IUB International University Bremen

Explicit 8-Descent on Elliptic Curves

by

Sebastian Karl Michael Stamminger

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in Mathematics

Approved, Thesis Committee:

____

Professor Michael Stoll

____

Professor Dierk Schleicher

____

Professor John E. Cremona

Date of Defense : December 09, 2005

School of Engineering and Science

Abstract

In this thesis I will describe an explicit method for performing an 8-descent on elliptic curves. First I will present some basics on descent, in particular I will give a generalization of the definition of n-coverings, which suits the needs of higher descent. Then I will sketch the classical method of 2-descent, and the two methods that are known for doing a second 2-descent, also called 4-descent.

Next I will locate the starting position for 8-descent and supplement the exposition of 4-descent by some more detailed geometric information.

In Chapter 3, I will describe the construction of the descent map. It is very similar to Cassels' method for doing a 4-descent, however there are some differences that make our situation more complicated.

This descent map can be used to give an explicit description of a subset of the 8-Selmer group. However, the set we get is only close to the right one, so I call it the fake Selmer set. The methods for computing the fake Selmer set are described in Chapter 4.

The elements of the fake Selmer set are algebraic objects. One would like to have them represented by geometric objects, for example by n-coverings. Finding a method for representing elements of the fake Selmer set as n-coverings is one of my main results. The description of this method is the content of Chapter 5.

The most important result I achieved is the Galois cohomological interpretation of 4- and 8-descent. The relation of Merriman, Siksek, and Smart's method of 4-descent to Galois cohomology has been open for almost ten years now. The methods with which I could solve that problem could immediately be transferred to the Galois cohomology of 8-descent. I guess that these methods, which I will explain in Chapter 6, can be used for giving the Galois cohomological interpretation of most higher descent that might be developed in future.

Being able to compute new examples was the driving force for developing this method of 8-descent. With the program I wrote I was able to find explicit equations for curves of order 8 in the Shafarevich-Tate group of an elliptic curve—the first of such high order—and I was able to prove the Birch- and Swinnerton-Dyer conjecture at the prime 2 for several elliptic curves, where previous methods could not succeed. I will present some examples which illustrate the methods nicely.

Finally, I will conclude by giving some directions for further work related to 8-descent and beyond.

Acknowledgments

I am grateful to my advisor Michael Stoll for his help and encouragement and for suggesting the topic of the thesis to me. I would also like to thank IUB for their financial support and Dierk Schleicher for guiding me to the arithmetic geometry at IUB.

For many helpful discussions I would like to thank Nils Bruin, Hans-Christian Graf v. Bothmer, John Cremona, Steve Donnelly, Tom Fisher, Florian Hess, Cathy O'Neil, Bjorn Poonen, Ed Schaefer, Samir Siksek, Denis Simon, William A. Stein, and Mark Watkins, many of whom I met the first time at the Institut Henri Poincaré, Paris, during the research trimester on Explicit Methods in Number Theory. I also want to thank the IHP and the organizers of this trimester for offering me this great opportunity and some financial support.

For the computations I used the programs Magma, PARI/GP, SAGE, and John Cremona's mwrank. The thesis was typeset using LaTeX and the diagrams were produced with Paul Taylor's commutative diagrams package.

Finally, I would like to thank my wife, Gabi, for her support through the turbulent times during which this thesis was written; and also Julian, for providing most of the turbulence.

Contents

Introduction

Mathematics is often particularly exciting and fruitful when several areas of research meet. This can be seen in mathematical fields such as algebraic topology, where the powerful machinery from algebra supports the study in topology, differential geometry, where methods from differential analysis yield deep geometrical theorems, or complex dynamics, where the methods from complex analysis allow to make significant progress on dynamical systems. The field of research I pursued is arithmetic geometry, which is the combination of number theory, algebra, and geometry.

The number theoretic problem of finding rational solutions to algebraic equations can be interpreted as the problem of finding points on varieties. With this interpretation, number theoretic problems can be solved with the help of the elaborate techniques from algebraic geometry. Thus arithmetic geometry can be understood as the art of finding rational points on algebraic varieties over number fields.

Among the favorite varieties of study of arithmetic geometers are elliptic curves. There has been an enormous amount of research on the arithmetic geometry of elliptic curves, and some of the deepest results in mathematics are concerned with them, such as Andrew Wiles proof of Fermat's Last Theorem. Elliptic curves also have important applications. For example in cryptography they can be used for encrypting messages, and also for the converse, to attack other cryptographic methods. The main fascination of elliptic curves is that they lie at the intersection of different branches of mathematics: as curves they belong to the field algebraic geometry, in analysis they show up in the study of doubly periodic functions on the complex plane $\mathbb{C}$, and in number theory they have been studied for centuries as one of the most interesting Diophantine equations.

An elliptic curve E over a number field K can be given by the following equation in two variables $y^2 = x^3 + Ax + B$, where $A, B \in K$. The set of solutions (x, y), with $x, y \in K$, of this equation—in geometrical language the K-rational points on the curve—is the object of study. It is one of the miracles about elliptic curves that this set carries the structure of an abelian group. Number theory alone would not have indicated that; it was the connection to complex analysis and algebraic geometry that gave rise to this observation.

Already the fact that this group is finitely generated is a deep theorem due to Mordell and Weil. This finitely generated abelian group is called *Mordell-Weil group* and denoted by $E(K)$. By another deep theorem of Mazur [13, 14], the torsion part of $E(\mathbb{Q})$ has only one of fifteen possibilities:

$$\begin{array}{ll} \mathbb{Z}/N\mathbb{Z}, & 1 \leq N \leq 10 \quad \text{or} \quad N = 12, \\ \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2N\mathbb{Z}, & 1 \leq N \leq 4. \end{array}$$

In addition, the torsion part can be effectively determined. The free part of the Mordell-Weil group makes the difficulties. It is conjectured that the rank of this group can be any non-negative integer; while most elliptic curves have rank 0 or 1, examples with ranks up to 24 are known. There is no method known that guarantees to determine the rank. Even if we know the rank, it might be a practical problem to find points on E that generate $E(K)$.

One simplification is known: for finding $E(K)$ it is enough to find the group $E(K)/nE(K)$ for any integer $n \geq 2$, since then $E(K)$ can be obtained by the so called theory of heights. However, so far there is no algorithm known that guarantees to find the Mordell-Weil group or equivalently $E(K)/nE(K)$.

At least in principle there is a method called *n-descent* to get a bound on $E(K)/nE(K)$, $n \geq 2$, and the larger n is the more information it produces. What n-descent actually does is to compute the so called n-Selmer group of E. This group contains $E(K)/nE(K)$, but sometimes it can be greater than $E(K)/nE(K)$; in the latter case, there is no method known to determine which subgroup $E(K)/nE(K)$ is.

The cases where the n-Selmer group is strictly greater than $E(K)/nE(K)$ are rare, but exciting. The difference between them is measured by the Shafarevich-Tate group, which became one of the most fascinating objects in arithmetic geometry and is the main unknown in the Birch and Swinnerton-Dyer conjecture.

In practice, even over $K = \mathbb{Q}$ the only n for which n-descent is feasible are $n = 2, 3, 4$, and by the work I have done $n = 8$. In special cases other n might be possible, e.g. if there is a K-rational n-torsion point on E as in the method described in [21]. This special case was already solved for $n = 2$ by Pierre de Fermat (1601 - 1665), who invented the name descent. The method of 2-descent was used in Birch and Swinnerton-Dyer's extensive computations which led to their famous conjecture. For $n = p$ an odd prime there is a theoretical description which is feasible in practice only for $n = 3$ [19].

The first practical method for doing a second 2-descent, also called 4-descent, was presented by Merriman, Siksek, and Smart [15] in 1996 based on Sikseks thesis [20]. A different method was described by Cassels [6] two years later. With these methods they were able to compute examples where 2-descent was known to fail because of a theoretical obstruction. This obstruction comes from non-trivial 2-torsion in the Shafarevich-Tate group. The method of Merriman, Siksek, and Smart has another feature: they are able to represent the elements of the 4-Selmer group as geometric objects, namely as the intersection of two quadrics in $\mathbb{P}^3$, which is an important starting position for 8-descent. However, what is missing in both expositions is a Galois cohomological interpretation of their methods.

The *aim* of my research was to extend the methods of 4-descent to 8-descent. Starting with the intersection of two quadrics produced by a 4-descent, I wanted to perform a further 2-descent on them, to get parts of the 8-Selmer group. The next goal was to find a method for representing the elements of the 8-Selmer group as geometric objects. Under the condition that the first two steps can be done, the last objective was to develop methods for minimizing these geometric objects.

I *succeeded* with the first two steps. Finding a method to compute parts of the 8-Selmer group could almost be done with the method my advisor Michael Stoll proposed to me, by adjusting Cassels' method of 4-descent to our situation. The differences between Cassels' starting position and our situation turned out to cause some unforeseen difficulties. For example, handling the set of bad primes is one of these problems.

As one of my main results I consider the exploration of a method for representing the algebraic elements of the fake Selmer set as geometric objects, namely as n-coverings. I had the main idea for this when I extensively studied a particular example, which turned out to be a lucky choice because of its beauty and simplicity. It still took me a few months to understand that my methods produce not one geometric object, but the union of two, which I would have to separate.

The most important result I achieved is the Galois cohomological interpretation of 4- and 8-descent. It turned out to be much more complicated than I first thought. The existing methods for the cohomological interpretation of explicit descent could not be applied, since all these methods work with writing down short exact sequences of groups and taking their long exact cohomology sequence. However, this cannot work in our setting, since we do not have groups, but only cosets of groups. That is the typical situation for higher descent, and no higher descent had a Galois cohomological interpretation so far. I have been trying to solve this problem for more than two years, and only a few weeks before finishing this thesis I found the last missing ingredient, which made my previous attempts work. The importance of the methods I developed is first, that I can relate 4- and 8-descent to cohomology and safely make statements about Shafarevich-Tate groups, and secondly, that it provides a framework for the Galois cohomological interpretation of most higher descents that might be developed in future.

The method of 8-descent can be used to compute examples where 4-descent is known to fail because of the obstruction coming from non-trivial 4-torsion in the Shafarevich-Tate group. The program I wrote should be able to compute all of these examples, and I tested it successfully on some of them. Another *application* is to construct elements of order 8 in the Shafarevich-Tate group, and in fact with these methods I am able to get explicit equations for curves of order 8 in the Shafarevich-Tate group—the first of such high order. To really see that these curves are in the Shafarevich-Tate group, we need some additional information, such as the rank of $E(\mathbb{Q})$ being zero or one. For using 8-descent to search for points on the elliptic curve, first a method for minimization has to be found.

Chapter 1

Basics on Descent

1.1 What is Descent?

The first account to this question is the following. An *n-descent* on an elliptic curve E over a number field K is a method to compute the n-Selmer group, $\mathrm{Sel}^{(n)}(E/K)$, of E over K. The n-Selmer group is defined to be the kernel of the homomorphism α where α is defined by the following diagram:

$$\begin{tikzcd}
0 \arrow[r] & E(K)/nE(K) \arrow[r, "\delta_n"] \arrow[d] & H^1(K, E[n]) \arrow[r] \arrow[d] \arrow[dr, "\alpha"] & WC(E/K)[n] \arrow[r] \arrow[d] & 0 \\
0 \arrow[r] & \prod_v E(K_v)/nE(K_v) \arrow[r] & \prod_v H^1(K_v, E[n]) \arrow[r] & \prod_v WC(E/K_v)[n] \arrow[r] & 0.
\end{tikzcd}$$

This diagram is quite standard in Galois cohomology and a good exposition can be found in [21, X.4]. The group on the right, $WC(E/K)$, is the Weil-Châtelet group of E over K, which we will encounter again in the next section, where I will give its definition. This answer to the question, what an n-descent is, is precise, but it does not tell us much about how descent works, and why it is called descent.

A completely different answer to this question was given to me by Samir Siksek: “Factor whenever you can! If you cannot factor, enlarge the field!” I was very surprised by this answer, but now I understood the guiding principle in the tricky calculations of Pierre de Fermat. There is a very nice exposition of the calculations that Pierre de Fermat called descent in [28], where Fermat’s letters and marginalia are cited. Let us look at an example what factoring means in this context. Suppose we are searching rational solutions to the equation $y^2 = f(x)$ where $f(x)$ is a polynomial of degree 3 over $\mathbb{Z}$. Now we try to factor. Suppose we can decompose $f(x)$ into linear factors $f(x) = (x - e_1)(x - e_2)(x - e_3)$ over $\mathbb{Z}$. If now (x, y) is a hypothetical solution, then we can look at the square free part ξ_i of $(x - e_i)$, hence $x - e_i = \xi_i u_i^2$ for some u_i. For a given $f(x)$ there is a method to restrict the set of possible ξ_i's to a finite set. This can be done by elementary but tricky divisibility arguments, or more conceptually by using a descent map. One obvious restriction on the possible ξ_i's is that the product $\xi_1\xi_2\xi_3$ is a square, say η^2. Then for each possible (ξ_1, ξ_2, ξ_3) we get a new system of equations

$$\begin{aligned} \xi_1 u_1^2 &= x - e_1 \\ \xi_2 u_2^2 &= x - e_2 \\ \xi_3 u_3^2 &= x - e_3 \end{aligned} \tag{1.1}$$

with variables x, u_1, u_2, and u_3. In fact, the system is redundant and we can leave away the third equation and the variable u_3. Going from the equation $y^2 = f(x)$ to a finite set of systems of equations (1.1) is a typical example of descent. What do we win by this process? First, it might be easier to find solutions to (1.1) than to $y^2 = f(x)$, and secondly, it might be easier to rule out the existence of solutions to (1.1) than to $y^2 = f(x)$. Thus we have a tool to help us finding solutions and proving that there are no more, but there are cases where this tool will not suffice. If $f(x)$ does not factor we have to use the suggestion to enlarge the field until $f(x)$ factors. To get back the information over $\mathbb{Q}$ we need some algebraic number theory that Fermat did not know.

The third answer to our initial question is geometric: Doing a descent means searching for unramified coverings. We have already seen an example for that above, but I did not use the geometric language. The equations (1.1) define a curve, say C, and with $(x, u_1, u_2, u_3) \mapsto (x, \eta u_1 u_2 u_3)$, where $\eta^2 = \xi_1\xi_2\xi_3$, we get a map from C to $E : y^2 = f(x)$. This covering corresponds to adjunction of the square roots of the functions $x - e_i$, and since they have double zeros and poles on E the covering is unramified. With this geometric language we will have a concrete guide for the method of descent in cases where it is not so clear anymore what factoring should mean, for example on a system of equations of several variables. I will make the notion of coverings more precise in the next section, where we will also see how these geometrical objects are naturally related to cohomology, so that we are led back to the first answer.

For the rest of this thesis I will take the number field $K = \mathbb{Q}$. Most of what follows smoothly generalizes to arbitrary number fields, but I want to emphasize that the following methods work over $\mathbb{Q}$ without having to enlarge the base field. Also for practical computations the most interesting case is $K = \mathbb{Q}$.

1.2 n-Coverings

To make the distinction better visible between maps that are defined over $\mathbb{Q}$ and maps defined over $\bar{\mathbb{Q}}$ we use dotted arrows for the latter ones.

On an elliptic curve E the multiplication by n map is an example of a so called n-covering, and the twists of that are defined to be the n-coverings of E, see [9] for details of the definition and the meaning of twists in this situation. However, we need a more general definition, since we want to do higher descent, hence we start with a (principal) homogeneous space (C, μ) of E, i.e. a smooth curve C and a morphism $\mu : C \times E \to C$ over $\mathbb{Q}$ which induces a simply transitive action of E on C. By a standard abuse of notation, we refer to a homogeneous space as C rather than (C, μ), the morphism μ to be understood. An *isomorphism of homogeneous spaces* (C, μ) and (C', μ') is an isomorphism of curves $\pi : C \dashrightarrow C'$ such that $\pi(Q + P) = \pi(Q) + P$ for every $P \in E(\bar{\mathbb{Q}})$ and every $Q \in C(\bar{\mathbb{Q}})$, where the left plus sign means the action of E on C and the right one means the action of E on C'. The set of homogeneous spaces of E modulo $\mathbb{Q}$-isomorphy is called the *Weil-Châtelet group*, $WC(E/\mathbb{Q})$, of E. This set can be given the structure of a group, see [21]. A homogeneous space is $\bar{\mathbb{Q}}$-isomorphic to E, since for every $P_0 \in C(\bar{\mathbb{Q}})$ we have the isomorphism $\psi_C : E \to C$, $P \mapsto P_0 + P$. Its inverse can be written as $C \to E$, $Q \mapsto [Q - P_0]$ where $[Q - P_0]$ denotes the class represented by the degree 0 divisor $Q - P_0$ in $\mathrm{Pic}^0(C) = E$. Now we can give the generalized definition of an n-covering.

An *n-covering* of the homogeneous space C is a homogeneous space D of E and a morphism $\phi : D \to C$ over $\mathbb{Q}$ such that

$$\phi(Q + P) = \phi(Q) + [n]P$$

for any $P \in E(\bar{\mathbb{Q}})$ and $Q \in D(\bar{\mathbb{Q}})$. This means that geometrically ϕ is multiplication by n, since the following diagram commutes

$$\begin{tikzcd}
D \arrow[r, "\phi"] & C \\
E \arrow[u, dotted, "\psi_D"] \arrow[r, "{[n]}"] & E, \arrow[u, dotted, "\psi_C"']
\end{tikzcd}$$

where $\psi_C : E \dashrightarrow C$, $P \mapsto P_0 + P$, for any choice of $P_0 \in C$, and $\psi_D : E \dashrightarrow D$, $P \mapsto Q_0 + P$, for any choice of $Q_0 \in \phi^{-1}(P_0)$. That is the reason for the name n-covering.

An *isomorphism* of two n-coverings $\phi : D \to C$ and $\phi' : D' \to C$ of C is an isomorphism of homogeneous spaces $\pi : D \dashrightarrow D'$ such that the following diagram

$$\begin{tikzcd}
D \arrow[r, "\phi"] \arrow[d, dotted, "\pi"'] & C \arrow[d, equal] \\
D' \arrow[r, "\phi'"] & C
\end{tikzcd}$$

commutes. Then $\phi' : D' \to C$ is called a *twist* of $\phi : D \to C$. Two n-coverings are called $\mathbb{Q}$*-isomorphic* or just *isomorphic* if there is an isomorphism π over $\mathbb{Q}$.

An n-covering $\phi : D \to C$ is called *everywhere locally solvable*, if $D(\mathbb{Q}_v) \neq \emptyset$ for every place v, i.e. v runs through the primes and ∞, and we write $\mathbb{Q}_\infty := \mathbb{R}$.

In this language, 2-descent on an elliptic curve E means to compute the set of all everywhere locally solvable 2-coverings $\phi_2 : C_2 \to E$ of E up to $\mathbb{Q}$-isomorphism. Doing a 4-descent on E means to compute the set of all everywhere locally solvable 4-coverings $C_4 \to E$ of E. However, since every 4-covering can be split into $C_4 \to C_2 \to E$ for some 2-covering $\phi_2 : C_2 \to E$, a 4-descent can be done in two steps: First, do a 2-descent to get a set of 2-coverings, next, for every 2-covering $\phi_2 : C_2 \to E$ find all 2-coverings of C_2. This is meant by a *second 2-descent*. Combining all this information gives the set of all 4-coverings of E.

The same holds for an 8-descent. We want to compute all everywhere locally solvable 8-coverings of E. We can do that in three steps: First, do a 2-descent, then a second 2-descent, and finally a *third 2-descent*, i.e. for every 4-covering $C_4 \to E$ find all 2-coverings $\phi_8 : C_8 \to C_4$ of C_4. The following diagram should give an overview:

$$\begin{tikzcd}
C_8 \arrow[r, "\phi_8"] \arrow[d, dotted, "\psi_8"'] & C_4 \arrow[r, "\phi_4"] \arrow[d, dotted, "\psi_4"'] & C_2 \arrow[r, "\phi_2"] \arrow[d, dotted, "\psi_2"'] & E \arrow[d, equal] \\
E \arrow[r, "{[2]}"] & E \arrow[r, "{[2]}"] & E \arrow[r, "{[2]}"] & E
\end{tikzcd}$$

where ψ_8, ψ_4, and ψ_2 are the $\bar{\mathbb{Q}}$-isomorphisms to E.

What is the relation to cohomology? There is a general principle relating twists of geometric objects to Galois cohomology. The following proposition can be found in [26, Prop. 1.3].

Proposition 1.2.1. *Let X be some sort of algebraic or geometric object, defined over $\mathbb{Q}$. Then the set of twists of X, i.e., objects Y defined over $\mathbb{Q}$ such that X and Y are isomorphic over $\bar{\mathbb{Q}}$, up to $\mathbb{Q}$-isomorphism, is parameterized by $H^1(\mathbb{Q}, \mathrm{Aut}_{\bar{\mathbb{Q}}}(X))$.*

What does this mean in our context? Let us first look at n-coverings of E. These are twists of $[n] : E \to E$. The automorphisms of $[n] : E \to E$ are the translations by n-torsion points (acting on the left E), hence the n-coverings up to $\mathbb{Q}$-isomorphism are parameterized by $H^1(\mathbb{Q}, E[n])$ by Proposition 1.2.1. The subset of everywhere locally solvable n-coverings corresponds to $\mathrm{Sel}^{(n)}(E/\mathbb{Q})$.

Now let us look at general n-coverings $\phi : D \to C$. The automorphisms of $\phi : D \to C$ are given by the $\bar{\mathbb{Q}}$-isomorphisms $D \to D$ induced by the action of n-torsion points of E on D. Hence they are parameterized by $H^1(\mathbb{Q}, E[n])$, too.

This has an important consequence for higher descent. Take for example a 2-covering $\phi_2 : C_2 \to E$ obtained by a 2-descent. Then by a second 2-descent on C_2 we would like to get some information about the coset above C_2 in $\mathrm{Sel}^{(4)}(E/\mathbb{Q}) \subset H^1(\mathbb{Q}, E[4])$. However, as we have just seen the set of 2-coverings of C_2 is parameterized by $H^1(\mathbb{Q}, E[2])$, and not by a subset of $H^1(\mathbb{Q}, E[4])$. The same holds for 8-descent: A third 2-descent on a curve C_4, where $C_4 \to C_2 \to E$ is a 4-covering, produces 2-coverings of C_4, which are parameterized by $H^1(\mathbb{Q}, E[2])$, and not by a subset of $H^1(\mathbb{Q}, E[8])$. This is not a problem since we have canonical maps

$$0 \to \frac{E(\mathbb{Q})[m]}{nE(\mathbb{Q})[mn]} \longrightarrow H^1(\mathbb{Q}, E[n]) \longrightarrow H^1(\mathbb{Q}, E[mn]).$$

We will see this effect later in the section about the Galois cohomological interpretation of 4- and 8-descent.

Let us see how to interpret the superfluous part coming from the subgroup $E(\mathbb{Q})[m]/nE(\mathbb{Q})[mn]$ in $H^1(\mathbb{Q}, E[n])$ in terms of n-coverings. In the case of a second 2-descent this is $E(\mathbb{Q})[2]/2E(\mathbb{Q})[4]$. If we have a 2-covering $\phi_4 : C_4 \to C_2$ and a point $S \in E(\mathbb{Q})[2]$, then $\phi_4' := \tau_S \circ \phi_4$ is another 2-covering of C_2, where $\tau_S : C_2 \to C_2$, $P \to P + S$, using the action of E on C_2. ϕ_4 and ϕ_4' are not isomorphic as 2-coverings unless there is a rational 4-torsion point $S' \in E(\mathbb{Q})[4]$ with $[2]S' = S$ which we could use to define a $\mathbb{Q}$-isomorphism $\tau_{S'} : C_4 \to C_4$, $Q \mapsto Q + S'$. However, as 4-coverings of E the maps $\phi_2 \circ \phi_4 : C_4 \to E$ and $\phi_2 \circ \phi_4' : C_4 \to E$ are isomorphic, since the action of S on C_2 gets annihilated by ϕ_2.

Analogously, in the case of 8-descent the difference between isomorphy of 2-coverings of C_4 and isomorphy of 8-coverings of E is measured by $E(\mathbb{Q})[4]/2E(\mathbb{Q})[8]$.

It would be interesting to make this observation more explicit, to be able to remove the part coming from $E(\mathbb{Q})[m]/nE(\mathbb{Q})[mn]$ in practical computations.

1.3 Sketch of the Method of 2-Descent

In this section I will sketch the method of 2-descent. The method I will describe is usually called the number field method. There is also an invariant theory based method, which I will not present. An extensive description of the number field method of 2-descent can be found in [5, 22].

We want to perform a 2-descent on the elliptic curve $E : y^2 = f(x)$ over $\mathbb{Q}$. For that, one uses a descent map, which is the so called $x - T$-map. Let $A := \mathbb{Q}[\theta] := \mathbb{Q}[T]/(f(T))$ be the étale algebra. Then we define

$$\begin{aligned} x - T : E(\mathbb{Q}) &\longrightarrow A^*/A^{*2} \\ P &\longmapsto x(P) - \theta \end{aligned}$$

If there is rational 2-torsion on E, then one has to modify the $x - T$-map a little bit. For simplicity, we just stick to the generic case, when there is no rational 2-torsion. One can show either directly using the geometrical group law on E or using more abstract arguments involving Picard groups, that the $x - T$-map is a homomorphism, and that its kernel is $2E(\mathbb{Q})$.

It is easy to see that the image of the $x - T$-map is contained in the kernel of the norm

$$\begin{aligned} N : A^*/A^{*2} &\longrightarrow \mathbb{Q}^*/\mathbb{Q}^{*2} \\ \xi &\longmapsto N(\xi). \end{aligned}$$

Let $H := \ker(N)$, then one can identify H with $H^1(\mathbb{Q}, E[2])$ using the Weil pairing. Thus with H we have a very concrete description of a cohomology group. The 2-Selmer group, which is a subgroup of $H^1(\mathbb{Q}, E[2])$, can now be identified with a subgroup of H, which can be computed in practice. It is the intersection of all the local images of the $x - T$-map, i.e.

$$\mathrm{Sel}^{(2)}(E/\mathbb{Q}) = \bigcap_v \mathrm{res}_v^{-1}((x - T)(E(\mathbb{Q}_v)))$$

where $\mathrm{res}_v : A^*/A^{*2} \to A_v^*/A_v^{*2}$, where $A_v := A \otimes \mathbb{Q}_v$, is the canonical map, and v runs through all primes and ∞. Here $\mathbb{Q}_\infty$ means $\mathbb{R}$. By an additional argument we can restrict to a finite set of primes, so that the 2-Selmer group can actually be computed.

The next step is to represent the elements in the 2-Selmer group as 2-coverings of E. Suppose we have an element in $\mathrm{Sel}^{(2)}(E/\mathbb{Q}) \subset A^*/A^{*2}$, represented by $\xi \in A^*$, then we can construct a 2-covering $\phi_2 : C_2 \to E$ by the following procedure:

The hypothesis that ξ is the image of a point $P = (x, y) \in E(\mathbb{Q})$ under the $x - T$-map implies

$$x - \theta = \xi\eta^2 \tag{1.2}$$

for some $\eta \in A^*$. Write $\xi = \xi_1 + \theta\xi_2 + \theta^2\xi_3$ and $\eta = y_1 + \theta y_2 + \theta^2 y_3$. Since we do not know x, y, and y_1, y_2, y_3, we interpret them as variables. Multiplying out the right hand side of (1.2) gives

$$x - \theta = Q_1(y_1, y_2, y_3) + \theta Q_2(y_1, y_2, y_3) + \theta^2 Q_3(y_1, y_2, y_3)$$

for some quadratic forms Q_1, Q_2, and Q_3 depending on ξ. Sorting by powers of θ implies $Q_2 = -1$ and $Q_3 = 0$. This defines a curve C_2 in $\mathbb{A}^3$ with variables y_1, y_2, y_3 as the intersection of two quadrics. The morphism $\phi_2 : C_2 \to E$ is given by $x = Q_1(y_1, y_2, y_3)$ and $y = \pm rN(\eta)$ where $N(\eta)$ is a polynomial in the variables y_1, y_2, y_3 and $r \in \mathbb{Q}^*$ is a square root of $N(\xi)$. Recall that ξ is in the kernel of $N : A^*/A^{*2} \to \mathbb{Q}^*/\mathbb{Q}^{*2}$. The choice of the sign does not matter, since it gives isomorphic 2-coverings.

This is one possible model of the 2-descendent C_2. Another model for C_2 is of the form

$$y^2 = g(x),$$

with a degree 4 polynomial g, which one can get by using the fact that the quadric $Q_3 = 0$ is singular. It has a singularity at infinity. The isomorphism between these two models can be given explicitly, see [5].

1.4 Two Methods for 4-Descent

There are two methods for doing a 4-descent, i.e. a second 2-descent on C_2. One method uses the model $y^2 = g(x)$ for C_2 and the other uses the model which describes C_2 as the intersection of two quadrics.

1.4.1 Merriman, Siksek, and Smart's Method

Merriman, Siksek, and Smart's method [15] for performing a second 2-descent uses the model $y^2 = g(x)$ for C_2. For the descent map they adapt the $x - T$-map, which we again denote by $x - T$.

$$\begin{aligned} x - T : C_2(\mathbb{Q}) &\longrightarrow A^*/A^{*2}\mathbb{Q}^* \\ P &\longmapsto x(P) - \theta \end{aligned}$$

where $A := \mathbb{Q}[\theta] := \mathbb{Q}[T]/(g(T))$. Notice that here A has degree 4 instead of 3 compared to the 2-descent case. Notice also that we have to divide out $\mathbb{Q}^*$, which corresponds to the fact that C_2 is unramified above infinity, whereas E is ramified there—as coverings of the projective line.

This $x - T$-map does not have all the nice properties that the one from 2-descent has. The norm condition has to be adjusted, but computing the intersection of the local images is almost the same. Taking the intersection of the local images, we get a finite coset of $A^*/A^{*2}\mathbb{Q}^*$, which we call the *fake 2-Selmer set* of C_2, denoted by $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q})$. As I will show in Section 6.1.3 $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q})$ is related to a coset in $\mathrm{Sel}^{(4)}(E/\mathbb{Q})$.

By a similar argument as in the 2-descent one can construct a 2-covering $\phi_4 : C_4 \to C_2$ out of an element in $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q})$ represented by $\xi \in A^*$. For details see [15]. Again C_4 is given by the intersection of two quadrics, this time in $\mathbb{P}^3$,

$$C_4 : Q_1(x_1, x_2, x_3, x_4) = Q_2(x_1, x_2, x_3, x_4) = 0.$$

Similarly to the construction above we get the map $\phi_4 : C_4 \to C_2$ where we have to choose the sign for the y-coordinate. However, this time the choice of sign does matter, hence we get in fact two different 2-coverings $\phi_4^{\pm} : C_4 \to C_2$.

Another difference to the situation of 2-descent is that the two quadrics $Q_1 = 0$ and $Q_2 = 0$ are both non-singular. Thus it is not possible to get a model for C_4 of the form $y^2 = \text{quartic}$ as above.

