PAIRING FOR ELLIPTIC CURVES WITH EMBEDDING DEGREE 1 DEFINED OVER PRIME ORDER FIELDS

Bogdan TUDOR
INTRASOFT INTL., BUCHAREST
25-29 DECEBAL BLV., bogdantudor74@yahoo.com

Abstract: In the last years, prime order pairing-based cryptography has received much attention. The embedding degree 1 is very appealing because the computations can be performed without any extension fields, but the efficiency parameter and this particular value of the embedding degree imposes some limits about the possible curves. This paper presents an analysis of the existing algorithms for constructing such curves and how they are related to the concept of pairing friendly. The original contribution of this paper consists in the proposal of two new algorithms. Their results are analyzed from the efficiency and computational point of view. In addition, it is presented the results of pairing implementation in Java language for the curves generated with the successful algorithm.

Keywords: *computer software, polynomial methods, data privacy, algorithms, elliptic curves.*

I. INTRODUCTION

One possible choice in implementing data privacy is to encrypt the data. There are multiple possibilities in doing this, but the advances in the mathematical theory suggest that the asymmetric encryption with elliptic curves presents some advantages. Although almost every classic asymmetric system can be modeled with these curves, there are many cryptographic constructions that can be implemented only with them. This area is mainly represented by pairing based operations, which require a special form of the elliptic curve equation (lower embedding degree). The most used construction theory for these is represented by the complex multiplication theory, which gives a framework and not an implementation.

The pairing implementation is heavily dependant on the selection of the right parameters. Their computation can be performed in the same field or in the extension fields. The theories that optimize the computations are well established for the extension fields of degree 2 or 3. For the degree 1, the computations are performed in the same field since the extension is included in it and thus the computation complexity is reduced. This lowered complexity is balanced by the amount of required computations.

For simplifying the generation, once the parameters are known, a search among the possible discriminant values shows that the first value which is suitable for this purpose is D=-7. The new generating algorithms are proposed for this particular discriminant. The generation of suitable values for different bit sizes is not expensive and the values for several bit sizes have been computed. For some of these, the running time of the corresponding pairing algorithm, implemented in Tate form with Java language, is presented. Although the point generation becomes harder when the bit size increases, choosing prime numbers with $q \equiv 3 mod 4$ simplifies the procedure.

II. PAIRING FRIENDLY CURVES

1. Definition

An elliptic curve with small embedding degree and large subgroup field is called pairing friendly. This name denotes that in this curve the pairing computations are optimal. If it is considered the curve $E(F_q)$ with embedding degree k, the discrete logarithm problem gives the minimum bit sizes for the elliptic curve parameters. From the pairings point of view there are two conditions that must be met: discrete logarithm problem must be unfeasible in the field $E(F_q)$ and the same problem must be unfeasible for the extension field $E(F_{q^k})$.

The later condition is derived from the general definition of the pairings (there are many extensions for this definition and not all involves the extension as it is presented):

$$e : E(F_q) \times E(F_{q^k}) \rightarrow E(F_{q^k}) \quad (1)$$

If the base field for the point has the order r, the efficiency of the computations is:

$$\rho = \frac{log\, q}{log\, r} \quad (2)$$

Obviously, this parameter must be as low as possible ($\rho = 1$ is the ideal case that can be attained only in special cases, as outlined below).
The second condition gives the generalized ρ coefficient:

$$\rho = k\frac{log\, q}{log\, r} \quad (3)$$

Manuscript received June 4, 2011; revised January 29, 2012

Regarding the embedding degree, there is a result due to Luca and Shparlinski given in [1], that basically says that the probability of constructing an elliptic curve with a small embedding degree for field of order r is much higher for $r < \sqrt{q}$ and this probability becomes almost negligible otherwise. Based on this result, in [1] it is defined the pairing friendly curve through the existence of this subfield of order r and the condition that $k < log_2(r)/8$. The last condition comes from the approximation of the boundaries given by the security conditions (computationally equivalence between elliptic curve cryptography represented by pairings and symmetric cryptography represented by Advanced Encryption System schemas for 128, 192 and 256 bits) as in [2].

