Assignment for the Postgraduate Course Cryptography over Elliptic Curves: Factorization of Semi-Primes using Lenstra's and SQUAR Method

Paschalis Aggelidis
Department of Mathematics
Aristotle University of Thessaloniki

June 26, 2023

Contents

Abstract

This assignment provides a comprehensive study and comparison of two factorization methods, namely Lenstra's factorization method and the SQUAR algorithm, both of which are based on elliptic curves. The objective of this study is to to explore the effectiveness and efficiency of these algorithms in factoring semi-primes. We begin by introducing the fundamentals of elliptic curves over fields of characteristic $\neq 2, 3$ and the structure of the abelian group defined on them. Next the Lenstra's method is introduced, which relies on repeated addition, and analyzed in detail. Following the study of Lenstra's algorithm, we introduce the SQUAR algorithm, which is based on counting points over an elliptic curve. We proceed by highlighting the main differences of the algorithms and provide some numerical results to compare their time efficiency. Finally an acceleration of SQUAR algorithm is discussed.

Keywords— elliptic curves, factorization, Lenstra's Method, SQUAR Method, acceleration

1 Introduction

The study of elliptic curves represents an intersection of algebra, geometry and number theory. The two algorithms that are presented in this assignment are a perfect illustration of how these mathematical fields intersect. There are several algorithms that factorize a semi-prime $n = pq$, where n is known, and p, q are integer prime factors to be found. In this assignment, two methods are concerned for factoring semi-primes, which are Lenstra's factorization method and the SQUAR method of factoring semi-primes. Before discussing the algorithms and their functionality we introduce some basic concepts of elliptic curves and the group structures on which the algorithms perform. Next we discuss about Lenstra's method and its characteristics which are based on repeated addition of a pair of points (x, y) on an elliptic curve. After that, we introduce the SQUAR algorithm which is based on counting how integer points (x, y) satisfy these curves. Crypto-immunity of various cryptographic protocols is mostly based on hardness of either the integer factorization or the discrete logarithm problem.

2 Fundamentals of Elliptic Curves

Here we define elliptic curves over a field $\mathbb{K}$, where $char(\mathbb{K}) \neq 2, 3$.

Definition 1. *Let $\mathbb{K}$ be a field of either characteristic 0 or characteristic greater than 3. Then we define an elliptic curve E over $\mathbb{K}$ to be the projective closure of a nonsingular curve over $\mathbb{K}$ of the form*

$$y^2 = x^3 + ax + b$$

where $a, b \in \mathbb{K}$.

To describe this projective closure, we set $f(x, y) = y^2 - x^3 - ax - b$, then consider the homogenization of $f(x, y)$:

$$F(X, Y, Z) = Z^3 f\left(\frac{X}{Z}, \frac{Y}{Z}\right) = Y^2Z - X^3 - aXZ^2 - bZ^3$$

The projective closure is the set of solutions to $F(X, Y, Z) = 0$. Since in projective space any point $(X : Y : Z)$ is equivalent to $(\lambda X : \lambda Y : \lambda Z)$ where $\lambda \in \mathbb{K}$ is nonzero, the solutions where $Z \neq 0$ are equivalent the solutions where $Z = 1$: the affine solutions $f(x, y) = 0$.

Now we consider when $Z = 0$. Plugging in, we get $X^3 = 0$, indicating that the only remaining solution is $(0 : 1 : 0)$. We will call this the point at infinity, referred to with the letter O, and the rest of the points will be referred to by their corresponding affine point.

We give one more definition.

Definition 2. *We define the set $\mathbb{K}$-rational points of an elliptic curve E as the set of points on E whose coordinates all lie in $\mathbb{K}$, as well as the point O, and denote it $E(\mathbb{K})$.*

3 The Group Law

We now introduce the construction of the abelian group structure on the $\mathbb{K}$-rational points of an elliptic curve E over a field $\mathbb{K}$. To accomplish this, we first give such an abelian group structure over the algebraic closure $\overline{\mathbb{K}}$. We then show that $E(\mathbb{K})$ is a subgroup of $E(\overline{\mathbb{K}})$, which implies that $E(\mathbb{K})$ itself is an abelian group.

