error vectors (red) versus error vectors causing decoding failures (blue). The left plot is for non-weak keys (with threshold $T = 3$); the right plot is for unfiltered random keys.

The simulations suggest that syndrome weights of generic vectors tend to follow a normal distribution while the error vectors causing decoding failures have syndrome weights that are more concentrated around the mean, which we hypothesise to be lower than that of the generic vectors; see Figure 6.5 for the case $r = 587$, where we compare the syndrome weights of the **TODO: check** 128 vectors which caused decoding failures with the syndrome weights of the $10^5$ randomly generated vectors of the same weight $t = 18$.

**TODO: might need to generate r587df again**



(a) Decoding failure vectors

(b) Randomly generated vectors

**Figure 6.5**: A comparison of syndrome weights for $r = 587$ between the 128 error vectors which were found to be involved in decoding failures and $10^5$ random vectors. Vertical axis is frequency, and horizontal axis is syndrome weight.

Using this data, we explore whether or not there is convincing evidence that the syndrome weights of error vectors causing decoding failures are lower than those of generic vectors. The null hypothesis is that there is no difference between the two groups in consideration while the alternative hypothesis is that the generic vectors have higher syndrome weights. Both data come from random, independent sampling and have data sets with more than 30 observations. The difference in sample means may be modeled using a $t$-distribution. For each $r$, one could compute the point estimates $m_{\text{generic}} - m_{\text{DF}}$ of population difference $\mu = \mu_{\text{generic}} - \mu_{\text{DF}}$ and standard errors of the point estimate

$$SE = \sqrt{\frac{\sigma^2_{\text{generic}}}{N_{\text{generic}}} + \frac{\sigma^2_{\text{DF}}}{N_{\text{DF}}}}.$$

With this information, one could compute the test statistic for this (one-tailed) test by the formula $T = \frac{\mu - 0}{SE}$. Using either a $t$-table or statistics software, we can find appropriate degrees of freedom and from there, the $p$-value, for each $r$. Our conclusion is that for the sixteen $r$-values in the range $509 \leq r \leq 827$, the $p$-value is less than the significance value $\alpha = 0.01$, and therefore we reject the null hypothesis, i.e., syndrome weights of error vectors causing decoding failures are lower than those of generic vectors.

A general summary of the test statistic values $m_{\text{generic}} - m_{\text{DF}}$ and the corresponding $p$-values can be found in Table 6.2.**TODO: check calculations**

| $r$ | $m_{\text{generic}} - m_{\text{DF}}$ | $p$ |
|---|---|---|
| 509 | 9.29 | $< 0.00001$ |
| 523 | 8.60 | $< 0.00001$ |
| 541 | 9.79 | $< 0.00001$ |
| 547 | 9.29 | $< 0.00001$ |
| 557 | 6.20 | $< 0.00001$ |
| 563 | 8.61 | $< 0.00001$ |
| 587 | 6.56 | $< 0.00001$ |
| 613 | 10.92 | $< 0.00001$ |
| 619 | 8.86 | $< 0.00001$ |
| 653 | 15.99 | $< 0.00001$ |
| 659 | 11.49 | $< 0.00001$ |
| 661 | 9.40 | $< 0.00001$ |
| 677 | 14.45 | $< 0.00001$ |
| 701 | 17.58 | $< 0.00001$ |
| 757 | 16.25 | 0.00278 |
| 827 | 17.53 | 0.00002 |

**Table 6.2**: Hypothesis test results for $509 \leq r \leq 827$, with the corresponding test statistic values and $p$-values, indicating the vectors causing decoding failures do have lower syndrome weights than generic vectors for $509 \leq r \leq 701$, notably a selection of $r$-values where the waterfall region meets the error floor in the DFR graph of Figure 6.1.

