MATHEMATICS OF COMPUTATION
Volume 00, Number 0, Pages 000–000
S 0025-5718(XX)0000-0

# FACTORIZATION WITH GENUS 2 CURVES

ROMAIN COSSET

ABSTRACT. The elliptic curve method (ECM) is one of the best factorization methods available. It is possible to use hyperelliptic curves instead of elliptic curves but it is in theory slower. We use special hyperelliptic curves and Kummer surfaces to reduce the complexity of the algorithm. Our implementation GMP-HECM is faster than GMP-ECM for factoring big numbers.

## INTRODUCTION

The elliptic curve method (ECM) introduced in 1985 by H. W. Lenstra, Jr. [6] plays an important role in factoring integers. ECM is used to find "medium sized" (up to 60 digits) prime factors of "random" numbers. It is also used by other factoring algorithms like sieving methods for cofactoring [5]. ECM is a generalization of Pollard's $p-1$ algorithm: instead of working in $\mathbb{F}_p^*$, we work in the group of points on an elliptic curve. We generalize it by using hyperelliptic curves of genus 2 instead of elliptic curves of genus 1.

At first sight the "hyperelliptic curve method" (HECM) seems slower that ECM because of two reasons: first the arithmetic of hyperelliptic curves is slower compared to the arithmetic of elliptic curves, and secondly its probability of success is smaller than that of ECM.
Indeed this probability depends on the probability of the cardinality of the Jacobian of the curve modulo a prime factor $p$ of $n$ being smooth. But the probability of a number being smooth decreases with its size. By Weil's theorem, the cardinality of a Jacobian of a genus 2 curve on $\mathbb{F}_p$ is around $p^2$ whereas the cardinality of an elliptic curve over the same field is around $p$. This seems like a major deterrent to using HECM. To avoid it we use special hyperelliptic curves whose Jacobians are isogenous to the product of two elliptic curves: they are called decomposable. One run of HECM with this kind of curves is equivalent to two simultaneous runs of ECM, thus the probability of success of HECM with decomposable hyperelliptic curves is comparable to that of two ECM.
The complexity of stage 1 is dominated by the cost of the arithmetic on the curves. In genus 1, the quickest arithmetic is obtained by using Montgomery formulæ [8] or the new Edwards curves [1]. In genus 2, the best known formulæ use Kummer surfaces [3]. The Kummer surface is a variety obtained by identifying opposite points of the Jacobian. However, not all genus 2 hyperelliptic curves map to a rational Kummer surface.

2000 *Mathematics Subject Classification.* Primary 11Y05; Secondary 11Y16, 11Y40.

The curves used in HECM are built as reduction modulo $n$ of curves defined over $\mathbb{Q}$ which are selected for satisfying different constraints. We need a large number of possible trial curves in order to factor $n$, hence the requirement that there exists an infinite family of curves over $\mathbb{Q}$. This requires some work; finally we obtain a parametrization of a subfamily of hyperelliptic curves: the parameters live on an elliptic curve over a function field.
We have implemented our algorithm by using many functions of GMP-ECM a free ECM program. During the computation, there are many multiplications by parameters of the hyperelliptic curves so the use of small parameters (i.e. which fit into one machine word) makes these multiplications negligible before the cost of full-length modular multiplications. This makes our software faster than GMP-ECM.

In Section 1 we give some facts about ECM, decomposable genus 2 hyperelliptic curves and Kummer surfaces. Details of the parametrization are given in Section 2 while Section 3 focuses on the HECM algorithm. In Section 4 we describe our implementation and give some numerical results.

In the following, $n$ denotes the number to be factored.

## 1. Background

1.1. **ECM.** Following a common notation abuse in factoring algorithms, we work over $\mathbb{Z}/n\mathbb{Z}$ as if it were a field. The only operation that might fail is "field" inversion which is calculated using the Euclidean algorithm. If an inversion fails, we find a factor of $n$.

The ECM method starts by choosing a random elliptic curve $\mathcal{E}$ over $\mathbb{Z}/n\mathbb{Z}$ and a point $P$ on it. Let $k$ be a positive integer, we compute $Q = [k]P$ and we hope $Q$ to be the zero of the curve modulo a prime $p$ dividing $n$ but not modulo $n$. If this is the case, then one division fails during the computation and we find the factor $p$. This happens if the cardinality $|\mathcal{E}(\mathbb{F}_p)|$ divides $k$. There are two different phases (also called stages or steps) in ECM: in phase 1 we hope that the cardinality is $B_1$-smooth (i.e. multiple of prime powers less than $B_1$) for some bound $B_1$. In phase 2 we try to cover the "close-miss" case where the cardinality is a product of a $B_1$-smooth part by a prime cofactor. If this fails, we take another elliptic curve. ECM is a probabilistic algorithm whose complexity is $O(L(p)^{\sqrt{2}+o(1)}M(\log(n)))$ with $L(p) = e^{\sqrt{\log(p)\log(\log(p))}}$ and $M(\log(n))$ is the complexity of multiplications modulo $n$. The complexity of ECM is dominated by the size of the smallest factor $p$ of $n$ rather than the size of the number $n$ to be factored. Note however that ECM does not always find the smallest factor.

In stage 1 we hope that the cardinality of the elliptic curve is $B_1$-smooth. We take $k = \prod_{\pi \leq B_1} \pi^{[\log(B_1)/\log(\pi)]} = \mathrm{lcm}(1, 2, \ldots, B_1)$ so that all $B_1$-smooth numbers divide $k$. The main cost of stage 1 is the arithmetic; different methods are used to reduce this cost. We focus on ECM with Montgomery formulæ [8] since it shares several traits with the HECM method which is the subject of this work. Let $\mathcal{E}$ be an elliptic curve in Montgomery's form: $y^2 = x^3 + a\,x^2 + x$. We use projective coordinates $(x : y : z)$ to avoid divisions which are expensive. Moreover we disregard the $y$-coordinate which means that we identify a point with its opposite. It is still possible to "double" a point (i.e. computing $\pm 2P$) and if the difference $\pm(P-Q)$ between two points is known then we can compute their sum $\pm(P+Q)$: this operation is called pseudo-addition. To compute the multiple of a point we need to find chains of doubling and pseudo-additions which is a special case of addition chains called "Lucas Chains" [7]. For instance

$$1 \to 2 \to 3 \to 4 \to 7 \to 10 \to 17$$

is a chain for 17. One way to find such chains is to note that if we know $[n]P$ and $[n+1]\,P$ then we can compute $[2n]\,P$, $[2n+1]\,P$ and $[2n+2]\,P$. Hence we have binary chains: at each step we choose the point to double according to the binary expansion of the multiplier. For instance the following chain is the binary chain for 17:

$$1 \to 2 \to 3 \to 4 \to 5 \to 8 \to 9 \to 17.$$

Binary chains are not the shortest chain of doubling and pseudo-addition. Montgomery's PRAC algorithm finds short Lucas chains [7].

A description of stage 2 falls out of scope of the present article since for reasons which will be explained later, stage 1 is the only phase done by HECM.

1.2. **Decomposable curves.**

**Definition 1.1.** A genus 2 hyperelliptic curve $C$ is said to be decomposable if there exists an isogeny $\phi$ from $Jac\,(\mathcal{C})$ to the product of two elliptic curves $\mathcal{E}_i$:

$$\phi : \quad Jac\,(\mathcal{C}) \longrightarrow \mathcal{E}_1 \times \mathcal{E}_2.$$

If the kernel of the isogeny is $\mathbb{Z}/k\mathbb{Z} \times \mathbb{Z}/k\mathbb{Z}$ we say that the curve is $(k,k)$-decomposable.

There are characterizations for a curve to be $(k,k)$-decomposable [11]. We will only look at $(2,2)$-decomposable curves since the conditions are the simplest.