1.4.2 Cassels' Method

Cassels' method [6] for performing a second 2-descent uses the model of C_2 given by the intersection of two quadrics produced by a 2-descent. Remember that one of them was singular. When we take its projective closure, we can write

$$C_2 : Q_1(x_1, x_2, x_3, x_4) = Q_2(x_1, x_2, x_3, x_4) = 0$$

for two quadratic forms Q_1 and Q_2 where w.l.o.g. Q_1 is singular. C_2 is contained in the whole pencil of quadrics $\lambda_1 Q_1 + \lambda_2 Q_2 = 0$ where $(\lambda_1 : \lambda_2) \in \mathbb{P}^1$. Let M_1 and M_2 be the symmetric matrices corresponding to Q_1 and Q_2, i.e. $Q_1(x_1, \ldots, x_4) = \vec{x} M_1 \vec{x}^t$ and $Q_2(x_1, \ldots, x_4) = \vec{x} M_2 \vec{x}^t$. Then $f(T) := \det(T M_1 + M_2)$ is a polynomial of degree 3 in the variable T.

Moreover, f is in fact (equivalent to) the cubic polynomial of the underlying elliptic curve $E : y^2 = f(x)$.

As étale algebra we take $A := \mathbb{Q}[\theta] := \mathbb{Q}[T]/(f(T))$. For simplicity, let us assume that f is irreducible, hence A is a number field.

Since $\det(\theta M_1 + M_2) = 0$, the quadric $Q_\theta := \theta Q_1 + Q_2$ is singular. Projection from the singularity of Q_θ gives a conic C. Since C_2 is everywhere locally solvable, so is C, hence by the Hasse-Principle it has a $\mathbb{Q}[\theta]$-point.

Take the tangent line at this point and lift it under the projection map to a tangent plane at Q_θ. Denote the linear form defining this tangent plane by L_1. The same argument works for the singular quadric $Q_1 = 0$, and since Q_1 is defined over $\mathbb{Q}$, so is the linear form, which we denote by L_0.

The descent map then is

$$\begin{aligned} C_2(\mathbb{Q}) &\longrightarrow A^*/A^{*2} \\ P &\longmapsto \frac{L_1}{L_0}(P) \end{aligned}$$

Here again one has to compute the intersection of the local images to get a finite subset, which should somehow be related to a subset of the 4-Selmer group. This is not the path Cassels treads in [6], he uses this construction to give an explicit definition of the so called Cassels-Tate pairing instead. However, using this descent map for computing the 4-Selmer group should be very similar to the methods I will describe for 8-descent.

Chapter 2

The Starting Position for 8-Descent

We want to perform an 8-descent on an elliptic curve E, which we can do in three steps, first, a 2-descent to get a set of everywhere locally solvable 2-coverings $\phi_2 : C_2 \to E$, and next, a second 2-descent on each C_2 using the method of Merriman, Siksek, and Smart to get everywhere locally solvable 2-coverings $\phi_4 : C_4 \to C_2$. The last step will be to perform a third 2-descent on each C_4.

The first two stepsare already implemented. For doing a 2-descent we can use John Cremona's program mwrank or a recent different implementation in Magma, and for 4-descent we can use Tom Womack's implementation in Magma, which was part of his thesis [29] in 2003.

So we concentrate only on the last step—the third 2-descent. Thus our starting position is that we are given an everywhere locally solvable 2-covering $\phi_4 : C_4 \to C_2$ of a 2-descendent C_2. Since C_4 is given by the intersection of two quadrics, we will adapt Cassels' method of a second 2-descent, however our situation is more difficult as we will see.

2.1 The Pencil of Quadrics

The 4-descendent C_4 is given by the intersection of two quadrics in $\mathbb{P}^3$, i.e. $C_4 : Q_1 = Q_2 = 0$ for quadratic forms over $\mathbb{Q}$ in four variables. Then C_4 is contained in the whole pencil of quadrics $\mathbb{Q}_\lambda : \lambda_1 Q_1 + \lambda_2 Q_2 = 0$ where $\lambda := (\lambda_1 : \lambda_2) \in \mathbb{P}^1$.

Let M_1 and M_2 be the symmetric matrices corresponding to Q_1 and Q_2, i.e. $Q_1(x_1, \ldots, x_4) = \vec{x} M_1 \vec{x}^t$ and $Q_2(x_1, \ldots, x_4) = \vec{x} M_2 \vec{x}^t$. Then $g(T) := \det(TM_1 + M_2)$ is a polynomial of degree 4 in the variable T. Let $\theta_1, \ldots, \theta_4$

be the zeros of g. Then the matrices $\theta_i M_1 + M_2$, $i = 1, \dots, 4$ are singular, hence the quadrics $Q_{\theta_i} : \theta_i Q_1 + Q_2 = 0$ are the singular quadrics in the pencil.

In addition, the curve $C_2 : y^2 = g(x)$ is the underlying 2-descendent of C_4, at least equivalent to it, see [15].

2.2 The Étale Algebra A

Let $g(T) := \det(TM_1 + M_2)$ be as above, then we define the étale algebra to be

$$A := \mathbb{Q}[\theta] := \mathbb{Q}[T]/(g(T)).$$

A is a product of number fields $A \cong \prod_{i=1}^{t} K_i$ and there can be three cases:

1. $t = 1$, i.e. $A \cong K_1$ is a degree 4 number field,
2. $t = 2$, and K_1 and K_2 are degree 2 number fields,
3. g has a linear factor.

The first case is the generic case. The second case can also happen, and we call it the *split case*. However, the third case is not interesting, since if g has a linear factor, say $T - a$, $a \in \mathbb{Q}$, then $(a, 0)$ is a rational point on C_2, and we are done.

Notice, that this is the same étale algebra as in Merriman, Siksek, and Smart's method of 4-descent. This will have a positive effect as we will see in Theorem 3.6.5.

The Magma[1] code for computing the étale algebra is as follows.

```
function EtaleAlgebra(C4)
  // C4 is the intersection of two quadrics produced by a 4-descent.
  QT<T> := PolynomialRing(Rationals());
  bool, M := IsQuadricIntersection(C4);
  // M[1] and M[2] are the corresponding symmetric matrices.
  g := Determinant(T*M[1] + M[2]);
  // g = HyperellipticPolynomials(AssociatedHyperellipticCurve(C4));
  if assigned C4`EtaleAlgebra then
    return C4`EtaleAlgebra, C4`EtaleAlgebra`AbsoluteMap, g;
  else
    QNF := NumberField(Polynomial([0,1]) : DoLinearExtension);
    //Q as a number field, since pSelmerGroup needs that.
    A<theta> := quo<PolynomialRing(QNF) | g >;
    Aabs, iso := AbsoluteAlgebra(A);
    //The decomposition of A into number fields.
```

[1] All the programs are written for Magma version 2.12-12.

```
    C4`EtaleAlgebra := A; // We store A with C4.
    return A, iso, g;
  end if;
end function;
```

A remark on the implementation: I store the étale algebra A with the curve C_4, since we do not want to recompute the number fields in the decomposition of A. The main reason for that is not only to save time, but also to work over one and the same number field and not to have many different, but isomorphic ones.

A different interpretation of A, which we will need in the section about the Galois cohomology, is the following: Let $\mathcal{R} := \{(\theta_i, 0) \mid i \in \{1, \ldots, 4\}\}$ be the set of ramification points of C_2. Let $\bar{A} := A \otimes \bar{\mathbb{Q}}$, then $\bar{A}^*$ can be interpreted as the set of maps from $\mathcal{R}$ to $\bar{\mathbb{Q}}^*$,

$$\bar{A}^* = \mathrm{Map}(\mathcal{R}, \bar{\mathbb{Q}}^*),$$

and A^* can be interpreted as the Galois-equivariant subset

$$A^* = \mathrm{Map}(\mathcal{R}, \bar{\mathbb{Q}}^*)^{\mathrm{Gal}(\bar{\mathbb{Q}}/\mathbb{Q})},$$

where the action of the Galois group on $\mathrm{Map}(\mathcal{R}, \bar{\mathbb{Q}}^*)$ is defined by $\phi^\sigma := (R^\sigma \mapsto \phi(R)^\sigma)$ for $\phi \in \mathrm{Map}(\mathcal{R}, \bar{\mathbb{Q}}^*)$, hence ϕ is Galois-equivariant if $\phi(R)^\sigma = \phi(R^\sigma)$ for all $R \in \mathcal{R}$. In addition, we write $A_v := A \otimes \mathbb{Q}_v$ and $\bar{A}_v := A \otimes \bar{\mathbb{Q}}_v$. We will also use the second roots of unity $\mu_2(\bar{A}) = \mathrm{Map}(\mathcal{R}, \mu_2)$.

2.3 The Four Singular Quadrics in the Pencil

The singular quadrics in the pencil can be computed by the following function. It computes one singular quadric for each Galois orbit. The other singular quadrics are the conjugates of these.

```
function SingularQuadricsInThePencil(C4)
  bool, M := IsQuadricIntersection(C4);
  A := C4`EtaleAlgebra;
  iso := A`AbsoluteMap;
  // A.1 is a generic zero of g.
  thetas := iso(A.1);
  // thetas are the zeros of g in the number fields,
  // one for each Galois orbit.
  Msing := <theta*M[1] + M[2] : theta in thetas>;
  return <Quadric(M) : M in Msing>;
end function;
```

where we used the following function to get the quadric represented by the symmetric matrix.

```
function Quadric(M);
  // M is an (n+1)x(n+1) symmetric matrix.
  // Return the corresponding quadric.
  K := CoefficientRing(M);
  n := NumberOfColumns(M)-1;
  Pn<[x]> := ProjectiveSpace(K,n);
  Kx := CoordinateRing(Pn);
  xvec := Matrix([x]);
  // q := x*M*Transpose(x); //the quadratic form.
  q := (xvec*ChangeRing(M,Kx)*Transpose(xvec))[1,1];
  return Scheme(Pn,q);
end function;
```

Let $Q_{\theta_1}, \dots, Q_{\theta_4}$ be the four singular quadrics in the pencil. Projection from the singular point of Q_{θ_i} gives a conic C_{θ_i} over $\mathbb{Q}[\theta_i]$. Since C_4 is everywhere locally solvable, so is C_{θ_i}. By the Hasse-Principle C_{θ_i} has a $\mathbb{Q}[\theta_i]$-rational point, hence is isomorphic to $\mathbb{P}^1_{\mathbb{Q}[\theta_i]}$. In the following section we will analyze this map given by projection from the singular point.

2.4 A Nice Geometric Fact About 4-Descent

In this section I will present a further analysis of the geometry of 4-descent. The main result is Proposition 2.4.1, which we will need in Section 3.6. This proposition is implicitly used in [15] without mentioning it.

Let $\phi_4 : C_4 \to C_2$ be the 2-covering constructed in the 4-descent. With the notations from above we have $C_4 : Q_1 = Q_2 = 0$, $C_2 : y^2 = g(x)$, and ϕ_4 is constructed by invariant theory as described in [15]. Let $\theta_1, \dots, \theta_4$ be the roots of g and $Q_{\theta_i} : \theta_i Q_1 + Q_2 = 0$ the four singular quadrics in the pencil.

Let $\pi_{\theta_i} : C_4 \to \mathbb{P}^1$ be the double cover defined over $\mathbb{Q}[\theta_i]$ given by projection from the singular point of Q_{θ_i}. By Hurwitz' formula π_{θ_i} has four ramification points. They belong to the point $(\theta_i, 0)$ on C_2 by the following

Proposition 2.4.1. *The four ramification points of π_{θ_i} coincide with the preimage of $(\theta_i, 0)$ under ϕ_4, $i = 1, \dots, 4$.*

Proof. W.l.o.g. $i = 1$. Since this is a geometric question, we can work over $\mathbb{C}$. Thus we can assume $Q_1 = x_1^2 + ax_2^2 + bx_3^2 + cx_4^2$, and $Q_2 = x_1^2 + x_2^2 + x_3^2 + x_4^2$ for $a, b, c \in \mathbb{C}^*$, and $\theta_1 = -1$.

Now by [15] $\phi_4 : C_4 \to C_2$ is given by $(x_1 : \ldots : x_4) \mapsto (-F_1/F_2, G/F_2^2)$, where

$F_1 = (a+b+c)x_1^2 + (ab+ac+a)x_2^2 + (ab+bc+b)x_3^2 + (ac+bc+c)x_4^2$,
$F_2 = (ab+ac+bc)x_1^2 + (abc+ab+ac)x_2^2 + (abc+ab+bc)x_3^2 + (abc+ac+bc)x_4^2$,
and
$G = dx_1x_2x_3x_4$, where $d = -(a-1)(b-1)(c-1)(a-b)(a-c)(b-c)$.

The formulas for computing F_1, F_2, and G can be obtained by the recipe given in [15], which uses invariant theory. Notice that F_1 and F_2 are non-zero, since a, b, c are non-zero. And G is non-zero, since $d^2 = \mathrm{disc}(g)$.

Let r be a square root of $\frac{b-c}{a-b}$ and s a square root of $\frac{-a+c}{a-b}$. Then one can check that the four points $(0 : \pm r : \pm s : 1)$ are the preimages of $(-1, 0)$ under ϕ_4. Notice, that $1, a, b, c$ must be pairwise distinct, since else C_4 is singular.

To see that these are also the ramification points of π_{θ_1}, one observes that the singular quadric $Q_{\theta_1} : \theta_1 Q_1 + Q_2 = 0$ corresponding to $\theta_1 = -1$ has $(1 : 0 : 0 : 0)$ as singular point. Thus π_{θ_1} is projection from $(1 : 0 : 0 : 0)$. Now the tangent lines at C_4 through $(0 : \pm r : \pm s : 1)$ are given by $x_2 \mp rx_4 = x_3 \mp sx_4 = 0$, which obviously go through $(1 : 0 : 0 : 0)$. Hence $(0 : \pm r : \pm s : 1)$ are the ramification points of π_{θ_1}. □

In particular we get the

Corollary 2.4.2. *If S is a preimage of $(\theta_i, 0)$ under ϕ_4, then S is a hyperosculating point.*

Proof. The tangent plane to the singular quadric at S meets C_4 four times in S. □

Remark 2.4.3. *The proof of the proposition shows that the four ramification points of π_{θ_i} lie on a plane. In fact, $(0 : \pm r : \pm s : 1)$ lie on $x_1 = 0$, and since we just need a linear change of variables to bring Q_1 and Q_2 simultaneously in diagonal form, this holds in general.*

There is a more abstract point of view on this setting. It is enlightening to see this, too. For that we change from the curves to their Picard groups. Since C_4 has genus 1, we can identify its points with $\mathrm{Pic}^1(C_4)$. By the group law in the 4-Selmer group we can add two homogenous spaces to get another one, in particular we have $[C_4] + [C_4] = [C_2]$, where $[C_n]$ means the class of the n-covering $C_n \to E$ in the n-Selmer group. This means that we can identifiy C_2 with $\mathrm{Pic}^2(C_4)$ and the following diagram

$$\begin{tikzcd}
C_4 \arrow[r, "\phi_4"] \arrow[d, leftrightarrow] & C_2 \arrow[d, leftrightarrow] \\
\mathrm{Pic}^1(C_4) \arrow[r, "\cdot 2"] & \mathrm{Pic}^2(C_4)
\end{tikzcd}$$

commutes. In addition, we can identify $\mathrm{Pic}^4(C_4)$ with $\mathrm{Pic}^0(C_4)$. This is possible, since C_4 has degree 4, hence a plane H meets C_4 in four points counting multiplicities, thus we have the bijection

$$\mathrm{Pic}^4(C_4) \to \mathrm{Pic}^0(C_4), \quad D \mapsto D - (H)_{C_4},$$

where we subtract the divisor cut out by H. Finally, we have the well known identification of $\mathrm{Pic}^0(C_4)$ and E.

With these identifications, we have the diagram

$$\begin{tikzcd}
C_4 \arrow[r, "\phi_4"] \arrow[d, leftrightarrow] & C_2 \arrow[rr, "\phi_2"] \arrow[d, leftrightarrow] & & E \arrow[d, leftrightarrow] \\
\mathrm{Pic}^1(C_4) \arrow[r, "\cdot 2"] & \mathrm{Pic}^2(C_4) \arrow[r, "\cdot 2"] & \mathrm{Pic}^4(C_4) \arrow[r, "\cong"] & \mathrm{Pic}^0(C_4)
\end{tikzcd}$$

which commutes by Proposition 2.4.1. This diagram coincides with the intuition that ϕ_2 and ϕ_4 correspond to multiplication by 2.

Another possibility to prove Proposition 2.4.1 would be to show that the map $w \circ \Delta$ coincides with ϕ_4 where $\Delta : C_4 \to C_4 \times C_4$, $P \mapsto (P, P)$, and $w : C_4 \times C_4 \to C_2$ is the map described in [28] and [15] by the recipe: if P_1, P_2 denote points of C_4 there is a unique point $\lambda = (\lambda_1 : \lambda_2) \in \mathbb{P}^1$ such that the line through P_1 and P_2 lies in the quadric Q_λ, and then $G(\lambda_1, \lambda_2)$ is a square, hence gives a point on $C_2 : y^2 = G(x, z)$ where G is the homogenization of g.

Now if $w \circ \Delta = \phi_4$, then Proposition 2.4.1 would follow by the nice geometrical argument: Let S be a ramification point of π_{θ_i}, then the tangent line at S goes through the singularity of Q_{θ_i}, hence is contained in Q_{θ_i}, thus $w(S, S) = (\theta_i, 0)$

Finally, I want to conclude that the 16 hyperosculating points correspond to the 4-torsion points on E, a result which might be well known to the experts.

Corollary 2.4.4. *If we consider C_4 as an elliptic curve over $\bar{\mathbb{Q}}$ with a hyperosculating point as the origin for the group law, then the hyperosculating points are the 4-torsion points on the elliptic curve C_4.*

Proof. By Proposition 2.4.1 we can choose $\bar{\mathbb{Q}}$-isomorphisms $\psi_2 : C_2 \dashrightarrow E$ by $(\theta_1, 0) \mapsto O$ and $\psi_4 : C_4 \dashrightarrow E$ by $S \mapsto O$ for a hyperosculating point

$S \in \phi_4^{-1}(\theta_i, 0)$ to get the commutative diagram

$$\begin{tikzcd}
C_4 \arrow[r, "\phi_4"] \arrow[d, dotted, "\psi_4"'] & C_2 \arrow[d, dotted, "\psi_2"'] \\
E \arrow[r, "{[2]}"] & E.
\end{tikzcd}$$

Hence ψ_2 maps the points $(\theta_i, 0)$ to $E[2]$ and ψ_4 maps the hyperosculating points to $E[4]$. □

A different proof for the corollary can be found in the end of [1]

Chapter 3

The Descent Map

3.1 Construction of F

For doing 8-descent, we want to use functions on the curve C_4. In the following I describe how to construct these functions and how to put them together to get a descent map F.

Let $Q_{\theta_1}, \ldots, Q_{\theta_4}$ be the four singular quadrics in the pencil and $\pi_{\theta_i} : \mathbb{P}^3 \to \mathbb{P}^2$ be the projection from the singular point of Q_{θ_i}. The image of Q_{θ_i} under π_{θ_i} is a conic C_{θ_i} over $\mathbb{Q}[\theta_i]$, which has a point since C_4 is everywhere locally solvable, compare Section 2.3. We can find this point by diagonalizing C_{θ_i} and solving a norm equation as we will see in the next section.

If we found a point on C_{θ_i}, we can compute the tangent line to it. Its preimage under the projection gives a tangent plane at Q_{θ_i}, which is given by a linear form, say L_i. For technical reasons L_i should have integral coefficients, i.e. coefficients in the ring of integers of K_i, so I just scale it. There might be better methods, but for our purpose this seems to be good enough.

For each Galois orbit of $\mathcal{R} = \{(\theta_i, 0) \mid i \in \{1, \ldots, 4\}\}$ we compute one tangent plane given by L_i, and for the conjugates in the orbit we take the conjugates of L_i, i.e. we take the map $\theta_i \mapsto L_i$ Galois-equivariant and get $L := (L_i) \in A[C_4]$. For the map F we then take

$$\begin{aligned} F : C_4(\mathbb{Q}) &\longrightarrow A^*/A^{*2}\mathbb{Q}^*, \\ P &\longmapsto L(P). \end{aligned}$$

Here $L(P)$ is an abbreviation for $(L_i(P))_i \in \prod K_i \cong A$ where we take one L_i from each conjugacy class. If $L_i(P) = 0$ for some i, then one of the tangent planes meets C_4 in a $\mathbb{Q}$-rational point, which we can easily check beforehand. In this case we do not need to do any descent anymore, since we already found a point on C_4. We also need this map locally. For that we apply the canonical map $A \to A_v$, where $A_v := A \otimes \mathbb{Q}_v$, to the coefficients of L and get $L_v := (L_{v,i}) \in A_v[C_4]$, and $F_v : C_4(\mathbb{Q}_v) \to A_v^*/A_v^{*2}\mathbb{Q}^*$, $P_v \mapsto L_v(P_v)$. However, here it can happen, that $L_{v,i}(P_v) = 0$ fore some i, and then we have to adjust the definition of F, see Section 4.2.

The idea for using these tangent planes for doing a descent comes from Cassels method [6] for doing a 4-descent. However, the difference between his setting and our situation is, that in his case the étale algebra is of the form $\mathbb{Q}[T]/(f(T))$, where f is the defining polynomial of the underlying elliptic curve $y^2 = f(x)$ and one of the linear forms, say L_4, is already defined over $\mathbb{Q}$. Hence he can use the functions L_i/L_4 on the curve to define a map into $(\mathbb{Q}[T]/(f(T)))^*$ modulo squares. In contrast to that, we do not have functions on the curve, but only linear forms to build up F. That is the reason why we have to divide by $\mathbb{Q}^*$ in addition, since a point on the projective curve C_4 is only defined up to a scalar.

However, one can consider F to be defined by functions on C_4, too. If we divide L_i by a linear form over $\mathbb{Q}$, for example x_4, then L_i/x_4 is a function on C_4 and F is equal to the map

$$C_4(\mathbb{Q}) \longrightarrow A^*/A^{*2}\mathbb{Q}^*, \quad P \longmapsto \left(\frac{L_i}{x_4}(P)\right)_i .$$

In fact, $\frac{L_i}{x_4}(P) \equiv L_i(P) \mod \mathbb{Q}^*$. Here the role of $\mathbb{Q}^*$ can be interpreted as absorbing the poles of the function L_i/x_4. For some proofs it is useful to know that F can be defined by functions on C_4 rather than just by linear forms.

The code for computing F in Magma is very nice due to the high level of the language. If we have a singular quadric, we have to project from the singular point to get a conic, on which we have to search a point.

```
function ConicOfSingularQuadric(Qsing)
  // Qsing is a singular quadric.
  pt := SingularPoints(Qsing)[1];
  C<[z]>, projection := Projection(Qsing,pt);
  // This is a conic and the projection P3 -> P2.
  return ImprovedIntegralModel(C), projection;
end function;
```

If we are able to compute a point on a conic, we can continue by taking the tangent line to the conic at this point, and lift it under the projection map. With `ImprovedIntegralModel` I just multiply through with the lowest common denominator to make the coefficients integral.

```
function TangentPlaneAt(Qsing : Point := Point)
```

```
  // Qsing is a singular quadric in the pencil.
  C, projection := ConicOfSingularQuadric(Qsing);
  vprintf EightDescent, 1 :
  "Projection from the singularity leads to a conic given by %o.\n",
  DefiningPolynomial(C);
  pt := PointOnConic(C : Point := Point);
  line := TangentLine(pt);
  vprintf EightDescent, 1 : "The tangent line to this point is %o, ",
  DefiningPolynomial(line);
  t := line @@ projection; // t is the tangent plane.
  // Make it integral:
  t := ImprovedIntegralModel(t);
  vprintf EightDescent, 1 :
  "which lifts under the projection to the tangent plane %o.\n\n",
  DefiningPolynomial(t);
  return t;
end function;
```

If we now evaluate the linear forms L_i, which define the tangent planes, at points of C_4, we get the map F. For technical reasons, I define F as a map starting at the set of sequences rather than C_4, because I want to avoid that every point at which I evaluate F is checked to be in C_4 and gets normalized by Magma.

```
function TheMapF(C4,L)
  // L = <L[1]> or <L[1],L[2]>.
  C4seq := PowerSequence(Integers());
  A := C4`EtaleAlgebra; iso := A`AbsoluteMap;
  F := map<C4seq -> A | pt :-> <Evaluate(L[i],pt) : i in [1..#L]>@@iso >;
  return F;
end function;
```

3.2 Finding a Point on a Conic

For constructing our map F, we have to find a point on a conic over a number field. This is a crucial step, which needs much computational power.

Over $\mathbb{Q}$ there are very fast algorithms for finding points on conics. However, we have to solve a conic over a number field, which generically has degree 4.

3.2.1 Diagonalizing a Conic

Diagonalizing a quadratic form is standard. I write it up here, since we are interested in finding points on a conic, and when diagonalizing, there might show up some easy points. In addition, from the explicit formulas we can read off, what happens to the coefficients, and why they get blown up so much.

Let $C : q = 0$, where $q := ax_1^2 + bx_1x_2 + cx_1x_3 + dx_2^2 + ex_2x_3 + fx_3^2$, $a, b, c, d, e, f \in K$ be a conic in $\mathbb{P}^3$. We want to find an isomorphism of $\mathbb{P}^3$ such that C is isomorphic to $C_{\text{diag}} : \alpha x_1^2 + x_2^2 + \beta x_3^2 = 0$, $\alpha, \beta \in K$.

If $a = 0$ then $(1 : 0 : 0)$ is a point on C. Similar for d and f. If $b^2 - 4ad = 0$ then $(b : -2a : 0)$ is a point on C. And if $4adf - ae^2 - b^2f + bce - c^2d = 0$, then $(2cd - be : 2ae - bc : b^2 - 4ad)$ is a point on C.

In all the other cases we can define the diagonalized conic by

$$C_{\text{diag}} : \alpha x_1^2 + x_2^2 + \beta x_3^2 = 0$$

with $\alpha := 4ad - b^2$ and $\beta := 4a(4adf - ae^2 - b^2f + bce - c^2d)$, and the isomorphism $\phi : C \to C_{\text{diag}}$ is given by $(2ax_1 + bx_2 + cx_3 : (4ad - b^2)x_2 + (2ae - bc)x_3 : x_3)$.

Notice, that if a, b, c, d, e, f are integral, so are α and β. The coefficient β is a polynomial of degree 4 in a, b, c, d, e, f, thus the order of magnitude of its height is about 4 times as large as the one of a, b, c, d, e, f. This blow up of the coefficients might make a nice looking conic into one with huge coefficients. However, now we can apply the machinery for solving norm equations, if our coefficients and the number field are in the range of practicability.

3.2.2 Finding a Point by Solving a Norm Equation

If we have a conic in diagonal form, then we can use the machinery for solving norm equations to find a point on it. This is standard knowledge, however, for completeness I include it.

We can assume that the conic is of the form $C : \alpha x_1^2 + x_2^2 + \beta x_3^2 = 0$, $\alpha, \beta \in K$, since we can divide by the coefficient of x_2^2. If $-\alpha$ is a square in K, say $\xi^2 = -\alpha$, then $(\xi : \alpha : 0)$ is a point on C. Else, $K[\sqrt{-\alpha}]|K$ is a field extension of degree 2, and if we can find a solution to the norm equation

$$N_{K[\sqrt{-\alpha}]|K}(\,\_\,) = -\beta,$$

say $\eta \in K[\sqrt{-\alpha}]$, then we have $\eta = \eta_1 + \sqrt{-\alpha}\eta_2$, $\eta_1, \eta_2 \in K$, such that $N_{K[\sqrt{-\alpha}]|K}(\eta) = \eta_1^2 + \alpha\eta_2^2 = -\beta$, hence $(\eta_2 : \eta_1 : 1)$ is a point on C.

Solving norm equations is an own area of research. There are algorithms for doing this, however, they are quite limited. Number fields of degree 8 such as $K[\sqrt{-\alpha}]$ appear to be feasible most times in practice. One obstacle is, that for computing the norm equation, we have to know the class group of $K[\sqrt{-\alpha}]$. This might be a serious problem. However, we do not need a provable result about the class group. It is enough if we computed it under the Generalized Riemann Hypothesis or even using a lower bound. We do not even care if the class group we computed is wrong, if in the end we get a point on the conic. If we have a point, it is trivial to check that it is on the conic, no matter how we got it.

```
function PointOnDiagonalizedConic(Cdiag)
  // Cdiag is defined by alpha*z[1]^2 + z[2]^2 + beta*z[3]^3 = 0.
  P2<[z]> := AmbientSpace(Cdiag);
  K := BaseField(Cdiag);
  fdiag := DefiningPolynomial(Cdiag);
  alpha := MonomialCoefficient(fdiag,z[1]^2);
  beta := MonomialCoefficient(fdiag,z[3]^2);
  // If -alpha is a square, we have a trivial solution,
  // the same for -beta,
  // else we go on with NormEquation.
  bool1, sqrt1 := IsSquare(-alpha);
  bool2, sqrt2 := IsSquare(-beta);
  if bool1 then
    pt := [sqrt1,alpha,0];
    vprintf EightDescent,1: "Negative of first coefficient is a square, ";
  elif bool2 then
    pt := [0,beta,sqrt2];
    vprintf EightDescent,1: "Negative of third coefficient is a square, ";
  else
    L := NumberField(PolynomialRing(K)![alpha,0,1]);
    // i.e. L = K[sqrt(-alpha)]
    // To speed up the class group computation we use a small bound:
    vprintf EightDescent, 2 :
    "Starting computation of the class group with bound = 200.\n";
    vtime EightDescent,2: _ := ClassGroup(AbsoluteField(L) : Bound:=200);
    vprintf EightDescent, 2 : "Starting solving the norm equation.\n";
    vtime EightDescent,2: bool, solutions := NormEquation(L,-beta);
    assert bool;
    vprintf EightDescent, 2 : "Found a solution to the norm equation,";
    s := solutions[1];
    pt := Reverse(Eltseq(s)) cat [1];
  end if;
  return Cdiag!pt;
end function;
```

The above code works fine in the split case, however in the generic case, this might be too slow. Denis Simon has a more elaborate code using norm equations for solving conics in his PARI/GP program `bnfqfsolve2`, which worked in all examples I tried. It is a good idea to reduce the coefficients of C before starting to apply the norm equations machinery. This idea was worked out by Denis Simon and implemented in the function `bnfqfsolve`. These programs are contained in the file ell.gp available at Denis Simon's web page.

Over $\mathbb{Q}$ there are other methods to solve conics, which do not need diagonal forms. Thus one can avoid the blowing up of the coefficients when diagonalizing. Denis Simon found a method for solving non-diagonal conics over $\mathbb{Q}$. However, this algorithm is hardly extendible to number fields.

3.3 Independence of the Choice of the Point on the Conic

In the construction of F there is some choice involved, namely the choice of the point on the conic. If we take a different point on the conic, we get different tangent planes, hence a different looking map F. However, F should be independent of this choice. This is what we will see below.