## 6.5 Conclusion and future work

In the analysis of the BIKE cryptosystem at the 20-bit security level, we have reproduced the error floor phenomenon and obtained large amount of data for analysis. We found that decoding failure error vectors have lower syndrome weights than those of random vectors. Furthermore, as identified in [Vas21a, Vas21b], the three classes of problematic error vectors $\mathscr{C}, \mathscr{N}, 2\mathscr{N}$ and their proximity sets $\mathscr{A}_{t,\ell}(\mathscr{S})$ contain many elements that cause decoding failures. However, our experiments showed these sets are not responsible the bulk of decoding failures.

It therefore remains to further identify classes of error vectors causing decoding failures in our experiments. As part of an ongoing work, the small parameters allow us to adopt a graph theoretic approach to study the Tanner graph representations of these QC-MDPC codes, which allow us to study interesting behaviours coming from absorbing and trapping sets.

# Bibliography

[ABB$^+$17] Nicolas Aragon, Paulo Barreto, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Shay Gueron, Tim Güneysu, Carlos Aguilar Melchor, Rafael Misoczki, Edoardo Persichetti, Nicolas Sendrier, Jean-Pierre Tillich, and Gilles Zémor. BIKE: Bit flipping key encapsulation - spec v1.0. `https://bikesuite.org/files/BIKE.2017.11.30.pdf`, 2017.

[ABB$^+$21] Nicolas Aragon, Paulo Barreto, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Santosh Ghosh, Shay Gueron, Tim Güneysu, Carlos Aguilar Melchor, Rafael Misoczki, Edoardo Persichetti, Jan Richter-Brockmann, Nicolas Sendrier, Jean-Pierre Tillich, Valentin Vasseur, and Gilles Zémor. BIKE: Bit flipping key encapsulation - spec v4.2. `https://bikesuite.org/files/v4.2/BIKE_Spec.2021.07.26.1.pdf`, 2021.

[ABH$^+$] Sarah Arpin, Tyler Raven Billingsley, Daniel Rayor Hast, Jun Bo Lau, Ray Perlner, and Angela Robinson. Raw data and decoder in the paper "a study of error floor behavior in qc-mdpc codes". `https://github.com/HastD/BIKE-error-floor`. Accessed: 2022-05-23.

[Ass20] Eran Assaf. Computing classical modular forms for arbitrary congruence subgroups. *arXiv: Number Theory*, 2020.

[BBK10] Jennifer S. Balakrishnan, Robert W. Bradshaw, and Kiran S. Kedlaya. Explicit coleman integration for hyperelliptic curves. In Guillaume Hanrot, François Morain, and Emmanuel Thomé, editors, *Algorithmic Number Theory*, pages 16–31, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[BD17] Jennifer Balakrishnan and Netan Dogra. Quadratic chabauty and rational points ii: Generalised height functions on selmer varieties. *International Mathematics Research Notices*, 04 2017.

[BD18] Jennifer S. Balakrishnan and Netan Dogra. Quadratic Chabauty and rational points, I: $p$-adic heights. *Duke Math. J.*, 167(11):1981–2038, 2018. With an appendix by J. Steffen Müller.

[BDM$^+$19] Jennifer Balakrishnan, Netan Dogra, J. Steffen Müller, Jan Tuitman, and Jan Vonk. Explicit Chabauty–Kim for the split Cartan modular curve of level 13. *Annals of Mathematics*, 189(3):885 – 944, 2019.

[BDM$^+$21] Jennifer Balakrishnan, Netan Dogra, Jan Müller, Jan Tuitman, and Jan Vonk. Quadratic chabauty for modular curves: Algorithms and examples. *Preprint, 2101.01862*, 01 2021.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1+1=0$ improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 520–536, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[BMvT78] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[BN20] François Brunault and Michael Neururer. Fourier expansions at cusps. *The Ramanujan Journal*, 53(2):423–437, Nov 2020.

[Bos] Boston University Shared Computing Cluster. `https://www.bu.edu/tech/support/research/computing-resources/scc/`. Accessed: 2022-02-18.