**Theorem 1.2.** *Let $\mathcal{C}$ be a hyperelliptic curve of genus 2 given by the equation*

$$y^2 = x\,(x-1)\,(x-\lambda)\,(x-\mu)\,(x-\nu)$$

*where $\lambda$, $\mu$, $\nu$ are the Rosenhain invariants of the curve. $\mathcal{C}$ is $(2,2)$-decomposable if and only if its Rosenhain invariants are linked by*

$$\lambda = \mu\frac{1-\nu}{1-\mu}.$$

The proof is given in [4]. The two underlying elliptic curves have equation

$$\chi\, y^2 = (x-1)\left(x-x_2^2\right)\left(x-x_3^2\right)$$

with

$$q = \pm\sqrt{\mu\,(\mu-\nu)}, \quad x_2 = \frac{\mu+q}{\mu-q},\; x_3 = \frac{1-\mu-q}{1-\mu+q},\; \chi = -q\,\mu\,(\mu-1)\,.$$

Over non algebraically closed fields $K$, the Rosenhain invariants may be non rational. Thus a $(2,2)$-decomposable curve can have a different equation. If $\mathcal{C}$ is a $(2,2)$-decomposable curve in Rosenhain form then the two underlying elliptic curves are rational if and only if $\mu\,(\mu-\nu)$ is a square in $K$.

The morphisms between the hyperelliptic curves and the elliptic curves are defined on $K(q)$. They are given by

$$(x,y) \longrightarrow \left(\left(\frac{x-\mu-q}{x-\mu+q}\right)^2, \frac{w\,y}{(x-\mu+q)^3}\right) \text{ with } w = \frac{8q}{(\mu-q)(-1+\mu-q)}.$$

1.3. **Kummer surfaces.** Gaudry proposes in [3] to use theta functions to perform the arithmetic in the Jacobian of some genus 2 curves. Working with theta functions allows the use of the numerous classical formulæ that were found using complex analysis: the reader is referred to Mumford's books [9, 10] for details. In fact all the formulæ we need are algebraic and can be used on finite fields (of characteristic different from 2).

Let $\Omega$ be a matrix in the Siegel half space of dimension 2. The theta functions are functions from $\mathbb{C}^2/\left(\mathbb{Z}^2+\Omega\mathbb{Z}^2\right)$ to $\mathbb{C}$. There are 16 theta functions with half-integer characteristic: 10 are even and 6 are odd. The scalars obtained by evaluating them in $z=(0,0)$ are called theta constants. The main difference with Gaudry's article is that we use the squares of the coordinates. We are principally interested in only four squares of theta functions and the corresponding square theta constants $(\alpha:\beta:\gamma:\delta)$.

A Kummer surface $\mathcal{K}_{(\alpha:\beta:\gamma:\delta)}$ can be defined by the choice of four "theta constants" $(\alpha:\beta:\gamma:\delta)$ in $\mathbb{P}^3(\mathbb{C})$. We write $(X:Y:Z:T)$ for the coordinates of a point on $\mathcal{K}_{(\alpha:\beta:\gamma:\delta)}$. The equation of the Kummer surface is:

$$\begin{aligned} 4E'^2\alpha\beta\gamma\delta XYZT &= \left(\left(X^2+Y^2+Z^2+T^2\right)-\right.\\ &\qquad \left.-F(XT+YZ)-G(XZ+YT)-H(XY+ZT)\right)^2 \end{aligned}$$

where $E'$, $F$, $G$, $H$ are constants that can be calculated from $(\alpha:\beta:\gamma:\delta)$ by the formulæ given in the appendix.

Let $\frac{\epsilon}{\phi}$ be the ratio of two theta constants which can be calculated from the theta constants $(\alpha:\beta:\gamma:\delta)$. Let

$$\lambda := \frac{\alpha\gamma}{\beta\delta}, \qquad \mu := \frac{\gamma\epsilon}{\delta\phi}, \qquad \nu := \frac{\alpha\epsilon}{\beta\phi},$$

$$\mathcal{C}: \qquad y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu).$$

Then $Jac(\mathcal{C})/\{\pm1\}$ is isomorphic to $\mathcal{K}_{(\alpha:\beta:\gamma:\delta)}$ over $\mathbb{C}$:

$$\psi: \qquad Jac(\mathcal{C})/\{\pm1\} \longrightarrow K_{(\alpha:\beta:\gamma:\delta)}$$

Over non algebraically closed fields, let $\tilde{\mathcal{C}}$ be the quadratic twist of $\mathcal{C}$. Then $Jac(\mathcal{C})/\{\pm1\}$ and $Jac(\tilde{\mathcal{C}})/\{\pm1\}$ are included in $\mathcal{K}_{(\alpha:\beta:\gamma:\delta)}$:

$$\mathcal{K}_{(\alpha:\beta:\gamma:\delta)} = \psi\left(Jac(\mathcal{C})/\{\pm1\}\right)\bigcup\tilde{\psi}\left(Jac\left(\tilde{\mathcal{C}}\right)/\{\pm1\}\right)$$

with the 2-torsion points shared by the two Jacobians. Usually, we use the Mumford coordinates $(u,v)$ on $Jac(\mathcal{C})$, from a point $P$ in $K_{(\alpha:\beta:\gamma:\delta)}$, it is possible to calculate $u$ and $v^2$ of its image by the morphism [3, 12]. Of course $v$ is defined up to its sign since we can't distinguish between one point and its opposite. This generalizes the idea of Montgomery's formulæ for elliptic curves.

The arithmetic on $Jac(C)$ transports to arithmetic on $\mathcal{K}_{(\alpha:\beta:\gamma:\delta)}$. Given a point $P=\psi(D)$ on the Kummer surface it is possible to double it (i.e. to compute $\psi([2]D)$): see algorithm 1. Given two points on a Kummer surface $P=\psi(D)$ and

**Algorithm 1** Doubling on a Kummer surface

**Input:** a point $P = (X : Y : Z : T)$ on $\mathcal{K}_{(\alpha:\beta:\gamma:\delta)}$.
**Output:** the point $[2]P = (X_2 : Y_2 : Z_2 : T_2)$.

$$\begin{array}{rclrcl} X' &=& (X+Y+Z+T)^2 \frac{1}{A}, & Y' &=& (X+Y-Z-T)^2 \frac{1}{B}, \\ Z' &=& (X-Y+Z-T)^2 \frac{1}{C}, & T' &=& (X-Y-Z+T)^2 \frac{1}{D} \\ X_2 &=& (X'+Y'+Z'+T')^2 \frac{1}{\alpha}, & Y_2 &=& (X'+Y'-Z'-T')^2 \frac{1}{\beta}, \\ Z_2 &=& (X'-Y'+Z'-T')^2 \frac{1}{\gamma}, & T_2 &=& (X'-Y'-Z'+T')^2 \frac{1}{\delta} \end{array}$$

**Algorithm 2** Pseudo-addition on a Kummer surface

**Input:** two points $P = (X : Y : Z : T)$ and $Q = (\underline{X} : \underline{Y} : \underline{Z} : \underline{T})$ on $\mathcal{K}_{(\alpha:\beta:\gamma:\delta)}$, and the point $R = \left(\bar{X} : \bar{Y} : \bar{Z} : \bar{T}\right)$ equal to $P - Q$ such that $\bar{X}\bar{Y}\bar{Z}\bar{T} \neq 0$.
**Output:** the point $P + Q = (x : y : z : t)$.

$$\begin{array}{rcl} X' &=& (X+Y+Z+T)(\underline{X}+\underline{Y}+\underline{Z}+\underline{T}) \frac{1}{A}, \\ Y' &=& (X+Y-Z-T)(\underline{X}+\underline{Y}-\underline{Z}-\underline{T}) \frac{1}{B}, \\ Z' &=& (X-Y+Z-T)(\underline{X}-\underline{Y}+\underline{Z}-\underline{T}) \frac{1}{C}, \\ T' &=& (X-Y-Z+T)(\underline{X}-\underline{Y}-\underline{Z}+\underline{T}) \frac{1}{D} \\ x &=& (X'+Y'+Z'+T')^2 \frac{1}{\bar{X}}, \\ y &=& (X'+Y'-Z'-T')^2 \frac{1}{\bar{Y}}, \\ z &=& (X'-Y'+Z'-T')^2 \frac{1}{\bar{Z}}, \\ t &=& (X'-Y'-Z'+T')^2 \frac{1}{\bar{T}} \end{array}$$

$Q = \psi(D')$, we can't add them since we don't know whether we must compute $\psi(D + D')$ or $\psi(D - D')$. However if one of these quantities is known then it is possible to compute the other, this is called a pseudo-addition (algorithm 2). We note $[2]P$ and $P + Q$ these two operations. Note that these formulæ do not hold when one coordinate is zero.