Let E be an elliptic curve over a field $\mathbb{K}$, where $char(\mathbb{K}) = 0$ or $char(\mathbb{K}) \neq 2, 3$, and let $\overline{\mathbb{K}}$ be the algebraic closure of $\mathbb{K}$. Since $\mathbb{K} \subset \overline{\mathbb{K}}$, we can also consider E over $\overline{\mathbb{K}}$. Let L be any line in projective space, let S be any $\overline{\mathbb{K}}$-rational point, and let m_S denote the multiplicity of the intersection of E and

L at S. As $\overline{\mathbb{K}}$ is algebraically closed, and $\deg(E) = 3$, the total multiplicity over all points is 3, i.e., $\sum_S m_s = 3$. Without loss of generality, and bearing in mind that due to multiplicity, any two or all three points could be equal, we refer to the three points of intersection of L with E as P, Q and R. Also, for any point $S \in E(\overline{\mathbb{K}})$ with $S = (X_S : Y_S : Z_S)$, we define the point $-S$ to be the point $(X_S : -Y_S : Z_S)$. Note that by the definition of an elliptic curve, $S \in E(\overline{\mathbb{K}})$ implies that $-S \in E)\overline{\mathbb{K}}$. Furthermore, since $(0 : 1 : 0) = (0 : -1 : 0)$, we can see that $-O = O$. With these facts in mind, we define addition over $E(\overline{\mathbb{K}})$ by saying $P, Q, R \in E(\overline{\mathbb{K}})$ are colinear, then

$$P + Q = -R$$

To find the sum of two points, we examine the line L between them, find the third point of intersection of L with E, and then invert that point. Now we shoe that this definition of addition is an abelian group.

First of all, consider the intersection of the line $Z = 0$ with an elliptic curve E over $\overline{\mathbb{K}}$ given by the equation $Y^2Z - X^3 - aXZ^2 - bZ^3$. As touched on before, the point of intersection O satisfies $X^3 = 0$. But now notice that this intersection has multiplicity 3. Therefore $O + O = -0$, i.e., $O + O = O$.

Addition over $E(\overline{\mathbb{K}})$ must be closed, since we determined that any $P, Q, R \in E(\overline{\mathbb{K}})$ and that $S \in E(\overline{\mathbb{K}})$ implies that $-S \in E(\overline{\mathbb{K}})$.

Note also that it must be the case that $\forall P, Q \in E(\overline{\mathbb{K}})$, $P + Q = Q + P$, since P, Q determine exactly one line. Therefore the operation is commutative.