We look at the generic case $A \cong K_1$, the split case is analogous. If we have a point on the conic, we get a tangent plane $L_1 = 0$ to the singular quadric as described above. If we take a different point on the conic, we get a different tangent plane, say $L_1' = 0$. L_1 and L_1' are defined over K_1. Let $F : C_4(\mathbb{Q}) \to A^*/A^{*2}\mathbb{Q}^*$, $P \mapsto L_1(P)$, and $F' : C_4(\mathbb{Q}) \to A^*/A^{*2}\mathbb{Q}^*$, $P \mapsto L_i'(P)$ be the corresponding maps.

One would like to have that F and F' coincide. This is almost true. They coincide up to a translation.

Lemma 3.3.1. *There is a $\gamma \in A^*$ such that for every point $P \in C_4(\mathbb{Q})$*

$$F(P) = \gamma F'(P).$$

Proof. We just look at the generic case, when $A \cong K_1$ is a number field. The split case is analogous. Write $(L_1)_{C_4} = 2(S_1 + S_2)$ and $(L_1')_{C_4} = 2(S_1' + S_2')$ the divisors cut out by L_1, L_1' respectively. Then S_1, S_2, S_1', and S_2' lie on a plane. In fact, the planes $L_1 = 0$ and $L_1' = 0$ are tangent to the singular quadric $Q_{\theta_1} = 0$, hence the two lines $L_1 = Q_{\theta_1} = 0$ and $L_1' = Q_{\theta_1} = 0$ meet in the singularity of $Q_{\theta_1} = 0$, thus are contained in a plane defined over K_1, say $H = 0$. Thus $S_1 + S_2 + S_1' + S_2' = (H)_{C_4}$ the divisor cut out by H, hence $(L_1 L_1')_{C_4} = (H^2)_{C_4}$, which means $L_1 L_1' = \gamma H^2$ in $K_1[C_4]$ for some $\gamma \in K_1$. Hence modulo squares $L_1(P) \equiv \gamma L_1'(P)$ for every $P \in C_4(\mathbb{Q})$. □

This means that doing a descent using F gives the same information as doing a descent using F'.

We will use this lemma in practice to reduce the set of bad primes, see Section 4.3. The γ and the H in $L_1 L_1' = \gamma H^2$ can be computed with the following functions. $H = 0$ is the plane containing the two tangent lines, thus we take three points on these two lines, the point of intersection and one on each line, and compute the plane through these three points.

```
function PlaneThrough(ThreePoints)
  // Given three points in P3 over K.
  // Return the plane over K through the three points.
  P3<[x]> := Ambient(Scheme(Rep(ThreePoints)));
  K := BaseField(P3);
  Kc<[c]> := PolynomialRing(K,4);
  // c[1],...,c[4] will be the coefficients of H.
  KcX<[X]> := PolynomialRing(Kc,4);
  H := &+[c[i]*X[i] : i in [1..4]];
  // the unknown plane.
  HPs := [Evaluate(H,Eltseq(P)) : P in ThreePoints];
  // H(P) must be zero for P in ThreePoints.
  coeffs := [[MonomialCoefficient(HP,c[i]) : i in [1..4]] : HP in HPs];
  Ker := Kernel(Transpose(Matrix(coeffs)));
  vprintf EightDescent,3: "Number of possible planes: %o. Expected 1.
  \n",Dimension(Ker);
  c := Eltseq(Basis(Ker)[1]);
  H := &+[c[i]*x[i] : i in [1..4]];
  return H;
end function;
```

Now we are able to compute the plane $H = 0$ going through S_1, S_2, S_1', and S_2'. Then the constant γ in the equation $L_1L_1' = \gamma H^2$ could be obtained by evaluating L_1L_1' and H^2 at any point of $C_4(\bar{\mathbb{Q}})$ different from S_1, S_2, S_1', and S_2' and take the quotient of these two. A different method, which we use here is to take the unique normal form of L_1L_1' and H^2 with respect to $I(C_4)$ and take the quotient of them[1].

```
function TheGamma(C4,L1,L11,Qsing)
  // L1/L1' = gamma*q^2 on C4. This function computes the gamma.
  // Or: L1*L1' = gamma*H^2. (q = H^2/L1'^2)
  // Notation: L1' = L11.
  vprint EightDescent,3: "Computing gamma...";
  P3<[x]> := Ambient(Qsing);
  K := BaseField(P3);
  Kx := CoordinateRing(P3);
  L1 := Kx!L1;
  Q1,Q2 := Explode(ChangeUniverse(DefiningPolynomials(C4),Kx));
  pt1 := SingularPoints(Qsing)[1];
  i := 1; // If pt1[i] = 0, then we take a different i.
  while pt1[i] eq 0 do i+:=1; end while;
  ThreePoints := {pt1} join Points(Scheme(Qsing,[L1,x[i]]))
  join Points(Scheme(Qsing,[L11,x[i]]));
  vprint EightDescent,3: "Computing the plane through the three points.";
  vtime EightDescent,3: H := PlaneThrough(ThreePoints);
```

[1] γ can be computed already using the tangent lines on the conic without having to compute H. This might be a little bit faster.

```
  I := ideal<Kx | DefiningPolynomials(C4)>;
  vprintf EightDescent,3:
  "The plane H with L1 L1' = gamma H^2 is\n%o\n", H;
  // Now gamma:
  return K!( NormalForm(L1*L11,I)/NormalForm(H^2,I) );
end function;
```

3.4 F as a Homomorphism on $\mathrm{Pic}(C_4)$

Since C_4 has genus 1, we can identify $C_4(\mathbb{Q})$ with $\mathrm{Pic}^1(C_4)$. And $\mathrm{Pic}^1(C_4)$ is a coset in the group $\mathrm{Pic}(C_4)$. Is F compatible with this group law?

Yes, it is. We can even define F as a homomorphism on all of $\mathrm{Pic}(C_4)$. The definition is canonical, the well-definedness follows from Weil-Reciprocity as we will see below.

Let $[D]$ be a divisor class in $\mathrm{Pic}(C_4)$. By the Moving Lemma, we can assume that D has disjoint support with the divisor $(L_1L_2L_3L_4)_{C_4}$ cut out by the four planes.

Write $D = \sum_P n_P P$. Then we can define $F([D]) := \prod_P F(P)^{n_P}$. Now F is clearly a homomorphism, if we can show that it is well-defined.

Lemma 3.4.1. $F : \mathrm{Pic}(C_4) \to A^*/A^{*2}\mathbb{Q}^*$ *is well-defined.*

Proof. Let D_1 and D_2 be linearly equivalent divisors, i.e. $D_1 - D_2 = \mathrm{div}(h)$ for some $h \in \mathbb{Q}(C_4)$. Write $(L_i)_{C_4} = 2(S_1^{\theta_i} + S_2^{\theta_i})$ the divisor cut out by L_i, and $(x_4)_{C_4} = S_\infty$ the divisor cut out by x_4. By Weil-Reciprocity

$$\begin{aligned} F(\mathrm{div}(h)) &= (L_i/x_4(\mathrm{div}(h)))_i \\ &= (h(\mathrm{div}(L_i/x_4)))_i \\ &= (h(2(S_1^{\theta_i} + S_2^{\theta_i}) - S_\infty))_i \\ &= (h(S_1^{\theta_i} + S_2^{\theta_i})^2 h(S_\infty)^{-1})_i \equiv 1 \mod A^{*2}\mathbb{Q}^*, \end{aligned}$$

since h and S_∞ are defined over $\mathbb{Q}$. Hence $F(D_1) = F(D_2)$. □

Corollary 3.4.2. *If* $C_4(\mathbb{Q}) \neq \emptyset$*, then* $F(C_4(\mathbb{Q}))$ *is a coset in* $A^*/A^{*2}\mathbb{Q}^*$*.*

Proof. $C_4(\mathbb{Q})$ can be identified with $\mathrm{Pic}^1(C_4)$, which is a coset in $\mathrm{Pic}(C_4)$, and F is a homomorphism. □

3.5 F Modulo the Action of $2E(\mathbb{Q})$

Since C_4 is a homogeneous space for E, we have an action of E on C_4. Since we want to perform a third 2-descent on C_4, we want that F is not affected by the action of $2E(\mathbb{Q})$ on $C_4(\mathbb{Q})$. With our interpretation of F as a homomorphism on all of $\mathrm{Pic}(C_4)$, this just means that $2E(\mathbb{Q})$ is in the kernel of F.

Lemma 3.5.1. *We have*

$$2E(\mathbb{Q}) \subset \ker(F).$$

Proof. Let $P \in E(\mathbb{Q})$. By the identification $E(\mathbb{Q}) \cong \mathrm{Pic}^0(C_4)$ we can consider P as a degree 0 divisor class on C_4. Now by definition $F(2P) = F(P)^2$, which is trivial in $A^*/A^{*2}\mathbb{Q}^*$. □

We will see later that $2E(\mathbb{Q})$ has index 2 in $\ker(F)$. If $\ker(F)$ was too large, then F would not be useful for a descent. It is analogous to the situation of 4-descent, where the kernel of the $x-T$-map is a little bit larger than $2E(\mathbb{Q})$, and just leads to the identification of ±1.

3.6 Comparison of F and the $x - T$-map

In the following we identify $\mathrm{Pic}^0(C_4)$ and $\mathrm{Pic}^0(C_2)$ with E. For the class of a divisor D we write $[D]$.

Lemma 3.6.1. *Let $P_1, P_2 \in C_4(\bar{K})$. Then $[2][P_1 - P_2] = [\phi_4(P_1) - \phi_4(P_2)]$ on E.*

Proof. For any homogenous space and the corresponding $\bar{K}$-isomorphism $\gamma : C \to E$ we have $[\gamma(P_1) - \gamma(P_2)] = [P_1 - P_2]$.

We have the commuting diagram

$$\begin{tikzcd}
C_4 \arrow[r, "\phi_4"] \arrow[d, dotted, "\alpha"'] & C_2 \arrow[d, dotted, "\beta"] \\
E \arrow[r, "{[2]}"] & E,
\end{tikzcd}$$

thus $[\phi_4(P_1) - \phi_4(P_2)] = [\beta^{-1}[2]\alpha(P_1) - \beta^{-1}[2]\alpha(P_2)] = [2][\alpha(P_1) - \alpha(P_2)] = [2][P_1 - P_2]$. □

Now we interpret $F : C_4(K) \to A^*/A^{*2}K^*$ as given by functions on C_4, namely L_i/x_4. Let $(L_i)_{C_4} = 2(S_1^{\theta_i} + S_2^{\theta_i})$ and $(x_4)_{C_4} = S_\infty$. We think of $2(S_1^{\theta_i} + S_2^{\theta_i})$, $i = 1, \ldots, 4$ as the zeros, and S_∞ as the poles of F. Let S^{θ_i} be a ramification point of π_{θ_i}. Then we have the following

Lemma 3.6.2. $S_1^{\theta_i} + S_2^{\theta_i} \sim 2S^{\theta_i}$.

Proof. Let H_1 be the plane through $S_1^{\theta_i}$, $S_2^{\theta_i}$, S^{θ_i}, and H_2 the hyperosculating plane at S^{θ_i}. Since H_1 contains the tangent line at S^{θ_i}, we have $\operatorname{div}(H_1/H_2) = S_1^{\theta_i} + S_2^{\theta_i} - 2S^{\theta_i}$. □

Let R^{θ_i} be the point $(\theta_i, 0)$ on C_2. Thus $\operatorname{div}(x - \theta_i) = 2R^{\theta_i} - R_\infty$ where R_∞ is the divisor above infinity.

Theorem 3.6.3. *$[S_1^{\theta_i} + S_2^{\theta_i} - (S_1^{\theta_j} + S_2^{\theta_j})] = [R^{\theta_i} - R^{\theta_j}]$ on E.*

Proof. Let S^{θ_i} be a ramification point of the double covering $\pi_{\theta_i} : C_4 \to \mathbb{P}^1$ given by projection from the singular point of $Q_{\theta_i} : \theta_i Q_1 + Q_2 = 0$. By the previous lemma $S_1^{\theta_i} + S_2^{\theta_i} \sim 2S^{\theta_i}$, thus $[S_1^{\theta_i} + S_2^{\theta_i} - (S_1^{\theta_j} + S_2^{\theta_j})] = [2][S^{\theta_i} - S^{\theta_j}] = [\phi_4(S^{\theta_i}) - \phi_4(S^{\theta_j})] = [R^{\theta_i} - R^{\theta_j}]$ by Lemma 3.6.1 and Proposition 2.4.1. □

Corollary 3.6.4. *Let $T_i := [S_1^{\theta_i} + S_2^{\theta_i} - (S_1^{\theta_4} + S_2^{\theta_4})]$, $i = 1, \ldots, 4$. Then $\{T_1, \ldots, T_4\} = E[2]$.*

Proof. By the previous theorem $T_i = [R^{\theta_i} - R^{\theta_4}]$. And in the commutative diagram

$$\begin{tikzcd}
C_2 \arrow[r, "\phi_2"] \arrow[d, dotted, "\psi_2"'] & E \arrow[d, equal] \\
E \arrow[r, "{[2]}"] & E
\end{tikzcd}$$

the $\bar{\mathbb{Q}}$-isomorphism ψ_2 maps $\{R^{\theta_1}, \ldots, R^{\theta_4}\}$ bijectively to $E[2]$. □

This corollary is an important statement. It shows, that the tangent planes defining F correspond to the 2-torsion points on E. This fact will be an important ingredient in the proof, that F is the right map for doing an 8-descent.

The next theorem shows that the image of F coincides with the image of the $x - T$-map up to translation. Here $x - T : C_2(\mathbb{Q}) \to A^*/A^{*2}\mathbb{Q}^*$, $(x, y) \mapsto x - \theta$, as in Section 1.4.1. This result is very useful to determine the local image of F in practice, since we have to compute just one point on C_4 and for the rest we can use the local image of $x - T$, we computed already during the 4-descent. We will also use this result for the cohomological interpretation of the map F.

Theorem 3.6.5. *Let K be $\mathbb{Q}$ or $\mathbb{Q}_v$. If $C_4(K) \neq \emptyset$, then*

$$\operatorname{im}(F) = \alpha \cdot \operatorname{im}(x - T)$$

for some $\alpha \in A^*$.

Proof. Since C_4 has a K-rational point, so has C_2, and hence they are both isomorphic to E, thus we have a K-isomorphism $\psi : C_4 \to C_2$. Let $R^{\theta_i} := (\theta_i, 0)$ on C_2 and $\psi^* R^{\theta_i}$ its preimage. The divisor $S_1^{\theta_i} + S_2^{\theta_i} - \psi^* R^{\theta_i}$ has degree 1, hence is linearly equivalent to a unique point P^{θ_i}, thus there is a function $H_{\theta_i} \in K[\theta_i](C_4)$ such that $\operatorname{div}(H_{\theta_i}) = S_1^{\theta_i} + S_2^{\theta_i} - \psi^* R^{\theta_i} - P^{\theta_i}$.

P^{θ_i} is independent of θ_i, in particular $P^{\theta_i} \in C_4(K)$: By the previous theorem we have $[S_1^{\theta_i} + S_2^{\theta_i} - (S_1^{\theta_j} + S_2^{\theta_j})] = [R^{\theta_i} - R^{\theta_j}] = [\psi^*(R^{\theta_i}) - \psi^*(R^{\theta_j})]$, thus $S_1^{\theta_i} + S_2^{\theta_i} - \psi^* R^{\theta_i} \sim S_1^{\theta_j} + S_2^{\theta_j} - \psi^* R^{\theta_j}$, hence $P^{\theta_i} = P^{\theta_j} =: P$.

Now

$$\begin{aligned} \operatorname{div}\left(\frac{F_{\theta_i}}{H_{\theta_i}^2} \cdot \frac{1}{\psi^*(x - \theta_i)}\right) &= 2(S_1^{\theta_i} + S_2^{\theta_i}) - S_\infty - 2(S_1^{\theta_i} + S_2^{\theta_i} - \psi^* R^{\theta_i} - P) \\ &\quad - 2\psi^* R^{\theta_i} + \psi^* R_\infty \\ &= -S_\infty + 2P + \psi^* R_\infty \end{aligned}$$

is independent of θ_i, thus equal to $\operatorname{div}(G)$ for some $G \in K(C_4)$. This means $\operatorname{div}(\frac{F_{\theta_i}}{H_{\theta_i}^2 G}) = \operatorname{div}(\psi^*(x - \theta_i))$, thus $\frac{F_{\theta_i}}{H_{\theta_i}^2 G} = \alpha_i \psi^*(x - \theta_i)$ for some $\alpha_i \in K[\theta_i]$. In $A^*/A^{*2}K^*$ we get $F(C_4(K)) \equiv (\frac{F_{\theta_i}}{H_{\theta_i}^2 G}(C_4(K)))_i = (\alpha_i(x - \theta_i)(\psi(C_4(K))))_i = \alpha(x - T)(C_2(K))$ for $\alpha := (\alpha_i)$. □

Chapter 4

The Fake Selmer Set

4.1 Definition

Let $\mathrm{res}_v : A^*/A^{*2}\mathbb{Q}^* \to A_v^*/A_v^{*2}\mathbb{Q}_v^*$ be the canonical restriction map. Then we define the *fake 2-Selmer set* of C_4 as the intersection of the local images of F:

$$\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q}) := \bigcap_v \mathrm{res}_v^{-1}(F(C_4(\mathbb{Q}_v))).$$

In the following sections we will find conditions that show that $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q})$ is a finite and computable set. One reason why it is so interesting to be able to compute the fake Selmer set is that it gives us a tool to prove that C_4 does not have a rational point in certain cases.

Proposition 4.1.1. *We have*

$$F(C_4(\mathbb{Q})) \subset \mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q}).$$

In particular, if $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q}) = \emptyset$*, then* $C_4(\mathbb{Q}) = \emptyset$*.*

Proof. Clear. □

4.2 The Norm Condition

Let $L = (\theta_i \mapsto L_i)$ the linear forms defining F. Then the product $L_1L_2L_3L_4$ can be considered as the norm of L. The aim of this section is to show that $L_1L_2L_3L_4$ is a square up to a constant.

The following lemma is a weaker version of the theorem we want to have. A similar statement for the 4-descent situation can be found in [6]. The linear forms L_i are not functions on C_4. To get functions on C_4 we divide by x_4. There is nothing special about x_4; one could replace it by any linear form which is defined over $\mathbb{Q}$.

Lemma 4.2.1. *There exists a function $q \in \mathbb{Q}(C_4)$ and a constant $c \in \mathbb{Q}$ such that*

$$\frac{L_1}{x_4} \cdot \ldots \cdot \frac{L_4}{x_4} = c \cdot q^2 \quad \text{on } C_4.$$

Proof. We have $\operatorname{div}(\frac{L_1}{x_4} \cdot \ldots \cdot \frac{L_4}{x_4}) = 2D$ where $D = \sum_{i=1}^4 S_1^{\theta_i} + S_2^{\theta_i} - 2\,(x_4)_{C_4}$. Since $(x_4)_{C_4} \sim (L_4)_{C_4}$, we have $D \sim \sum_{i=1}^4 (S_1^{\theta_i} + S_2^{\theta_i} - (S_1^{\theta_4} + S_2^{\theta_4}))$, which sums up to 0 by corollary 3.6.4, hence $D = \operatorname{div}(q)$ for some $q \in \mathbb{Q}(C_4)$. Therefore $\operatorname{div}(\frac{L_1}{x_4} \cdot \ldots \cdot \frac{L_4}{x_4}) = \operatorname{div}(q^2)$, hence $\frac{L_1}{x_4} \cdot \ldots \cdot \frac{L_4}{x_4} = c \cdot q^2$ for some $c \in \mathbb{Q}$. □

The divisor $(L_1 L_2 L_3 L_4)_{C_4}$ cut out by the four planes is twice a divisor, say D_8, since the planes are tangent to C_4, i.e.

$$(L_1 L_2 L_3 L_4)_{C_4} = 2D_8.$$

Lemma 4.2.2. *We have the linear equivalence*

$$D_8 \sim \left(x_4^2\right)_{C_4}.$$

Proof. By Lemma 4.2.1 we have $2D_8 - 2\,(x_4^2)_{C_4} = \operatorname{div}(q^2)$, hence $D_8 - (x_4^2)_{C_4} = \operatorname{div}(q)$. □

Next we want to show that D_8 is the divisor cut out by a surface of degree 2. For that, we need the following

Lemma 4.2.3. *The linear system $L_{C_4}(2)$ of degree 2 surface sections is complete.*

Proof. There are 10 forms $x_1^2, x_1x_2, \ldots, x_3x_4, x_4^2$ of degree 2, and there are two linear relations $Q_1 = Q_2 = 0$ among them, hence they span an at least 8 dimensional vector space, thus $L_{C_4}(2)$ is at least of dimension 7.

$L_{C_4}(2)$ is contained in the complete linear system $|(x_4^2)_{C_4}|$, which is of dimension $\deg((x_4^2)_{C_4}) - 1 = 7$, hence they must coincide. □

Since the linear system $L_{C_4}(2)$ is complete, we can prove a stronger version of Lemma 4.2.1. Instead of an identity of rational functions, we even get an identity of polynomials.

Theorem 4.2.4. *There exists a quadratic form $Q_3 \in \mathbb{Q}[x_1, \ldots, x_4]$ and a constant $c \in \mathbb{Q}$ such that*

$$L_1 L_2 L_3 L_4 - c \cdot Q_3^2 \in I(C_4)$$

where $I(C_4)$ is the homogeneous ideal of C_4.

Proof. By Lemma 4.2.2 and Lemma 4.2.3, $D_8 \in L_{C_4}(2)$, hence $D_8 = (Q_3)_{C_4}$ for some quadratic form Q_3. Since D_8 is defined over $\mathbb{Q}$, we can choose Q_3 over $\mathbb{Q}$. Hence $\operatorname{div}(\frac{L_1L_2L_3L_4}{Q_3^2}) = 0$, thus $\frac{L_1L_2L_3L_4}{Q_3^2} = c$ is constant. $L_1L_2L_3L_4$ and Q_3 are defined over $\mathbb{Q}$, so is c. This means $L_1L_2L_3L_4 - c \cdot Q_3^2 \in I(C_4)$. □

This has a consequence for the descent map F.

Corollary 4.2.5. *Let $N : A^*/A^{*2}\mathbb{Q}^* \to \mathbb{Q}/\mathbb{Q}^{*2}$ be induced by the norm $A \to \mathbb{Q}$. Then*

$$N(F(C_4(\mathbb{Q}))) = c.$$

The same is true locally.

Proof. Let $P \in C_4(\mathbb{Q})$. Then by Theorem 4.2.4 or even Lemma 4.2.1 we have $N(F(P)) = L_1L_2L_3L_4(P) = c \cdot Q_3(P)^2 = c$ in $\mathbb{Q}^*/\mathbb{Q}^{*2}$. This is true globally, in particular locally. □

In fact, F was not yet defined at zeros of the L_i, but the Corollary tells us how we have to define it. For a point P on C_4 such that some L_i vanishes at P, we have to replace $L_i(P)$ by any ξ such that $\xi \cdot \prod_{j \neq i} L_j(P) = c$. The same locally.

A method for computing Q_3 and c will be described below.

4.2.1 Algorithm for Finding Q_3 and c

The four planes intersect C_4 in the eight points D_8. The quadric $Q_3 = 0$ must go through these eight points. This gives us restrictions on Q_3, which will enable us to find the coefficients of Q_3.

The set of quadratic forms in 4 variables is a 10-dimensional $\mathbb{Q}$-vector space, spanned by the monomials $x_1^2, x_1x_2, \ldots, x_3x_4, x_4^2$. We want to determine the subspace of quadrics vanishing on D_8.

The eight equations: Generically, D_8 consists of one Galois orbit, else one has to work with one point from each Galois orbit. Let us consider the generic case. There we compute one point P of D_8. For that, we do a Groebner basis calculation where we enlarge the field step by step until we get a point on D_8. P is defined over a degree 8 number field K_8, and D_8 consists of all conjugates of P. A quadric Q in the space of quadrics vanishing on D_8 must fulfill the equation $Q(P) = 0$. This is an equation in the coefficients of Q. It is defined over K_8, and since we want Q to be defined over $\mathbb{Q}$, restriction of scalars gives 8 linear equations over $\mathbb{Q}$. These 8 equations determine the subspace of quadratic forms vanishing on D_8.

On the first sight, one might expect that this space is 2-dimensional, since it is a subspace of a 10-dimensional space determined by 8 equations.

However, already Q_1 and Q_2 are contained in this subspace, since they go through the eight points – they vanish even on all of C_4. Q_1 and Q_2 span a 2-dimensional space, namely the pencil of quadrics containing C_4.

By Theorem 4.2.4 there exists a third quadric Q_3, which goes through D_8, but does not vanish on all of C_4. Hence the vector space of quadrics that vanish on D_8 is at least 3-dimensional, which means that the 8 equations are not independent.

Now we can take any quadric in this 3-dimensional space, which is not already in the pencil of C_4, as our Q_3. Then Q_3 goes through the eight points, but does not vanish on all of C_4. It does not meet C_4 in more points, since a degree 4 curve and a degree 2 surface meet in exactly 8 points.

Now we found Q_3, but how do we get the constant c? For that, one can evaluate $L_1L_2L_3L_4$ and Q_3^2 at a point on $C_4(\bar{\mathbb{Q}})$, but outside D_8. Then the fraction of these two values is the constant c. Or, one can take the unique normal forms of $L_1L_2L_3L_4$ and Q_3^2 with respect to $I(C_4)$. The fraction of these two again is c.

4.2.2 The Implementation

The implementation is as follows: First, we compute one point in each Galois orbit of D_8. This can be done for general zero dimensional affine schemes. Note, that the main tool here is a Groebner basis calculation, and the factorization of the last element of the Groebner basis. Each irreducible factor defines a number field, and its generic zero is the last coordinate of the point. The rest is recursion.

```
function OnePointInEachGaloisOrbit(Z)
  vprint EightDescent,3: "Computing one point in each Galois orbit.";
  // Z is a zero dimensional affine scheme over a number field.
  // We want to find one point in each Galois orbit of Z(Qbar).
  L := BaseField(Z);
  Gr := GroebnerBasis(Z);
  g := Gr[#Gr]; // We take the last element.
  if g eq 1 then return []; end if; // If Z is empty.
  bool, g := IsUnivariate(g);
  assert bool; // g should be univariate, since Z is zero dimensional.
  fact := Factorization(g);
  // The roots of g in suitable number fields:
  fields := <NumberField(f[1] : DoLinearExtension) : f in fact>;
  d := Rank(Ideal(Z));
  //Now the d-th coordinate of our point is L.1.
  if d eq 1 then return [* <L.1> : L in fields *]; end if;
  // terminating condition.
  // Recursion:
```

```
  pts := < <OnePointInEachGaloisOrbit(Scheme(Spec(P), [Evaluate(h, x cat
  [L.1]) : h in Gr])),   L.1>  where x := [P.i : i in [1..d-1]] where P :=
  PolynomialRing(L, d-1)  :  L in fields >;
  points := [* <Append(tup, pt[2]) : tup in pt[1]>[1] : pt in pts *];
  // Turn the tuples into sequences:
  return [* [pt[i] : i in [1..#pt]] : pt in points *];
end function;
```

When a quadric Q has to go through a certain point, this gives some constraints on Q. The constraints are some linear equations in the coefficients of Q, which we store as a matrix.

```
function ConstraintsOnQuadricsThrough(P)
  vprintf EightDescent,3:
  "Computing the constraints on quadrics through the point... \n";
  // P is a point on C4 over some number field.
  // We search a quadric Q over the rationals
  // which goes through P.
  P := Eltseq(P);
  L := Universe(P);
  Labs := AbsoluteField(L);
  Lc<[c]> := PolynomialRing(Labs,10);
  // c[1],...,c[10] are the coefficients of Q.
  Lcx := PolynomialRing(Lc,4);
  xx := MonomialsOfDegree(Lcx,2);
  // xx = {x1^2, x1*x2, ..., x3*x4, x4^2}
  Q := &+[c[i]*xx[i] : i in [1..10]];
  // Q is the unknown quadric.
  QP := Evaluate(Q,P); // = Q(P).
  coeffs := [Eltseq(MonomialCoefficient(QP,c[i])) : i in [1..10]];
  return Matrix(coeffs);
end function;
```

To get Q_3, we take the constraints coming from all the points together, compute the space of solutions, which is the kernel of the corresponding matrix, and quotient out by Q_1 and Q_2.

```
function ThirdQuadric(C4,L)
  vprint EightDescent,3:
  "Computing the vector space of quadrics through the eight points.";
  ts := [Scheme(Proj(Parent(Li)),Li) : Li in L];
  Zs := [Scheme(t,DefiningPolynomials(C4)) : t in ts];
  error if &or[Dimension(Scheme(Z,Z.4)) gt -1 : Z in Zs],
  "ERROR in ThirdQuadric: We take the wrong affine patch.";
  Zs := [AffinePatch(Z,1) : Z in Zs];
  pts := &cat[ OnePointInEachGaloisOrbit(Z) : Z in Zs ]; // affine
  pts := [* P cat [1] : P in pts *]; // projective
  Ms := [ConstraintsOnQuadricsThrough(P) : P in pts];
  Mat := HorizontalJoin(Ms);
  vprint EightDescent,3:"Computing the kernel of the constraints matrix.";
  Ker := KernelMatrix(Mat);
  L := Lattice(Ker);
  L1 := PureLattice(L);
  // We mod out by Q1 and Q2.
  xx := MonomialsOfDegree(CoordinateRing(Ambient(C4)),2);
  L2, qmap := quo<L1 | [[MonomialCoefficient(Q,xx[i]) : i in [1..10]] : Q
  in DefiningPolynomials(C4)]>;
  L3 := TorsionFreeSubgroup(L2);
  assert Ngens(L3) ge 1;//space of quadrics mod <Q1,Q2> is at least 1-dim.
  c := Eltseq(L3.1 @@ qmap); // The coefficients of Q3.
  Q3 := &+[c[i]*xx[i] : i in [1..10]];
  return Q3;
end function;
```

The constant c is obtained by

```
function TheConstant(C4,L,Q3)
  Qx := CoordinateRing(Ambient(C4));
  I := Ideal(C4);
  FourPlanes := &*[Qx!ProductOfConjugates(Li) : Li in L];
  c := Rationals()!(NormalForm(FourPlanes,I)/NormalForm(Q3^2,I));
  return c;
end function;
```

where the product of the conjugates of the polynomial $L_i \in (\mathbb{Q}[\theta_i])[x_1,\dots,x_4]$ is computed by taking its norm when considered as an element of the ring $(\mathbb{Q}[x_1,\dots,x_4])[\theta_i]$.

```
function ProductOfConjugates(L1)
  Kx := Parent(L1);
  _, m := SwapExtension(Kx);
  return Norm(m(L1));
end function;
```

4.3 The Set of Bad Primes

For defining the set of bad primes we need the following notations. Let $\tilde{K}$ be the splitting field of g. For a prime $\mathfrak{P}$ of $\tilde{K}$ denote the residue class field by $\kappa_{\mathfrak{P}}$. We can consider L_i as a polynomial over the ring of integers of $\tilde{K}$, thus as a polynomial over $\kappa_{\mathfrak{P}}$. We denote by $\{L_i = 0 \mod \mathfrak{P}\}$ the $\kappa_{\mathfrak{P}}$-points of $L_i = 0$. In addition, we consider $C_4(\mathbb{F}_p)$ as a subset of $C_4(\kappa_{\mathfrak{P}})$.