Ideally, the numbers involved have to be Solina's primes (equal to a sum or difference of powers of 2) and to be of low Hamming weight (the representation have a small number of digits different from 0). This definition of Solina's primes can be extended by using other base instead of 2 [3] (for example 3 as in [4], [5]) and thus using a variant of the original Tate algorithm that uses this base (mainly uses the tripling point and derives a tripling formula for the divisors).

A central point in the elliptic curve construction is represented by the cyclotomic polynomials. The kth cyclotomic polynomial is noted as Φ_k and can be computed recursively from the first value:

$$\Phi_1(x) = x - 1 \text{ and } x^k - 1 = \prod_{d \backslash k} \Phi_d(x) \text{ for } k > 1 \quad (4)$$

There is a proposition in [1] that links this polynomial with the elliptic curve that has embedding degree k with respect to an subgroup of size r. Basically it says that E has embedding degree k if and only if $\Phi_k(q) \equiv 0 \, mod \, r$. The case of embedding degree 1 for the elliptic curve is very appealing because the computations are simplified (the efficiency is maximum as $\rho = 1$ and cannot be reached for this degree value).

2. Complex multiplication methods

There are two main methods for constructing elliptic curves with predefined parameters: Cocks-Pinch and Dupont-Enge-Morain [6]). The former is more suitable from the computations point of view because the subgroup order is more restricted in the latter (it has to be of the form of a resultant).

The general Cocks-Pinch method:

- Fix a positive integer k and a positive squarefree integer D
- Let r be a prime such that $k \mid r-1$ and $\left(\frac{-D}{r} = 1\right)$
- Let z be a root of unity in Z/rZ, such a z exists because of the above condition
- Let $t' = z + 1$
- Let $y' = (t' - 2)/\sqrt{-D}(mod\, r)$
- Let $t \equiv t'$ and $y \equiv y'$ mod r
- Let $q = (t^2 + Dy^2)/4$

If q represents a prime integer, then there exists an elliptic curve E defined over F_q with an subgroup with order r and embedding degree k.

The Cocks-Pinch method is defined mostly for the case in which:

$$\left(\frac{-D}{r} = 1\right) \quad (5)$$

so the equation $-D = a^2$ has solutions in Z_r.

III. ALGORITHMS

1. Sample curve

The same curve equation gives different types of curves depending on the underlying field order:

$$y^2 = x^3 - x + 1 \quad (6)$$

Field order	Curve order	Curve type
11	10	ordinary
97	98	supersingular
101	100	ordinary
103	112	ordinary

Table 1. Field order dependency of curve type

From the Table 1 it is easily to see that sometimes $N = q - 1$ (this happens for values 11 and 101).

The general idea is to find an elliptic curve with certain properties:

- $\# E(F_q) = q - 1$ or equivalent
- $\# E(F_q) = 2 * A$, A – prime number
- A factor has a low Hamming weight.

It is noted: $E(F_q) = F(E_q) + G(E_q)$ (this is a formal addition), where $\# F(E_q) = A$ and $\# G(F_q) = 2A$ where it is noted with # the order of the points from one set (it is an extension of the normal order). The above curve has two obvious points: (0, 1) and (1, 1). These points belong to the second set (they have the same order 2A) for q=11. As a result, the addition operation between any points from the first set with a point from the second set will give a point from the second set. The Tate pairing requires a point to be from the first set and the second point to be independent from the first one. This requirement comes from the factor $g_{jP,kP}(D_Q)$ that appears on the denominator of the Tate step. Here jP and kP represents points from the first set, and their sum belongs to the first set. D_Q represents the divisor of the second point from the pairing (Q_1-Q_2) and these points are from the second set. From the above discussion the possibility of dividing by 0 is very low. The following algorithm can achieve the generation of all these points:

- Find a point P with order A
- Add to this point (0,1): Q = P + (0,1)

The pairing $\tau(P,P) = \tau(P,D_Q)$, where $D_Q = (Q) - (0,1)$

The above algorithm is possible in this case because (0,1) has the order 2A (the order of the curve). In the general case step 2 in replaced by: find a point with order 2A and add this point to P.