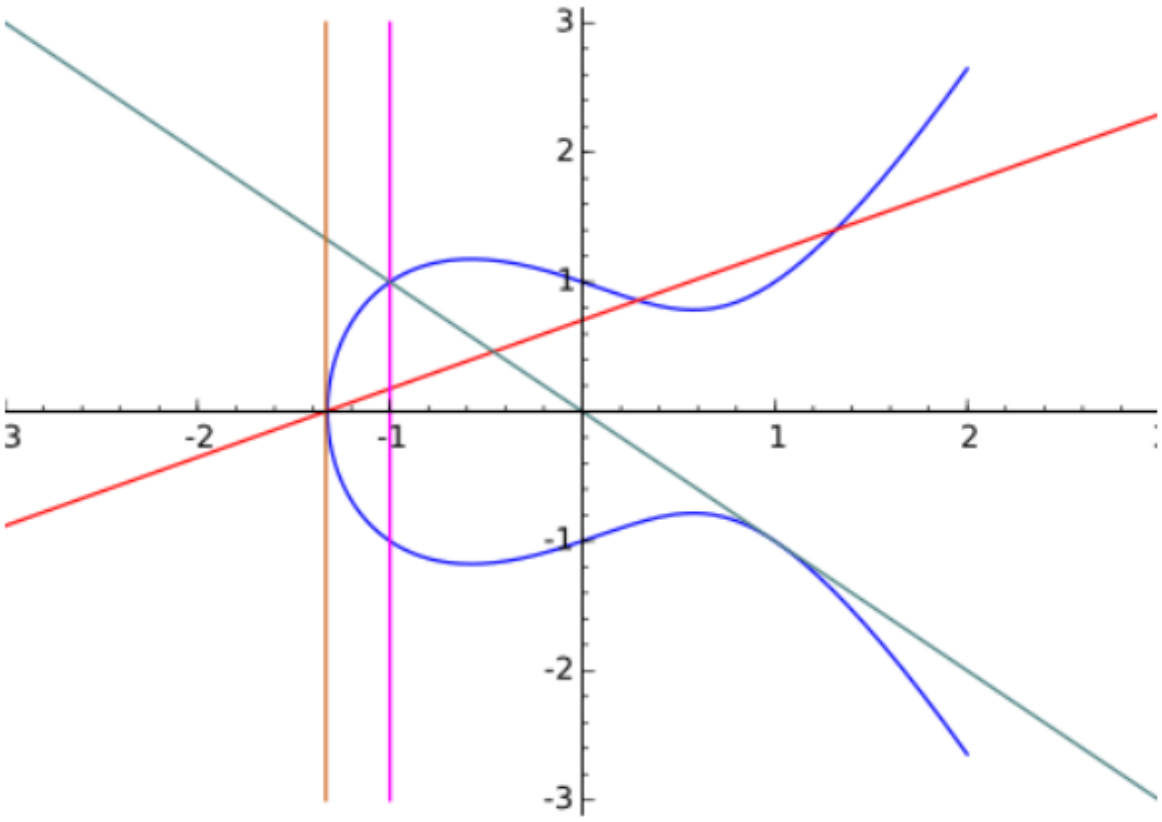

Figure 1: Some potential intersections of a line with an elliptic curve.

Next, we show that for a point $P \in E(\overline{\mathbb{K}})$, the additive inverse is indeed $-P$. Let $P = (x_P, y_P)$ where $x_P, y_P \in \overline{\mathbb{K}}$. Then the only line intersecting both P and $-P$ is $X = x_P Z$. The third point at which this line intersects E is O, so $P + (-P) = -O = O$.

Thus, we have demonstrated all conditions for $E(\overline{\mathbb{K}})$ to be an abelian group. We derive an explicit formulae to calculate the sum of any two points. Let $P, Q \in E(\overline{\mathbb{K}})$ such that $P = (x_1, y_1)$ and $Q = (x_2, y_2)$.

First we assume that $P \neq Q$. If $x_1 \neq x_2$ then we can easily find λ, the slope of the line between them:

$$\lambda = \frac{y_1 - y_2}{x_1 - x_2}$$

Next we calculate the y-intercept of the line, μ, as

$$\mu = y_1 - \lambda x_1 = y_2 - \lambda x_2$$

Now that we have the equation for the line, we can substitute $\lambda x + \mu$ in for any y in our elliptic curve:

$$y^2 = (\lambda x + \mu)^2 = x^3 + ax + b$$
$$0 = x^3 - \lambda^2 x^2 + (a - 2\lambda\mu)x + (b - \mu^2)$$

By the fundamental theorem of algebra, and since $\overline{\mathbb{K}}$ is algebraically closed, this cubic equation must have a third root, namely $x_3 \in \overline{\mathbb{K}}$, so

$$0 = x^3 - \lambda^2 x^2 + (a - 2\lambda\mu)x + (b - \mu^2) = (x - x_1)(x - x_2)(x - x_3).$$

When we expand the right-hand side expression, the coefficient of the x^2 term is $-x1 - x_2 - x_3$. Therefore,

$$\lambda^2 = x_1 + x_2 + x_3.$$

That means that we can isolate x_3 in the equation, and subsequently find y_3:

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = -\lambda x_3 - \mu.$$

As P, Q and R are colinear, recall that for $R = (x_3, y_3)$, we have that $P + Q = -R = (x_3, -y_3)$. If x_1, x_2, x_3 are all distinct, the line connecting P, Q and R will resemble the red line in Figure 1.

Now suppose, contrary to our earlier assumption, that $x_1 = x_2$. If $P \neq Q$, $y_2 = -y_1$ and the line connecting them is simply the vertical line $x = x_1$, so $Q = -P$ and $P + Q = O$. This case resembles the pink line in Figure 1.

If $P = Q$, then the line with multiplicity 2 at P is the tangent line to E at P, so to find the slope λ of the tangent line at P, we perform implicit differentiation:

$$y^2 = x^3 + ax + b$$
$$2y\frac{dy}{dx} = 3x^2 + a$$
$$\frac{dy}{dx} = \frac{3x^2 + a}{2y}$$
$$\lambda = \frac{3x_1^2 + a}{2y_1}$$

We then proceed as before with the prior formulae and arrive at the same equations for x_3 and y_3, unless $y = 0$. If $y = 0$, we again have a vertical line, so the third point of the intersection is again O. Thus, $P + P = O$. Note that in this case, $P = -P$. For $y \neq 0$, the $P = Q$ case resembles the blue line in Figure 1. If $y = 0$, the $P = Q$ case resembles the orange line.