We work with projective coordinates so divisions can be replaced by multiplications. Moreover the constants $1/\alpha$, $1/\beta$, $1/\gamma$, $1/\delta$, $1/A$, $1/B$, $1/C$ and $1/D$ can be precomputed. The cost of pseudo-addition is 4 divisions, 4 multiplications, 4 squares and 4 multiplications by constants (thereafter denoted $4I + 4M + 4S + 4d$) or if the divisions are replaced by multiplications the cost becomes $14M + 4S + 4d$. The cost of doubling is $8S + 8d$. Note that by using the properties of projective coordinates, it is possible to save some multiplications [3].

For the multiplication algorithm we want to avoid divisions since they are costly. This imposes that the inverses of the coordinates of $P - Q$ are always known when we want to pseudo-add $P$ and $Q$. Thus it is impossible to use "short" Lucas chains and the PRAC algorithm, we must use binary chains where $P - Q$ is always the initial point (algorithm 3). For each bit of the multiplier $k$, we do one doubling and one pseudo-addition on the Kummer surface. The total cost of a multiplication by $k$ is $7M + 9S + 9d$ per bit of $k$ since some computations can be shared between the two operations (it is $12M + 4S + 16d$ if we don't share them).

*Remark* 1.3. The coordinates of the initial point $P$ affect the cost of computation. For instance if two coordinates of $P$ are equal or opposite then we gain, for each

**Algorithm 3** Multiplication algorithm

**Input:** a point $P$ on $\mathcal{K}_{(\alpha:\beta:\gamma:\delta)}$ and an integer $k$.
**Output:** the point $[k]P$.
**if** $k = 2$ **then**
  **return** $[2]P$
**else**
  Let $k = \sum_{i=0}^{l} k_i 2^{l-i}$ be the binary expansion of $k$ with $k_0 = 1$ the most significant bit.
  $P_m = P$; $P_p = [2]P$;
  **for** $i$ from 2 to $l$ **do**
    $Q = P_p + P_m$ {note that we have $P_p - P_m = P$}
    **if** $k_i = 1$ **then**
      $P_p = [2]P_p$; $P_m = Q$;
    **else** $\{k_i = 0\}$
      $P_m = [2]P_m$; $P_p = Q$;
    **end if**
  **end for**
  **return** $P_m$
**end if**

bit of the multiplier, one multiplication and another one which in fact is a square. This is a speed up of 5%.

## 2. Parametrization

In this Section, we find suitable curves for HECM. We obtain curves over $\mathbb{Q}$ having the desired properties and consider their reduction modulo $n$. We want to have an infinite number of curves over $\mathbb{Q}$ so that we have enough curves over $\mathbb{Z}/n\mathbb{Z}$. Note that throughout the construction, we should never compute a square root because of two reasons: first this square root could be in an extension of $\mathbb{Q}$, thus when reducing modulo $n$ we could be working in an extension of $\mathbb{Z}/n\mathbb{Z}$ and the arithmetic would be slower. Moreover, computing square roots modulo the composite integer $n$ is equivalent to find factors of $n$. To sum up, we require that all constants are defined by rational functions in several variables.

### 2.1. Parametrization of the hyperelliptic curves.

**Lemma 2.1.** *Let $\mathcal{C}$ be a genus 2 curve over $\mathbb{Q}$ with equation*

$$\mathcal{C}: \qquad \chi y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu).$$

*$\mathcal{C}$ can be used in HECM (i.e. $\mathcal{C}$ is $(2,2)$-decomposable with rational underlying elliptic curves and the rational Kummer surface) if and only if*

$$\lambda = \mu\frac{1-\nu}{1-\mu}, \qquad \mu(\mu-\nu) = \square, \qquad \lambda\mu\nu = \square$$

*where $\square$ means that the quantity must be a square.*

*Proof.* The first two conditions mean that $\mathcal{C}$ is $(2,2)$-decomposable with rational underlying elliptic curves. The Kummer surface is rational if and only if the squares

$(\alpha : \beta : \gamma : \delta)$ of the four theta constants are rational. They are linked with the Rosenhain invariants by:

$$\lambda = \frac{\alpha\gamma}{\beta\delta}, \qquad \mu = \frac{\gamma\epsilon}{\delta\phi}, \qquad \nu = \frac{\alpha\epsilon}{\beta\phi}$$

where $\epsilon$ and $\phi$ are two other squares of theta constants whose ratio is rational when $(\alpha : \beta : \gamma : \delta)$ are rational. Note that this implies that the Rosenhain invariants must also be rational so it gives a justification to the choice of the equation of the hyperelliptic curve. From these three equations, we obtain:

$$\frac{\alpha}{\beta} = \frac{\sqrt{\lambda\mu\nu}}{\mu}, \qquad \frac{\gamma}{\delta} = \frac{\sqrt{\lambda\mu\nu}}{\nu}.$$

So the product $\lambda\mu\nu$ must be a square in $\mathbb{Q}$.
When we write the equation $\lambda = \mu\frac{1-\nu}{1-\mu}$ in terms of theta constants (use the formulæ from [4] and [12]), it yields to $\alpha^2 = \delta^2$. Moreover $\beta$, $\gamma$, $\delta$ are functions of $\lambda$, $\mu$, $\nu$ and $\alpha$ and thus are rational. This proves that the conditions are sufficient. $\square$

*Remark* 2.2. The Rosenhain invariants $(\lambda, \mu, \nu)$ are defined up to the action of the group $PGL(2, 5)$ see [4]. Our choice here is one that leads to an equality between two of the first four theta constants. With a different choice of order, we would have had another square root to handle. Moreover the fact that $\alpha^2 = \delta^2$ is a main advantage for the arithmetic: it saves two scalar multiplications per multiplier's bit when computing the multiple of a point.

Since $(\alpha : \beta : \gamma : \delta)$ live in $\mathbb{P}^3(\mathbb{Q})$, we can take $\alpha = 1$. We choose also $\delta = \alpha = 1$. The choice $\delta = -\alpha = -1$ leads to an isomorphic Kummer surface.
Take $\mu = 1 - \frac{\nu(1-\nu)}{s^2}$ with $s \in \mathbb{Q}$ so that $\lambda\mu\nu = \mu^2 s^2$. The second equation becomes

$$\mu\left(\mu - \nu\right) = \frac{1}{s^4}\left(-\nu + \nu^2 + s^2\right)\left(\nu - 1\right)\left(\nu - s^2\right) = \square.$$

Assume that $\frac{\nu - s^2}{\nu - 1}$ is a square $u^2$ (then $\nu = \frac{s^2 - u^2}{1 - u^2}$) so that the equation above rewrites as:

$$1 + (-3/s^2 + 1/s^4)u^2 + u^4/s^2 = \square.$$

Let $v^2$ be the square, the point $(u, v)$ lies on an elliptic curve over $\mathbb{Q}(s)$. This elliptic curve is in the Jacobi model [2], is of rank 1 and we have a non torsion point $P = \left(1, 1 - \frac{1}{s^2}\right)$ on it.