Now we can define the set S of *bad primes* as the set of primes containing 2, ∞, and

1. the primes dividing $\operatorname{disc}(g) \cdot c$,
2. p such that there exists a prime $\mathfrak{P}$ of $\tilde{K}$ above p such that

$$C_4(\mathbb{F}_p) \cap \{L_i = 0 \mod \mathfrak{P}\} \cap \{L_j = 0 \mod \mathfrak{P}\} \neq \emptyset$$

   for some $i \neq j$.

The last condition means that two different tangent planes do not meet in a point on C_4 mod p.

The next question is, how to actually compute the set S. The following lemma shows that S is contained in a set, which can be computed explicitly. Let Q_3 be the third quadric as in the previous section and let $P_8 := \mathbf{Proj}(\mathbb{Z}[x_1, \ldots, x_4]/(Q_1, Q_2, Q_3))$. Recall that Q_3 intersects C_4 in the eight points where the four tangent planes $L_1, \ldots, L_4$ meet C_4, hence P_8 is the scheme of these eight points over $\mathbb{Z}$.

Lemma 4.3.1. *Let S' be the set of primes containing 2, ∞, and*

1. *the primes dividing* $\operatorname{disc}(g) \cdot c$*,*
2. *p such that a prime ideal above p divides all coefficients of L_1 (or L_2),*
3. *p such that $P_8 \mod p$ is singular.*

Then $S \subset S'$.

Proof. Let $p \notin S'$. Suppose $p \in S$, hence p must fulfill the last condition on S. Thus there is a point

$$P \in C_4(\mathbb{F}_p) \cap \{L_i = 0 \mod \mathfrak{P}\} \cap \{L_j = 0 \mod \mathfrak{P}\}$$

for some $i \neq j$ and some $\mathfrak{P}|p$.

Since $\mathfrak{P}$ does not divide all coefficients of L_i or L_j, P must be a singular point of P_8 mod $\mathfrak{P}$. Since P is $\mathbb{F}_p$-rational, it is a singular point of P_8 mod p, which contradicts the last property of S'. □

With the following Magmacode one can compute a set containing the set of bad primes. The primes from condition 2 can be computed by the following function. In fact, it might compute a few more than that. To avoid working with prime decompositions in number fields, we work with the norms of the coefficients, which lie in $\mathbb{Q}$, and look for their prime divisors. However, we might get more primes than condition 2 needs, since different conjugates above p may divide different coefficients. Thus we might have to check local solvability at more primes than necessary, but this seems to be faster than excluding superfluous primes.

```
function PrimesDividingTheNormsOfAllCoefficients(L1)
  // we assume L1 is integral.
  coeffs := Coefficients(L1);
  norms := [Norm(c) : c in coeffs];
  bad := GCD(ChangeUniverse(norms,Integers()));
  return Set(PrimeDivisors(bad));
end function;
```

A more conceptual way of phrasing condition 3 is that we take the closed points of the image of the projection from the singular subscheme of P_8 to $\mathrm{Spec}(\mathbb{Z})$. Projecting to $\mathrm{Spec}(\mathbb{Z})$ can be done by eliminating all variables.

```
function ProjectionToSpecZ(X)
  // X is a scheme over Z.
  vprint EightDescent,3: "Projecting to Spec(Z)";
  if IsAffine(X) then
    I := Ideal(X);
    idl := EliminationIdeal(I,{}); // eliminating all variables.
    bool, n := IsPrincipal(idl); assert bool;
    n := Integers()!n;
    if n ne 0 then return n; else return 1; end if;
  else // X is projective.
    d := Dimension(Ambient(X));
    return LCM([ProjectionToSpecZ(AffinePatch(X,i)) : i in [1..d]]);
  end if;
end function;
```

The set of primes fulfilling condition 3 could now simply be computed as the prime divisors of $n :=$ `ProjectionToSpecZ(SingularSubscheme(`P_8`))`. However, there can occur a serious problem: n can be so large that we cannot factor it. This is a very common situation, it already happens in the example in Section 7.2.1.

However, one can use a trick to avoid this factorization problem in practice. The trick is: Take a different point on the conic and the corresponding tangent plane to get a map F'. In Section 3.3 I showed that F and F' differ by a constant γ.

Let us assume for the moment that $\gamma = 1$. If one can compute the set of bad primes for F', then one can just use F' and forget F. Usually this is not the case and one cannot factor the n' from F', too. However, since the images of F and F' coincide, they must involve the same bad primes. Hence we can take the greatest common divisors of n and n', which happens to be a very small number in practice, which we can easily factor. In many examples it turns out that $\gcd(n, n')$ does not contribute new primes. I even guess, that the primes from condition 3 are superfluous. I will refer to this conjecture as the *Bad Primes Hypothesis*. It would be nice to have a proof for that. If $\gamma \neq 1$, we also have to take the prime divisors of its norm into account.

The following function combines all that

```
function MyBadPrimes(C4,L,Q3 : BadPrimesHypothesis := BadPrimesHypothesis)
  g := Modulus(C4`EtaleAlgebra);
  d := LCM([Denominator(c): c in Coefficients(g)]);
  g := ChangeRing(g*d,Integers());
  c := TheConstant(C4,L,Q3);
  // Here we could check, whether c is too big for factorization.
  // If so, we could start again with different tangent planes
  // until c is nice enough.
  vprint EightDescent,3: "Factoring c...";
  fc := Factorization(Numerator(c))*Factorization(Denominator(c));
  vprintf EightDescent,3: "Factorization of c: %o\n",fc;
  S1 := Set(PrimeDivisors(SquareFree(fc)))
  join Set(PrimeDivisors(Discriminant(g)));
  vprint EightDescent,3: "Bad primes from the coefficients of Li...";
  S2 := &join{PrimesDividingTheNormsOfAllCoefficients(Li) : Li in L};
  vprint EightDescent,3: S2;
  vprintf EightDescent,3:
  "Under BadPrimesHypothesis S = %o\n",{2} join S1 join S2;
  if BadPrimesHypothesis then
    S3 := {};
  else
    vprint EightDescent,3:
    "Projection of the singular subscheme of P_8 to Spec(Z):";
    P8 := ChangeRing(Scheme(C4,Q3),Integers());
    badP8 := ProjectionToSpecZ(SingularSubscheme(P8));
    vprint EightDescent,3: badP8;
    // if badP8 is to big (60 digits or more), we try to make it smaller:
    if Log(badP8)/Log(10) ge 60 then
      //Rem: we could try several different tangent planes.
      vprint EightDescent,3: "Trying to reduce the set of bad primes.";
      Qsing := SingularQuadricsInThePencil(C4)[1];
      L11 := DifferentTangentPlane(Qsing,L[1]);
      vprintf EightDescent,3: "The different tangent plane is L1' =
      \n%o.\n",L11;
      Lnew := L; Lnew[1] := L11;
      // We only change the first one.
      // Evtl. better results when changing both in the split case,
      // but then TheGamma has to be computed for both.
      Q31 := ThirdQuadric(C4, Lnew);
      vprintf EightDescent,3: "The quadric Q3' is \n%o.\n",Q31;
      bad1 := BadPrimesUnfactored(C4,Lnew,Q31);
      vprint EightDescent,3: "Computed bad primes for L1'.";
      gamma := TheGamma(C4,L[1],Lnew[1],Qsing);
      vprintf EightDescent,3: "gamma =\n%o.\n",gamma;
      Ngamma := Norm(gamma);
      badP8 := GCD(badP8, bad1*Numerator(Ngamma)*Denominator(Ngamma));
    end if;
    vprintf EightDescent,3: "Factoring %o...\n", badP8;
    S3 := Set(PrimeDivisors(badP8));
    S := {2} join S1 join S2 join S3;
    if S eq {2} join S1 join S2 then
      vprintf EightDescent,3: "We proved now:\
      \nBadPrimesHypothesis is true.\n";
    end if;
  end if;
  return {2} join S1 join S2 join S3;
end function;
```

where `DifferentTangentPlane` takes a different random point on the conic, to get a different tangent plane $L_1' = 0$, and `BadPrimesUnfactored` returns the unfactored version of the bad primes. For details see the programs in the appendix.

4.4 Unramifiedness Outside S

Let S be the set of bad primes as above. Let K be a local field. An element $\xi \in K^*$ is called *unramified* if $K[\sqrt{\xi}]|K$ is an unramified extension. For $v \nmid 2$ this is equivalent to $v(\xi) \equiv 0 \mod 2$ where v is the normalized valuation of K. We have to generalize this to our étale algebra. Let $A_v = \prod K_i$ be the decomposition of A_v into local fields K_i. If $\xi \in A_v$ we denote by $\xi_i \in K_i$ the image of ξ in K_i. An element $\xi \in A_v^*$ is called *unramified* if every ξ_i is unramified. An element of $A_v^*/A_v^{*2}\mathbb{Q}_v^*$ is called *unramified* if it has a representative $\xi \in A_v^*$ which is unramified.

An element ξ of $A^*/A^{*2}\mathbb{Q}^*$ is said to be *unramified outside* S if $\mathrm{res}_v(\xi)$ is unramified for each $v \notin S$. Let us denote the subset of A^*/A^{*2} of elements unramified outside S by $A(S,2)$. Then the subset of $A^*/A^{*2}\mathbb{Q}^*$ of elements unramified outside S is isomorphic to $A(S,2)/\mathbb{Q}(S,2)$ by [17, Proposition 12.8]. If one replaces $\mathbb{Q}$ by a number field, this set gets more complicated. $A(S,2)/\mathbb{Q}(S,2)$ is a finite computable set and we will show that the fake Selmer set is contained in it.

Theorem 4.4.1. *Let $p \notin S$. Then $F_p(C_4(\mathbb{Q}_p))$ is unramified.*

Proof. Let $p \notin S$. Let $A_p = \prod K_i$ the decomposition into local fields, and let F_p be given by $L_p = (\theta_i \mapsto L_i)$. Let $\tilde{K}$ be the splitting field of g over $\mathbb{Q}_p$.

Fix i_0. Let $\mathfrak{p} \subset K_{i_0}$ be the prime above p, and $\mathfrak{P} \subset \tilde{K}$ be the prime above $\mathfrak{p}$. Then $\mathfrak{P}|p$ is unramified since p does not divide $\mathrm{disc}(g)$. Let $P \in C_4(\mathbb{Q}_p)$, $P = (x_1 : \ldots : x_4)$ for $x_i \in \mathbb{Z}_p$ coprime. Let $v_{\mathfrak{P}}$ be the normalized valuation corresponding to $\mathfrak{P}$. By the norm condition we have $v_{\mathfrak{P}}(L_1(P) \cdot \ldots \cdot L_4(P)) \equiv v_{\mathfrak{P}}(c) \mod 2$, and since $\mathfrak{P}$ does not divide c, we get

$$v_{\mathfrak{P}}(L_1(P)) + \ldots + v_{\mathfrak{P}}(L_4(P)) \equiv 0 \mod 2.$$

Since the L_i are integral, we have $v_{\mathfrak{P}}(L_i(P)) \geq 0$ for all i. If $v_{\mathfrak{P}}(L_i(P)) > 0$, this means $L_i(P) = 0 \mod \mathfrak{P}$. Then for all $j \neq i$ we have $L_j(P) \neq 0 \mod \mathfrak{P}$ by the last condition on S. That means $v_{\mathfrak{P}}(L_j(P)) = 0$ for all $j \neq i$, thus

$$v_{\mathfrak{P}}(L_i(P)) = 0 \mod 2.$$

Since this holds for every $\mathfrak{P}$ above $\mathfrak{p}$, we get $v_{\mathfrak{p}}(L_{i_0}(P)) = 0 \mod 2$. □

By this theorem and the section about the norm condition we get the following computable description of the fake Selmer group.

Corollary 4.4.2.

$$\begin{aligned}\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q}) = \{\xi \in A^*/A^{*2}\mathbb{Q}^* \mid \xi \textit{ is unramified outside } S,\\ N(\xi) = c \mod \mathbb{Q}^{*2},\\ \mathrm{res}_v(\xi) \in F_v(C_4(\mathbb{Q}_v)) \textit{ for all } v \in S\}.\end{aligned}$$

Proof. For "⊂" the only thing that remains to show is that for a given $\xi \in \mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q})$ we have $N(\xi) = c \mod \mathbb{Q}^{*2}$. If not, then by the Chebotarev Density Theorem applied to the Kummer extension $\mathbb{Q}[\sqrt{N(\xi)/c}]|\mathbb{Q}$, we would find $N(\xi)/c \notin \mathbb{Q}_v^{*2}$ for some v, which would contradict Corollary 4.2.5.

The other inclusion follows from a counting argument using Theorem 3.6.5 and the results about the local image of the $x - T$-map. □

4.5 Implementation of the Fake Selmer Set

The subset of elements of $A^*/A^{*2}\mathbb{Q}^*$ unramified outside S, which is isomorphic to $A(S, 2)/\mathbb{Q}(S, 2)$ can be computed with the following function. In addition, this function computes the subset of $A(S, 2)/\mathbb{Q}(S, 2)$ fulfilling the norm condition. I call this set H, since it corresponds to the cohomology group $H^1(\mathbb{Q}, E[2]; S)$ as we will see later.

```
function TheSetH(C4,c,S);
  A := C4`EtaleAlgebra;
  // S must be a set of prime ideals in QNF.
  S_QNF := {p*MaximalOrder(BaseRing(A)) : p in S};
  vprintf EightDescent, 2: "Construction of A(S,2)\n";
  vtime EightDescent, 2:
```

```
  AS2,A_to_AS2,toVec,bU := pSelmerGroup(A,2,S_QNF:Raw);
  vprintf EightDescent, 2:
  "Computing the image of S in A(S,2), to get A(S,2)/Q(S,2).\n";
  vtime EightDescent,2: ImageOfS := {A_to_AS2(t) : t in {-1} join Set(S)};
  AS2Q, modQstar := quo<AS2 | ImageOfS>;
  A_to_AS2Q := A_to_AS2 * modQstar;
  vprintf EightDescent, 2: "Construction of the norm\n";
  bU := Eltseq(bU);
  t := Cputime();
  NbU := [Norm(b) : b in bU];
  QS2,Q_to_QS2 := pSelmerGroup(2,S_QNF);
  BruinNorm := hom<AS2 -> QS2 |
  [Q_to_QS2(PowerProduct(NbU,[c mod 2 : c in Eltseq(toVec(g))])) :
  g in OrderedGenerators(AS2)]>;
  MyNorm := hom<AS2Q -> QS2 | [BruinNorm(a@@modQstar) : a in
  OrderedGenerators(AS2Q)]>;
  // MyNorm: A(S,2)/Q(S,2) -> Q(S,2).
  vprintf EightDescent, 2: "%o\n", Cputime(t);
  vprintf EightDescent, 2:"Computing the preimage of c under the norm.\n";
  t := Cputime();
  bool, c1 := HasPreimage(Q_to_QS2(c), MyNorm);
  if bool then
    Ker := Kernel(MyNorm);
    vprintf EightDescent,2: "%o\n", Cputime(t);
    vprintf EightDescent,2: "Bound on #Sel after norm condition: 2^%o.\n",
    Round(Log(#Ker)/Log(2));
    // Representation of the coset c1+Ker:
    _, quomap := quo<AS2Q | Ker>;
    H := <quomap, {quomap(c1)} >;
  else
    vprint EightDescent, 2: "Sel was killed by the norm condition.";
    H := <{},{}>;
  end if;
  return H, A_to_AS2Q, toVec, bU;
end function;
```

Now the fake Selmer set is just the intersection of the local images, where we only have to consider the primes in S. Notice that I did not yet implement local solvability at infinity. How to compute the local image of F will be described in the next section.

```
function FakeSelmerSet(C4,L,g,Q3,S,H,A_to_AS2Q,toVec,bU)
  // Preliminaries:
  Zx := PolynomialRing(Integers(),4);
  F := TheMapF(C4,L);
  c := TheConstant(C4,L,Q3);
  Q1,Q2 := Explode(DefiningPolynomials(C4));
  EightPts := Scheme(Proj(Zx),[Q1,Q2,Q3]);
```

```
// Now we start:
Sel := H;
for p in S do
  vprintf EightDescent, 2: "\nLooking at prime p = %o.\
  \nComputing the local image.\n", p;
  vtime EightDescent, 2: localimage, res_p, Fpt:= LocalImage(C4,F,p,g);
  vprintf EightDescent, 2:
  "Setting up res_p:A(S,2)/Q(S,2)->Q(S,2) as a homomorphism.\n";
  t := Cputime();
  lbU := [res_p(b) : b in bU];
  AS2Q := Codomain(A_to_AS2Q);
  modQstar := Components(A_to_AS2Q)[2];//A_to_AS2Q = A_to_AS2*modQstar.
  // Now Res_p = res_p, but as a homomorphism A(S,2)/Q(S,2) -> Q(S,2).
  Res_p := hom<AS2Q -> Codomain(res_p)| [ &+[v[i]*lbU[i] : i in [1..#v]]
  where v := Eltseq(toVec(a @@ modQstar)) :
  a in OrderedGenerators(AS2Q)]>;
  vprintf EightDescent, 2: "%o\n", Cputime(t);
  vprintf EightDescent, 2:
  "Computing the preimage of the local image under res_p:\n";
  t := Cputime();
  modU := localimage[1];
  // modU is the quotient map with the information about the subgroup.
  xi_p := localimage[2]; //the coset representative.
  modURes_p := hom<AS2Q->Codomain(modU) | [modU(Res_p(a)) : a in
  OrderedGenerators(AS2Q)]>;
  bool, xi := HasPreimage(xi_p, modURes_p);
  if not bool then
    vprintf EightDescent, 1:
    "Local image does not have a preimage under res_p,\
    \ni.e. the preimage is not unramified outside S.\n";
    return {}, _;
  end if;
  V := Kernel(modURes_p);
  // We use the following representation of the coset xi+V:
  _, quomap := quo<AS2Q | V>;
  xiV := <quomap, {quomap(xi)}>;
  vprintf EightDescent, 2: "%o\n", Cputime(t);
  vprintf EightDescent, 2: "and intersecting with Sel.\n";
  vtime EightDescent, 2: Sel := CosetIntersection(Sel,xiV);
  vprintf EightDescent, 2: "Bound on #Sel now: 2^%o.\
  \nCoset representative: %o\n",
  Round(Log(#Kernel(Sel[1]))/Log(2)), Sel[2];
  if IsEmpty(Sel[2]) then
    return {}, _;
  end if;
end for;
// Sel is a coset, represented as Sel = <quomap,{zeta}>.
// We want to have it as an actual set.
quomap, setzeta := Explode(Sel);
```

```
  assert #setzeta eq 1; //Only one coset. (not empty: checked before)
  zeta := Rep(setzeta);
  z := zeta @@ quomap;
  U := Kernel(quomap);
  Sel := {z + u : u in U};
  vprintf EightDescent, 0: "Attention: I did not check p = infinity.\n\n";
  return Sel, A_to_AS2Q;
end function;
```

4.6 Computing the Local Image of F

Now we will see how we can find the local image of F, i.e. the image of the function $F_v : C_4(\mathbb{Q}_v) \to A_v^*/A_v^{*2}\mathbb{Q}_v^*$. First, we compute the image of one local point.

4.6.1 One Local Point

The curve C_4 is everywhere locally solvable, since it is an element of the 4-Selmer group. Thus we know that there exist points in $C_4(\mathbb{Q}_v)$ for every place v. Also explicitly finding such a point is not a problem in theory. However, a theoretical problem is, that one can compute such a point only to a finite precision. Thus we have to know up to which precision we have to compute it. In addition, we want to compute this point efficiently.

So what do we do? Let $v = p$ be a finite prime. The curve C_4 is given by two quadrics Q_1 and Q_2, which we can assume to have integer coefficients, i.e. $Q_1, Q_2 \in \mathbb{Z}[x_1, \dots, x_4]$. We look for a $\mathbb{Z}_p$-integral point in the affine patches. We suppose that there is one in $\{x_4 \neq 0\}$. Thus we dehomogenize Q_1 and Q_2 to get $Q_1^{(0)}(X,Y,Z) := Q_1(X,Y,Z,1)$ and $Q_2^{(0)}(X,Y,Z) := Q_2(X,Y,Z,1)$ in $\mathbb{Z}[X,Y,Z]$ and set $C_{4,\text{affine}} : Q_1^{(0)} = Q_2^{(0)} = 0$.

Now we reduce $C_{4,\text{affine}}$ mod p and take an $\mathbb{F}_p$-point $P_0 = (\bar{a}_0, \bar{b}_0, \bar{c}_0)$, with integers $0 \leq a_0, b_0, c_0 < p$, on it. If P_0 is smooth, it lifts to a p-adic point. Next we set $Q_i^{(1)} := \frac{1}{p} Q_i^{(0)}(a_0 + pX, b_0 + pY, c_0 + pZ)$, $i = 1, 2$. Then $Q_1^{(1)}$ and $Q_2^{(1)}$ are again in $\mathbb{Z}[X,Y,Z]$, since mod p we have $Q_i^{(0)}(a_0 + pX, b_0 + pY, c_0 + pZ) \equiv Q_i^{(0)}(P_0) \equiv 0$. Now we reduce the affine curve $Q_1^{(1)} = Q_2^{(1)} = 0$ mod p and take a point $P_1 = (\bar{a}_1, \bar{b}_1, \bar{c}_1)$, $0 \leq a_0, b_0, c_0 < p$, on it. If there is none, we have to start again with a different P_0. Next we set $Q_i^{(2)} := \frac{1}{p} Q_i^{(1)}(a_1 + pX, b_1 + pY, c_1 + pZ)$, $i = 1, 2$. Then $Q_1^{(2)}$ and $Q_2^{(2)}$ are again in $\mathbb{Z}[x_1, x_2, x_3]$, since mod p, P_1 is a zero of $Q_i^{(1)}$. Now we take a point on the affine curve $Q_1^{(2)} = Q_2^{(2)} = 0$ mod p, and so on. Then with $a := a_0 + pa_1 + p^2 a_2 + \dots$, $b := b_0 + pb_1 + p^2 b_2 + \dots$, and $c := c_0 + pc_1 + p^2 c_2 \dots$

we get the p-adic point $P := (a : b : c : 1)$ on C_4.

The following lemma tells us how long we have to proceed.

Lemma 4.6.1. *Let $p \neq 2$. Suppose we computed a p-adic point $P \in C_4(\mathbb{Q}_p)$ up to precision N, and that $v_p(N(F(P))) < N$, then $F(P)$ is independent of the precision higher than N.*

In other words: Every point that reduces to $P \mod p^N$ has the same image under F.

Proof. Locally $A_p \cong \prod_{i=1}^{t_p} K_i$ where the K_i are local fields corresponding to the irreducible factors of g over $\mathbb{Q}_p$. Write $F_p(P) = (F_{p,i}(P))_i$ with $F_{p,i}(P) \in K_i$. Recall, that F is given by linear forms with integral coefficients.

Hence

$$F_{p,i}(P) = F_{p,i}(a_0, b_0, c_0, 1) + pF_{p,i}(a_1, b_1, c_1, 0) + p^2 F_{p,i}(a_2, b_2, c_2, 0) + \dots$$

in K_i. Let e be the ramification index of K_i over $\mathbb{Q}_p$. Let π be a uniformizing element of K_i, i.e. $(\pi) = \mathfrak{p}$ for the prime ideal above p, hence $u\pi^e = p$ for a unit u. Hence

$$\begin{aligned} &F_{p,i}(a_0, b_0, c_0, 1) + pF_{p,i}(a_1, b_1, c_1, 0) + p^2 F_{p,i}(a_2, b_2, c_2, 0) + \dots \\ &\quad = F_{p,i}(a_0, b_0, c_0, 1) + u\pi^e F_{p,i}(a_1, b_1, c_1, 0) + u\pi^{2e} F_{p,i}(a_2, b_2, c_2, 0) + \dots \\ &\quad = \pi^k x_k + \pi^{k+1} x_{k+1} + \pi^{k+2} x_{k+2} + \dots \end{aligned}$$

for some $x_{k+j} \in K_i$ with $v_{\mathfrak{p}}(x_{k+j}) \geq 0$ and $v_{\mathfrak{p}}(x_k) = 0$.

Since $v_p(N(F(P))) < N$, we have $k < N$. Hence the terms a_N, b_N, c_N and higher cannot affect x_k. Since modulo squares $\pi^k x_k + \pi^{k+1} x_{k+1} + \dots$ is uniquely determined by k and x_k, we are done. □

Remark 4.6.2. *We could get a better estimate for the precision we need, when we take the ramification indices and inertia degrees of the involved number fields into account. However, lifting a point to higher precision costs almost nothing. So for practical purposes this estimate is enough.*

For $p = 2$ we have to compute not only the first non-zero digit x_k in the π-adic expansion, but at least the first three digits, more precisely the first $2e_i + 1$, where e_i is the ramification index of $K_i | \mathbb{Q}_2$.

Let $e := \max e_i$, for the fields in the decomposition $A_2 \cong \prod_{i=1}^{t_2} K_i$.

Lemma 4.6.3. *If we know the point $P \in C_4(\mathbb{Q}_2)$ up to precision N and $v_2(N(F_2(P))) < N - 2e - 1$, where e is the maximum of the ramification indices as above, then $F_2(P)$ is determined.*

Proof. To know $F_2(P) = \pi^k x_k + \pi^{k+1}x_{k+1} + \pi^{k+2}x_{k+2} + \ldots$ in K_i^* modulo squares, it is enough to know k and $x_k, \ldots, x_{k+2e_i+1}$.

Since $v_2(N(F(P))) < N - 2e - 1$, we have $k < N - 2e - 1$. Hence for $n \geq N$, $F(a_n, b_n, c_n, 0)\pi^{ne_i}$ has valuation at least $ne_i \geq N > k + 2e + 1$, hence the terms a_n, b_n, c_n cannot affect $x_k, \ldots, x_{k+2e+1}$. □

Corollary 4.6.4. *If we know the point $P \in C_4(\mathbb{Q}_2)$ up to precision N and $v_2(N(F_2(P))) \leq N - 10$, then $F_2(P)$ is determined.*

Proof. The degree of g is four, hence $e \leq 4$, and the corollary follows. □

With this theoretical result, we are able to compute the local image of a point. We just compute a local point, look whether the precision is enough, and if not, lift it to higher precision and so on.

Let me conclude with a remark on the practical computation of a local point. Computing a local point on a variety can be done in the way I described above for the curve C_4 by reduction mod p and deducing equations for the next level. Nils Bruin has implemented this method for very general varieties. He also has some improvements of this method. This is the Magma-intrinsic `IsLocallySolvable`. However, he needs to compute the whole set of points mod p, which can take a long time for large p.

Since we know that C_4 is locally solvable, and we just want to get one point, we can compute a random point mod p instead of all points. How to do this efficiently is the content of the next section.

4.6.2 Random $\mathbb{F}_p$-points on the Intersection of Two Quadrics

For finding a random $\mathbb{F}_p$-point on $C_{4,\text{affine}}$ we set the first coordinate to a random $\mathbb{F}_p$-value a. This is equivalent to intersecting $C_{4,\text{affine}}$ with the plane $\{X = a\}$. Usually this is a zero-dimensional scheme, on which we can easily find a point by a Groebner basis computation. If there is no point, we take a different random value a.

This procedure is very fast, and the following heuristics shows why it is so fast. If the reduction of C_4 mod p is smooth, it is an elliptic curve over $\mathbb{F}_p$, thus has about p many points by Hasse-Weil. If it has one singularity, it has genus 0, thus is parameterized by $\mathbb{P}^1_{\mathbb{F}_p}$, hence has about p many points, too. It can also happen, that it has two singular points, but then we use Nils Bruin's function `IsLocallySolvable`, which is very fast in this case, since it first looks into the singular locus. So we just work with the first two cases, where we have about p many points on C_4.

On the other hand, we have p many planes of the form $\{X = a\}$, $a \in \mathbb{F}_p$. Since C_4 has degree 4, a plane meets C_4 in at most 4 points, thus at least every fourth plane must meet C_4 in an $\mathbb{F}_p$-point. In the examples it turned out to be about every second plane. Thus usually we have to try two values for a and compute the points of the zero-dimensional scheme $C_{4,\text{affine}} \cap \{X = a\}$, to get an $\mathbb{F}_p$-point on C_4. This is much faster than computing all points of $C_4(\mathbb{F}_p)$.

4.6.3 The Whole Image

By Theorem 3.6.5, we know that the image of F_v and the image of the local $x - T$-map coincide up to a translation, i.e.

$$\operatorname{im}(F_v) = \alpha \cdot \operatorname{im}(x - T).$$

for some $\alpha \in A_v^*/A_v^{*2}\mathbb{Q}_v^*$. Thus if we know $\operatorname{im}(x - T)$ in $A_v^*/A_v^{*2}\mathbb{Q}_v^*$, and one element $\xi \in \operatorname{im}(F_v)$, then we get the whole $\operatorname{im}(F_v)$ in the following way.

By the local version of Corollary 3.4.2, $\operatorname{im}(F_v)$ is a coset in $A_v^*/A_v^{*2}\mathbb{Q}_v^*$, and so is $\operatorname{im}(x - T)$, hence $\operatorname{im}(x - T) = \eta \cdot U$ for some η and a subgroup U of $A_v^*/A_v^{*2}\mathbb{Q}_v^*$. Then $\operatorname{im}(F_v) = \xi \cdot U$.