Suppose that $P, Q \in E(\mathbb{K})$, and $P + Q = -R$. Assume $R \neq O$. Then using our formulae for x_3 and y_3, in all cases, x_3 and y_3 are arithmetic expressions of x_1, x_2, y_1, y_2, and a, all of which are in $\mathbb{K}$, meaning that $x_3, y_3 \in \mathbb{K}$. If $R = O$, then by definition $R \in E(\mathbb{K})$. Since $R \in E(\mathbb{K})$, $-R \in E(\mathbb{K})$, so $E(\mathbb{K})$ is closed under addition of points. That makes $E(\mathbb{K})$ a subgroup of $E(\overline{\mathbb{K}})$. Consequently, $E(\mathbb{K})$ is an abelian group.

4 Lenstra's Method

The algorithm (1) is based on the multiplication of a point P of an elliptic curve E over a finite field $\mathbb{F}_p$, k times. In order to see how this algorithm works instead of $\mathbb{F}_p$ we will be doing the multiplication of a random point $P \in E$ over $\mathbb{Z}/n\mathbb{Z}$, as if $\mathbb{Z}/n\mathbb{Z}$ is the field over which we are working, which is problematic. Indeed, $\mathbb{Z}/n\mathbb{Z}$ is not a field since $\exists g \in \mathbb{Z}/n\mathbb{Z}$ where $\gcd(g, n) \neq 1$ and those elements lack multiplicative inverse. But this is what we are looking for when we are hunting for a factor of n. Recall the formulae of addition of two points P, Q of an elliptic curve E, where $R = (x_3, y_3)$ and $P + Q = -R$:

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = -\lambda x_3 - \mu$$

For $x_1 \neq x_2$ we had

$$\lambda = (y_1 - y_2)(x_1 - x_2)^{-1}$$

making $x_1 - x_2$ the value that may not have an inverse. When $x_1 = x_2$, we instead used the formulae

$$\lambda = (3x_1^2 + a)(2y_1)^{-1}$$

in the equations of x_3 and y_3 to find the sum of P and Q, and it will instead be $2y_1$ that may lack a multiplicative inverse. For simplicity, we will call d the value, of either $x_1 - x_2$ or $2y_1$, that lacks multiplicative inverse.

When d lacks multiplicative inverse in $\mathbb{Z}/n\mathbb{Z}$, it means that $D = \gcd(d, n) > 1$. Since $D < n$ then $D|n$. This is the mechanism by which we will find a factor of n. But when d lacks an inverse?

We will denote the number of points of an elliptic curve E over $\mathbb{F}_p$ as $\#E(\mathbb{F}_p)$. From Lagrange's Theorem every point $P \in \mathbb{F}_p$, has an order where $ord(P)|\#E(\mathbb{F}_p)$. When $\#E(\mathbb{F}_p)|k$ it means that $kP = O$ for every point on E when reduced mod$p$. This means that p devides the denominator d of kP. So as $p|d$, and p divides n, then $p|\gcd(d, n)$. Therefore we will have this lack of inverse exactly when $\#E(\mathbb{F}_p)|k$.

There are two restrictions to the above. As discussed before the characteristic of the group over which we are using the formulas for addition must not be 2 or 3, so we need to check that $2, 3 \nmid n$ before we begin. Also, if there exist m, r such that $m^r = n$, we will not find the factors of n using this method, so we need to check that the roots of n, starting from the square root and stopping once a root evaluates to be less than 2, are not natural numbers.

With this explanation of the basic method done, we now proceed with the actual steps of the algorithm:

Algorithm 1 Lentra's Method for factoring semi-primes.

```
1: procedure LENSTRA(n, limit)
2:     if 2, 3|n then
3:         return "Integers 2 or 3 must not divide n."
4:     end if
5:     if ∃m, r ∈ ℤ : m^r = n then
6:         return "This method is not applicable."
7:     end if
8:     A, x_1, y_1 ← random values: 1 < A, x_1, y_1 < n
9:     B ← y_1^2 − x_1^3 − b x_1
```

10: **if** $\gcd(4A^3 + 27B^2, n) = n$ **then**
11: **return** "Go back to step 8."
12: **else if** $\gcd(4A^3 + 27B^2, n) > 1$ **then**
13: **return** $\gcd(4A^3 + 27B^2, n)$
14: **else**
15: **for** $k \leftarrow lcm(1, \ldots, \text{limit})$ **do**
16: $P \leftarrow kP$
17: **if** kP fails **then**
18: **return** $\gcd(d, n)$
19: **end if**
20: **end for**
21: **end if**

As we can see, the algorithm can be really exhausting, since we have to compute the multiplication of kP multiples times before we find a factor of n. If we get lucky, the algorithm could end in two multiplications of kP, which depends on the randomized point P of E. Lenstra's algorithm is the third fastest method for factoring large numbers and the best in handling low factors.