**Theorem 2.3.** *A subfamily of* $(2, 2)$*-decomposable hyperelliptic genus* 2 *curves with rational underlying elliptic curves and with rational Kummer surfaces is given by the following parametrization:*

$$\mathcal{C}: \qquad \chi y^2 = x\left(x - 1\right)\left(x - \lambda\right)\left(x - \mu\right)\left(x - \nu\right),$$

$$\lambda = \mu\frac{1-\nu}{1-\mu}, \qquad \mu = 1 - \frac{\nu\left(1-\nu\right)}{s^2}, \qquad \nu = \frac{s^2 - u^2}{1 - u^2}$$

*where* $(u, v)$ *lies on the following elliptic curve*

$$1 + (-3/s^2 + 1/s^4)u^2 + u^4/s^2 = v^2.$$

*A non-torsion point on this curve is* $(1, 1 - \frac{1}{s^2})$*. The parameters* $s$*,* $\chi$*,* $u$*,* $v$ *are rational numbers that must verify the following conditions.*

**Genericity condition 2.4.** With the notations of the theorem, the curve $\mathcal{C}$ is of genus 2 if and only if 0, 1, $\infty$, $\lambda$, $\mu$ and $\nu$ are distinct and $\chi$ is not zero. This is equivalent to

$$\chi \neq 0, \qquad s \neq 0, \pm 1, \qquad u \neq 0, \pm 1, \qquad v \neq 0,$$
$$s \neq \pm u, \qquad s^2 - 2u^2 + u^4 \neq 0.$$

In particular, condition 2.4 implies that we can't use the point $(1, 1 - \frac{1}{s^2})$ on the Jacobi curve. We must take a multiple of this point: for instance its double $(2, 1 + \frac{2}{s^2})$.

**Genericity condition 2.5.** For the arithmetic on the Kummer surface, all the theta constants must be non zero. The parameters must satisfy condition 2.4 and

$$s \neq \pm u^2.$$

The parameter $\chi$ determines whether we are on the curve or on its quadratic twist. However it is not chosen during the parametrization of the curve: since the curve and its twist are both included in the same Kummer surface, the choice of a point on it determines whether we work on the curve or on its twist. In practice, the $y$-coordinate of the point is never used, so $\chi$ is never computed.

We could have taken $\frac{\nu - s^2}{\nu - 1} = l\, u^2$ (here $l = 1$) but that would have led to more complicated equations. For instance with $l = -1$, we found an elliptic curve with no point on it. In this case ($l = -1$), we had to assume that $s$ was on a conic for the curve to become of rank 1.

*Remark* 2.6. ECM supposes that the cardinality of elliptic curves on $\mathbb{F}_p$ behaves like a random integer of size around $p$ with some additional divisibility conditions. In HECM, we work simultaneously with two elliptic curves. Their cardinalities are not independent: points of 4-torsion cannot exist on only one curve. However, experimentally, it seems that this is the only noticeable link between their cardinalities.

2.2. **Finding a point on the Kummer surface.** In ECM we need a generic point (i.e. a non torsion point) in the group. Contrary to the Brent-Suyama parametrization for elliptic curves, it is possible here to find a point after having chosen a hyperelliptic curve. However, the initial point $P$ must be a non torsion point of the two underlying elliptic curves over $\mathbb{Q}$ and not only of the Jacobian of the hyperelliptic curve. Remember that we don't want to take square roots or to work in an extension field so we can't take three random coordinates and solve the equation of the Kummer surface to find the last one.

The first method to find a point is to take one coordinate to be zero. Then the three other lie on a conic. Since conics are birationally equivalent to $\mathbb{P}^1$, we have an infinite number of points on the Kummer surface. In general, these points are not torsion points. Note however that these points cannot be used directly for the multiplication algorithm: we must double them until we find a point with no zero coordinate; in general, one doubling is enough.
There is a better solution to find points: we look for points with two equal or opposite coordinates. Such points have the advantage of saving multiplications (see remark 1.3): the cost of the multiplication algorithm becomes $12M + 10S$

per bit of the multiplier. It turns out that not all choices of coordinate pairs are possible: some choices lead only to torsion points or to impossible equations like $-1 = \square$. The choice $Y = -X$ yields to the equation of an elliptic curve with many points on it. These points lead to points on the Kummer surface which are of infinite order (over $\mathbb{Q}$). For instance we use the following point:

$$\begin{aligned} X &= \frac{u^2\left(u^2-1\right)\left(su^4-3su^2+u^2+s^3-s^2+s\right)}{(s-u^2)s^4v^2}, & Y &= -X \\ T &= \frac{\left(s^2-u^2\right)\left(u^2-1\right)}{s^4v^2}, & Z &= 1. \end{aligned}$$

## 3. Algorithm and ameliorations

3.1. **HECM.** In ECM, the big product $k = \prod_{\pi \leq B_1} \pi^{[\log(B_1)/\log(\pi)]}$ is not computed as such [13]. Instead, for each prime $\pi$ we compute $l = [\log(B_1)/\log(\pi)]$ and then we do $l$ times $Q = [\pi]Q$ and go to the next prime. This avoids the computation of a big integer product and PRAC finds shorter chains if it works with prime multipliers [7]. If we work with Kummer surfaces we can't use PRAC because changing the initial point is costly (we would have divisions). Moreover the advantage of choosing a good initial point is lost if we don't use binary Lucas chains. Therefore in HECM we begin by computing $k = \prod_{\pi \leq B_1} \pi^{[\log(B_1)/\log(\pi)]}$. In theory this computation is costly but it practice it is negligible: we use product tree and fast multiplication. Moreover this computation is shared for many curves.

We hope to encounter the zero of one of the underlying elliptic curves so it is important to have explicit morphisms between the Kummer surface and the two underlying elliptic curves. In fact, it is impossible to obtain real points on the curves since we can't distinguish between a point and its opposite and we don't even know if we are on the curves or on their twists. If the elliptic curves are in the Weierstrass form, we only need the coordinates $(x :: z)$ to test if the point is zero: just compute $\gcd(z, n)$. The morphisms between the Kummer surface and the underlying elliptic curves (identified with their quadratic twists) are rational over $\mathbb{Q}$. However their building blocks are defined over an extension field of $\mathbb{Q}$: they use square roots that disappear in the global morphisms. Since we are not allowed to use square roots we must rewrite the global morphisms. As a result their expressions become complicated however, they are computed only once for every run of HECM, thus they are negligible before the cost of computing $[k]P$. Note that we only need the $(x :: z)$ coordinates which are invariant under the isomorphism from the curve (in the Weierstrass form) to its twist. See the appendix for a description of the different elementary blocks. The first stage of the HECM algorithm is summarized in algorithm 4.

We now turn to what is called stage 2. The initial point for the arithmetic is $Q = [k]P$, thus we don't have the benefit of a "good" initial point. Moreover, stage 2 needs the $x$-coordinate of many points on the elliptic curve in Weierstrass form [13], therefore if we use hyperelliptic curves we need to apply morphisms a lot and, in this case, their cost would not be negligible. In addition to that, if we want to use Brent-Suyama's extension, we can't use pseudo-additions but real additions. For all these reasons, it seems that hyperelliptic curves should not be used for stage 2. Instead, we can apply the ECM stage 2 to the two underlying elliptic curves.

**Algorithm 4** HECM (stage 1)

**Input:** the number $n$ to be factor. The smoothness bound $B_1$.
**Output:** a factor $p$ of $n$.

1: Compute $k = \operatorname{lcm}(1, 2, \ldots, B_1)$.
2: Choose a random decomposable curve $\mathcal{C}$ over $\mathbb{Z}/n\mathbb{Z}$ and a point $P$ on its Kummer surface.
3: Compute $Q = [k]P$.
4: Map $Q$ to the two underlying elliptic curves $\mathcal{E}_i$.
5: Hope that $Q = \mathcal{O}_{(\mathcal{E}_i)} \mod p$ for one $\mathcal{E}_i$ (test whether $\gcd(z, n) \neq 1$).
6: Else go to 2.