4.6.4 Implementation

With the following functions we can compute the local image of F. The first function computes a random point on an affine curve as described above.

```
function RandomPoint(C)
  // C is an affine curve over Fp in A3.
  vprintf EightDescent,3: "Computing a random F_p-point";
  assert Dimension (C) eq 1;
  Fp := BaseField(C);
  pts := {};
  while IsEmpty(pts) do
    vprintf EightDescent,3: ".";
    // The number of dots is the number of trials, which is
    // expected to be less than 4 on average.
    x := Random(Fp);
    Z := Scheme(C, C.1 - x);
    //some hyperplane section (C.1 must have some value x).
    pts := Points(Z); // Z is 0-dim.
  end while;
  vprintf EightDescent,3: "\n";
  return Random(pts);
end function;
```

Using an $\mathbb{F}_p$-point, we compute the equations for the next level as described in the beginning of Section 4.6.1.

```
function NextLevel(C,pt_p)
  // C an affine scheme over the integers.
  // pt_p a point mod p.
  Fp := Parent(pt_p[1]);
  p := Characteristic(Fp);
  pt := ChangeUniverse(Eltseq(pt_p),Integers());
  newpols := [Evaluate(f,[pt[i] + p*C.i : i in [1..3]]) div p : f in
  DefiningPolynomials(C)];
  return Scheme(Ambient(C),newpols);
end function;
```

With the following function we compute the image of one local point under F. For the cases where the singularities are bad or for small primes, i.e. $p \leq 5$, we hand over to `ImageOfOnePointAtVeryBadOrSmallPrime` which uses Nils Bruin's function `IsLocallySolvable`.

```
function ImageOfOneLocalPoint(C4,F,p)
  if p le 5  or  Dimension(SingularSubscheme(ChangeRing(C4,GF(p)))) ge 1
  or  #SingularPoints(ChangeRing(C4,GF(p))) ge 2 then
    return ImageOfOnePointAtVeryBadOrSmallPrime(C4,F,p);
  else
    Fp := GF(p);
    C_i := AffinePatch(ChangeRing(C4,Integers()),1);
    // Might be a bad affine patch (e.g. at p=2).
    error if Dimension(C_i) lt 1,
    "ERROR in MyImageOfOneLocalPoint: We took the wrong affine patch.";
    pt := [0,0,0];
    for i in [0..1000] do
      C_i_Fp := BaseChange(C_i,Fp);
      if Dimension(SingularSubscheme(C_i_Fp)) ge 1  or
      #SingularPoints(C_i_Fp) ge 2 then // hand over to IsLocallySolvable.
        return ImageOfOnePointAtVeryBadOrSmallPrime(C4,F,p);
      end if;
      pt_i := RandomPoint(C_i_Fp);
      // If pt_i is smooth, it lifts. Thus we take only smooth points.
      while IsSingular(pt_i) do pt_i := RandomPoint(C_i_Fp); end while;
      C_i := NextLevel(C_i,pt_i);
      pt := [pt[j] + p^i*Integers()!pt_i[j] : j in [1..3]] cat [1];
      v := Valuation(Integers()!Norm(F(pt)),p);
      if v lt i then
        //Then precision i is enough to determine F(pt) mod A^*2.
        return F(pt), pt, i;
      end if;
    end for;
    error
    "ERROR: Precision 1000 was not enough to compute the local image.";
  end if;
end function;
```

The whole local image of F can now be computed using the local image of the $x-T$-map. This is already implemented in Magma as part of Tom Womack's FourDescent-routine with some improvements by Mark Watkins. We only have to modify the function LocalPoints there to get the local image $(x-T)(C_2(\mathbb{Q}_p))$, which I call LocalImageOfC2.

```
function LocalImage(C4,F,p,g)
  A := Codomain(F);
  g := ChangeRing(g,Rationals()); // needed for local image of C2.
  Fpt := ImageOfOneLocalPoint(C4,F,p);
  kgens, res_p := LocalImageOfC2(g,p : Algebra:=A);
  xi_p := res_p(Fpt);
  //The local image of F is the coset xi*U, where U is generated by kgens.
  _, quomap := quo<Codomain(res_p)|kgens>;
  localimage := <quomap, quomap(xi_p)>;
  return localimage, res_p, Fpt;
end function;
```

Chapter 5

Representation as 2-Coverings

So far we computed the fake Selmer set as a coset of $A^*/A^{*2}\mathbb{Q}^*$. What we really want to have are 2-coverings of C_4. In this chapter I will show how one can construct these geometrical objects out of the algebraic ones. I did the main work for this in Paris at the Institut Henri Poincaré during the trimester on "Explicit Methods in Number Theory" in fall 2004.

5.1 Abstract Geometrical Construction

Let $K := \bar{\mathbb{Q}}(C_4)$ be the function field of C_4 over $\bar{\mathbb{Q}}$. Let $L_1, \ldots, L_4$ be the tangent planes as constructed in the 8-descent. Then $t_i := L_i/L_4$, $i = 1, \ldots, 4$ are functions on C_4. On $E = \mathrm{Pic}^0(C_4)$ we have $[\frac{1}{2}\mathrm{div}(t_i)] = T_i$, with $\{T_1, T_2, T_3\} = E[2] \setminus O$ and $T_4 = O$ by Corollary 3.6.4. If we adjoin the square roots of the t_i, we get a field extension of degree 4:

Proposition 5.1.1. *The extension $K[\sqrt{t_1}, \sqrt{t_2}, \sqrt{t_3}]|K$ is Galois with Galois group $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$.*

Proof. The proposition follows immediately from the two facts $\sqrt{t_i} \notin K$ for $i \neq 1, 2, 3$, and $\sqrt{t_1 t_2 t_3} \in K$. □

Later we will construct an unramified covering of C_4 with this function field. Then the following proposition, which is well known to the experts, compare e.g. [1, 6], shows that this will be a 2-covering.

Proposition 5.1.2. *Let C_8 be a curve defined over $\mathbb{Q}$ and $\phi_8 : C_8 \to C_4$ be an unramified morphism defined over $\mathbb{Q}$ such that the extensions of function fields $\bar{\mathbb{Q}}(C_8)|\phi_8^*(\bar{\mathbb{Q}}(C_4))$ is Galois with Galois group $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$, then $\phi_8 : C_8 \to C_4$ is a 2-covering.*

Proof. Since ϕ_8 is unramified C_8 has genus 1 by Riemann-Hurwitz. If we consider C_4 as an elliptic curve with the point $(\theta_1, 0)$ as zero, and C_8 as an elliptic curve with a preimage of $(\theta_1, 0)$ under ϕ_8 as zero, then ϕ_8 is an isogeny. The kernel of ϕ_8 is $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$, since this is the Galois group of the function field extension.

By the uniqueness of an isogeny with predescribed kernel [21, III, Prop. 4.12], ϕ_8 must be multiplication by 2 up to a $\bar{\mathbb{Q}}$-isomorphism. □

Next we will show how one can construct a model for C_8 explicitly.

5.2 Explicit Construction of ϕ_8

In this section we will show how to construct a model for C_8 in $\mathbb{P}^3$ and the equations for the 2-covering $\phi_8 : C_8 \to C_4$.

5.2.1 The First Two Quartics

We will first look at the generic case, when A is a number field. We start with an element $\xi \in \mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q})$ and we want to represent it as a 2-covering of C_4. In fact, we will see that we can construct two 2-coverings from ξ.

Recall that we were working with the map $F : C_4(\mathbb{Q}) \to A^*/A^{*2}\mathbb{Q}^*$. What do we know about F? Usually we do not know the set $C_4(\mathbb{Q})$, else we would not have to do a descent on C_4. This means, that we have a map F whose domain we do not know. However, we know that F is given by a linear form L_1, and we know a possible image ξ, but we know ξ only up to a square and a $\mathbb{Q}^*$-scalar. Thus we are searching a point $P \in C_4(\mathbb{Q})$ and an element $y \in A$ such that

$$F(P) = \xi y^2. \tag{5.1}$$

The $\mathbb{Q}^*$-scalar is absorbed by the left hand side of (5.1), since F is given by a homogeneous polynomial.

Equation (5.1) contains the main information for constructing the 2-covering. Write $P = (x_1 : \ldots : x_4)$, $\xi = \xi_1 + \theta\xi_2 + \theta^2\xi_3 + \theta^3\xi_4$, $\xi_1, \ldots, \xi_4 \in \mathbb{Q}$, and $y = y_1 + \theta y_2 + \theta^2 y_3 + \theta^3 y_4$. Since we do not know $x_1, \ldots, x_4$ and $y_1, \ldots, y_4$, we interpret them as variables. Then equation (5.1) reads

$$L_1(x_1, \ldots, x_4) = (\xi_1 + \theta\xi_2 + \theta^2\xi_3 + \theta^3\xi_4)(y_1 + \theta y_2 + \theta^2 y_3 + \theta^3 y_4)^2. \tag{5.2}$$

If we sort the left hand side of (5.2) by powers of θ, we get

$$L_1(x_1,\ldots,x_4) = l_1(x_1,\ldots,x_4) + \theta l_2(x_1,\ldots,x_4) + \theta^2 l_3(x_1,\ldots,x_4) + \theta^3 l_4(x_1,\ldots,x_4)$$

for some linear forms $l_i \in \mathbb{Q}[x_1, \ldots, x_4]$. If we multiply out the right hand side of (5.2) and sort by powers of θ, we get

$$\begin{aligned}&(\xi_1 + \theta\xi_2 + \theta^2\xi_3 + \theta^3\xi_4)(y_1 + \theta y_2 + \theta^2 y_3 + \theta^3 y_4)^2 = \\ &\qquad q_1(y_1, \ldots, y_4) + \theta q_2(y_1, \ldots, y_4) + \theta^2 q_3(y_1, \ldots, y_4) + \theta^3 q_4(y_1, \ldots, y_4)\end{aligned}$$

for some quadratic forms $q_i \in \mathbb{Q}[x_1, \ldots, x_4]$ depending on ξ. Thus equation (5.2) reads

$$\begin{aligned}&l_1(x_1, \ldots, x_4) + \theta l_2(x_1, \ldots, x_4) + \theta^2 l_3(x_1, \ldots, x_4) + \theta^3 l_4(x_1, \ldots, x_4) = \\ &\qquad q_1(y_1, \ldots, y_4) + \theta q_2(y_1, \ldots, y_4) + \theta^2 q_3(y_1, \ldots, y_4) + \theta^3 q_4(y_1, \ldots, y_4).\end{aligned}$$

Restriction of scalars gives the system of equations

$$\begin{aligned}l_1(x_1, \ldots, x_4) &= q_1(y_1, \ldots, y_4),\\ l_2(x_1, \ldots, x_4) &= q_2(y_1, \ldots, y_4),\\ l_3(x_1, \ldots, x_4) &= q_3(y_1, \ldots, y_4),\\ l_4(x_1, \ldots, x_4) &= q_4(y_1, \ldots, y_4),\end{aligned} \tag{5.3}$$

over $\mathbb{Q}$, which we can write as

$$M \begin{pmatrix} x_1 \\ \vdots \\ x_4 \end{pmatrix} = \begin{pmatrix} q_1(y_1, \ldots, y_4) \\ \vdots \\ q_4(y_1, \ldots, y_4) \end{pmatrix}$$

with the matrix M of coefficients of the l_i. Generically M is invertible, see also 5.2.5. In this case we can multiply with its inverse, and get

$$\begin{pmatrix} x_1 \\ \vdots \\ x_4 \end{pmatrix} = M^{-1} \begin{pmatrix} q_1(y_1, \ldots, y_4) \\ \vdots \\ q_4(y_1, \ldots, y_4) \end{pmatrix} =: \begin{pmatrix} r_1(y_1, \ldots, y_4) \\ \vdots \\ r_4(y_1, \ldots, y_4) \end{pmatrix} \tag{5.4}$$

for some quadratic forms $r_i \in \mathbb{Q}[x_1, \ldots, x_4]$.

In addition, $P = (x_1 : \ldots : x_4)$ stands for a point on C_4, i.e. $(x_1, \ldots, x_4)$ must be a zero of Q_1 and Q_2, thus substituting $x_i = r_i(y_1, \ldots, y_4)$, $i = 1, \ldots, 4$, gives two quartics

$$G_1(y_1, \ldots, y_4) := Q_1(r_1(y_1, \ldots, y_4), \ldots, r_4(y_1, \ldots, y_4))$$

and

$$G_2(y_1, \ldots, y_4) := Q_2(r_1(y_1, \ldots, y_4), \ldots, r_4(y_1, \ldots, y_4))$$

Let $C_8 : G_1 = G_2 = 0$ in $\mathbb{P}^3$, then equation (5.4), defines a rational map

$$\phi_8 : C_8 \to C_4.$$

by $(y_1, \ldots, y_4) \mapsto (x_1 : \ldots : x_4) = (r_1(y_1, \ldots, y_4), \ldots, r_4(y_1, \ldots, y_4))$.

However, C_8 consists of two components, and is not yet the 2-covering we want to have. Michael Stoll pointed out to me that I did not use the norm condition so far, which should give me a third quartic. In fact, this third quartic separates the two components of C_8, and each of these two components is a 2-covering as we will see below.

Before I describe how to find the third quartic, I want to show how one finds $\phi_8 : C_8 \to C_4$ in the split case $A \cong K_1 \times K_2$. Here we have $F(P) = (L_1(P), L_2(P))$ and $\xi = (\xi_1 + \theta_1\xi_2, \xi_3 + \theta_2\xi_4) \in K_1 \times K_2$, $\xi_1, \ldots, \xi_2 \in \mathbb{Q}$, and a generic element of $K_1 \times K_2$ can be written as $y = (y_1 + \theta_1 y_2, y_3 + \theta_2 y_4)$. Thus equation (5.1) reads

$$\begin{aligned} l_1(x_1, \ldots, x_4) + \theta_1 l_2(x_1, \ldots, x_4) &= L_1(x_1, \ldots, x_4) = (\xi_1 + \theta_1\xi_2)(y_1 + \theta_1 y_2)^2 \\ l_3(x_1, \ldots, x_4) + \theta_2 l_4(x_1, \ldots, x_4) &= L_2(x_1, \ldots, x_4) = (\xi_3 + \theta_2\xi_4)(y_3 + \theta_2 y_4)^2. \end{aligned}$$

Multiplying out the right hand side and sorting by powers of θ_1 and θ_2 gives the a system of the form (5.3) and we can continue as above.

5.2.2 The Third Quartic

In this section we show how to find the third quartic, which separates the two components of C_8. We get the third quartic from the norm condition. Let us look at the generic case where A is a number field, the split case is analogous. By Theorem 4.2.4 we have $N(L_1(x_1, \ldots, x_4)) = c \cdot Q_3(x_1, \ldots, x_4)^2$, for some $Q_3 \in \mathbb{Q}[x_1, \ldots, x_4]$ and some $c \in \mathbb{Q}^*$. In addition, we have $N(\xi) = ca^2$ for some $a \in \mathbb{Q}^*$, since $\xi \in \mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q})$.

Now, taking norms on each side of $\xi y^2 = F(P) = L_1(x_1, \ldots, x_4)$, we get

$$N(\xi)N(y)^2 = c \cdot Q_3(x_1, \ldots, x_4)^2.$$

Here $y = y_1 + \theta y_2 + \theta^2 y_3 + \theta^3 y_4$, and $N(y)$ is a polynomial in $y_1, \ldots, y_4$ over $\mathbb{Q}$. Using $N(\xi) = ca^2$ and substituting $x_i = r_i(y_1, \ldots, y_4)$, $i = 1, \ldots, 4$, gives

$$ca^2 N(y)^2 = c \cdot Q_3(r_1(y_1, \ldots, y_4), \ldots, r_4(y_1, \ldots, y_4))^2$$

Cancelling c and taking square roots, gives the two equations

$$aN(y) = \pm Q_3(r_1(y_1, \ldots, y_4), \ldots, r_4(y_1, \ldots, y_4)),$$

The polynomials $aN(y) \mp Q_3(r_1(y_1,\dots,y_4),\dots,r_4(y_1,\dots,y_4))$ are quartic forms over $\mathbb{Q}$ in the variables $y_1,\dots,y_4$, say G_3^+ and G_3^-.

Let $C_8^+ : G_1 = G_2 = G_3^+ = 0$ and $C_8^- : G_1 = G_2 = G_3^- = 0$, and $\phi_8^+ : C_8^+ \to C_4$ and $\phi_8^- : C_8^- \to C_4$ be the restrictions of ϕ_8. We will see in Section 5.2.3 that ϕ_8^+ and ϕ_8^- are in fact 2-coverings of C_4. Thus we constructed two geometrical objects out of one algebraic object $\xi \in \mathrm{Sel}^{(2)}_{\text{fake}}(C_4/\mathbb{Q})$.

Compare this with the situation of 4-descent, where one gets two 2-coverings of C_2 from one element $\xi \in \mathrm{Sel}^{(2)}_{\text{fake}}(C_2/\mathbb{Q})$. However, by a 4-descent we get *one* curve C_4 and two different morphisms to C_2, $\phi_4^+ : C_4 \to C_2$, $(x_1,\dots,x_4) \mapsto (x,y)$, and $\phi_4^- : C_4 \to C_2$, $(x_1,\dots,x_4) \mapsto (x,-y)$. In my construction of the 8-descendents, we get two different curves C_8^+ and C_8^-, which are the two components of C_8, but the two morphisms $\phi_8^+ : C_8^+ \to C_4$ and $\phi_8^- : C_8^- \to C_4$ are given by the same equations $x_i = r_i(y_1,\dots,y_4)$, $i = 1,\dots,4$, since they are just the restrictions of ϕ_8.

Remark 5.2.1. *The schemes C_8^+ and C_8^- are not saturated. If one takes their saturations, one gets four more quintics. That was pointed out to me by v. Bothmer.*

5.2.3 The Function Field of $C_8^\pm$

The construction above produces in fact 2-coverings of C_4 by the following

Theorem 5.2.2. *The morphisms $\phi_8^+ : C_8^+ \to C_4$ and $\phi_8^- : C_8^- \to C_4$ constructed above are 2-coverings of C_4.*

Proof. The function field of C_8^+ over $\bar{\mathbb{Q}}$ can be obtained by looking at the affine patch $\{L_4 \neq 0\}$, i.e. we consider $L_4 = 0$ as the plane at infinity. Over $\bar{\mathbb{Q}}$ the four planes given by $L_1,\dots,L_4$ are rational, thus the equation $F(P) = \xi y^2$, which we used to construct C_8^+, reads

$$L_1 = y_1^2, \qquad L_2 = y_2^2, \qquad L_3 = y_3^2, \qquad L_4 = y_4^2. \tag{5.5}$$

Here we can neglect ξ, since we can put its square root to the y_i's, since we are working over $\bar{\mathbb{Q}}$. Now we consider $L_4 = 0$ as the plane at infinity, and in the affine patch $\{L_4 \neq 0\}$, equations (5.5) mean

$$\frac{L_1}{L_4} = \left(\frac{y_1}{y_4}\right)^2, \qquad \frac{L_2}{L_4} = \left(\frac{y_2}{y_4}\right)^2, \qquad \frac{L_3}{L_4} = \left(\frac{y_3}{y_4}\right)^2.$$

Writing $t_i := L_i/L_4$, $i = 1,\dots,4$, we see that the function field of C_8^+ is $K[\sqrt{t_1},\sqrt{t_2},\sqrt{t_3}]$, where $K = \bar{\mathbb{Q}}(C_4)$ is the function field of C_4, and ϕ_8^+ corresponds to the inclusion $K \subset K[\sqrt{t_1},\sqrt{t_2},\sqrt{t_3}]$. The same holds for ϕ_8^-.

In addition, ϕ_8^+ and ϕ_8^- are unramified, since the zeros and poles of t_i, $i = 1, \ldots, 3$, have multiplicity 2, thus their square root exists in a neighborhood of a zero or pole.

Thus ϕ_8^+ and ϕ_8^- are 2-coverings of C_4 by Proposition 5.1.1 and Proposition 5.1.2. □

5.2.4 Local Solvability of $C_8^+ \cup C_8^-$

Now we want to see how much we can say about the local solvability of our 2-coverings $\phi_8^\pm : C_8^\pm \to C_4$. At this stage we can only note that the union $C_8 = C_8^+ \cup C_8^-$ has points everywhere locally. That each component is everywhere locally solvable will follow from the Galois cohomological interpretation.

Let $\xi \in \mathrm{Sel}_{\mathrm{fake}}(C_4/\mathbb{Q})$. Thus $\mathrm{res}_v(\xi) \equiv F_v(P_v) \mod A_v^{*2}\mathbb{Q}_v^*$ for a point $P_v \in C_4(\mathbb{Q}_v)$. Hence there exists an element $y = y_1 + \theta y_2 + \theta^2 y_3 + \theta^3 y_4 \in A_v^*$ such that $L_v(P_v) = \mathrm{res}_v(\xi) y^2$ for a suitable scaling of P_v. Then $(y_1 : \ldots : y_4)$ is a v-adic point on C_8 by the construction of C_8. However, we cannot decide on which component of C_8 it lies. So we just know that one of the components has a v-adic point. To conclude that the other component also has a v-adic point, we will use Corollary 6.2.8.

5.2.5 Remark on the Invertibility of M

We just consider the case where A is number field, the split case is analogous. Let $L_1 = l_1 + \theta_1 l_2 + \theta_1^2 l_3 + \theta_1^3 l_4$, and M be the matrix of the coefficients of $l_1, \ldots, l_4$, thus

$$M \begin{pmatrix} x_1 \\ \vdots \\ x_4 \end{pmatrix} = \begin{pmatrix} l_1 \\ \vdots \\ l_4 \end{pmatrix}.$$

Let $L_1, \ldots, L_4$ be the conjugates of L_1.

Theorem 5.2.3. *The following are equivalent:*

(a) M is invertible.

(b) The four planes in $\mathbb{P}^3$ given by $L_1, \ldots, L_4$ do not intersect.

(c) $L_1, \ldots, L_4$ are linearly independent in the $\bar{\mathbb{Q}}$-vector space $\bar{\mathbb{Q}}[x_1, \ldots, x_4]$.

Proof. The intersection of the L_i is a subspace of $\mathbb{P}^3(\bar{\mathbb{Q}})$ which is defined over $\mathbb{Q}$, and is precisely the kernel of M. Then the equivalence of (a), (b), and (c) is obvious. □

Generically, three planes in $\mathbb{P}^3$ intersect in one point, and a fourth plane does not meet this point. Thus if $L_1, \dots, L_4$ behave like randomly chosen planes, M is invertible. However, $L_1, \dots, L_4$ are conjugate, so they do not look so independent.

In fact, if L_1 meets C_4 in a $\mathbb{Q}$-rational point, then L_2, L_3, and L_4 have to go through this point, too. And this situation really occurs in practice, but it is not a problem, since then we found a point in $C_4(\mathbb{Q})$ and we are done.

However, this is not the only case when M is non-invertible. It is true, that if $L_1, \dots, L_4$ meet in one point P, that P must be $\mathbb{Q}$-rational, however, P need not lie on C_4. It is not clear to me whether one can use this situation to find a rational point on C_4. What one can do in practice in this case, is to choose a different point on the conic to get different tangent planes, which might not meet in a point.

Alternatively, one can try to get equations for the 2-covering, even if M is not invertible. Working this out might also lead to an idea how to use the non-invertibility of M for finding a rational point on C_4.

5.2.6 Twenty Quadrics in $\mathbb{P}^7$

It is known that an 8-covering of an elliptic curve can be represented by 20 quadrics in $\mathbb{P}^7$. We can get such a model from our $C_8^{\pm}$, which is given by three quartics in $\mathbb{P}^3$, in the following way.

Let $D := (H)_{C_8}$ be the divisor cut out by a plane, for example the plane given by $y_1 = 0$. Then D has degree 8, hence the complete linear system $|D|$ induces a rational map $\phi_{|D|} : C_8^{\pm} \to \mathbb{P}^7$. The image of $\phi_{|D|}$ is then given by 20 quadrics by Riemann-Roch.

The code for doing this in Magma is

```
function TwentyQuadrics(C8)
  // Another good name would be QuadricIntersection(C8).
  // C8 is the eightdescendent given by three quartics
  D := Divisor(C8, Scheme(C8,C8.1));
  // D is the divisor cut out by the plane y_1 = 0.
  phi_D := DivisorMap(D);
  C20<[y]> := Image(phi_D);
  return C20, phi_D;
end function;
```

5.3 Implementation of 8-Descent

With the following functions we can compute the representations of an element of the fake Selmer set as 2-coverings.

```
function TheMatrix(L)
  // The matrix M of the coefficients.
  Ms := [Matrix([Eltseq(MonomialCoefficient(Li,Parent(Li).i))
  : i in [1..4]]) : Li in L];
  return HorizontalJoin(Ms);
end function;
```

With the function CorrespondingSubstitution we can get the quadratic forms from equation (5.4), which define ϕ_8.

```
function CorrespondingSubstitution(L,xi)
  // L = <L[1],L[2]> or <L[1]>.
  // xi = <xi[1],xi[2]> or <xi[1]> in K1 (x K2).
  // we compute "(xi*y^2) * M^-1" in the following sense:
  M := TheMatrix(L);
  error if not IsInvertible(M), //Should be checked before.
  "ERROR: M is not invertible. Choose a different point on the conic!";
  // The following is like SwapExtension,
  // but over the product of 1 or 2 fields.
  Ks := <BaseRing(Parent(Li)) : Li in L>;
  gs := <DefiningPolynomial(K) : K in Ks>;
  Qy<[y]> := PolynomialRing(Rationals(),4);
  Qyths := [quo<PolynomialRing(Qy) | g/LeadingCoefficient(g)> : g in gs];
  m := [hom<Ks[i] ->Qyths[i] | Qyths[i].1 > : i in [1..#L]];
  gen := <&+[y[i+2*j-1]*Qyths[j].1^i : i in [0..Degree(Ks[j])-1]] :
  j in [1..#L]>;
  // generic element of K1 (x K2) = y = y_1 + y_2 theta_1 + ...
  rhs := [Eltseq(m[i](xi[i])*gen[i]^2) : i in [1..#L]];
  rhs := Matrix([ChangeUniverse(&cat rhs, Qy)]); // as matrix.
  subst := Eltseq(rhs * ChangeRing(M^-1,Qy));
  // We return in addition Norm(y), since we have it already.
  return subst, &*[Norm(gen[i]) : i in [1..#L]];
end function;
```

Now we can compute the three quartics that define $\phi_8^\pm : C_8^\pm \to C_4$.

```
function ThreeQuartics(C4,L,Q3,xi)
  // C4 is the 4-descendent.
  // L are the linear forms defining the tangent planes.
  // Q3 is the third quadric through the 8 points.
  Qx<[x]> := CoordinateRing(AmbientSpace(C4));
  vprintf EightDescent, 2:
  "\nComputing the substitution corresponding to xi = %o.\n",xi;
  subst, NormOfy := CorrespondingSubstitution(L,xi);
  vprintf EightDescent, 2: "Computing the first two quartics.\n";
  G1, G2 := Explode([Evaluate(f,subst) : f in DefiningPolynomials(C4)]);
  // G1 and G2 are two quartics, which define C8.
  // C8 consists of the two components C8plus and C8minus.
  // They are separated by a third quartic, which comes from
  // the norm condition (L1L2L3L4 = c*Q3^2).
  // Since L1L2L3L4 = Norm(xi*y^2), this is equiv. to
  // Norm(xi)*(Norm(y))^2 = c*Q3^2, and since Norm(xi) = c*a^2 for some a,
  // equiv. to a*Norm(y) = +/- Q3(subst).
  c := TheConstant(C4,L,Q3);
  b := &*[Norm(xi[i]) : i in [1..#xi]]/c;
  vprintf EightDescent, 2: "Computing the square root of Norm(xi)/c.\n";
  bool, a := IsSquare(b);
  assert bool; // Norm(xi) = c*a^2.
  vprintf EightDescent, 2: "Computing the third quartic.\n";
  G3plus := a*NormOfy + Evaluate(Q3,subst);
  vprintf EightDescent,2: "Magma is checking that C8plus is a curve.\n";
  vtime EightDescent, 2 :
  C8plus := Curve(AmbientSpace(C4),[Qx|G1,G2,G3plus]);
  G3minus := a*NormOfy - Evaluate(Q3,subst);
  vprintf EightDescent,2: "Magma is checking that C8minus is a curve.\n";
  vtime EightDescent,2:
  C8minus := Curve(AmbientSpace(C4),[Qx|G1,G2,G3minus]);
  phi1 := map<C8plus ->C4 | subst>;
  phi2 := map<C8minus->C4 | subst>;
  return [phi1,phi2];
end function;
```

Putting everything together we can perform a third 2-descent on C_4, i.e. an 8-descent. I implemented the following optional parameters:

`UsePari`: When this parameter is turned on, then the necessary data for computing a point on a conic is written into the file "conic.gp". When you open the file "conic.gp" with PARI/GP (you need Denis Simon's PARI file "ell.gp" available at his web page) the point is computed and written into the file "solution.m". Then you can load "solution.m" and get a point, which you can use as the optional parameter `Point`.

`Point`: This parameter can be set to a point (as sequence) on the conic which is used in the 8-descent. E.g. if it was computed with PARI.

`StopWhenFoundPoint`: When this parameter is turned on, the function stops if it happens to find a point on C_4 during the 8-descent, and returns this point.

`BadPrimesHypothesis`: If this is set to true, then the program assumes that the set of bad primes consists just of $S_1 :=$ prime divisors of $2c\,\mathrm{disc}(g)$, and $S_2 :=$ primes dividing the norms of all coefficients of L_1 (and L_2). The primes coming from projection of the singular subscheme of P_8 to $\mathrm{Spec}(\mathbb{Z})$ are disregarded.