5 SQUAR Algorithm

This factoring algorithm (2) is based on the analysis of several elliptic curves E and counting the points $P \in E$.

We consider $n = pq$, where p and q are multi-digit long primes. We will study only two of the following three cases:

$$p = q = 1 \bmod 4$$
$$(p + q)(\text{mod}4) = 0$$
$$p = q = 3(\text{mod}4)$$

In this assignment we discuss the factorization algorithm for the first two cases only.

We consider a sequence of elliptic curves modulo n:

$$E(n, b) : \; y^2 = x\left(x^2 + b^2\right)(\text{mod}n)$$

where $b \geq 1$ is an integer control parameter and let $P(n, b)$ denote the number of points on the elliptic curve $E(n, b)$. We will ow illustrate the SQUAR algorithm:

Algorithm 2 SQUAR Factorization Algorithm for factoring semi-primes

1: **procedure** SQUAR(n)
2: **if** $n(\text{mod}4) = 3$ **then**
3: $b \leftarrow$ random integer
4: compute $P(n, b)$
5: $p \leftarrow \gcd\left[P(n, b), n\right], \; q \leftarrow n/p$
6: **return** $p, \; q \;\; = 0$

```
else if n(mod4) = 1 then
    compute P(n, 1)
    if P(n, 1) = n then
        return "The algorithm is not applicable"
     else
        for b ← 2, 3, 5, 7, 11, ... do
            compute P(n, b)
            if b|n then
                p ← b, q ← n/p
            else if P(n, b_i) ≠ P(n, b_j), i ≠ j, i, j ∈ {1, 2, 3, 4} then
                Q ← max(A, Q, R, U)
                U ← min(A, Q, R, U)
                S ← (Q − U − |A − R|)/4
                p ← gcd(n, S), q ← n/p
                return p, q
            end if
        end for
    end if
 end if
```

After presenting the algorithm itself we will now show the validation of the algorithm. We begin with this simple definition that anyone familiar with basic Number Theory should know:

Definition 3. *A non-zero integer a is called a quadratic residue* mod$p$ *if-f* $\exists z \in \mathbb{Z}$*:*

$$z^2 = a(\mathrm{mod}p)$$

By the Euler criterion, if p is prime, then a is a quadratic residue if-f

$$a^{(p-1)/2}(\mathrm{mod}p) = 1$$

and the algorithm is based on the following proposition:

Conjecture 1. If $p = q = 1(\mathrm{mod}4)$ then there exist two positive integers $c < p$ and $d < q$, and four sets S_1, S_2, S_3 and S_4 such that for every $i \neq j,\ S_i \cap S_j = \emptyset$, where:

$$\begin{aligned} S_1 &= \{b : P(n,b) = U := (p-c)(q-d)\}; \\ S_2 &= \{b : P(n,b) = A := (p-c)(q+d)\}; \\ S_3 &= \{b : P(n,b) = R := (p+c)(q+d)\}; \\ S_4 &= \{b : P(n,b) = Q := (p+c)(q+d)\} \end{aligned}$$

and $S_1 \cup S_2 \cup S_3 \cup S_4 = \{1, 2, 3, 5, 7, 11, 13, \dots\}$ (set of all primes).

Example 1. For the semi-prime $n = 1352513$ the sets S_1, S_2, S_3 and S_4 are as follows:

$$\begin{aligned} S_1 &= \{b = 1, 2, 7, 41, \dots; U = 1267905\}; \\ S_2 &= \{b = 5, 13, 17, 43, 61, 67, 71, 79, \dots; A = 1313817\}; \\ S_3 &= \{b = 3, 11, 23, 29, 37, 47, 59, 73, \dots; R = 1389325\}; \\ S_4 &= \{b = 19, 31, 53, \dots; Q = 1439305\} \end{aligned}$$

and so if we set $S = (Q - U - |A - R|)/4$, we have $p = \gcd(n, S) = 569$ and $q = n/p = 2377$. Notice, at the beginning of this example, that $n(\mathrm{mod}4) = 1$

A priori it is not obvious how this conjecture would help to find the factors of n. But as we seen above that is the case. From the definitions above, we get that $Q > \max(A, R)$ and $U < \min(A, R)$, i.e., that $n + (pd + qc) + cd > \max(A, R)$ and $\min(A, R) < n - (pd + qc) + cd$. Let $R = A$, then $pd = qc$. Since both p and q are primes, the latter equation holds only if $c = p$ and $d = q$, which is impossible by the conjecture.