3.2. **Torsion of the curves.** To improve the probability of success, we can force the group order of the elliptic curves modulo primes $p$ to be divisible by small numbers. Mazur's theorem states that the torsion group $E_{tor}(\mathbb{Q})$ of any elliptic curve over $\mathbb{Q}$ is isomorphic to one of the following groups:

$$E_{tor}(\mathbb{Q}) \cong \begin{cases} \mathbb{Z}/m\mathbb{Z} & 1 \leq m \leq 10 \textit{ or } m = 12 \\ \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2m\mathbb{Z} & 1 \leq m \leq 4 \end{cases}$$

In our case the underlying elliptic curves have at least four 2-torsion points which means that $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ is a subgroup of the torsion group.

In theory it should be possible to find decomposable hyperelliptic curves with underlying elliptic curves having larger torsion group. However this leads to more complicated equations for the parametrization. There is also a theoretical obstruction to improving the torsion: to have a torsion point of order more than 2 means to be able to find a starting point on the Kummer surface corresponding to a point on $Jac(\mathcal{C})$ and not on its twist $Jac(\tilde{\mathcal{C}})$. This theoretical obstruction could be removed if the curve and its twist had the same torsion. However this increases the number of equations needed for the parametrization.

Using the parametrization presented in Section 2, we have table 1 where letters indicate whether and where 4-torsion points exist: $C$ for the curve and $T$ for its quadratic twist. With the same probability, we work on the curve or on its twist. Suppose that the Legendre symbols of $s^2 - u^2$, $s^2 - 1$, $u^2 - 1$ are independent which experimentally is a valid approximation. Then if $p \equiv 1$ [4] we have a point of 4-torsion with probability 1/2, and if $p \equiv 3$ [4], the probability is 3/4. Computer experiments show that for $p \equiv 3$ [4] the power of two dividing the cardinality is, on average, 3.48 (instead of 3.5) and for $p \equiv 1$ [4] it is 3.15 (instead of 3).

Experimentally, the order of the curves is as likely to be smooth as a random integer about 1/7.75 in value (compared with 1/23.7 for Suyama's curves [13]). The parametrization $s = \frac{3+t^2}{3-t^2}$, $u = 2$, $v = 1 + \frac{2}{s^2}$ provides better torsion (the power of two is on average 3.66) but there are few such curves with low parameters.

3.3. **Small parameters.** For a chosen Kummer surface and a chosen initial point, the arithmetic on Kummer surfaces uses many multiplications by fixed parameters. Remember from section 1.3 that the cost of a scalar multiplication is $4M+12S+16d$ per bit of the multiplier; i.e., 4 multiplications, 12 squares and 16 multiplications by constants. Suppose that these constants are small in comparison to the number $n$ to

Table 1. 4-torsion points on the underlying elliptic curves

| $s^2-u^2$ | $s^2-1$ | $u^2-1$ | $-1$: $\square$ | $-1$: $\boxtimes$ |
|---|---|---|---|---|
| $\square$ | $\square$ | $\square$ | $C$ | $C$ $T$ |
| | | $\boxtimes$ | $C$ | $C$ $T$ |
| | $\boxtimes$ | $\square$ | $C$ | |
| | | $\boxtimes$ | $T$ | $C$ $T$ |
| $\boxtimes$ | $\square$ | $\square$ | $T$ | $C$ $T$ |
| | | $\boxtimes$ | $C$ | |
| | $\boxtimes$ | $\square$ | $C$ | $C$ $T$ |
| | | $\boxtimes$ | $C$ | $C$ $T$ |

be factored, then the cost of multiplications by them will be negligible with respect to the cost of the full-length multiplications. The cost per bit of the multiplier becomes $4M+12S$ and if we use the rule of thumb $1S=0.8M$ then it is $13.6M$. For one run, GMP-ECM (which uses Montgomery coordinates and the PRAC algorithm [7]) uses approximately $6M+3S$. Thus two runs of ECM cost $12M+6S$ or $16.8M$ with $S=0.8M$. This is a speed-up of 20%. Of course this is only valid for large $n$. EECM (ECM with Edwards curves) uses signed sliding window to compute $[k]P$ (see [1]). These chains use 1 elliptic curve doubling and $\epsilon$ elliptic curve additions for each bit of $k$ where $\epsilon$ converges to 0 when $k$ increases. Doubling in Edwards coordinates uses only $3M+4S$, while additions use $10M+1S+1d$. In theory, two runs of EECM would be faster than one run of HECM if $\epsilon$ is less than $1/12$ (if $1S=0.8M$ then $\epsilon$ should be less than $1/20$). However this requires that the width of the window be very large and thus precomputations and memory usage might be not negligible. The authors of [1] claim that with $B_1=16384$, EECM uses 195111 multiplications in stage 1 (with a width-6 signed sliding window). For the same $B_1$, HECM uses 379334 multiplications but since it does two curves in parallel, it uses only 189667 multiplications for stage 1 for one elliptic curve. Therefore, more experimentations are needed but to our knwolegde there is no public implementation of EECM.

By a small constant we mean a number which fits into a signed long. This constraint limits the number of curves we can use: if we work with a 64 bits processor there are $185,399$ useful hyperelliptic curves which is sufficient for finding factors of more than 65 digits. In the multiplication algorithm, the $16d$ could be reduced to $12d$ since we work in the projective space but we would have rational numbers on $\mathbb{Q}$ that, modulo $n$, must each fit into a long which is more complicated.
More explicitly, our parametrization gives $(\alpha:\beta:\gamma:\delta)$ in terms of rational functions of small degree in $(s,u)\in\mathbb{Q}^2$. The same is true for $(1/A:1/B:1/C:1/D)$. We work in projective coordinates so we can clear the denominators. The constants used in the multiplications become:

$$\left(\frac{1}{\alpha}:\frac{1}{\beta}:\frac{1}{\gamma}:\frac{1}{\delta}\right)=\left(s^4v^2:\ s^5v^2:\ s\left(u^2-s^2\right)\left(u^2-1\right):\ s^4v^2\right),$$

$$\begin{aligned}\left(\frac{1}{A}:\frac{1}{B}:\frac{1}{C}:\frac{1}{D}\right) \quad = \quad & \Big((s-1)^2\left(s^2-2u^2+u^4\right)\left(s+u^2\right)^2 : \\ & \left(u^2-1\right)\left(s+u^2\right)^2\left(s-u^2\right)^2 : \\ & -\left(u^2-1\right)\left(s+u^2\right)^2\left(s-u^2\right)^2 : \\ & (s+1)^2\left(s^2-2u^2+u^4\right)\left(s-u^2\right)^2\Big).\end{aligned}$$

Note that $s$ is rational so we take $s = a/b$ for integers $a$,$b$. After generating a hyperelliptic curve, we test if these constants are small. We then need to find a point on the Kummer surface for which the inverses of its coordinates are small. Unfortunately we have found no point such that the polynomial expressions for the inverses of the coefficients have degree equal or lower than the degree of the constants above. In practice, we have several generic points so we check whether those points fit, if not we try with another curve.