2. First variant of complex multiplication

The complex multiplication has as the starting parameter the value: $-D = a_q^2 - 4q$ where $\#E(F_q) = q + 1 - a_q$.
From the constraint $\#E(F_q) = q - 1$ follows a possible value of $a_q = 2$, so $-D = 4 - 4q$.
As the q value gets bigger, the discriminant grows to computationally unfeasible values. For example D=1000003 gives a polynomial of order 105 with very big coefficients. The single possibility of handling big values for the field order (required by the cryptographic strength) is to control the discriminant of the elliptic curve.
The following algorithm follows this idea:

- find the numbers q, a_q, A with the property $q + 1 - a_q = 2 * A$, q prime, A low Hamming weight prime
- the discriminant of the elliptic curve $D < 2000000$ - in order to be computationally feasible.

For this algorithm it is fixed the interval for q (val1…val2), limit1 represents the limit for the discriminant value and limit2 represents the limit for the Hamming weight of A.

```
for q in (val1, val2):
  for $a_q$ with $|a_q| < 2\sqrt{q}$ , $-D = a_q^2 - 4q$
    if D < limit1
      if $q + 1 - a_q = 2 * A$, A –prime number with low Hamming weight < limit2
        print q,$a_q$
```

As proposed in [7], there is a variant of Cocks-Pinch method of generating ordinary pairing-friendly curves.

3. Modified cyclotomic polynomial

In [1] is presented an exception for the first cyclotomic polynomial: $\Phi_1(\chi) = \chi$ and not $\chi - 1$ as expected. All the generated curves are sparse families (there are two parameters that control the curve generation: l and χ). The first curve is the simplest one and is especially good for computations due to her special group structure:

$$k = 1, \rho = 1, D = 1 \quad (7)$$
$$p(\chi) = l^2\chi^2 + 1$$
$$r(\chi) = \Phi_1(\chi) = \chi$$
$$t(\chi) = 2$$

Form these parameters two curves can appear:

$$l\chi \equiv 0 \bmod 4 \Rightarrow E : y^2 = x^3 - x$$
$$l\chi \equiv 2 \bmod 4 \Rightarrow E : y^2 = x^3 - 4x$$

The curve field has the structure $E(F_p) \cong Z_{tr} \oplus Z_{tr}$
The other generated curves have the structure:
$E(F_p) \cong Z_{r^2/2} \oplus Z_{r/2}$ and $E(F_p) \cong Z_r \oplus Z_{r/3}$.
From the above results, it is easily to see that for $t \geq 1$ on optimal case is the first one. The independence of the fields assures that the final divisors have no points in common and the Tate divisors will give no errors in the computation. This is not exactly true, because some points have not the proper order (in fact they belong to a subgroup with different order so they appear in expansions of different points).
Again it is tested the Tate pairing of a point with itself, i.e. $\tau(P, P)$. For the example it is chosen the field with

$$p = 101 = 10^2 + 1 \quad (8)$$

From this choice results $r = 5, t = 2$ and $E : y^2 = x^3 - 4x$. These values respect the initial conditions because $\Phi_1(101) \equiv 0 \bmod 5$. Also as a corollary, from the t value it results immediately this condition. To assure the independence of the divisor it is chosen a point with order 10 (r*t). A quick search reveals point (93, 5). The second point is chosen also to have order 10: (88, 73):
$\tau((93,5),(93,5)) = \tau((93,5),(8,51) - (88,73)) = 6$.