As a result of the algorithm, we derive a system of three equations with four integer unknowns p, q, c and d:

$$\begin{gathered} pq = n;\ (p + c)(q + d) = Q;\ (p - c)(q - d) = U; \\ \text{i.e., } n + (pd + qd) + cd = Q; \\ \text{and } n - (pd + qc) + cd = U \end{gathered}$$

Now we consider another system of three equations with the same integer unknowns:

$$\begin{gathered} pq = n,\ (p - c)(q + d) = A;\ (p + c)(q - d) = R; \\ \text{i.e., } n + (pd - qc) - cd = A; \\ \text{and } n - (pd - qc) - cd = R \end{gathered}$$

Then from the above equations we get:

$$pd - qc = (A - R)/2$$

and as a result,

$$qc = \frac{\left[\frac{Q-U}{2} - \frac{A-R}{2}\right]}{4}$$

Finally, from $qc \neq n$, it follows that

$$q = \gcd(n, qc),\ p = n/q$$

In fact, one can easily see that one of the factors is the average of two gratest common divisors

$$p = \left[\gcd(A, Q) + \gcd(R, U)\right]/2,\ q = n/p$$

But this computation is twice as complicated as the previous one.

The SQUAR factorization algorithm can be generalized and based on one of the following conjectures. Consider a set of elliptic curves as follows:

$$E(n, a) :\ y^2 = x(x^2 + a)(\mathrm{mod}n)$$

where $a \neq 0$, $n = pq$, $p = q = 1(bmod4)$ and $P(n, a)$ denotes the number of points on $E(n, a)$.

Conjecture 2. If a is a quadratic residue mod$n$, then there exist two positive integers $c < p$ and $d < q$, and four sets S_1, S_2, S_3 and S_4 such that for every $i \neq j$, $S_i \cap S_j = \emptyset$, where

$$\begin{aligned} S_1 &= \{a : P(n, a) = u_1 := (p - c)(q - d)\} \\ S_2 &= \{a : P(n, a) = u_2 := (p - c)(q + d)\} \\ S_3 &= \{a : P(n, a) = u_3 := (p + c)(q - d)\} \\ S_4 &= \{a : P(n, a) = u_4 := (p + c)(q + d)\} \end{aligned}$$

and $S_1 \cup S_2 \cup S_3 \cup S_4$ is the set of all quadratic residue mod$n$.

Conjecture 3. If a is a quadratic non-residue mod$n$, then there exist two positive integers $g < p$ and $h < q$, and four sets T_1, T_2, T_3 and T_4 such that

$$T_1 = \{a : P(n,a) = w_1 := (p+g)(q+h)\}$$
$$T_2 = \{a : P(n,a) = w_2 := (p-g)(q+h)\}$$
$$T_3 = \{a : P(n,a) = w_3 := (p+g)(q-h)\}$$
$$T_4 = \{a : P(n,a) = w_4 := (p-g)(q-h)\}$$

where for every $i \neq j$, $T_i \cap T_j = \emptyset$ and $T_1 \cup T_2 \cup T_3 \cup T_4$ is the set of all quadratic non-residue mod$n$.

6 SQUAR Algorithm Acceleration

We begin this section by providing some numerical results of the SQUAR algorithm in order to thoroughly understand the acceleration procedure.

Table 1: **Number of point $P(n, b_k)$ on four elliptic curves**

n	$P(n, b_{k_1}); k_1 = 1$	$P(n, b_{k_2}); k_2$	$P(n, b_{k_3}); k_3$	$P(n, b_{k_4}); k_4$	max
24869	**37981;1**	**34713;2**	**13993;3**	**12789;4**	4
3813809	**3850233;1**	**3774993;2**	**3674789;3**	**3955221;11**	11
3858521	**3996001;1**	**3652173;3**	**3717945;4**	**4067965;17**	17
4549289	**4255713;1**	**4558669;3**	**4852633;4**	**4530141;7**	7

Table 2: **Major steps of SQUAR algorithm.**

n	A	Q	R	U	S	$p; q$
24869	37981	34713	13993	12789	**1118**	**13; 1913**
3813809	3955221	3850233	3774993	3674789	**51298**	**1973; 1933**
3858521	4067965	3996001	3717945	3652173	**34434**	**1913; 2017**
4549289	4852633	4558669	4530141	4255713	**142098**	**2153; 2113**

The computations provided in Table 1 show that for $n = 3813809$ it is necessary to compute $P(n, b)$ six times for $b = 1, 2, 3, 5, 7$ and 11 until four distinct integers A, Q, U, R are found. Hence, the acceleration process will focus on reducing the times $P(n, b)$ has to be computed.