`DontTestLocalSolvabilityAt`: This parameter can be set to a set of primes. At these primes local solvability will not be tested. You can test the resulting curves in the end for local solvability if you want. E.g. with `IsLocallySolvable(Projection(C8),p)`.

```
function EightDescent(C4 : UsePari := false, Point := [],
StopWhenFoundPoint := false, BadPrimesHypothesis := false,
DontTestLocalSolvabilityAt := {})
  P3<[x]> := Ambient(C4); // for nicer output.
  A<theta>, iso, g := EtaleAlgebra(C4);
  error if not g/LeadingCoefficient(g) eq Parent(g)!Modulus(A),
  "ERROR: The assigned etale algebra is wrong.";
  vprintf EightDescent,1:
  "The etale algebra is A = \\Q[T]/(g(T)) where g = %o\n", g;
  g := ChangeRing(g,Integers());
  // for nicer output we assign the names theta_1 and theta_2:
  K1<theta_1> := Codomain(iso)[1];
  vprintf EightDescent,1: "It is isomorphic to ";
  if #iso(theta) eq 2 then
    K2<theta_2> := Codomain(iso)[2];
    vprintf EightDescent,1:
    "K_1\\times K_2 where K_1 = \\Q[T]/(%o) and K_2 = \\Q[T]/(%o).\n",
    DefiningPolynomial(K1), DefiningPolynomial(K2);
  else
    vprintf EightDescent,1: "K_1 where K_1 = \\Q[T]/(%o).\n",
    DefiningPolynomial(K1);
  end if;
  Qsing := SingularQuadricsInThePencil(C4);
  vprintf EightDescent,1: "The singular quadrics in the pencil are \n%o ",
  DefiningPolynomial(Qsing[1]);
  if #Qsing eq 2 then
    vprintf EightDescent,1: "and\n%o ", DefiningPolynomial(Qsing[2]);
  end if;
  vprintf EightDescent,1: "and their conjugates.\n";
  vprintf EightDescent,4: "As symmetric matrices: \n%o\n",
  <SymmetricMatrix(DefiningPolynomial(C)) : C in Qsing>;
  if UsePari then
    error if #Qsing gt 1, "do not use PARI.
    Magma should be able to find a point.";
    C, pr := ConicOfSingularQuadric(Qsing[1]);
    C := ImprovedIntegralModel(C);
    diagmap := diag(C);
    m := InputForPARI(Codomain(diagmap));
    // writes the necessary data into the file "conic.gp".
    print "\n\n\nOpen conic.gp with PARI/GP,";
    print "it writes the solution to the file solution.m.";
    print "(You need Denis Simon's file ell.gp)";
    print "Then load the file solution.m.";
    print "It contains the point you can use as optional parameter.";
    return " ";
  end if;
```

```
vprintf EightDescent,2: "\nComputation of the tangent planes.\n";
ts := [TangentPlaneAt(Q : Point:=Point) : Q in Qsing];
L := <DefiningPolynomial(t) : t in ts>;//The linear forms for the map F.
if not IsInvertible(TheMatrix(L)) and StopWhenFoundPoint then
  vprintf EightDescent,2: "Checking the intersection of the four planes
  with C4 for Q-rational points.\n";
  trivialpts := Points(Scheme(C4, &*[Qx!ProductOfConjugates(Li)
  : Li in L])) where Qx := CoordinateRing(Ambient(C4));
  if not IsEmpty(trivialpts) then
    vprintf EightDescent,1: "Tangent planes meet C4 in a Q-point:\n";
    return trivialpts;
  end if;
end if;
while not IsInvertible(TheMatrix(L)) do
  vprint EightDescent, 2 : "M is not invertible.";
  L := <DifferentTangentPlane(Qsing[i],L[i]) : i in [1..#L]>;
  vprintf EightDescent,2: "We chose a different point.\nNow L = %o\n",L;
end while;
vprintf EightDescent,2: "Computing the third quadric ... \n";
vtime EightDescent,2: Q3 := ThirdQuadric(C4,L); //(C4,L);
vprintf EightDescent,1: "The third quadric in the pencil is %o.\n", Q3;
vprintf EightDescent,4:"As symmetric matrix:\n%o\n",SymmetricMatrix(Q3);
c := TheConstant(C4,L,Q3);
vprintf EightDescent,1:
"The constant in L_1L_2L_3L_4 = c*Q3^2 is c = %o.\n", c;
vprintf EightDescent,2: "\nComputing the bad primes ...\n";
t := Cputime();
S := MyBadPrimes(C4,L,Q3 : BadPrimesHypothesis := BadPrimesHypothesis);
vprintf EightDescent,2:
"Time: %o for computing the set of bad primes.\n", Cputime(t);
vprintf EightDescent,1: "The set of bad primes is %o.\n", S;
vprintf EightDescent,2: "\nComputing the subset H of A(S,2)/Q(S,2),
fulfilling the norm condition.\n";
t := Cputime();
H,A_to_AS2Q,toVec,bU := TheSetH(C4,c,S);
vprintf EightDescent,2:
"Time: %o for computing the set H.\n", Cputime(t);
// if H is empty then we can stop.
if IsEmpty(H[2]) then
  return [];
end if;
vprintf EightDescent,2:
"\nComputing the intersection of the local images\n";
S := S diff DontTestLocalSolvabilityAt;
if not IsEmpty(DontTestLocalSolvabilityAt) then
  vprintf EightDescent,3:
  "Attention: We do not test local solvability at %o.\n",
  DontTestLocalSolvabilityAt;
end if;
```

```
  t := Cputime();
  Sel, mSel := FakeSelmerSet(C4,L,g,Q3,S,H,A_to_AS2Q,toVec,bU);
  vprintf EightDescent,2: "Time: %o
  for computing the intersection of the local images\n", Cputime(t);
  Xi := {iso(xi @@ mSel) : xi in Sel};
  vprintf EightDescent,1: "The fake Selmer set consists of \n%o\n", Xi;
  return &cat[ThreeQuartics(C4,L,Q3,xi) : xi in Xi];
end function;
```

Chapter 6

Cohomological Interpretation of 4- and 8-Descent

6.1 Galois Cohomology of 4-Descent

6.1.1 The Main Tool

Let $\phi_4 : C_4 \to C_2$ be an everywhere locally solvable 2-covering of the 2-descendent $C_2 : y^2 = g(x, z)$, if there is one. Let $R_i := (\theta_i, 0) \in C_2(\bar{\mathbb{Q}})$, $i = 1, \ldots, 4$, and $\mathcal{R} := \{R_1, \ldots, R_4\}$ the set of ramification points on C_2. Then the divisor cut out by the vertical line $x - \theta_i z = 0$ is $(x - \theta_i z)_{C_2} = 2R_i$. The pullback $\phi_4^*(R_i)$ is the divisor cut out by a surface $(G_i)_{C_4} = \phi_4^*(R_i)$ for some form $G_i \in \mathbb{Q}[\theta_i][x_1, \ldots, x_4]$. For the model of C_4 as intersection of two quadrics in $\mathbb{P}^3$ and $\phi_4 : C_4 \to C_2$ given by invariant theory, we can even tell explicitly what $G_i = 0$ is: It is the plane through the four hyperosculating points $\phi_4^{-1}(R_i)$. See Remark 2.4.3 for the fact that they lie on a plane. Hence we have the following equality in the coordinate ring $\mathbb{Q}[\theta_i][C_4]$

$$G_i^2 \xi_i = \phi_4^*(x - \theta_i z) \tag{6.1}$$

for some constant $\xi_i \in \mathbb{Q}[\theta_i]$. We choose $R_i \mapsto G_i$ Galois-equivariant and write $G := (G_i) \in A[C_4]$. Let $\xi := (\xi_i) \in A^*$.

This ξ is the most important ingredient for the whole Galois cohomological interpretation. We will see that we have a kind of correspondence between 2-coverings of C_2 and elements of A^*

$$(\phi_4 : C_4 \to C_2) \longleftrightarrow \xi.$$

Before we get into the Galois cohomology, I want to note that already at this stage we can prove that 4-descent [15] really finds all everywhere locally solvable 2-coverings of C_2.

Theorem 6.1.1. *Let $\phi_4 : C_4 \to C_2$ be an everywhere locally solvable 2-covering of C_2. Let ξ as above. Then*

$$\xi \in \mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q}).$$

Proof. Let $\phi_4 : C_4 \to C_2$ and ξ be given, such that $G^2\xi = \phi_4^*(x - Tz)$ in $A[C_4]$. Then for every place v and every point $Q_v \in C_4(\mathbb{Q}_v)$ we have $(x - T)(\phi_4(Q_v)) \equiv \xi G(Q_v)^2 \equiv \xi \in A_v^*/A_v^{*2}\mathbb{Q}_v^*$, hence $\xi \in \mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q})$. □

Remark 6.1.2. *If $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q})$ happens to be empty as for the 2-descendents in the example of the elliptic curve $E : y^2 + y = x^3 - x^2 - 929x - 10595$ given in [15], then this theorem shows that there is no everywhere locally solvable 2-covering of C_2. Thus this theorem verifies Merriman, Siksek, and Smart's claim that they have computed the 2-primary part of the Shafarevich-Tate group of the elliptic curve. They did show that for every 2-descendent the set of points $C_2(\mathbb{Q})$ is empty, hence $C_2 \in Ш(E/\mathbb{Q})$, and $\#Ш(E/\mathbb{Q})$ is at least 4, but they did not show that all everywhere locally solvable 2-coverings of C_2 can be obtained by their method and hence $\#Ш(E/\mathbb{Q}) = 4$.*

6.1.2 More on the Correspondence

In the following I want to say some more words on the correspondence between 2-coverings of C_2 and elements of A^*

$$(\phi_4 : C_4 \to C_2) \longleftrightarrow \xi.$$

For a given $\xi \in A^*$, we can construct two different 2-coverings $\phi_4^\pm : C_4 \to C_2$ as we have seen in Section 1.4.1. C_4 and $\phi_4^\pm$ are constructed from the equation

$$(x - Tz)(P) = \xi\eta^2 \tag{6.2}$$

for a generic point $P = (x : y : z)$ on C_2 and a generic element $\eta = y_1 + \theta y_2 + \theta^2 y_3 + \theta^3 y_4 \in A$. The curve C_4 is deduced from this such that η leads to a generic point $Q = (y_1 : \ldots : y_4)$ on C_4. The map $\phi_4^\pm$ is defined by mapping Q to $(x : \pm y : z)$. With $G(y_1, \ldots, y_4) := y_1 + \theta y_2 + \theta^2 y_3 + \theta^3 y_4 \in A[C_4]$ equation (6.2) reads

$$(x - Tz)(\phi_4^\pm(y_1 : \ldots : y_4)) = \xi G(y_1, \ldots, y_4)^2,$$

which is nothing else than equation (6.1) with the same ξ as the one we started with.

For the converse, if we have a 2-covering $\phi_4 : C_4 \to C_2$, then we get an element $\xi \in A^*$ and a linear form $G \in A[C_4]$ such that equation (6.1) holds

$$G^2\xi = \phi_4^*(x - Tz).$$

Further, we can assume by a linear change of variables that G is given by $G(y_1, \dots, y_4) = y_1 + \theta y_2 + \theta^2 y_3 + \theta^3 y_4$ and C_4 is embedded in $\mathbb{P}^3$ with variables $y_1, \dots, y_4$. Then reversing the above argument our initial $\phi_4 : C_4 \to C_2$ is one of the two 2-coverings that we can construct out of ξ.

6.1.3 Cohomological Interpretation of $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q})$

Now we turn our attention to the Galois cohomological interpretation of 4-descent. We need some maps, which we will define below. Assume that there exists an everywhere locally solvable 2-covering $\phi_4 : C_4 \to C_2$. The following map is a kind of twisted version of the connecting homomorphism

$$\delta_2 : E(\mathbb{Q}) \longrightarrow H^1(\mathbb{Q}, E[2])$$

of the long exact cohomology sequence. To indicate that the following map depends on the choice of the 2-covering $\phi_4 : C_4 \to C_2$, I denote it by δ_{ϕ_4}. Let

$$\begin{aligned} \delta_{\phi_4} : C_2(\mathbb{Q}) &\longrightarrow H^1(\mathbb{Q}, E[2]) \\ P &\longmapsto (\sigma \mapsto [Q^\sigma - Q]) \end{aligned}$$

where $Q \in C_4(\bar{\mathbb{Q}})$ with $\phi_4(Q) = P$. Let e_2 be the Weil pairing, and fix $R_1 \in \mathcal{R}$. Now we interpret $\bar{A}$ as $\mathrm{Map}(\mathcal{R}, \bar{\mathbb{Q}})$ and A as the Galois-equivariant subset $\mathrm{Map}(\mathcal{R}, \bar{\mathbb{Q}})^{\mathrm{Gal}(\bar{\mathbb{Q}}/\mathbb{Q})}$. Let

$$\begin{aligned} \varepsilon : E[2] &\longrightarrow \mu_2(\bar{A})/\mu_2 \\ T &\longmapsto (R_i \mapsto e_2(T, [R_i - R_1])). \end{aligned}$$

Remark 6.1.3. *The map above is a special case of the map ε introduced by Poonen and Schaefer in the end of Section 6 in [17].*

This can be seen by applying Proposition 7.1 in [17], which reads in our notation: If $T \in E[2]$, then $\varepsilon(T) = (R_i \mapsto e_2(T, R_i))$. Poonen and Schaefer extended the Weil pairing such that they are able to write $e_2(T, R_i)$. The bilinearity of their extended pairing implies $e_2(T, [R_i - R_1]) = e_2(T, R_i)/e_2(T, R_1)$, hence in $\mu_2(\bar{A})/\mu_2$ we have the equality

$$(R_i \mapsto e_2(T, [R_i - R_1])) = (R_i \mapsto e_2(T, R_i)) = \varepsilon(T).$$

Notice that ε is independent of the choice of R_1 since we divide by μ_2, thus ε is a homomorphism of Galois modules, hence it induces a map

$$\varepsilon_* : H^1(\mathbb{Q}, E[2]) \longrightarrow H^1(\mathbb{Q}, \mu_2(\bar{A})/\mu_2).$$

Variations of this map are also studied by Lopez-Neumann [11]. We also need the injective homomorphism from the Kummer sequence

$$\begin{aligned} q_* : A^*/A^{*2}\mathbb{Q}^* &\hookrightarrow H^1(\mathbb{Q}, \mu_2(\bar{A})/\mu_2) \\ a &\longmapsto (\sigma \mapsto \alpha^\sigma/\alpha) \text{ with } \alpha^2 = a \end{aligned}$$

which can also be found in [17]. With the $\xi \in A^*$ from (6.1) above we have the translation by ξ-map

$$A^*/A^{*2}\mathbb{Q}^* \to A^*/A^{*2}\mathbb{Q}^*, \quad \alpha \mapsto \alpha\xi.$$

In addition, we have all these maps locally, i.e. over $\mathbb{Q}_v$. To relate them we have the canonical restriction maps

$$\mathrm{res}_v : H^1(\mathbb{Q}, E[n]) \longrightarrow H^1(\mathbb{Q}_v, E[n])$$

and on the corresponding groups from the étale algebra modulo squares and scalars

$$\mathrm{res}_v : A^*/A^{*2}\mathbb{Q}^* \longrightarrow A_v^*/A_v^{*2}\mathbb{Q}_v^*$$

where $A_v := A \otimes \mathbb{Q}_v$, which we also denote by res_v. Now we can set up the main diagram for the Galois cohomological interpretation of 4-descent.

Theorem 6.1.4. *The following diagram commutes.*

$$\begin{tikzcd}
C_2(\mathbb{Q}) \arrow[r, "x - T"] \arrow[dd, "\delta_{\phi_4}"'] & A^*/A^{*2}\mathbb{Q}^* \arrow[d, "\cdot\xi"] \\
 & A^*/A^{*2}\mathbb{Q}^* \arrow[d, hook, "q_*"] \\
H^1(\mathbb{Q}, E[2]) \arrow[r, "\varepsilon_*"] & H^1(\mathbb{Q}, \mu_2(\bar{A})/\mu_2).
\end{tikzcd}$$

The same holds over $\mathbb{Q}_v$. For that, we take the local versions of the maps $\delta_{\phi_4,v}$, $\varepsilon_{*,v}$, $q_{*,v}$, and multiplication by $\mathrm{res}_v(\xi)$. Notice that ξ is defined globally.

Proof. The main tool for the proof is the following short cut map, going from the upper left corner to the lower right one.

$$\begin{aligned} s : C_2(\mathbb{Q}) &\longrightarrow H^1(\mathbb{Q}, \mu_2(\bar{A})/\mu_2) \\ P &\longmapsto \left(\sigma \mapsto \left(R_i \mapsto \frac{G_i(X + [Q^\sigma - Q])}{G_i(X)}\right)\right) \end{aligned}$$

where G_i is as in equation (6.1), Q is a point of $C_4(\bar{\mathbb{Q}})$ with $\phi_4(Q) = P$, and X is any point on C_4 such that G_i is non-zero at X and $X + [Q^\sigma - Q]$. Here the plus sign means the action of E on C_4.

By a slight generalization of the definition of the Weil pairing we get (up to a global change of sign for all i simultaneously)

$$e_2(T, [R_i - R_1]) = \frac{G_i(X+T)}{G_i(X)},$$

which shows that the lower triangle commutes, i.e. $\varepsilon_* \circ \delta_{\phi_4} = s$.

To show that the upper triangle commutes, let $P \in C_2(\mathbb{Q})$ and $Q \in C_4(\bar{\mathbb{Q}})$ such that $\phi_4(Q) = P$. First, assume that $P \notin \mathcal{R}$. Then $(x-T)(P) = \xi G(Q)^2$ in $A^*/A^{*2}\mathbb{Q}^*$, and putting $X = Q$, we get

$$\begin{aligned} s(P) &= \left(\sigma \mapsto \left(R_i \mapsto \frac{G_i(Q^\sigma)}{G_i(Q)}\right)\right) \\ &= (\sigma \mapsto (\alpha^\sigma/\alpha)) \end{aligned}$$

where $\alpha = (R_i \mapsto G_i(Q)) = G(Q) \in \bar{A}^* = \mathrm{Map}(\mathcal{R}, \bar{\mathbb{Q}}^*)$. Hence

$$\begin{aligned} q_*(\xi \cdot ((x-T)(P))) &= q_*(\xi^2 G^2(Q)) \\ &= q_*(G^2(Q)) \\ &= (\sigma \mapsto \alpha^\sigma/\alpha) \\ &= s(P). \end{aligned}$$

The case $P \in \mathcal{R}$ does not happen over $\mathbb{Q}$, since else C_2 has a trivial $\mathbb{Q}$-rational point, and we do not do a further descent. However, over $\mathbb{Q}_v$ this can happen, and then we use linearity.

For that, let $P_E \in E(\mathbb{Q}_v)$ be any point such that $P + P_E \notin \mathcal{R}$. The plus sign means action of E on C_2. Then $(x-T)(P) = ((x-T)(P+P_E))/((x-T)(P_E)$, where $(x-T)(P_E)$ means evaluation of the $x-T$-map at a suitable degree 0 divisor representing P_E. Remember that the $x-T$-map can be defined on all of $\mathrm{Pic}(C_2)$. On the other hand, $\delta_{\phi_4}(P+P_E) = \delta_{\phi_4}(P) + \delta_2(P_E)$ by the following lemma. Hence the claim follows from $\varepsilon_*\delta_2(P_E) = q_*((x-T)(P_E))$, which holds by [17, Theorem 9.4], see also Theorem 6.1.8 below. □

Lemma 6.1.5. *Let $P \in C_2(\mathbb{Q})$ and $P_E \in E(\mathbb{Q})$. Then*

$$\delta_{\phi_4}(P + P_E) = \delta_{\phi_4}(P) + \delta_2(P_E).$$

The same holds over $\mathbb{Q}_v$.

Proof. Let $Q_E \in E(\bar{\mathbb{Q}})$ with $[2]Q_E = P_E$, then $\delta_2(P_E) = (\sigma \mapsto (Q_E^\sigma - Q_E))$. On the other hand, $\phi_4(Q + Q_E) = P + P_E$, hence

$$\begin{aligned}\delta_{\phi_4}(P + P_E) &= (\sigma \mapsto [(Q + Q_E)^\sigma - (Q + Q_E)]) \\ &= (\sigma \mapsto [Q^\sigma - Q]) + (\sigma \mapsto (Q_E^\sigma - Q_E)) \\ &= \delta_{\phi_4}(P) + \delta_2(P_E).\end{aligned}$$

□

Lemma 6.1.6. *If there exists an everywhere locally solvable 2-covering $\phi_4 : C_4 \to C_2$ of C_2, then*

$$\bigcap_v \operatorname{res}_v^{-1}(\delta_{\phi_4,v}(C_2(\mathbb{Q}_v))) = \operatorname{Sel}^{(2)}(E/\mathbb{Q})$$

in $H^1(\mathbb{Q}, E[2])$.

Proof. Take a point $Q_v \in C_4(\mathbb{Q}_v)$ as the origin for the group law on C_4, and $P_v := \phi_4(Q_v)$ as the origin for the group law on C_2, then $C_4 \cong C_2 \cong E$, and ϕ_4 is multiplication by 2. With these identifications $\delta_{\phi_4,v}$ and $\delta_{2,v}$ are exactly the same. The claim follows from $\operatorname{Sel}^{(2)}(E/\mathbb{Q}) = \bigcap_v \operatorname{res}_v^{-1}(\delta_{2,v}(E(\mathbb{Q}_v)))$ □

The short exact sequence

$$0 \to E[2] \xhookrightarrow{i} E[4] \xrightarrow{[2]} E[2] \to 0,$$

where i is the inclusion, induces a long exact sequence

$$0 \to \frac{E(\mathbb{Q})[2]}{2E(\mathbb{Q})[4]} \longrightarrow H^1(\mathbb{Q}, E[2]) \xrightarrow{i_*} H^1(\mathbb{Q}, E[4]) \xrightarrow{[2]_*} H^1(\mathbb{Q}, E[2]).$$

Let $\pi_2^4 : \operatorname{Sel}^{(4)}(E/\ \mathbb{Q}) \to \operatorname{Sel}^{(2)}(E/\mathbb{Q})$ be the restriction of $[2]_*$ to the Selmer groups, then we can compute the fibers of π_2^4 by the following

Proposition 6.1.7. *If there exists an everywhere locally solvable 2-covering $\phi_4 : C_4 \to C_2$ of C_2, then*

$$(\pi_2^4)^{-1}([C_2]) = [C_4] + i_*(\operatorname{Sel}^{(2)}(E/\mathbb{Q})),$$

where $[C_n]$ denotes the class in $\operatorname{Sel}^{(n)}(E/\mathbb{Q})$ *represented by the n-covering C_n.*

Proof. Clear. □

This shows that we can parametrize the coset $(\pi_2^4)^{-1}([C_2])$, which we actually want to compute, by $\operatorname{Sel}^{(2)}(E/\mathbb{Q})$. And $\operatorname{Sel}^{(2)}(E/\mathbb{Q})$ is related to $\operatorname{Sel}^{(2)}_{\text{fake}}(C_2/\mathbb{Q})$ by Theorem 6.1.4.

6.1.4 The Size of $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q})$

let us have a more detailed look at the size of the fake Selmer set. It is exactly 1/2 of the size of $\mathrm{Sel}^{(2)}(E/\mathbb{Q})$, since the map ε_* is 2:1. One can prove that by elementary methods, looking at the resolvent of $g(x)$.

However, it might be illuminating to deduce it from the theory about descent on Jacobians of cyclic covers of the projective line [17]. Our curve $C_2 : y^2 = g(x)$ is one of the simplest examples of a cyclic cover of the projective line. We can consider E as the Jacobian of C_2 and use the $x-T$-map on C_2 to do 2-descent on E. This is useless in practice, since we already did a 2-descent on E to compute C_2, however in theory it gives us a new point of view. Notice that we use the $x-T$-map on C_2, not the one on E. We just identify E with $\mathrm{Pic}^0(C_2)$ and evaluate the $x-T$-map on degree zero divisors of C_2. Poonen and Schaefer call the following subgroup of $A^*/A^{*2}\mathbb{Q}^*$ the *fake 2-Selmer group of E*:

$$\mathrm{Sel}^{(2)}_{\mathrm{fake}}(E/\mathbb{Q}) := \bigcap_v \mathrm{res}_v^{-1}((x-T)(E(\mathbb{Q}_v))).$$

Below we will identify $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(E/\mathbb{Q})$ with its image under q_*.

Theorem 6.1.8 (Poonen-Schaefer). *Let δ_2, ε_*, and q_* as above. Then the diagram*

$$\begin{tikzcd}
E(\mathbb{Q}) \arrow[r, "x-T", "\text{on } \mathrm{Pic}^0(C_2)"'] \arrow[d, "\delta_2"'] & A^*/A^{*2}\mathbb{Q}^* \arrow[d, hook, "q_*"] \\
H^1(\mathbb{Q}, E[2]) \arrow[r, "\varepsilon_*"] & H^1(\mathbb{Q}, \mu_2(\bar{A})/\mu_2)
\end{tikzcd}$$

commutes. The same holds locally.

Proof. This is a special case of Theorem 9.4 in [17]. Our map δ_2 is called ι there, $p=2$, and we restrict the domain of the $x-T$-map and δ_2 to $\mathrm{Pic}^0(C_2) = E(\mathbb{Q})$. These maps could be defined on a larger domain, but we do not need that. □

Now we come to the fact that the map ε_* is 2:1.

Theorem 6.1.9 (Poonen-Schaefer). *The sequence*

$$0 \to \mu_2 \longrightarrow \mathrm{Sel}^{(2)}(E/\mathbb{Q}) \xrightarrow{\varepsilon_*} \mathrm{Sel}^{(2)}_{\mathrm{fake}}(E/\mathbb{Q}) \to 0$$

is exact.

Proof. This is a special case of Theorem 13.2 in [17]. The conditions for exactness on the left side are satisfied: $g(x)$ never has factors of degree prime to $p = 2$, and the genus of C_2 is $g = 1$, hence is not even. Looking at the definition of the map $\mu_2 \to \mathrm{Sel}^{(2)}(E/\mathbb{Q})$ one even sees that $-1 \mapsto [C_2]$, where $[C_2]$ denotes the class of the 2-covering $\phi_2 : C_2 \to E$ in the 2-Selmer group, i.e. $\ker(\varepsilon_*) = \{0, [C_2]\}$. □

Next, we will see that our fake Selmer set $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q})$ is just a translation of $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(E/\mathbb{Q})$.

Lemma 6.1.10. *Let $\phi_4 : C_4 \to C_2$ be an everywhere locally solvable 2-covering of C_2. Let ξ as in equation (6.1). Let v be some place. Let $Q_{0,v} \in C_4(\mathbb{Q}_v)$ and $P_{0,v} := \phi_4(Q_v)$. Let $\psi_v : C_2(\mathbb{Q}_v) \to E(\mathbb{Q}_v)$, $P_v \mapsto [P_v - P_{0,v}]$ be the isomorphism depending on $P_{0,v}$. Then*

$$\begin{tikzcd}
C_2(\mathbb{Q}_v) \arrow[r, "x-T"] \arrow[r, "\text{on } \mathrm{Pic}^1(C_2)"'] \arrow[d, "\psi_v"'] & A_v^*/A_v^{*2}\mathbb{Q}_v^* \arrow[d, "\cdot\, \mathrm{res}_v(\xi)"] \\
E(\mathbb{Q}_v) \arrow[r, "x-T"] \arrow[r, "\text{on } \mathrm{Pic}^0(C_2)"'] \arrow[d, "\delta_{2,v}"'] & A_v^*/A_v^{*2}\mathbb{Q}_v^* \arrow[d, hook, "q_{v,*}"] \\
H^1(\mathbb{Q}_v, E[2]) \arrow[r, "\varepsilon_*"] & H^1(\mathbb{Q}_v, \mu_2(\bar{A}_v)/\mu_2)
\end{tikzcd}$$

commutes.

Proof. The commutativity of the lower square is the Theorem above, and the commutativity of the upper square follows from $(x - T)(P_{0,v}) = \mathrm{res}_v(\xi)$ by equation (6.1). □

This lemma can be used to give a different proof of Theorem 6.1.4. One just has to use $\delta_2 \circ \psi = \delta_{\phi_4}$, which holds by Lemma 6.1.5.

Remark 6.1.11. *Note that ξ is independent of v. The existence of such a global ξ depends on the existence of an everywhere locally solvable 2-covering of C_2.*

If we do not have an everywhere locally solvable 2-covering of C_2, then we can still get isomorphisms $\psi'_v : C_2(\mathbb{Q}_v) \to E(\mathbb{Q}_v)$, $P_v \mapsto [P_v - P_{1,v}]$ by choosing *any point $P_{1,v} \in C_2(\mathbb{Q}_v)$. But then we have to choose $\xi_v := x - T(P_{1,v})$ to get a commutative diagram*

$$\begin{tikzcd}
C_2(\mathbb{Q}_v) \arrow[r, "x-T", "\text{on } \mathrm{Pic}^1(C_2)"'] \arrow[d, "\psi'"'] & A^*/A^{*2}\mathbb{Q}^* \arrow[d, "\cdot\xi_v"] \\
E(\mathbb{Q}) \arrow[r, "x-T", "\text{on } \mathrm{Pic}^0(C_2)"'] & A^*/A^{*2}\mathbb{Q}^*.
\end{tikzcd}$$

Different ξ_v's might move the local images to different cosets, such that the intersection of them is empty, i.e. $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q}) = \emptyset$.

Only if there exists an everywhere locally solvable 2-covering of C_2, then all the local ξ_v's patch together to a global ξ, and we get the following

Corollary 6.1.12. *If there exists an everywhere locally solvable 2-covering of C_2, then*

$$\#\,\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q}) = \frac{1}{2}\#\,\mathrm{Sel}^{(2)}(E/\mathbb{Q}),$$

else $\quad \#\,\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q}) = 0.$

Proof. If there exists an everywhere locally solvable 2-covering of C_2, then by Lemma 6.1.10 $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q})$ is a translation of $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(E/\mathbb{Q})$, hence has the same size, and $\#\,\mathrm{Sel}^{(2)}_{\mathrm{fake}}(E/\mathbb{Q}) = \frac{1}{2}\#\,\mathrm{Sel}^{(2)}(E/\mathbb{Q})$ by Theorem 6.1.9.

If there is no everywhere locally solvable 2-covering of C_2, then $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q})$ is empty by Theorem 6.1.1. □

This shows why it makes sense that we can construct two 2-coverings out of one element $\xi \in A^*$.

6.2 Galois Cohomology of 8-Descent

The Galois cohomological interpretation of 8-descent is very similar to the one of 4-descent. We use the same tools and get analogous results.

6.2.1 The Main Tool

Let $\phi_4 : C_4 \to C_2$ be an everywhere locally solvable 2-covering of the 2-descendent C_2. Now we assume that there exists an everywhere locally solvable 2-covering $\phi_8 : C_8 \to C_4$ of C_4. Let $L \in A[C_4]$ be the linear form which defines the descent map F. Geometrically, $L = 0$ defines a tangent plane to the generic singular quadric in the pencil. Denote the divisor cut out by this plane by $(L)_{C_4} = 2(S_1^\theta + S_2^\theta)$. By the lemma below the pullback $\phi_8^*(S_1^\theta + S_2^\theta)$ is the divisor cut out by a hyperplane $(G)_{C_8} = \phi_8^*(S_1^\theta + S_2^\theta)$ for some linear form $G \in A[C_8]$. Hence we have the following equality in the coordinate ring $A[C_8]$

$$G^2\xi = \phi_8^*(L) \tag{6.3}$$

for some constant $\xi \in A^*$.

Lemma 6.2.1. *The divisor $\phi_8^*(S_1^\theta + S_2^\theta)$ is the divisor cut out by a hyperplane defined over A for a suitable model of C_8.*

Proof. Let $\theta_1, \ldots, \theta_4$ be the roots of g in $\bar{\mathbb{Q}}$, and $L_1, \ldots, L_4$ the corresponding linear forms. Let the divisor $(L_i)_{C_4} = 2(S_1^{\theta_i} + S_2^{\theta_i})$. Note that $\phi_8^*(S_1^{\theta_i} + S_2^{\theta_i}) = \sum_{T \in E[2]}((Q_1^{\theta_i} + T) + (Q_2^{\theta_i} + T))$ for some $Q_1^{\theta_i}, Q_2^{\theta_i} \in C_8(\bar{\mathbb{Q}})$ with $\phi_8(Q_1^{\theta_i}) = S_1^{\theta_i}$ and $\phi_8(Q_2^{\theta_i}) = S_2^{\theta_i}$. Here $Q_j^{\theta_i} + T$ is the action of E on C_8.

Then the class of the divisor

$$\begin{aligned}
[\phi_8^*((S_1^{\theta_i} + S_2^{\theta_i}) &- (S_1^{\theta_4} + S_2^{\theta_4}))] \\
&= \sum_{T \in E[2]} [(Q_1^{\theta_i} + T) + (Q_2^{\theta_i} + T) - (Q_1^{\theta_4} + T) - (Q_2^{\theta_4} + T)] \\
&= \sum_{T \in E[2]} [Q_1^{\theta_i} + Q_2^{\theta_i} - Q_1^{\theta_4} - Q_2^{\theta_4}] \\
&= 4[Q_1^{\theta_i} + Q_2^{\theta_i} - Q_1^{\theta_4} - Q_2^{\theta_4}] \\
&= 2[S_1^{\theta_i} + S_2^{\theta_i} - S_1^{\theta_4} - S_2^{\theta_4}] \\
&= O,
\end{aligned}$$

since $[S_1^{\theta_i} + S_2^{\theta_i} - S_1^{\theta_4} - S_2^{\theta_4}] \in E[2]$ by Corollary 3.6.4. Thus $\phi_8^*(S_1^{\theta_i} + S_2^{\theta_i})$ is linearly equivalent to $\phi_8^*(S_1^{\theta_4} + S_2^{\theta_4}) =: D$ for all i. Hence it is enough to show that D is the divisor cut out by a hyperplane for a suitable model of C_8.