3.4. **A numerical example.** Let's try to factor $n = 4,816,415,081$ with HECM. We use $B1 = 25$ and $B2 = 200$. Let's use the following parameters: $s = \frac{1}{2}$, $u = 2$ and $v = 9$. The Kummer surface is given by the parameters

$$(\alpha : \beta : \gamma : \delta) = \left(1 : 2 : \frac{9}{10} : 1\right).$$

One point on the Kummer surface is

$$P = (-272 : 272 : 63 : -140).$$

We compute $[k]\,P$ on the Kummer surface with $k = \operatorname{lcm}(2, B1) = 26,771,144,400$ and send the resulting point on the two underlying elliptic curves:

$$(3455587574, 1) \text{ on } 734346861\, y^2 = (x-1)\left(x-\frac{1}{25}\right)\left(x-\frac{1}{121}\right),$$

$$(3222355131, 1) \text{ on } 2791313056\, y^2 = (x-1)(x-25)(x-121).$$

We now use the ECM stage 2 on the two elliptic curves. The first one does not produce any factor but the second one produces the number 83003 which is a factor of $n = 4816415081 = 58027 * 83003$.
We inspected the calculus and found that we didn't work on the hyperelliptic curve but on its twist:

$$\chi y^2 = x(x-1)\left(x-\frac{9}{20}\right)\left(x-\frac{9}{4}\right)\left(x-\frac{5}{4}\right)$$

where $\chi$ is a non quadratic residue. The order of the initial point in the Jacobian of this curve modulo 83003 is $2\cdot3\cdot5\cdot11\cdot19\cdot73\cdot631$. Its order on the first elliptic curve is $2\cdot7\cdot631$ and on the second $2\cdot3\cdot5\cdot19\cdot73$.

## 4. Implementations and results

There are many implementations of ECM: GMP-ECM is a free program based on the GNU MP library. It is described in detail in [13]. By using many functions from GMP-ECM, we have built a new program called GMP-HECM which uses hyperelliptic curves. GMP-HECM will be distributed with GMP-ECM (currently it is distributed with the development version at http://gforge.inria.fr/projects/ecm/). Our software does stage 1 by generating a genus 2 hyperelliptic curve with small parameters, computes $[k]\,P$ on the Kummer surface and send the resulting point

TABLE 2. Optimal choice of the parameters for ECM and HECM

| Size of the factor | GMP-ECM Optimal $B_1$ | GMP-ECM Expected number of elliptic curves | GMP-HECM Optimal $B_1$ | GMP-HECM Default $B_2$ | GMP-HECM Expected number of elliptic curves |
|---|---|---|---|---|---|
| $10^{20}$ | $11,000$ | 74 | $14,000$ | $2.10^6$ | 75 |
| $10^{25}$ | $50,000$ | 214 | $60,000$ | $16.10^6$ | 214 |
| $10^{30}$ | $250,000$ | 430 | $260,000$ | $130.10^6$ | 491 |
| $10^{35}$ | $1.10^6$ | 904 | $1.10^6$ | $900.10^6$ | $1,116$ |
| $10^{40}$ | $3.10^6$ | $2,350$ | $3.10^6$ | $4.10^9$ | $2,871$ |
| $10^{45}$ | $11.10^6$ | $4,480$ | $11.10^6$ | $28.10^9$ | $5,425$ |
| $10^{50}$ | $43.10^6$ | $7,553$ | $43.10^6$ | $200.10^9$ | $9,003$ |
| $10^{55}$ | $110.10^6$ | $17,769$ | $110.10^6$ | $750.10^9$ | $21,183$ |
| $10^{60}$ | $260.10^6$ | $42,017$ | $260.10^6$ | $2.10^{12}$ | $49,534$ |
| $10^{65}$ | $850.10^6$ | $69,408$ | $850.10^6$ | $14.10^{12}$ | $81,387$ |

on the two underlying elliptic curves. Then, for stage 2, it uses GMP-ECM for the two underlying elliptic curves (This could be done in parallel). Indeed, GMP-ECM can use the results of stage 1 from an other software: it just needs the parameter $A$ of the elliptic curve in short Weierstrass form $y^2 = x^3 + Ax + B$ and the coordinates of the point on it.

Our elliptic curves are not, in general, Suyama curves so they don't have the same torsion groups as it is expected in GMP-ECM. This affects the probability of the cardinality of the curves being smooth and thus the choice of $B_1$ and $B_2$. GMP-ECM is able to choose the parameters $B_2$ for stage 2 from the value of $B_1$. This choice is not the same as the "standard continuation" (i.e. $B_2 = 100 * B_1$) but produces much bigger $B_2$. This reduces the number of expected curves for finding a factor. The choice of $B_1$ is heuristic, we tried to minimize the means of the expected time. Note that for finding small factors, HECM is not interesting due to the cost of initialization and morphisms. Table 2 compared the optimal value of $B_1$ for different sizes of the (usually unknown) factor. We see that for finding factors of at least 35 digits, we use the same $B_1$ than ECM. Moreover the ratio of the expected number of curves needed for finding a factor decreases with the size of the factor.

For not too small $B_1$, the theoretical and the real cost of the algorithm is linear in $B_1$. However its dependence in the size of the input number is not simple: modular multiplications are quadratic for small $n$ but become quasi-linear for large $n$. Moreover, there are a lot of other operations which complexity is not negligible for small $n$. Table 3 presents the time taken by the different operations during the computation of $[k]P$ for different sizes of $n$. We see that squares take the same time as multiplications. The reason is that GMP-ECM has special assembly code for general modular multiplications but not for squaring and thus squarings use the same assembly code as normal multiplications. For multiplications by small constants, we made special assembly code. The table also shows that additions are not negligible for small $n$.

Table 3. Fraction of time taken by the arithmetic operations for different sizes of $n$

| size of $n$ | $10^{100}$ | $10^{150}$ | $10^{200}$ | $10^{250}$ | $10^{300}$ | $10^{350}$ | $10^{400}$ | $10^{500}$ | $10^{1000}$ |
|---|---|---|---|---|---|---|---|---|---|
| M | 0.143 | 0.159 | 0.194 | 0.216 | 0.209 | 0.219 | 0.216 | 0.222 | 0.234 |
| S | 0.428 | 0.465 | 0.557 | 0.597 | 0.622 | 0.645 | 0.65 | 0.682 | 0.700 |
| d | 0.163 | 0.127 | 0.092 | 0.081 | 0.070 | 0.057 | 0.037 | 0.031 | 0.051 |
| additions | 0.237 | 0.179 | 0.116 | 0.109 | 0.074 | 0.057 | 0.059 | 0.041 | 0.017 |
| S/M | 1.00 | 0.98 | 0.96 | 0.93 | 0.99 | 0.98 | 1.04 | 1.02 | 1.00 |
| d/M | 0.26 | 0.20 | 0.12 | 0.093 | 0.084 | 0.066 | 0.043 | 0.035 | 0.055 |

Table 4. Comparison between stage 1 of GMP-ECM and GMP-HECM for different sizes of $n$ with a constant $B_1 = 10^7$ on a core 2 at 2.4Ghz.

| size of $n$ | $10^{100}$ | $10^{150}$ | $10^{200}$ | $10^{250}$ | $10^{300}$ | $10^{350}$ | $10^{400}$ | $10^{500}$ | $10^{1000}$ |
|---|---|---|---|---|---|---|---|---|---|
| $\frac{HECM}{2*ECM}$ | 1.15 | 1.09 | 1.03 | 1.0 | 0.97 | 0.94 | 0.95 | 0.93 | 0.89 |

Table 4 compares the running time of two runs of GMP-ECM versus GMP-HECM (remember that one run of HECM is equivalent to two runs of ECM) for a fixed $B_1$ and different sizes of $n$. This shows that for large $n$ (at least $10^{250}$), our software GMP-HECM is faster than GMP-ECM. Since HECM uses more squarings than ECM, having optimized assembly squaring code would be likely to provide a small improvement to the limit where GMP-HECM is more interesting than GMP-ECM. For very large $n$ the speed up of GMP-HECM compared to GMP-ECM is around 11%. This result agrees with the theoretically predicted value.

## Conclusion

Using a subfamily of hyperelliptic curves of genus 2 (i.e decomposable curves), we have built an algorithm for factorization which is equivalent to two simultaneous runs of ECM. Kummer Surfaces allowed us to choose small parameters that give an efficient arithmetic in the Jacobian of the curve. In practice, for large numbers, our implementation is faster than GMP-ECM. Special assembly code for squaring would provide another improvement to GMP-HECM.