[BP11] Yuri Bilu and Pierre Parent. Serre's uniformity problem in the split Cartan case. *Ann. of Math. (2)*, 173(1):569–584, 2011.

[BPR13] Yuri Bilu, Pierre Parent, and Marusia Rebolledo. Rational points on $X_0^+(p^r)$. *Ann. Inst. Fourier (Grenoble)*, 63(3):957–984, 2013.

[BT20] Jennifer S. Balakrishnan and Jan Tuitman. Explicit coleman integration for curves. *Mathematics of Computation*, 89:2965–2984, 2020.

[BT22] Jennifer Balakrishnan and Jan Tuitman. Magma code. `https://github.com/jtuitman/Coleman`, 2022.

[CdS88] Robert Coleman and Ehud de Shalit. $p$-adic regulators on curves and special values of $p$-adic $L$-functions. *Invent. Math.*, 93(2):239–266, 1988.

[Col85a] Robert F. Coleman. Effective Chabauty. *Duke Mathematical Journal*, 52(3):765 – 770, 1985.

[Col85b] Robert F. Coleman. Torsion points on curves and $p$-adic abelian integrals. *Ann. of Math. (2)*, 121(1):111–168, 1985.

[Col85c] Robert F. Coleman. Torsion points on curves and $p$-adic abelian integrals. *Ann. of Math. (2)*, 121(1):111–168, 1985.

[DGK20a] Nir Drucker, Shay Gueron, and Dusan Kostic. On constant-time QC-MDPC decoders with negligible failure rate. In Marco Baldi, Edoardo Persichetti, and Paolo Santini, editors, *Code-Based Cryptography*, pages 50–79. Springer, Cham, 2020.

[DGK20b] Nir Drucker, Shay Gueron, and Dusan Kostic. QC-MDPC decoders with several shades of gray. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography*, pages 35–50. Springer, Cham, 2020.

[DGK20c] Nir Drucker, Shay Gueron, and Dusan Kostic. Qc-mdpc decoders with several shades of gray. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography*, pages 35–50, Cham, 2020. Springer International Publishing.

[Dic] Leonard E. Dickson. *Linear groups with an exposition of the Galois field theory*. Leipzig, B.G. Teubner, 1901.

[DS05] Fred Diamond and Jerry Shurman. *A first course in modular forms*, volume 228 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2005.

[For81] Otto Forster. *Lectures on Riemann Surfaces*. Springer New York, NY, 1981.

[Gal96] Steven D. Galbraith. Equations for modular curves. *DPhil thesis, University of Oxford*, 1996.

[Gal99] Steven D. Galbraith. Rational points on $X_0^+(p)$. *Experiment. Math.*, 8(4):311–318, 1 1999.

[GJS16] Qian Guo, Thomas Johansson, and Paul Stankovski. A key recovery attack on mdpc with cca security using decoding errors. pages 789–815, 12 2016.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Ntru: A ring-based public key cryptosystem. In Joe P. Buhler, editor, *Algorithmic Number Theory*, pages 267–288, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[Joz01] R. Jozsa. Quantum factoring, discrete logarithms, and the hidden subgroup problem. *Computing in Science & Engineering*, 3(2):34–43, 2001.

[Ked01] Kiran S. Kedlaya. Counting Points on Hyperelliptic Curves using Monsky-Washnitzer Cohomology. *arXiv Mathematics e-prints*, page math/0105031, May 2001.

[LMF22] The LMFDB Collaboration. The L-functions and modular forms database. `http://www.lmfdb.org`, 2022.

[Maz77] B. Mazur. Rational points on modular curves. In Jean-Pierre Serre and Don Bernard Zagier, editors, *Modular Functions of one Variable V*, pages 107–148, Berlin, Heidelberg, 1977. Springer Berlin Heidelberg.

[Maz78] B. Mazur. Rational isogenies of prime degree (with an appendix by D. Goldfeld). *Invent. Math.*, 44(2):129–162, 1978.