For that, we can use the degree 8 divisor D, whose class is defined over $\mathbb{Q}$, to embed C_8 into a Severi-Brauer variety $S/\mathbb{Q}$ of dimension 7. Since C_8 is everywhere locally solvable, so is S, hence has a $\mathbb{Q}$-rational point by the Hasse principle, thus is $\mathbb{Q}$-isomorphic to $\mathbb{P}^7$. For this embedding of C_8 into $\mathbb{P}^7$ the divisor D obviously lies on a plane.

Since $\phi_8^*(S_1^{\theta_i} + S_2^{\theta_i})$ is defined over $\mathbb{Q}[\theta_i]$, the hyperplane $G_i = 0$ through it is defined over $\mathbb{Q}[\theta_i]$, moreover $\theta_i \mapsto \phi_8^*(S_1^{\theta_i} + S_2^{\theta_i})$ is Galois-equivariant, hence we can choose $\theta_i \mapsto G_i$ Galois-equivariant, then $G := (G_i)$ is defined over A. □

With this ξ we have again a kind of correspondence between 2-coverings of C_4 and elements of A^*

$$(\phi_8 : C_8 \to C_4) \longleftrightarrow \xi$$

We now prove the most important part of this correspondence, which shows that our method of doing an 8-descent finds all everywhere locally solvable 2-coverings of C_4.

Theorem 6.2.2. *Let $\phi_8 : C_8 \to C_4$ be an everywhere locally solvable 2-covering of C_4. Let ξ as above. Then*

$$\xi \in \mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q}).$$

In particular, if $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q}) = \emptyset$, then there cannot exist an everywhere locally solvable 2-covering of C_4.

Proof. Let $\phi_8 : C_8 \to C_4$ and ξ be given, such that $G^2\xi = \phi_8^*(L)$ in $A[C_8]$. Then for every place v and every point $Q_v \in C_8(\mathbb{Q}_v)$ we have $L(\phi_8(Q_v)) \equiv \xi G(Q_v)^2 \equiv \xi \in A_v^*/A_v^{*2}\mathbb{Q}_v^*$, hence $\xi \in \mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q})$. □

6.2.2 More on the Correspondence

The correspondence between 2-coverings of C_4 and elements of A^*

$$(\phi_8 : C_8 \to C_4) \longleftrightarrow \xi$$

is very similar to the case of 4-descent.

For a given $\xi \in A^*$, we can construct two different 2-coverings $\phi_8^{\pm} : C_8^{\pm} \to C_4$ as we have seen in Section 5. $C_8^{\pm}$ and $\phi_8^{\pm}$ are constructed from the equation

$$F(P) = \xi y^2 \tag{6.4}$$

for a generic point $P = (x_1 : \ldots : x_4)$ on C_4 and a generic element $y = y_1 + \theta y_2 + \theta^2 y_3 + \theta^3 y_4 \in A$, which leads to a generic point $Q = (y_1 : \ldots : y_4)$ on $C_8^+ \cup C_8^-$. The maps $\phi_8^{\pm}$ are defined by mapping Q to P. With $G(y_1, \ldots, y_4) := y_1 + \theta y_2 + \theta^2 y_3 + \theta^3 y_4 \in A[y_1, \ldots, y_4]$ equation (6.4) reads

$$L(\phi_8^{\pm}(y_1 : \ldots : y_4)) = \xi G(y_1, \ldots, y_4)^2,$$

which is nothing else than equation (6.3) with the same ξ as the one we started with.

For the converse, if we have a 2-covering $\phi_8 : C_8 \to C_4$, then there is an element $\xi \in A^*$ and a linear form $G \in A[C_8]$ by Lemma 6.2.1 such that

$$G^2\xi = \phi_8^*(L).$$

for a suitable model of $C_8 \subset \mathbb{P}^n$ with variables $z_1, \ldots, z_{n+1}$. The proof of the lemma shows $n = 7$. Write $G = G_1(z_1, \ldots, z_{n+1}) + \theta G_2(z_1, \ldots, z_{n+1}) + \ldots + \theta^3 G_4(z_1, \ldots, z_{n+1})$ with $G_i(z_1, \ldots, z_{n+1}) \in \mathbb{Q}[z_1, \ldots, z_{n+1}]$, $i = 1, \ldots, 4$. Then by the linear change of variables $y_i := G_i(z_1, \ldots, z_{n+1})$, $i = 1, \ldots, 4$, we get $G = y_1 + \theta y_2 + \theta^2 y_3 + \theta^3 y_4$. In addition, this linear change of variables should give an embedding of C_8 into $\mathbb{P}^3$, which coincides with one of the two 2-coverings we can construct out of ξ.

Then reversing the above argument our initial $\phi_8 : C_8 \to C_4$ would be one of the two 2-coverings that we can construct out of ξ.

6.2.3 Cohomological Interpretation of $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q})$

For the Galois cohomological interpretation of 8-descent we need almost the same maps as for the 4-descent. Assume that there exists an everywhere locally solvable 2-covering $\phi_8 : C_8 \to C_4$ of C_4. Let

$$\begin{aligned} \delta_{\phi_8} : C_4(\mathbb{Q}) &\longrightarrow H^1(\mathbb{Q}, E[2]) \\ P &\longmapsto (\sigma \mapsto [Q^\sigma - Q]) \end{aligned}$$

where $Q \in C_8(\bar{\mathbb{Q}})$ with $\phi_8(Q) = P$. Let ε_* and q_* the maps defined in Section 6.1.3.

Now the main diagram for the Galois cohomological interpretation of 8-descent is the following

Theorem 6.2.3. *The following diagram commutes, and the local versions, too.*

$$\begin{tikzcd}
C_4(\mathbb{Q}) \arrow[r, "F"] \arrow[dd, "\delta_{\phi_8}"'] & A^*/A^{*2}\mathbb{Q}^* \arrow[d, "\cdot\xi"] \\
 & A^*/A^{*2}\mathbb{Q}^* \arrow[d, hook, "q_*"] \\
H^1(\mathbb{Q}, E[2]) \arrow[r, "\varepsilon_*"] & H^1(\mathbb{Q}, \mu_2(\bar{A})/\mu_2).
\end{tikzcd}$$

Proof. The proof is completely analogous to the one of Theorem 6.1.4. Here we use

$$\begin{aligned} s : C_4(\mathbb{Q}) &\longrightarrow H^1(\mathbb{Q}, \mu_2(\bar{A})/\mu_2) \\ P &\longmapsto \left(\sigma \mapsto \left(R_i \mapsto \frac{G_i(X + [Q^\sigma - Q])}{G_i(X)}\right)\right) \end{aligned}$$

where G_i is as in equation (6.3), Q is a point of $C_8(\bar{\mathbb{Q}})$ with $\phi_8(Q) = P$, and X is any point on C_8 such that G_i is non-zero at X and $X + [Q^\sigma - Q]$. □

Lemma 6.2.4. *If there exists an everywhere locally solvable 2-covering $\phi_8 : C_8 \to C_4$ of C_4, then*

$$\bigcap_v \operatorname{res}_v^{-1}(\delta_{\phi_8,v}(C_4(\mathbb{Q}_v))) = \operatorname{Sel}^{(2)}(E/\mathbb{Q}).$$

Proof. Analogous to the proof of Lemma 6.1.6. □

The short exact sequence

$$0 \to E[2] \xhookrightarrow{i} E[8] \xrightarrow{[2]} E[4] \to 0,$$

where i is the inclusion, induces a long exact sequence

$$0 \to \frac{E(\mathbb{Q})[4]}{2E(\mathbb{Q})[8]} \longrightarrow H^1(\mathbb{Q}, E[2]) \xrightarrow{i_*} H^1(\mathbb{Q}, E[8]) \xrightarrow{[2]_*} H^1(\mathbb{Q}, E[4]).$$

Let $\pi_4^8 : \operatorname{Sel}^{(8)}(E/\ \mathbb{Q}) \to \operatorname{Sel}^{(4)}(E/\mathbb{Q})$ be the restriction of $[2]_*$ to the Selmer groups, then we can compute its fibers by the following

Lemma 6.2.5. *If there exists an everywhere locally solvable 2-covering $\phi_8 : C_8 \to C_4$ of C_4, then*

$$(\pi_4^8)^{-1}([C_4]) = [C_8] + i_*(\operatorname{Sel}^{(2)}(E/\mathbb{Q})).$$

Proof. Clear. □

This shows that we can parametrize the coset $(\pi_4^8)^{-1}([C_4])$, which we actually want to compute, by $\operatorname{Sel}^{(2)}(E/\mathbb{Q})$. And $\operatorname{Sel}^{(2)}(E/\mathbb{Q})$ is related to $\operatorname{Sel}^{(2)}_{\text{fake}}(C_4/\mathbb{Q})$ by Theorem 6.2.3.

6.2.4 The Size of $\mathrm{Sel}^{(2)}_{\textbf{fake}}(C_4/\mathbb{Q})$

The size of the fake Selmer set $\mathrm{Sel}^{(2)}_{\text{fake}}(C_4/\mathbb{Q})$ is 0 or half of the size of $\mathrm{Sel}^{(2)}(E/\mathbb{Q})$. It should be possible to show that the diagram

$$\begin{tikzcd}
E(\mathbb{Q}) \arrow[r, "F", "\text{on } \mathrm{Pic}^0(C_4)"'] \arrow[d, "\delta_2"'] & A^*/A^{*2}\mathbb{Q}^* \arrow[d, hook, "q_*"] \\
H^1(\mathbb{Q}, E[2]) \arrow[r, "\varepsilon_*"] & H^1(\mathbb{Q}, \mu_2(\bar{A})/\mu_2)
\end{tikzcd}$$

commutes, and then one can proceed as in Section 6.1.4.

However, we can use Theorem 3.6.5, which says that $\mathrm{im}(F)$ and $\mathrm{im}(x-T)$ coincide up to a translation.

Theorem 6.2.6. *Let $\phi_4 : C_4 \to C_2$ be a 2-covering of C_2, and let ξ_4 be the corresponding element in A^*. Assume there exists an everywhere locally solvable 2-covering $\phi_8 : C_8 \to C_4$, and let ξ_8 be the corresponding element in A^*. Let $Q_v \in \phi_8(C_8(\mathbb{Q}_v))$, and $P_v := \phi_4(Q_v)$. Then we have an isomorphism (of elliptic curves) $\psi_v : C_4 \to C_2$ defined by mapping $Q_v \mapsto P_v$ and the diagram*

$$\begin{tikzcd}
C_4(\mathbb{Q}_v) \arrow[r, "F_v"] \arrow[d, "\psi_v"'] & A_v^*/A_v^{*2}\mathbb{Q}_v^* \arrow[d, "\cdot\, \mathrm{res}_v(\xi_4/\xi_8)"] \\
C_2(\mathbb{Q}_v) \arrow[r, "x-T"] & A_v^*/A_v^{*2}\mathbb{Q}_v^*
\end{tikzcd}$$

commutes.

Proof. By the proof of Theorem 3.6.5 we have $F_v = \alpha_v \psi_v^*(x-T)$ for some α_v in A_v^*. Since $F_v(Q_v) = \mathrm{res}_v(\xi_8)$ and $(x-T)(\psi_v(Q_v)) = \mathrm{res}_v(\xi_4)$, we must have $\alpha_v = \mathrm{res}_v(\xi_4/\xi_8)$. □

The novelty here is that the translation can be obtained by a global object ξ_8/ξ_4.

Corollary 6.2.7. *If there exists an everywhere locally solvable 2-covering of C_4, then*

$$\#\,\mathrm{Sel}^{(2)}_{\text{fake}}(C_4/\mathbb{Q}) = \frac{1}{2}\#\,\mathrm{Sel}^{(2)}(E/\mathbb{Q}),$$

else $\quad \#\,\mathrm{Sel}^{(2)}_{\text{fake}}(C_4/\mathbb{Q}) = 0.$

Proof. If there exists an everywhere locally solvable 2-covering of C_4, then by the previous theorem $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q})$ is translation of $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q})$ by ξ_4/ξ_8, hence has the same size, and $\#\,\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_2/\mathbb{Q}) = \frac{1}{2}\#\,\mathrm{Sel}^{(2)}(E/\mathbb{Q})$ by Corollary 6.1.12.

If there is no everywhere locally solvable 2-covering of C_4, then $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q})$ is empty by Theorem 6.2.2. □

Again, this shows why it makes sense that we can construct two 2-coverings out of one element $\xi \in A^*$. By a counting argument we get the

Corollary 6.2.8. *The two 2-coverings $\phi_8^{\pm} : C_8^{\pm} \to C_4$ constructed out of an element $\xi \in \mathrm{Sel}_{\mathrm{fake}}(C_4/\mathbb{Q})$ are both everywhere locally solvable.*

Chapter 7

Examples

7.1 $Ш(E/\mathbb{Q}) \supset (\mathbb{Z}/8\mathbb{Z})^2$

The method of 8-descent can be used to show that there are elements of order 8 in $Ш(E/\mathbb{Q})$ for certain elliptic curves E. In the following example E has many isogenous curves, and the étale algebra splits into two number fields of degree 2. So this is an example for the split case, whereas the generic case will occur in the next example. The existence of isogenies has some nice effects. The first is, that the coefficients do not get blown up so much, so this example illustrates the method of 8-descent very nicely. Secondly, using isogenous curves we can actually prove that $Ш(E/\mathbb{Q}) = (\mathbb{Z}/8\mathbb{Z})^2$, whereas an 8-descent could only show $Ш(E/\mathbb{Q}) \supset (\mathbb{Z}/8\mathbb{Z})^2$. For proving equality by descent, one would have to do a fourth 2-descent, i.e. a 16-descent.

However, if there are no isogenies, then we are dependent on 8-descent to show $Ш(E/\mathbb{Q}) \supset \mathbb{Z}/8\mathbb{Z}$, which I could do for example for the elliptic curve which is referred to as 31252a1 in John Cremona's database.

7.1.1 The Elliptic Curve 1230f7

The elliptic curve $E : y^2 + xy + y = x^3 + x^2 - 14346720x - 20921901465$, which is referred to as 1230f7 in John Cremona's database, is known to have (analytical-) rank 0 and torsion subgroup $\{O, T\}$ with the 2-torsion point $T = (-8749/4, 8745/8)$. The Birch and Swinnerton-Dyer conjecture predicts that $\#Ш(E/\mathbb{Q}) = 64$, which would mean $Ш(E/\mathbb{Q}) = \mathbb{Z}/8\mathbb{Z} \times \mathbb{Z}/8\mathbb{Z}$, since the bound on the rank of E from the the 2-Selmer rank is 2.

It can be proven that the Birch and Swinnerton-Dyer conjecture is true for this curve. I learned the methods from William A. Stein at the workshop on Rational Points on Curves in Bremen, 2005. He gave a talk about the project of proving the Birch and Swinnerton-Dyer conjecture for all curves up to conductor 1000 with some additional assumptions. Our curve has conductor 1230, hence is not in this list, but the same methods can be applied:

The index of the Heegner point on the quadratic twist by -119 of E is 256, which can be computed with SAGE [23], hence by Kolyvagin's Theorem [12] $\#Ш(E/\mathbb{Q})$ must be a power of 2. Thus it remains to prove the Birch and Swinnerton-Dyer conjecture at the prime 2. By Cassels' Isogeny Theorem [4] it is enough to do that for an isogenous curve. So we look at the isogenous curves and find that the curve 1230f1 has conjecturally trivial Shafarevich-Tate group, which can be proven already by a 2-descent.

Now we know that $Ш(E/\mathbb{Q}) = \mathbb{Z}/8\mathbb{Z} \times \mathbb{Z}/8\mathbb{Z}$, but how do the curves in $Ш(E/\mathbb{Q})$ look like? Nobody has ever seen a curve of order 8 in the Shafarevich-Tate group of an elliptic curve. By an 8-descent, we can construct such a curve, and the following example illustrates the methods very nicely.

First, we start with a 2-descent on E and get three curves, one of which is

$$C_2 : y^2 = -3240x^4 + 13121x^2 - 13284.$$

Strictly speaking, we get a 2-covering $\phi_2 : C_2 \to E$.

By a second 2-descent on C_2, i.e. a 4-descent, we get four curves, one of which is

$$C_4 : \begin{cases} Q_1 := 2x_1x_2 + x_3^2 + 10x_4^2 = 0, \\ Q_2 := x_1^2 + 656x_2^2 - 162x_3x_4 = 0. \end{cases}$$

Again, strictly speaking, we get a 2-covering $\phi_4 : C_4 \to C_2$.

This is the starting position for us. We want to perform a third 2-descent on C_4, i.e. an 8-descent. For that we have to compute the descent map F.

7.1.2 The Descent Map

The étale algebra is $A = \mathbb{Q}[T]/(g(T))$ where $g = -10T^4 + 13121T^2 - 4304016$. It is isomorphic to $K_1 \times K_2$ where $K_1 := \mathbb{Q}[\theta_1] := \mathbb{Q}[T]/(T^2 - 10)$ and $K_2 := \mathbb{Q}[\theta_2] := \mathbb{Q}[T]/(T^2 + T - 10)$.

The singular quadrics in the pencil are

$$x_1^2 + 81/5\theta_1x_1x_2 + 656x_2^2 + 81/10\theta_1x_3^2 - 162x_3x_4 + 81\theta_1x_4^2 = 0$$

and

$$x_1^2 + (16\theta_2 + 8)x_1x_2 + 656x_2^2 + (8\theta_2 + 4)x_3^2 - 162x_3x_4 + (80\theta_2 + 40)x_4^2 = 0$$

and their conjugates.

Projection from the singularity of the first singular quadric leads to a conic given by $\theta_1 z_1^2 + 162z_1z_2 + 656\theta_1 z_2^2 + 81z_3^2 = 0$. Its diagonalization is $-4z_1^2 + z_2^2 - 1296\theta_1 z_3^2 = 0$. The negative of the first coefficient is a square, which gives the point $(-1/2 : 1 : 0)$ on the diagonalized conic, which maps back to the point $(-8\theta_1 : 1 : 0)$ on the original conic. The tangent line to this point is $2z_1 + 16\theta_1 z_2 = 0$, which lifts under the projection to the tangent plane $L_1 := x_1 + 8\theta_1 x_2 = 0$.

Projection from the singularity of the second singular quadric leads to a conic given by $z_1^2+(8\theta_2+4)z_2^2-162z_2z_3+(80\theta_2+40)z_3^2 = 0$. Its diagonalization is $(32\theta_2 + 16)z_1^2 + z_2^2 - 16z_3^2 = 0$. The negative of the third coefficient is a square, which gives the point $(0 : 4 : 1)$ on the diagonalized conic, which maps back to the point $(0 : 1/2(2\theta_2 + 1) : 1)$ on the original conic. The tangent line to this point is $2z_2 + (-2\theta_2 - 1)z_3 = 0$, which lifts under the projection to the tangent plane $L_2 := 2x_3 + (-2\theta_2 - 1)x_4 = 0$.

The descent map is then

$$\begin{aligned} F : C_4(\mathbb{Q}) &\to A^*/A^{*2}\mathbb{Q}^* \\ P &\mapsto (L_1(P), L_2(P)) \end{aligned}$$

7.1.3 The Fake Selmer Set

Let L_3 and L_4 be the conjugates of L_1 and L_2, then $L_1L_2L_3L_4 = 4x_1^2x_3^2 - 41x_1^2x_4^2 - 2560x_2^2x_3^2 + 26240x_2^2x_4^2$ satisfies the norm condition

$$L_1L_2L_3L_4 = cQ_3^2 \mod I(C_4)$$

for $Q_3 = x_1x_4 + 8x_2x_3$ and $c = -81$.

The set of bad primes consists of $S := \{\infty, 2, 3, 5, 41\}$, which are already the prime divisors of the discriminant of g. There are no common prime divisors of the coefficients of L_1 or L_2, and the prime divisors of c are already in S. The image of the projection of the singular subscheme of $\mathrm{Proj}(\mathbb{Z}[x_1, x_2, x_3, x_4]/(Q_1, Q_2, Q_3))$ to $\mathrm{Spec}(\mathbb{Z})$ is the ideal generated by 8501760, whose prime divisors are already contained in S.

Taking the intersection of the local images we get the fake Selmer set. It consists of the following subset of $A^*/A^{*2}\mathbb{Q}^*$

$$\{(1, -10\theta_2 - 37), (5, 10\theta_2 + 37), (5, -10\theta_2 - 37), (1, 10\theta_2 + 37)\}$$

represented by elements of $K_1 \times K_2$.

7.1.4 Representation as 2-coverings

Next, I want to show how one can represent the elements of the fake Selmer set as 2-coverings of C_4. let us take for example $\xi := (1, -10\theta_2 - 37)$. Then

ξ is a possible image of the map F, say the image of the hypothetical point $P = (x_1 : x_2 : x_3 : x_4)$. We know ξ only up to the square of an element $y = (y_1 + \theta_1 y_2, y_3 + \theta_2 y_4) \in K_1 \times K_2$. In this particular case the equation (5.1) $F(P) = \xi y^2$ reads

$$\begin{gathered} x_1 + 8\theta_1 x_2 = 1 \cdot (y_1 + \theta_1 y_2)^2 \\ 2x_3 + (-2\theta_2 - 1)x_4 = (-10\theta_2 - 37)(y_3 + \theta_2 y_4)^2 \end{gathered}$$

Since we do not know $x_1, \dots, x_4$ and $y_1, \dots, y_4$, we interpret them as variables. Multiplying out the right hand side and sorting by powers of θ_1 and θ_2 gives

$$\begin{aligned} x_1 + 8x_2\theta_1 &= y_1^2 + 10y_2^2 + 2y_1y_2\theta_1 \\ 2x_3 - x_4 - 2x_4\theta_2 &= -37y_3^2 - 200y_3y_4 - 270y_4^2 - (10y_3^2 + 54y_3y_4 + 73y_4^2)\theta_2 \end{aligned}$$

where we can read off immediately

$$\begin{aligned} x_1 &= y_1^2 + 10y_2^2, \\ x_2 &= \frac{1}{4}y_1y_2, \\ x_3 &= -16y_3^2 - \frac{173}{2}y_3y_4 - \frac{467}{4}y_4^2, \\ x_4 &= 5y_3^2 + 27y_3y_4 + \frac{73}{2}y_4^2. \end{aligned} \tag{7.1}$$

This corresponds to inverting the matrix $M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & -1 & -2 \end{pmatrix}$. In addition, $(x_1 : x_2 : x_3 : x_4)$ must be a point on C_4, thus substituting (7.1) into Q_1 and Q_2 gives two quartics

$$G_1 := \frac{1}{2}y_1^3y_2 + 5y_1y_2^3 + 506y_3^4 + 5468y_3^3y_4 + \frac{88633}{4}y_3^2y_4^2 + \frac{159631}{4}y_3y_4^3 + \frac{431249}{16}y_4^4$$

and

$$\begin{aligned} G_2 := y_1^4 + 61y_1^2y_2^2 + 100y_2^4 + 12960y_3^4 + 140049y_3^3y_4 + \frac{1135053}{2}y_3^2y_4^2 + \\ 1022139y_3y_4^3 + \frac{2761371}{4}y_4^4. \end{aligned}$$

Let $C_8 : G_1 = G_2 = 0$ in $\mathbb{P}^3$, then equations (7.1) define a rational map

$$\phi_8 : C_8 \to C_4.$$

C_8 consists of two components, which are separated by a third quartic coming from the norm condition. We have $N(\xi) = -1 = ca^2$ for $a = 1/9$ and

$N(y) = y_1^2y_3^2 - y_1^2y_3y_4 - 10y_1^2y_4^2 - 10y_2^2y_3^2 + 10y_2^2y_3y_4 + 100y_2^2y_4^2$. Substituting (7.1) into Q_3 gives $5y_1^2y_3^2 + 27y_1^2y_3y_4 + \frac{73}{2}y_1^2y_4^2 - 32y_1y_2y_3^2 - 173y_1y_2y_3y_4 - \frac{467}{2}y_1y_2y_4^2 + 50y_2^2y_3^2 + 270y_2^2y_3y_4 + 365y_2^2y_4^2$ and by Section 4.2.1 the third quadric is $G_3^{\pm} := aN(y) \pm Q_3$, hence

$$\begin{aligned} G_3^+ = \frac{46}{9}y_1^2y_3^2 + \frac{242}{9}y_1^2y_3y_4 + \frac{637}{18}y_1^2y_4^2 - 32y_1y_2y_3^2 - 173y_1y_2y_3y_4 - \\ \frac{467}{2}y_1y_2y_4^2 + \frac{440}{9}y_2^2y_3^2 + \frac{2440}{9}y_2^2y_3y_4 + \frac{3385}{9}y_2^2y_4^2. \end{aligned}$$

Then $C_8^+ : G_1 = G_2 = G_3^+ = 0$ and the 2-covering map $\phi_8^+ : C_8^+ \to C_4$ is given by the equations (7.1).

The three quartics are

$$C_8^+ : \begin{cases} 324y_1^3y_2 + 810y_1y_2^3 + 506y_3^4 + 10936y_3^3y_4 + 88633y_3^2y_4^2 + \\ \quad 319262y_3y_4^3 + 431249y_4^4, \\ 8y_1^4 + 122y_1^2y_2^2 + 50y_2^4 + 80y_3^4 + 1729y_3^3y_4 + 14013y_3^2y_4^2 + \\ \quad 50476y_3y_4^3 + 68182y_4^4, \\ 46y_1^2y_3^2 + 484y_1^2y_3y_4 + 1274y_1^2y_4^2 - 144y_1y_2y_3^2 - 1557y_1y_2y_3y_4 - \\ \quad 4203y_1y_2y_4^2 + 110y_2^2y_3^2 + 1220y_2^2y_3y_4 + 3385y_2^2y_4^2 \end{cases}$$

after the change of variables $(y_1, y_2, y_3, y_4) \to (6y_1, 3y_2, y_3, 2y_4)$ to minimize it a little bit.

Remark 7.1.1. *In this example we also see the effect of $E(\mathbb{Q})_{\text{tors}} = \mathbb{Z}/2\mathbb{Z}$. The pair of 2-coverings of C_4 corresponding to $(1, -10\theta_2 - 37)$ and the pair corresponding to $(1, 10\theta_2 + 37)$ coincide when considered as 8-coverings of E. Analogously for $(5, 10\theta_2 + 37)$ and $(5, -10\theta_2 - 37)$. Thus we get every 8-covering doubled. It would be nice to be able to remove this redundancy by dividing out the image of the 2-torsion point T under F. However, I do not know how to compute $F(T)$ explicitly.*

7.2 $Ш(E/\mathbb{Q})[2^\infty] = (\mathbb{Z}/4\mathbb{Z})^2$

With the following example I want to demonstrate, how one can use 8-descent to prove that there are no elements of order 8 in the Shafarevich-Tate group. For example, if the 2-primary part $Ш(E/\mathbb{Q})[2^\infty]$ of the Shafarevich-Tate group is conjectured to be $\mathbb{Z}/4\mathbb{Z} \times Z/4\mathbb{Z}$, then we can prove that with an 8-descent.

I documented the computations for the curves, which are referred to as 1309a1, 2045b1, and 2738c1 in John Cremona's database. For these curves we are dependent on 8-descent and could not argue with properties of isogenous curves, since there are none. However, our algorithm also works for curves which admit isogenies such as the smallest example with $Ш(E/\mathbb{Q}) = \mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$, which is the curve referred to as 210e7.

7.2.1 The Elliptic Curve 1309a1

The elliptic curve $E : y^2 + y = x^3 - 406957x - 99924251$, which is referred to as 1309a1 in John Cremona's database, is known to have (analytical-) rank 0 and trivial torsion subgroup. The Birch and Swinnerton-Dyer conjecture predicts that $\#Ш(E/\mathbb{Q}) = 16$, which would mean $Ш(E/\mathbb{Q}) = \mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$, since the bound on the rank of E from the 2-Selmer rank is 2.

This is an example of an elliptic curve where 2- and 4-descent do not suffice. Also arguments involving isogenous curves do not work, since there are none. However, with an 8-descent we can prove the Birch and Swinnerton-Dyer conjecture at 2 for E.

First, we start with a 2-descent on E and get three 2-coverings of E, one of which is

$$C_2 : y^2 = -49x^4 + 602x^3 - 564x^2 - 7896x - 8428.$$

By a second 2-descent on C_2, i.e. a 4-descent, we get two 2-coverings of C_2, one of which is

$$C_4 : \begin{cases} Q_1 := 2x_1x_4 + x_2^2 + 2x_2x_4 + x_3^2 + x_4^2 = 0, \\ Q_2 := 20x_1^2 + 54x_1x_2 + 30x_1x_3 - 42x_1x_4 - 7x_2^2 \\ \qquad - 10x_2x_3 + 16x_2x_4 + 9x_3^2 - 8x_3x_4 + 27x_4^2 = 0. \end{cases}$$

Now we want to perform a third 2-descent on C_4, i.e. an 8-descent. The étale algebra is $A = \mathbb{Q}[T]/(g(T))$ where $g = -T^4 + 94T^3 - 1104T^2 - 51952T - 305388$. It is isomorphic to $K_1 := \mathbb{Q}[\theta_1] = \mathbb{Q}[T]/(T^4 + 2T^3 - 8T^2 - 26T - 18)$. The singular quadrics in the pencil are $20x_1^2 + 54x_1x_2 + 30x_1x_3 + (22\theta_1^3 + 14\theta_1^2 - 216\theta_1 - 294)x_1x_4 + (11\theta_1^3 + 7\theta_1^2 - 108\theta_1 - 133)x_2^2 - 10x_2x_3 + (22\theta_1^3 + 14\theta_1^2 - 216\theta_1 - 236)x_2x_4 + (11\theta_1^3 + 7\theta_1^2 - 108\theta_1 - 117)x_3^2 - 8x_3x_4 + (11\theta_1^3 + 7\theta_1^2 - 108\theta_1 - 99)x_4^2 = 0$ and its conjugates.