Many thanks to Emmanuel Thomé for his help on this work and for his comments on the draft versions. Thanks also to the developers of GMP-ECM and in particular to Alexander Kruppa for his great assembly code.

## Appendix: Morphisms

A Kummer surface is defined by the first four theta constants $(\alpha : \beta : \gamma : \delta)$. Its equation is

$$\begin{aligned} 4E'^2\alpha\beta\gamma\delta XYZT &= \left(\left(X^2+Y^2+Z^2+T^2\right) - \right. \\ &\qquad \left. -F\left(XT+YZ\right) - G\left(XZ+YT\right) - H\left(XY+ZT\right)\right)^2 \end{aligned}$$

where

$$\begin{array}{rclcrcl} A & = & \alpha+\beta+\gamma+\delta, & & Y' & = & \alpha+\beta-\gamma-\delta, \\ C & = & \alpha-\beta+\gamma-\delta, & & T' & = & \alpha-\beta-\gamma+\delta, \end{array}$$

$$\begin{aligned} E' &= \frac{ABCD}{(\alpha\delta-\beta\gamma)(\alpha\gamma-\beta\delta)(\alpha\beta-\gamma\delta)}, \\ F &= \frac{\left(\alpha^2-\beta^2-\gamma^2+\delta^2\right)}{(\alpha\delta-\beta\gamma)}, \\ G &= \frac{\left(\alpha^2-\beta^2+\gamma^2-\delta^2\right)}{(\alpha\gamma-\beta\delta)}, \\ H &= \frac{\left(\alpha^2+\beta^2-\gamma^2-\delta^2\right)}{(\alpha\beta-\gamma\delta)}. \end{aligned}$$

In our case (decomposable curves with $\alpha=\delta$), the other theta constants are given by the following relations:

$$\theta_1^2(0)=\alpha, \quad \theta_2^2(0)=\beta, \quad \theta_3^2(0)=\gamma, \quad \theta_4^2(0)=\delta,$$

$$\theta_8^2(0)=\epsilon=\sqrt{\alpha\beta-\gamma\delta}\sqrt{\frac{\beta}{\alpha}}\sqrt{\nu}=\sqrt{\alpha\beta-\gamma\delta}\sqrt{\frac{\sqrt{\lambda\mu\nu}}{\lambda}},$$

$$\theta_{10}^2(0)=\phi=\frac{\epsilon\alpha}{\nu\beta}=\frac{\alpha\beta-\gamma\delta}{\epsilon},$$

$$\theta_7^2(0)=\sqrt{-\phi^2},$$

$$\theta_9^2(0)=\sqrt{-\epsilon^2},$$

$$\theta_5^2(0)=\frac{\gamma\epsilon-\delta\phi}{\theta_9^2(0)},$$

$$\theta_6^2(0)=\frac{\alpha\delta-\beta\gamma}{\theta_5^2(0)}.$$

The global morphism in HECM goes from the Kummer surface to the product of the underlying elliptic curves modulo the isomorphisms which identify a curve with its twist. However we only need the $(x::z)$ coordinates which are invariant under this morphisms. It is the composition of elementary blocks which use square roots:

$$\begin{tikzcd} \mathcal{C} \arrow[d, "f"'] & Jac(\mathcal{C}) \arrow[r, two heads] \arrow[d, "f_*"'] \arrow[rr, bend left, "\psi"] & Jac(\mathcal{C})/\{\pm 1\} \arrow[r] & \mathcal{K} \arrow[dll, dotted] \\ \mathcal{E}_1\times\mathcal{E}_2 & \mathcal{E}_1\times\mathcal{E}_2. & & \end{tikzcd}$$

Let $P=(X:Y:Z:T)$ a point on the Kummer surface $\mathcal{K}_{(\alpha:\beta:\gamma:\delta)}$ corresponding to the hyperelliptic curve $\mathcal{C}$ of equation $y^2=f(x)$. We want the Mumford coordinates $(u,v)$ of the two opposite divisors $\psi^{-1}(P)=\{\pm D\}$. The divisors $D$ and $-D$

share the same polynomial $u$ and have opposite $v$ polynomial. First, compute the following quantities where for readability we note $\theta_i$ for $\theta_i(0)$:

$$\begin{aligned}
\theta_7^2(z) &= \frac{-X\theta_5^2\theta_8^2+Y\theta_5^2\theta_{10}^2-Z\theta_6^2\theta_{10}^2+T\theta_6^2\theta_8^2}{\theta_6^4-\theta_5^4},\\
\theta_9^2(z) &= \frac{X\theta_6^2\theta_{10}^2-Y\theta_6^2\theta_8^2+Z\theta_5^2\theta_8^2-T\theta_5^2\theta_{10}^2}{\theta_8^4-\theta_{10}^4},\\
\theta_{11}^2(z) &= \frac{X\theta_6^2\theta_8^2-Y\theta_6^2\theta_{10}^2+Z\theta_5^2\theta_{10}^2-T\theta_5^2\theta_8^2}{\theta_6^4-\theta_5^4},\\
\theta_{12}^2(z) &= \frac{-X\theta_5^2\theta_{10}^2+Y\theta_5^2\theta_8^2-Z\theta_6^2\theta_8^2+T\theta_6^2\theta_{10}^2}{\theta_8^4-\theta_{10}^4},\\
\theta_{13}^2(z) &= \frac{-X\theta_8^2\theta_9^2+Y\theta_9^2\theta_{10}^2+Z\theta_7^2\theta_8^2-T\theta_7^2\theta_{10}^2}{\theta_7^4-\theta_9^4},\\
\theta_{14}^2(z) &= \frac{-X\theta_5^2\theta_9^2-Y\theta_6^2\theta_7^2+Z\theta_5^2\theta_7^2+T\theta_6^2\theta_9^2}{\theta_7^4-\theta_9^4},\\
\theta_{16}^2(z) &= \frac{-X\theta_6^2\theta_7^2-Y\theta_5^2\theta_9^2+Z\theta_6^2\theta_9^2+T\theta_5^2\theta_7^2}{\theta_7^4-\theta_9^4}.
\end{aligned}$$

In the general case, $\theta_{16}^2(z)$ is non zero. In this case $u(\xi)=\xi^2+u_1\xi+u_0$ is of degree 2 and $v(\xi)=\pm(v_1\xi+v_0)$ is of degree 1 with

$$u_0=\lambda\frac{\theta_8^2\,\theta_{14}^2(z)}{\theta_{10}^2\theta_{16}^2(z)},\qquad u_1=(\lambda-1)\frac{\theta_5^2\,\theta_{13}^2(z)}{\theta_{10}^2\theta_{16}^2(z)}-u_0-1,$$

$$\begin{aligned}
v_0^2 = & -\frac{\theta_1^4\theta_3^4\theta_8^2\,\theta_{14}^2(z)}{\left(\theta_2^2\theta_4^2\theta_{10}^2\,\theta_{16}^2(z)\right)^3}\left(\theta_2^2\theta_3^2\theta_9^4\,\theta_7^2(z)\,\theta_{12}^2(z)+\theta_1^2\theta_4^2\theta_7^4\,\theta_9^2(z)\,\theta_{11}^2(z)\right.\\
& +2\theta_1^2\theta_2^2\theta_3^2\theta_4^2(XZ+YT)-\frac{\theta_1^2\theta_3^2+\theta_2^2\theta_4^2}{E'}\left(\left(X^2+Y^2+Z^2+T^2\right)-\right.\\
& \left.\left.-F(XT+YZ)-G(XZ+YT)-H(XY+ZT)\right)\right).
\end{aligned}$$

Since $u$ divides $v^2-f$ we obtain formulæ to compute $v_1$. If $\theta_{16}^2(z)$ is zero and $(\lambda-1)\,\theta_5^2\,\theta_{13}^2(z)-\lambda\theta_8^2\,\theta_{14}^2(z)$ is non zero, then $u(\xi)=\xi+u_0$ is of degree 1 and $v(\xi)=\pm v_0$ is constant with

$$u_0=\frac{\lambda\theta_8^2\,\theta_{14}^2(z)}{(\lambda-1)\,\theta_5^2\,\theta_{13}^2(z)-\lambda\theta_8^2\,\theta_{14}^2(z)},\qquad v(\xi)=\pm v_0=\pm\sqrt{f(-u_0)}.$$

The last case is when $\theta_{16}^2(z)=(\lambda-1)\,\theta_5^2\,\theta_{13}^2(z)-\lambda\theta_8^2\,\theta_{14}^2(z)=0$. In that case, the divisor $D$ is the zero divisor.