[McE78] R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44:114–116, January 1978.

[Mer18] Pietro Mercuri. Equations and rational points of the modular curves $X_0^+(p)$. *Ramanujan J.*, 47(2):291–308, 2018.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 107–124, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[MSD74] B. Mazur and P. Swinnerton-Dyer. Arithmetic of weil curves. *Invent Math*, 25:1–61, 1974.

[MTSB13] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In *2013 IEEE International Symposium on Information Theory*, pages 2069–2073, 2013.

[NIE86] H. NIEDERREITER. Knapsack-type cryptosystems and algebraic coding theory. *Prob. Contr. Inform. Theory*, 15(2):157–166, 1986.

[Pat75] N. Patterson. The algebraic decoding of goppa codes. *IEEE Transactions on Information Theory*, 21(2):203–207, 1975.

[Pra62] E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[Ric03] Tom Richardson. Error floors of LDPC codes. In *Proc. 41st Annual Allerton Conf. on Communication, Control, and Computing*, pages 1426–1435, 01 2003.

[Ser72] J.-P. Serre. Propriétés galoisiennes des points d'ordre fini des courbes elliptiques. *Inv. Math.*, 15:259–3319, 1972.

[Shi94] Goro Shimura. *Introduction to the arithmetic theory of automorphic functions*, volume 11 of *Publications of the Mathematical Society of Japan*. Princeton University Press, Princeton, NJ, 1994. Reprint of the 1971 original, Kanô Memorial Lectures, 1.

[Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.

[Sik17] Samir Siksek. Quadratic chabauty for modular curves. *Preprint, 1704.00473*, 04 2017.

[Sta75] H. M. Stark. On complex quadratic fields with class-number two. *Mathematics of Computation*, 29(129):289–302, 1975.

[Ste89] Jacques Stern. A method for finding codewords of small weight. In Gérard Cohen and Jacques Wolfmann, editors, *Coding Theory and Applications*, pages 106–113, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

[Ste07] William Stein. *Modular forms, a computational approach*, volume 79 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2007. With an appendix by Paul E. Gunnells.

[SV19a] Nicolas Sendrier and Valentin Vasseur. About low DFR for QC-MDPC decoding. Cryptology ePrint Archive, Paper 2019/1434, 2019. `https://eprint.iacr.org/2019/1434`.

[SV19b] Nicolas Sendrier and Valentin Vasseur. On the decoding failure rate of QC-MDPC bit-flipping decoders. In *Post-quantum cryptography*, volume 11505 of *Lecture Notes in Comput. Sci.*, pages 404–416. Springer, Cham, 2019.

[The22a] The McEliece Developers. *Classic McEliece*, 2022. `https://classic.mceliece.org/index.html`.

[The22b] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.6.1)*, 2022. `https://www.sagemath.org`.

[Til18] Jean-Pierre Tillich. The decoding failure probability of MDPC codes. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 941–945. IEEE, 2018.

[Tui16] Jan Tuitman. Counting points on curves using a map to $\mathbf{P}^1$, i. *Math. Comput.*, 85:961–981, 2016.

[Tui17] Jan Tuitman. Counting points on curves using a map to $\mathbf{P}^1$, II. *Finite Fields and Their Applications*, 45:301–322, 05 2017.

[Vas21a] Valentin Vasseur. *Post-quantum cryptography: a study of the decoding of QC-MDPC codes*. PhD thesis, Université de Paris, Mar 2021.

[Vas21b] Valentin Vasseur. QC-MDPC codes DFR and the IND-CCA security of BIKE. Cryptology ePrint Archive, Paper 2021/1458, 2021. `https://eprint.iacr.org/2021/1458`.

[Wet97] Joseph L. Wetherell. *Bounding the number of rational points on certain curves of high rank*. PhD thesis, 1997.

[Zyw20] David Zywina. Computing actions on cusp forms. *arXiv: Number Theory*, 2020.