Let $p = q = 1(\text{mod}4)$; $n = pq$ and $M(n, b)$ denote the number of points on a dual elliptic curve:

$$y^2 = x(x^2 - b^2)(\text{mod}n)$$

The following conjecture and proposition holds:

Conjecture 4. If primes p and q are randomly selected, then with probability 3/4

$$P(n,1) \neq M(n,1)$$

and as a result, for every integer b

$$P(n,b) \neq M(n,b)$$

otherwise for every integer b

$$P(n,b) = M(n,b)$$

Proposition 1. If the factors $p, q = 1(\mathrm{mod} n)$, where $n = pq$, then the following identities hold for every positive integer m:

$$P(n,2^{m+1}) = M(n,2^{m-1}) \text{ and } M(n,2^{m+1}) = P(n,2^{m-1})$$

Proposition 2. If the factors $p, q = 1(\mathrm{mod} n)$, where $n = pq$, and $b_1 \neq b_2$ and $P(n,b1) \neq M(n,b_2)$ then $M(n,b_1) = P(n,b_2)$.

Let for example $n = 3813809$ and let $E_1 : \ y^2 = x(x^2 + b^2)(\mathrm{mod} n)$, $E_2 : \ y^2 = x(x^2 - b^2)(\mathrm{mod} n)$, two dual elliptic curves. We will use the above notation for the points of E_1 and E_2 accordingly. Hence, we have $P(3813809, 1) = 38500233$ and $M(3813809, 1) = 3774993$. Thus, Proposition 1 implies that there is no need to compute $P(3813809, 2)$. Instead, we compute $P(n, 3) = 3674789$, which is the third distinct value, hence Conjecture 4 imply that $M(n, 3) = 3955221$ is the fourth distinct value. Therefore, the algorithm requires only four basic steps instead of six. However, this acceleration procedure does not work in 25% of the cases. For instance, it does not work for $n = 3858521$, since neither of the first two properties of Conjecture 4 hold. Indeed, because $P(3858521, 1) = M(3858521, 1)$ the computation of $M(3858521, 3)$ provides no acceleration.

We will now provide the accelerated factorization algorithm. We also, assume that $n(\mathrm{mod} 4) = 1$ or else the algorithm is not applicable.

Algorithm 3 Accelerated SQUAR Algorithm

```
procedure SQUAR-Accelerated(n)
    compute P(n,1), M(n,1)
    if P(n,1) ≠ M(n,1) then
        for b = 3,5,7,11,... do
            if b|n then
                return p = b, q = n/p
            else if P(n,b*) ≠ M(n,1) and P(n,b*) ≠ P(n,1) then
                Compute P(n,b*)
            end if
        end for
        compute M(n,b*)
        U ← min[P(n,b*), M(n,1), M(n,b*), P(n,1)]
        Q ← max[P(n,b*), M(n,1), M(n,b*), P(n,1)]
        S ← (Q − U − |A − R|)/4
        p ← gcd(n,S)
        q ← n/p
        return p, q =0
```

```
else if P(n,1) = M(n,1) then
    for b = 3, 5, 7, 11, ... do
        compute P(n, b)
        if b|n then
            p ← b, q ← n/p
        else if P(n, b_i) ≠ P(n, b_j), i ≠ j, i, j ∈ {1, 2, 3, 4} then
            Q ← max(A, Q, R, U)
            U ← min(A, Q, R, U)
            S ← (Q − U − |A − R|/4)
            p ← gcd(n, S), q ← n/p
            return p, q
        end if
    end for
end if
```

For example let's consider the semi-prime $n = 6525401$, then the sets S_1, S_2, S_3, S_4 are as follows:

$$S_1 = \{b = 2, 5, 13, 23, 37, 59, \ldots; U = 6055665\}$$
$$S_2 = \{b = 3, 19, 47, \ldots; A = 6514053\}$$
$$S_3 = \{b = \mathbf{43}, 53, 67, 83, \ldots' R = 6519205\}$$
$$S_4 = \{b = 1, 7, 11, 17, 29, 31, 41, \ldots; Q = 7012681\}$$

Therefore, the original SQUAR algorithm requires at least fifteen basic steps, because 43 is the fourteenth prime. Yet, $P_1 \neq P_2$ and $P_2 = M_1$ imply that for every $k \geq 1$, $P_k \neq M_k$. Hence, instead of counting points $P_1, P_2, \ldots, P_{43}$ in fifteen elliptic curves, compute $M_3 = 6519205$. Thus, the algorithm requires only four basic steps instead of fifteen.