Let $P$ be a point on the $(2,2)$-decomposable hyperelliptic curve $\mathcal{C}$ given by the equation

$$\mathcal{C}:\quad \chi y^2=x(x-1)(x-\lambda)(x-\mu)(x-\nu),\quad \text{with } \lambda=\mu\frac{1-\nu}{1-\mu}.$$

We want to map $P$ to the elliptic curves. Let $\mathcal{C}'$ be the curve given by

$$\mathcal{C}':\quad \kappa y^2=\left(x^2-1\right)\left(x^2-x_2^2\right)\left(x^2-x_3^2\right)$$

where

$$q = \pm\sqrt{\mu(\mu-\nu)},\ x_2 = \frac{\mu+q}{\mu-q},\ x_3 = \frac{1-\mu-q}{1-\mu+q},\ \kappa = -\chi q\,\mu(\mu-1)\,.$$

The curves $\mathcal{C}$ and $\mathcal{C}'$ are isomorphic by the change of coordinates

$$\begin{array}{lcll} \mathcal{C} & \longrightarrow & \mathcal{C}' & \\ (x,y) & \longmapsto & \left(\frac{x-\mu-q}{x-\mu+q}, \frac{wy}{(x-\mu+q)^3}\right) & \text{with } w = \frac{8q}{(\mu-q)(-1+\mu-q)} \end{array}$$

The curve $\mathcal{C}'$ maps to the elliptic curve $\mathcal{E}:\ y^2 = (x-1)\left(x-x_2^2\right)\left(x-x_3^2\right)$ by the morphism $(x,y) \mapsto \left(x^2, y\right)$.
Changing $q$ to $-q$ changes the elliptic curve $\mathcal{E}$ to the other underlying elliptic curve.

Let $f$ be the map from $\mathcal{C}$ to the product of the two elliptic curves $\mathcal{E}_1 \times \mathcal{E}_2$ given by the product of the two maps defined above. The push forward $f_*$ of $f$ is defined by

$$f_*\left\{\begin{array}{lcl} Jac\,(\mathcal{C}) & \longrightarrow & Jac\,(\mathcal{E}_1) \times Jac\,(\mathcal{E}_2) \\ D = \sum_{i=1}^r P_i - rP_\infty & \longmapsto & \sum_{i=1}^r f\,(P_i) - r\,f\,(P_\infty) \end{array}\right.$$

where we define $f\,(P_\infty) = (\mathcal{O}_{\mathcal{E}_1}, \mathcal{O}_{\mathcal{E}_2})$ to be the zero of the two elliptic curves. Note that the divisors in the Jacobian of the elliptic curves are not reduced. For elliptic curves, the Jacobian of the curve is isomorphic to the set of points on the curve. Thus we can identify $Jac\,(\mathcal{E}_i)$ with $\mathcal{E}_i$. We can rewrite $f_*$ as

$$f_*\left\{\begin{array}{lcl} Jac\,(\mathcal{C}) & \longrightarrow & \mathcal{E}_1 \times \mathcal{E}_2 \\ D = \sum_{i=1}^r P_i - rP_\infty & \longmapsto & \sum_{i=1}^r f\,(P_i) \end{array}\right.$$

A generic divisor $D$ in the Jacobian of a genus 2 curve is the formal sum: $D = P_1 + P_2 - 2P_\infty$. Thus in general we have to add the two points $f\,(P_1)$ and $f\,(P_2)$ on the elliptic curves. In practice, to do this operation, we need complete formulæ for adding points on elliptic curves because we can't take a square root to get the $y$-coordinate which is needed for doubling in Weierstrass coordinates. Thus we use Jacobi or Edwards equations for the curves. Another solution to avoid square roots is to work in an extension "field" of degree 4.

For stage 2, GMP-ECM needs a point $(x_2, y_2)$ on the curve $y^2 = x^3 + Ax + B$ but we have a point $(x_1 :: z_1)$ on $\kappa z y^2 = x^3 + a_2 x^2 z + a_4 x z^2 + a_6 z^3$. First translate the point by $x \mapsto x - a_2 z/3$ and divide the $x$-coordinate by $z$ to get a point $(x_1', ?)$ on the curve $\kappa y^2 = f\,(x) = x^3 + a_4' x + a_6'$. Then the points $(x_1', \pm 1)$ are on the curve

$$Dy^2 = f\,(x) = x^3 + a_4' x + a_6' \quad \text{with } D = f\,(x_1')\,.$$

By the change of variable

$$(x,y) \longmapsto \left(\frac{x}{D}, \frac{y}{D}\right)$$

we get a point on the curve $y^2 = x^3 + Ax + B$ with $A = a_4'/D^2$ and $B = a_6/D^3$. Note that we can choose the sign of $y$ since a point and its opposite have the same order (remember that we cleared the power of 2 in stage 1).

## References


1. D. J. Bernstein, P. Birkner, T. Lange, and C. Peters, *ECM using Edwards curves*, Cryptology ePrint Archive, 2008, http://eprint.iacr.org/2008/016.
2. S. Duquesne, *Improving the arithmetic of elliptic curve in the Jacobi model*, Inform. Process. Lett. **104** (2007), 101–105.
3. P. Gaudry, *Fast genus 2 arithmetic based on theta functions*, J. Math. Cryptol. **1** (2007), 243–265.
4. P. Gaudry and É. Schost, *On the invariants of the quotients of the Jacobian of a curve of genus 2*, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (S. Boztaş and I. Shparlinski, eds.), Lecture Notes in Comput. Sci., vol. 2227, Springer-Verlag, 2001, pp. 373–386.
5. A. Kruppa, *Factoring into large primes with P-1, P+1 and ECM*, 2008, Available at http://cado.gforge.inria.fr/workshop/slides/kruppa.pdf.
6. H. W. Lenstra, Jr., *Factoring integers with elliptic curves*, Ann. of Math. (2) **126** (1987), 649–673.
7. P. L. Montgomery, *Evaluating recurrences of form $x_{m+n} = f(x_m, x_n, x_{m-n})$ via Lucas chains*, 1983, Available at ftp.cwi.nl/pub/pmontgom/Lucas.ps.gz.
8. ———, *Speeding the Pollard and Elliptic Curve Methods of Factorization*, Math. Comp. **48** (1987), no. 177, 243–264.
9. D. Mumford, *Tata lectures on theta I*, Progr. Math., vol. 28, Birkhäuser, 1983.
10. ———, *Tata lectures on theta II*, Progr. Math., vol. 43, Birkhäuser, 1984.
11. T. Shaska, *Genus 2 curves covering elliptic curves, a computational approach*, Lecture Notes Ser. Comput. **13** (2005), 151–195.
12. P. van Wamelen, *Equations for the Jacobian of a hyperelliptic curve*, Trans. Amer. Math. Soc. **350** (1998), no. 8, 3083–3106.
13. P. Zimmermann and B. Dodson, *20 years of ECM*, ANTS VII, Lecture Notes in Comput. Sci., vol. 4076, Springer-Verlag, 2006, pp. 525–542.


LORIA, Campus Scientifique - BP 239, 54506 Vandoeuvre-lès-Nancy , France
*E-mail address*: `romain.cosset@loria.fr`