Projection from the singularity leads to a conic given by $20z_1^2 + 54z_1z_2 + 30z_1z_3 + (11\theta_1^3 + 7\theta_1^2 - 108\theta_1 - 133)z_2^2 - 10z_2z_3 + (11\theta_1^3 + 7\theta_1^2 - 108\theta_1 - 117)z_3^2 = 0$. Its diagonalization is $(880\theta_1^3 + 560\theta_1^2 - 8640\theta_1 - 13556)z_1^2 + z_2^2 + (136320\theta_1^3 + 39040\theta_1^2 - 1218560\theta_1 - 1277440)z_3^2 = 0$. With Denis Simon's program `bnfqfsolve2` we find a point on it, which maps back to the point

$$\begin{aligned}(&-327434629966864\theta^3 - 279867784717716\theta^2 + 2939980077503565\theta + \\ &5146136309357024 \quad : \quad 109893505280293\theta^3 + 939104294026 42\theta^2 - \\ &9867930085508830\theta - 1727233694529083 \quad : \quad 306266215)\end{aligned}$$

on the original conic. Computing the tangent line to this point and lifting it under the projection gives the tangent plane

$$\begin{aligned} L_1 := (&-5676932556749437779\theta^3 - 20174071343161103321\theta^2 - \\ &21375728011753406000\theta + 10798550890055184 4) \, x_1 + \\ &(-4633338859802249972\theta^3 + 3353847492410018862\theta^2 + \\ &5811516661189486480\theta + 7792437403088126307) \, x_2 + \\ &1163029638512734945 \, x_3 + \\ &(2627985464363720840\theta^3 + 1100443093653600886\theta^2 - \\ &5388201851648867760\theta - 3764560453921021559) \, x_4 = 0. \end{aligned}$$

The third quadric in the pencil is

$$\begin{aligned} Q_3 := 2974787x_1^2 &+ 1424897x_1x_2 + 11362687x_1x_3 + 10342334x_1x_4 - \\ &1042924x_2^2 + 1138781x_2x_3 - 15355230x_2x_4 + 82454564x_3^2 - \\ &174019395x_3x_4 - 71385852x_4^2 = 0. \end{aligned}$$

The constant in $L_1L_2L_3L_4 = cQ_3^2$ is

$$c = -6292632057862009541394165597481327549166800563779834500$$

with factorization

$$2^2 \cdot 5^3 \cdot 64063^3 \cdot 3630893459603^3.$$

The discriminant of g is

$$\operatorname{disc}(g) = -11585215152896 = 2^8 \cdot 7^6 \cdot 11^3 \cdot 17^2.$$

The primes dividing the norms of the coefficients of L_1 are 5, 64063, and 3630893459603. Thus the set S of bad primes contains

$$\{\infty, 2, 5, 7, 11, 17, 64063, 3630893459603\}$$

and the primes p such that $P_8 := \operatorname{Proj}(\mathbb{Z}[x_1, \ldots, x_4]/(Q_1, Q_2, Q_3))$ is singular mod p. Theoretically, we can compute these primes as the prime divisors of the image of the projection of the singular subscheme of P_8 to $\operatorname{Spec}(\mathbb{Z})$, which is the ideal in $\mathbb{Z}$ generated by the 143 digit number

$$\begin{aligned} n := \; &75159089103525444849915608075230855073526033044987421582864 \\ &35168814308487932598175699881640740495922365158373377662844 \\ &30158166736216370370173750. \end{aligned}$$

However, in practice, we do not want to factor this number, especially, since the Bad Primes Hypothesis, see Section 4.3, says that we will not need it.

7.2.2 Verification of the Bad Primes Hypothesis for this example

With a trick we can avoid factoring this number and prove the Bad Primes Hypothesis for this particular example. For that, we take a different point on the conic to get another tangent plane

$$\begin{aligned}L_1' = {} & 93079924743617373664135184505043332\ x_1 + \\ & (-41587985210458895159952073693180025\theta_1^3 - \\ & 20307757545255174240267496519066603\theta_1^2 + \\ & 32664012310589324654637327822198631\theta_1 + \\ & 61558499476696659628173494903549206)\ x_2 + \\ & (15662281851490880343949871470707779\theta_1^3 - \\ & 32716973387200611029474313447730027\theta_1^2 - \\ & 93994311189907780851432833833033325\theta_1 + \\ & 10072085949969677685351150373055026)\ x_3 + \\ & (54967339636437132331079467174959906\theta_1^3 + \\ & 63518618311504679700280051683964447\theta_1^2 - \\ & 55024974987172814572137348711016743\theta_1 - \\ & 88706035846619387824366088271941562)\ x_4 = 0.\end{aligned}$$

We can apply the whole procedure to this L_1', too, and compute the set of bad primes corresponding to L_1' up to the point where we would have to factor again a large number. Instead of factoring it, we take the lowest common multiple of all the numbers involving bad primes – the unfactored version of the bad primes so to say – and get

$$\begin{aligned}s := \quad & 2679925772324413869637835756726706136126531605557792245364\,1 \\ & 016855527271370694207162288014835527565102997400492269311\,47 \\ & 3196196761342332912742882845604325486425278397223250994051\,7 \\ & 0309416515034623336416870257519459726367818054226491920813\,4 \\ & 0633299147931374966441175612207550884958289558593619622478\,0 \\ & 5380456964233614061692555220935498989095253050007367244921\,33 \\ & 67008629813651760632493\,6\end{aligned}$$

Now we have an even larger number, which we cannot factor. So why did we do that? The reason is, that we easily can compute the greatest common divisor of n and s, which we expect to be very small, hence easy to factor. And we only need the greatest common divisor, since the set of bad primes S

and S' coming from L_1 and L_1' respectively almost coincide. More precisely, by Lemma 3.3.1 we have $L_1/L_1' = \gamma$ modulo squares for some $\gamma \in K_1$. Hence S and S' coincide up to prime divisors of the Norm of γ, which is in this example

$$\begin{aligned}\gamma = -&52813263270376477895841325683230370441672627558662 78\theta_1^3 - \\ &187801236497559908079276826516233166315394073579919 72\theta_1^2 - \\ &199152335620595788213699295346701374846640430181796 00\theta_1 + \\ &990601659077279637788663075017675360032404708127808\end{aligned}$$

and $N(\gamma) \in \mathbb{Z}$. The prime divisors of n that contribute to S are at most the ones dividing the greatest common divisor of n and $sN(\gamma)$, which is 3925156600750. Now this number can easily be factored and all its prime factors are already contained in the set S which we got in the beginning.

Thus we showed that in this example the Bad Primes Hypothesis is true.

7.2.3 The Fake Selmer Set

Only four elements of $A(S, 2)/\mathbb{Q}(S, 2)$ fulfill the norm condition $N(\xi) = c$ mod squares. These four are killed by the local solvability condition at the prime 7, hence $\mathrm{Sel}^{(2)}_{\mathrm{fake}}(C_4/\mathbb{Q})$ is empty.

The same procedure can be applied to the other five 4-coverings of E, which would show that the 2-primary part of $\text{Ш}(E/\mathbb{Q})$ is $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$.

7.3 Searching Points

In this example I want to sketch how 8-descent might be used for finding large points on elliptic curves in future. let us consider the elliptic curve $E : y^2 = x^3 + 7823$. By analytical methods one can show that this curve has rank one, but for a long time the generator was not known. In 2002 Stoll could find that point by a 4-descent [25]. At that time already a method for minimizing the intersection of two quadrics had been developed by Cremona and Womack, but still the coefficients were too large for searching points. What was missing was a method of reduction, which Stoll could find.

Nowadays by a 4-descent on E we get a 4-covering with such small coefficients

$$C_4 : \begin{cases} Q_1 := 4x_1x_3 + 2x_1x_4 + x_2^2 + 4x_2x_4 - 2x_3^2 - 3x_4^2 = 0, \\ Q_2 := 4x_1^2 + 4x_1x_2 - 2x_1x_4 - x_2^2 - 2x_2x_4 - 2x_3^2 - 4x_3x_4 - 3x_4^2 = 0, \end{cases}$$

where we can find the point $(-681 : 539 : 116 : 125)$, which is considerably smaller than the corresponding point on the elliptic curve

$$\left(\frac{2263582143321421502100209233517777}{1435604977061909894854751519047 21}, \frac{186398152584623305624837551485596770028144776655756}{172009499810635335582100852593872795015977704348 1}\right).$$

The hope would be that another descent would further decrease the size of the point. So let us see what happens in a third 2-descent on C_4. The étale algebra is $A = \mathbb{Q}[T]/(g(T))$ where $g = 30T^4 + 12T^3 + 48T^2 - 116T - 18$. It is isomorphic to K_1 where $K_1 = \mathbb{Q}[T]/(15T^4 + 6T^3 + 24T^2 - 58T - 9)$. The singular quadrics in the pencil are $4x_1^2+4x_1x_2+4\theta_1x_1x_3+(2\theta_1-2)x_1x_4+(\theta_1-1)x_2^2+(4\theta_1-2)x_2x_4+(-2\theta_1-2)x_3^2-4x_3x_4+(-3\theta_1-3)x_4^2$ and their conjugates. Projection from the singularity leads to a conic, whose diagonalization is $(\theta_1 - 2)z_1^2 + z_2^2 + (-\theta_1^3 - \theta_1^2 + 2\theta_1 + 4)z_3^2$, on which we find a point with PARI/GP, which gives the tangent plane

$$\begin{aligned} L_1 := (&11900746487965000919915845350319881 0\theta_1^3- \\ &104477661987579603938632558377243336\theta_1^2+ \\ &105830717941159207323452731728268512\theta_1- \\ &642132239542681995498267662234672324)x_1+ \\ &(-368227970632471301668834754386164475\theta_1^3- \\ &155466273498067346450910280389246645\theta_1^2- \\ &598139387095368415715628557988732287\theta_1+ \\ &144525463721993358988021050264665985)x_2+ \\ &153058241129297835007003601141550 38x_3+ \\ &(-1070924087148514009079854271395572 60\theta_1^3- \\ &779682412974894335062858190452396689\theta_1^2- \\ &222673380399635009234741999908819785\theta_1+ \\ &319064911672182946221143886843892598)x_4 = 0 \end{aligned}$$

to the singular quadric.

The fake Selmer set consists of one element and the corresponding curves C_8^+ and C_8^- are given by

$$\begin{aligned} &-3035457906432394527910314701198744881140586892865520907358497246687971778 6 \\ &\quad 750621051643191959509221199996607184716117197257867424177051701\ x_1^4- \\ &\quad 74026521278365504445821390054117181809595945671464335832561539986150056898 \\ &\quad 7170073386307256377370724058363357435157835420841965126641274427/5\ x_1^3x_2- \\ &\qquad \dots \end{aligned}$$

where all the equations for $C_8^{\pm}$ and $\phi_8^{\pm} : C_8^{\pm} \to C_4$ would fill 15 pages. Thus this is not yet useful for searching points on $C_8^{\pm}$. However, we can have a look at the preimage of the point $(-681 : 539 : 116 : 125)$ of $C_4(\mathbb{Q})$ under ϕ_8^+ and ϕ_8^-. Under ϕ_8^+ it does not have a preimage in $C_8^+(\mathbb{Q})$, and under ϕ_8^- it has the preimage

$$\begin{aligned}(-&40985242083886589123010215324996749209563 :\\ &15639696777308000017162343540481421033935 :\\ &25368172243562802579394405343624028741O3 :\\ &10007084434492659884411645351813419277520)\end{aligned}$$

in $C_8^-(\mathbb{Q})$. The philosophy of descent tells us that there should be a much nicer model of C_8^- such that this point is even smaller than $(-681 : 539 : 116 : 125)$. So there should be much space for minimization and reduction, however, there is not yet a theory for minimizing such curves.

Chapter 8

Directions for Further Work

8.1 Improving the Implementation

There are some parts of the implementation that could be improved. First, it would be nice to have a Magma program for finding a point on a conic over a number field, which is as strong as Denis Simon's program.

Next, local solvability at ∞ should also be tested. I guess that one can adopt part of Nils Bruin's code for this. For local solvability at a finite prime p, I do not yet test the intersection of the tangent planes with C_4 for possible p-adic points. Using this, one might speed up the program. Another suggestion for speeding up the program is to store the local points already when doing 4-descent. Then one would not have to recompute them in the 8-descent.

Another point is that the whole routine depends on the choice of the tangent planes. It would be worth to have a method to find nice ones. Some experiments with different tangent planes corresponding to different points on the conic might be helpful.

8.2 Bad Primes Hypothesis

The set S of bad primes for an 8-descent depends on the choice of the tangent planes and often involves very large primes. The worst primes come from the last defining condition of S. However, in practice one can often avoid these primes, so I guess that they are superfluous. That is what I call the *Bad Primes Hypothesis*. It would be nice to have a proof for that. Or, at least it would be interesting to study how one can improve the conditions on S.

8.3 Minimization

When 4-descent [15] came up there was no theory for minimizing and reducing the intersection of two quadrics. So the curves obtained by a 4-descent had large coefficients and were not so useful for searching points on them. It took a few years until Michael Stoll could use a 4-descendent to find the generator of the elliptic curve $E : y^2 = x^3 + 7823$.

Minimizing the model of a genus 1 curve means to adjust the equations so that the invariants are small preferably the same as for the Jacobian. This of course begs the questions: What are the invariants, and how do we compute them? On these two questions Tom Fisher has done some work in a very general setting. He seems to be able to define the right invariants and knows a method to compute them. Even if this can be done, it is not clear how one should then go about minimizing. However, I expect this to be solved in the near future and then it makes sense to try to find points on 8-descendents.

8.4 9- and 16-Descent

Since 3-descent is now feasible one could try to do a second 3-descent, i.e. a 9-descent. This could be done by similar methods as the ones described in this thesis. One would have to replace the tangent planes to C_4 by tangent lines through the flex points of the plane cubic model of the 3-descendent. It would be interesting to work out the details to be able, to do some examples.

Also for 16-descent there is a perspective. One possibility is to try to do a fourth 2-descent on an 8-covering $C_8 \to E$. For that, one could try to find hyperplanes that are tangent to C_8. Another possibility could be to try to do a whole 4-descent on C_4 at once. By that I mean a method to construct 4-coverings of C_4 without first constructing 2-coverings of C_4. The idea for that is to take the hyperosculating points on C_4, see Section 2.4 and take tangent planes at them to get a descent map. The advantage of this method is that it is very easy to construct the descent map, in contrast to our 8-descent, where we have to find a point on a conic over a number field. However, the étale algebra A corresponding to the hyperosculating points is generically a number field of degree 16. The descent map would then be $C_4(\mathbb{Q}) \to A^*/A^{*4}\mathbb{Q}^*$ defined by the tangent plane to the generic hyperosculating point. Recall that the tangent plane meets the hyperosculating point four times, that is why we divide by fourth powers of A^*. For computing the fake Selmer set in $A^*/A^{*4}\mathbb{Q}^*$ one needs to be able to compute $A(S, 4)$. I have no experience how difficult this will get in practice; it would be interesting to see this.

8.5 The Big Goal

The long term perspective could be to find a proof for the finiteness of the Shafarevich-Tate group. I guess that the methods of higher and higher descent can contribute to that goal. So far we can do higher descent only at the prime 2, and only a second and a third 2-descent. So even for the 2-primary part of the Shafarevich-Tate group our knowledge of elements of higher order is very limited. It would be very helpful to see examples of even higher descent to study the general structure of further descents. A remarkable observation is that the coefficients of the 2^k-coverings get smaller and smaller, at least in the case $k = 1$ or 2. For 8-descendents there is no method of minimization yet, however I expect them to follow this trend as soon as one can minimize them. I guess that making this observation precise could be an important tool for proving the finiteness of the Shafarevich-Tate group.

Appendix A

Further Programs

```
// The following attachments are needed for LocalImageOfC2.
Attach("utils.m");
Attach("local_quartic1.m");
// The following attachment is needed for
// speeding up LocalTwoSelmerMap at large primes.
Attach("Mymynumfld.m");

AddAttribute(Crv, "EtaleAlgebra");

/*****************************************************************
MyIntegralModel just multiplies through with the common denominator.
*****************************************************************/

function MyIntegralModel(X)
  // X a scheme over a number field.
  // We make it integral by just multiplying through
  // with the lcm of the denominators of the coefficients.
  pols := DefiningPolynomials(X);
  K := BaseField(X);
  if Type(K) eq FldNum then
    OK := Integers(K);
    // The ideal c*OK is a fractional ideal. We take its denominator.
    d := [LCM([Denominator(c*OK) : c in Coefficients(f)]) : f in pols];
  elif Type(K) eq FldRat then
    n := [GCD([Numerator(c) : c in Coefficients(f)]) : f in pols];
    d := [LCM([Denominator(c) : c in Coefficients(f)]) : f in pols];
    d := [d[i]/n[i] : i in [1..#pols]];
  else
    error "Base field must be a number field or the rationals.";
  end if;
  return Scheme(Ambient(X), [pols[i]*d[i] : i in [1..#pols]]);
end function;

function MySize(X)
```

```
  // X is a scheme defined by one polynomial.
  OK := Integers(BaseField(X));
  c := ChangeUniverse(Coefficients(DefiningPolynomial(X)),OK);
  idl := ideal<OK|c>;
  return Norm(idl);
end function;

/**********************************************************************
We divide by one of the coefficients and look whether the integral model
is nicer than the other ones.
**********************************************************************/

function ImprovedIntegralModel(X)
  P3 := Ambient(X);
  L := DefiningPolynomial(X);
  coeffs := Coefficients(L);
  _,j := Min([MySize(MyIntegralModel(Scheme(P3,L/c))) : c in coeffs]);
  return MyIntegralModel(Scheme(P3,L/coeffs[j]));
end function;


/*******************************************************
Diagonalization of a conic (only in the generic case):
*******************************************************/

function diag(C)
  // C is a conic.
  P2<[z]> := AmbientSpace(C);
  q := DefiningPolynomial(C);
  a := MonomialCoefficient(q,z[1]^2);
  b := MonomialCoefficient(q,z[1]*z[2]);
  c := MonomialCoefficient(q,z[1]*z[3]);
  d := MonomialCoefficient(q,z[2]^2);
  e := MonomialCoefficient(q,z[2]*z[3]);
  f := MonomialCoefficient(q,z[3]^2);
  assert a*d*f ne 0;
  assert (4*a*d - b^2) ne 0;
  assert (4*a*d*f - a*e^2 - b^2*f + b*c*e - c^2*d) ne 0;
  fdiag := (4*a*d - b^2)*z[1]^2 + z[2]^2 +
  4*a*(4*a*d*f - a*e^2 - b^2*f + b*c*e - c^2*d)*z[3]^2;
  // The map between the conics:
  Cdiag := Scheme(P2,fdiag);
  phi := map< C -> Cdiag |
  [2*a*z[1] + b*z[2] + c*z[3],
  (4*a*d - b^2)*z[2] + (2*a*e - b*c)*z[3], z[3]]>;
  return phi;
end function;
```

```
/*********************************************************************
Finding a point on a conic by diagonalizing it and using NormEquation.
The trivial cases are done before. The parameter Point can be set to a
point on C, which we computed in advance, e.g. with PARI.
*********************************************************************/

function PointOnConic(C : Point := Point)
  P2<[z]> := Ambient(C);
  q := DefiningPolynomial(C);
  a := MonomialCoefficient(q,z[1]^2);
  b := MonomialCoefficient(q,z[1]*z[2]);
  c := MonomialCoefficient(q,z[1]*z[3]);
  d := MonomialCoefficient(q,z[2]^2);
  e := MonomialCoefficient(q,z[2]*z[3]);
  f := MonomialCoefficient(q,z[3]^2);
  // First we check whether the parameter Point is on C:
  if Point ne [] and Point in C then
    pt := Point;
  // If there is no trivial point on C, we diagonalize it.
  // First the trivial points:
  elif a eq 0 then
    bool, pt := [1,0,0] in C; assert bool;
  elif d eq 0 then
    bool, pt := [0,1,0] in C; assert bool;
  elif f eq 0 then
    bool, pt := [0,0,1] in C; assert bool;
  elif (4*a*d - b^2) eq 0 then
    bool, pt := [b,-2*a,0] in C; assert bool;
  elif (4*a*d*f - a*e^2 - b^2*f + b*c*e - c^2*d) eq 0 then
    bool, pt := [2*c*d-b*e,2*a*e-b*c,b^2-4*a*d] in C; assert bool;
  else
    // Now the nontrivial point (using diagonalization):
    diagmap := diag(C);
    Cdiag := Codomain(diagmap);
    vprintf EightDescent, 1 : "Its diagonalization is %o.\n",
    DefiningPolynomial(Cdiag);
    // Next, we check whether the parameter Point is on Cdiag:
    if Point ne [] then
      bool, ptdiag := Point in Cdiag;
      if not bool then // PARI output.
        K := BaseField(C);
        g := DefiningPolynomial(K);
        lc := LeadingCoefficient(g);
        // PARI needs monic polynomials for number fields.
        if lc ne 1 then
          g1<y> := MinimalPolynomial((K.1+1)*lc);
          K1<y> := NumberField(g1);
          m := hom<K->K1|(y/lc)-1>;
        else // lc eq 1
```

```
          g1<y> := MinimalPolynomial(K.1);
          K1<y> := NumberField(g1);
          m := hom<K->K1|K1.1>; // identity.
        end if;
        bool, ptdiag := [Reverse(x) @@ m : x in Point] in Cdiag;
        error if not bool, "Runtime error in PointOnConic:
        \nOptional parameter Point is not on the diagonalized conic.";
      end if;
    else
      ptdiag := PointOnDiagonalizedConic(Cdiag);
      vprintf EightDescent, 1 :
      "which gives the point %o on the diagonalized conic, ", ptdiag;
    end if;
    pt := Points(ptdiag @@ diagmap)[1];
    vprintf EightDescent, 1 :
    "which maps back to the point %o on the original conic.\n", pt;
  end if;
  return C!pt;
end function;


/**********************************************************************
Transforming a diagonal conic into the data that Denis Simon's PARI file
ell.gp (available at his web page) needs. The output is written to the
file conic.gp.
**********************************************************************/

function InputForPARI(C)
  // C is a conic over a number field
  // with defining polynomial a*x[1]^2 + x[2]^2 + b*x[3]^2.
  K := BaseField(C);
  g := DefiningPolynomial(K);
  lc := LeadingCoefficient(g);
  // PARI needs monic polynomials for number fields.
  if lc ne 1 then
    g1<y> := MinimalPolynomial((K.1+1)*lc);
    K1<y> := NumberField(g1);
    m := hom<K->K1|(y/lc)-1>;
  else // lc eq 1
    g1<y> := MinimalPolynomial(K.1);
    K1<y> := NumberField(g1);
    m := hom<K->K1|K1.1>; // identity.
  end if;
  C1 := BaseChange(C,m);
  q := DefiningPolynomial(C1);
  a := MonomialCoefficient(q,C1.1^2);
  b := MonomialCoefficient(q,C1.3^2);
  SetOutputFile("conic.gp" : Overwrite := true);
  printf "\\r ell.gp\n";
  printf "\\p 300 \n";
  printf "allocatemem(100 000 000) \n";
  printf "{g = %o}\n",g1;
  // Braces {..} are needed if the output is longer than one line.
  printf "{a = Mod(%o , g)}\n",a;
  printf "{b = Mod(%o , g)}\n",b;
  printf "s = bnfqfsolve2(bnfinit(g), -a, -b)\n";
  printf "[Vec(lift(s[2])),Vec(lift(s[1])),[0,0,0,1]] \n";
  printf "Str(%%,\";\")\n";
  printf "\\w solution.m \n";
  UnsetOutputFile();
  //print "Look into file conic.gp";
  // Remark: One could also use the function bnfqfsolve,
  // but then the third coefficient of the output is not 1.
  // Advantage: evtl. nicer solution.
  // Disadvantage: might take much longer.
  return m;
end function;


/**********************************************************************
DifferentTangentPlane computes a different tangent plane to the
singular quadric using parameterization of the points on the conic.
**********************************************************************/

function DifferentTangentPlane(Qsing1, L1 : PointOnP1:=[])
  // Qsing1 a singular quadric in the pencil with tangent plane L1.
  // We parametrize the conic, take a different point on it, and
  // compute the corresponding tangent plane.
  K := BaseField(Qsing1);
  C, pr := ConicOfSingularQuadric(Qsing1); // recomputed (does not matter).
  pt := Points(pr(Scheme(Qsing1,L1)))[1]; // The original point.
  param := Parametrization(Curve(C),pt);
  if PointOnP1 eq [] then
    PointOnP1 := [Random(K,10),1];
  end if;
  newpt := param(PointOnP1);
  line := TangentLine(newpt);
  t11 := line @@ pr;
  return DefiningPolynomial(ImprovedIntegralModel(t11));
end function;


/***********************************************************************
BadPrimes
***********************************************************************/

function BadPrimesUnfactored(C4,L,Q3)
  // Discriminant(g) is taken extra.
  c := TheConstant(C4,L,Q3);
  badc := Numerator(c)*Denominator(c);
  // Bad primes (unfactored) from coeffs:
  badcoeffs := LCM([GCD([Integers()!Norm(c) : c in Coefficients(Li)]) :
  Li in L]);
  // Bad primes from P8:
  P8 := ChangeRing(Scheme(C4,Q3),Integers());
  badP8 := ProjectionToSpecZ(SingularSubscheme(P8));
  vprintf EightDescent,4: "badP8 = %o.\n", badP8;
  bad := LCM([badc,badcoeffs,badP8]);
  return bad;
end function;

/**********************************************************************
The local image
**********************************************************************/

/**************************************
At very bad primes or small primes:
***************************************/

function MakeIntegral(pt)
  // pt is a p-adic point on a projective scheme.
  // We just scale it and
  // return an integral point (as sequence) with j-th coordinate = 1.
  min, j := Min([Valuation(pt[i]) : i in [1..#Eltseq(pt)]]);
  integralpt := [pt[i]/pt[j] : i in [1..#Eltseq(pt)]];
  assert integralpt in Scheme(pt); // over Zp:
  integralpt := ChangeUniverse(integralpt,Integers(Parent(pt[1])));
  return integralpt, j;
end function;

function InverseOfProjection(C4)
  C,pr := Projection(C4);
  phi := map<C4->C|DefiningEquations(pr)>;
  bool, inv := IsInvertible(phi);
  error if not bool, "Projection from (1:0:0:0) does not induce a
  birational map between C4 and its projection to the plane.
  We would have to project from a different point.";
  return inv;
end function;

// Local solvability test by projecting to P2 and taking the preimage.

function IsLocallySolvableByProjectionToPlaneCurve(C4,p)
  C := Curve(Projection(C4));
  bool, pt := IsLocallySolvable(C,p : Smooth);
  pt := LiftPoint(pt,50 : Strict:=false);
  inv := InverseOfProjection(C4);
  liftedpt := [Evaluate(f, Eltseq(pt)) : f in DefiningEquations(inv)];
  // same as inv(pt) (but over local field).
  return liftedpt in C4;
end function;

function ImageOfOnePointAtVeryBadOrSmallPrime(C4,F,p)
  bool, pt := IsLocallySolvableByProjectionToPlaneCurve(C4,p);
  // might get smashed by the way we take the preimage
  // under the projection.
  if not bool then
    vprint EightDescent,1:
    "ATTENTION: IsLocallySolvableByProjectionToPlaneCurve did not work.";
    bool, pt := IsLocallySolvable(C4,p);
    pt := LiftPoint(pt, 50 : Strict := false);
  end if;
  pti := MakeIntegral(pt);
  prec := Min([Precision(pti[i]) : i in [1..#pti]]);
  v := Valuation(Integers()!Norm(F(pti)),p);
  if (p eq 2) and (v le prec-10) then
    return F(pti), pti;
  elif (p ne 2) and (v lt prec) then
    return F(pti), pti;
  else
    error "ERROR: Lifted ",p,"-adic point to precision ", prec,
    ", which was not enough.";
    // TODO: automatically enlarging precision.
  end if;
end function;
```

Bibliography

[1] S. Y. An, S. Y. Kim, D. C. Marshall, S. H. Marshall, W. G. McCallum, and A. R. Perlis, Jacobians of Genus One Curves, *J. Number Th.* **90** (2001), 304-315.

[2] B. J. Birch and H. P. F. Swinnerton-Dyer, Notes on elliptic curves I, *J. reine angew. Math.* **212** (1963), 7-25.

[3] B. J. Birch and H. P. F. Swinnerton-Dyer, Notes on elliptic curves II, *J. reine angew. Math.* **218** (1965), 79-108.

[4] J. W. S. Cassels, Arithmetic on curves of genus 1. VIII. On conjectures of Birch and Swinnerton-Dyer, *J. reine angew. Math.* **217** (1965), 180-199.

[5] J. W. S. Cassels, Lectures on Elliptic Curves, London Math. Soc. Student Texts, Cambridge University Press, 1991.

[6] J. W. S. Cassels, Second descents for elliptic curves, *J. reine angew. Math.* **494** (1998), 101-127.

[7] H. Cohen, A Course in Computational Algebraic Number Theory, Springer, Berlin, 1993.

[8] J. E. Cremona, Algorithms for Modular Elliptic Curves, 2$^{\text{nd}}$ ed., Cambridge University Press, 1997.

[9] J. E. Cremona, T. A. Fisher, C. O'Neil, D. Simon, and M. Stoll, Explicit n-descent on elliptic curves I, *preprint.*

[10] J. R. Goldman, The Queen of Mathematics, A K Peters, 1998.

[11] V. G. Lopez Neumann, Descente explicite pour les Jacobiennes des courbes de genre 2, *preprint.*

[12] V. A. Kolyvagin, Euler systems, The Grothendieck Festschrift, Vol. II, Birkhäuser Boston, Boston, MA, 1990, pp. 435-483.

[13] B. Mazur, Modular curves and the Eisenstein ideal, *IHES Publ. Math.* **47** (1977), 33-186.

[14] B. Mazur, Rational isogenies of prime degree, *Invent. Math.* **44** (1978), 129-162.

[15] J. R. Merriman, S. Siksek and N. P. Smart, Explicit 4-descents on an elliptic curve, *Acta Arith.* **77** (1996), 358-404.

[16] L. J. Mordell, On the rational solutions of the indeterminate equations of 3rd and 4th degrees, *Proc. Camb. Phil. Soc.* **21** (1922), 179-192.

[17] B. Poonen and E. F. Schaefer, Explicit descent for Jacobians of cyclic covers of the projective line *J. reine angew. Math.* **488** (1997), 141-188.

[18] E. F. Schaefer, Computing a Selmer group of a Jacobian using functions on the curve, *Math. Ann.* **310** (1998), 447-471.

[19] E. F. Schaefer and M. Stoll, How to do a p-descent on an elliptic curve, to appear in *Trans. Amer. Math. Soc.* **356** (2004), 1209-1231.

[20] S. Siksek, Descents on curves of genus 1, PhD Thesis, Exeter, 1995.

[21] J. H. Silverman, The Arithmetic of Elliptic Curves, Springer, 1986.

[22] D. Simon, Computing the rank of elliptic curves over number fields, *LMS J. Comput. Math.* **5** (2002), 7-17.

[23] W. Stein, D. Joyner, SAGE: System for Algebra and Geometry Experimentation, *Comm. Computer Algebra* **39** (2005), to appear.

[24] M. Stoll, Implementing 2-descent for Jacobians of hyperelliptic curves, *Acta Arith.* **98** (2001), 245-277.

[25] M. Stoll, Explicit 4-descent on an elliptic curve, *unpublished.*

[26] M. Stoll, Descent on elliptic curves, Lecture notes available at http://212.201.48.1/stoll/talks/short-course-descent.pdf

[27] A. Weil, Sur un théorème de Mordell, *Bull. Sci. Math. (2)* **54** (1930), 182-191.

[28] A. Weil, Number Theory: an approach through history from Hamurapi to Legendre, Birkhäuser, Boston, 2001.

[29] T. Womack, Explicit descent on elliptic curves, PhD Thesis, Nottingham, 2003.