7 Numerical Results and Comparison of the two methods

In this chapter we will provide some numerical results of the above methods and we will compare these results. We begin with Lenstra's Method, where the table below shows us the number n that we are trying to factorize, the point P on the curve, the iterations (no. of multiplications) the algorithm required to finish the process and the time in seconds. For our example we chose the curves of the following form:

$$E(n): \ y^2 = x^3 + ax + b$$

Table 3: **Numerical Results of Lenstra's Method**

n	**P**	**a**	**b**	**iterations**	**p; q**	**time in sec.**
24869	(21601, 5103)	18244	1148	**63**	13;1913	**30.192**
3813809	(1418679, 3635283)	1353027	772586	**125**	1933;1973	**30.147**
3858521	(3706413, 476721)	1899186	3173143	**104**	1913;2017	**30.716**
4549289	(1402101, 959413)	3785220	2490211	**83**	2113;2153	**30.994**

Notice that while $n = 4549289$ is greater than $n = 3813809$, the algorithm required forty two less iterations to stop. Hence, luck factor is of great significance regarding these algorithms, as well in cryptography in whole. With that in mind, imagine picking a random point P and an elliptic curve E. If you get lucky, then the algorithm may end in just two iterations. Although, instead of choosing the luck factor, we should study the mean performances of these algorithms and then choose our favorite based on their performance with different semi-primes.

Next, we will take a look on how SQUAR algorithm performs. We built a code in Python3 where we defined a function for counting points on elliptic curves. The function was based on two successive loops where we counted how many points $1, \ldots, n-1$ satisfy the equation $y^2 = x^3 + ax + b$, hence the number of points on the elliptic curve. The time frame required by the function to count the points on an elliptic curve with such great semi-primes, was out of reach. So, we tried smaller semi-primes than the previous example.

Table 4: SQUAR and upgraded SQUAR algorithm results

n	**A**	**Q**	**R**	**U**	**iterations**	**p; q**	**time in sec.**
8077	6169	**10149**	9945	**6045**	10	41; 197	490.34
8077	6169	**10149**	9945	**6045**	**5**	41; 97	**278.38**

Table 4 clearly demonstrates the difference of SQUAR algorithm and the accelerated SQUAR algorithm. In the first case, one can notice, that the algorithm counted points on ten different elliptic curves in order to factor n. On the other hand, after performing the acceleration, the algorithm counted points on only **five** different curves, thus reducing the computational time almost in half. Although, as mentioned above the accelerated algorithm is effective in only 25% of the cases.

In conclusion, the SQUAR algorithm works in cases where $n(\text{mod}4) = 3$ and $n(\text{mod}4) = 1$. In the first case both the algorithms (the ordinary and the accelerated one) stop in one iteration, meaning we have to compute the points on only one elliptic curve. It is obvious that the SQUAR method requires an aglorithm that counts points on elliptic curves, sufficiently fast. Contrary to SQUAR algorithm, Lenstra's Method computes the factors of a large semi-prime relatively fast, thus, in terms of speed, someone would choose the Lenstra's Method.

8 Conclusion

Two algorithms for integers factorization were proposed. The first one, namely Lenstra's Method is based on elliptic addition and multiplication of a point P on an elliptic curve E over $\mathbb{Z}_n$. The algorithm may sometimes require numerous additions/multiplications, thus making it a complicated procedure. On the other hand the SQUAR method and its enhanced modification were introduced. Both of them are computationally efficient as an algorithm that counts points on elliptic curves. Finally, we provided some examples with numerical results to thoroughly understand both the factorization methods.

References

[1] D. PARKER, "Elliptic curves and lenstra's factorization algorithm," *University of Chicago: REU*, vol. 2014, 2014.

[2] B. S. Verkhovsky *et al.*, "Integer factorization of semi-primes based on analysis of a sequence of modular elliptic equations," *Int'l J. of Communications, Network and System Sciences*, vol. 4, no. 10, p. 609, 2011.