However not all the pairings can be computed here as it shows the following computation: $4 * (93,5) = (2,0)$.
The point (2, 0) has order 2. So it is not suitable for computing the Tate pairing (it falls when it is computed the vertical line on 2*P). The expansion of the point (93, 5) is: $(93,5),(22,37),(63,27),(82,61),(2,0),(82,40),(63,74),$ $(22,64),(93,96),\infty$. The orders of those points are: 10, 5, 10, 5, 2, 5, 10, 5, 10.

From the above values results that the t*r field is not suitable for implementing the Tate pairing on it. At most a point with order 5 can be implemented. The expansion of the point (22, 37) has the right properties. It does not contain false points (of order different from 5) and thus it is suitable for computation. The density of the points with different orders is represented in Table 2.

Order	Number of points
1	1
2	3
5	25
10	72

Table 2. Point order and generated field sizes

From the Table 2, a brute-force approach is the easiest for finding a point with the order r (in the specified field is almost one quarter of the total number of points). From the formula of p results that a p value with a small l it is desirable (as l grows). The minimum value of l is 2.

4. Second variant of complex multiplication

$$k = 1, \rho = 3, D = 1 \quad (9)$$
$$p(x) = \frac{x^6 + 3x^4 - x^2 + 1}{4}$$
$$r(x) = \Phi_4(x) = x^2 + 1$$
$$t(x) = -x^2 + 1$$
$$E : y^2 = x^3 + ax, a \in F_p$$

This one has $E(F_p) \cong Z_{r^2/2} \oplus Z_{r/2}$.
The equation of these curves can be deduced directly from the complex multiplication theory. The generalized equation is:

$4p = t^2 - Ds^2$, so D is the squarefree part. Following the discriminant values, the generalized values 1 and 3 defines special forms of elliptic curves:

$$D = 1 \Rightarrow E_a : y^2 = x^3 + ax, a \in F_p$$
$$D = 3 \Rightarrow E_b : y^2 = x^3 + b, b \in F_p \tag{10}$$

The above forms are important because there is no need in computing the Hilbert polynomial (or other polynomial variants as Weber ones) as in [8].
For embedding degree 1, $r \mid p-1$.

$$r \mid \#E(F_p) = p + 1 - a_p \tag{11}$$

it is obvious that a trivial case is one in which $a_p = 2$. In this case $\#E(F_p) = p - 1$. From the complex multiplication equation in the case of D=1 follows:

$$4p = t^2 - Ds^2 = 4 + s^2 \tag{12}$$

From here there are possible a number of choices:

- s represents the subgroup order and is a prime number. This possibility contradicts the integrality of p, because

$$p = \frac{4 + s^2}{4} = 1 + \left(\frac{s}{2}\right)^2 = 1 + l^2 x^2 \tag{13}$$

- s has in its composition the subgroup order.

This is describing the first curve in which $s = 2lx$ where x represents the subgroup order and it is prime. As x describe the order of the group in which the pairing will be computed, it is desirable that l to be very small. The smallest value for it is l=2, as l=1 does not give a prime number for p.
This represents the only possibility for fixed t (t=2). In the general case:

$$\#E(F_p) = p + 1 - a_p = (p-1) - (a_p - 2) \tag{14}$$

From the fact that $r \mid p-1$ and $r \mid \#E(F_p)$ results that $r \mid (a_p - 2)$ for $a_p \neq 2$. The rest of the parameterized functions fulfil this more general condition. The simplest case in this situation is one in which $r = a_p - 2$. For example:

$$r(x) = \Phi_6(x) = x^2 - x + 1 \tag{15}$$
$$t(x) = -x^2 + x + 1 \tag{16}$$

From the above equations the curve $E_c : y^2 = x^3 + x$ has the same properties (order 100 and same group structure). This fact is verified by computing the pairing:

$$\tau((68,84),(4,13)) = 87 \tag{17}$$

where the points belongs to a groups with order 5 (the same like in the case of a=-1). As discussed previously the efficiency of this type of curve in the prime order fields is represented by $\rho = \frac{\log_2(4x^2+1)}{\log_2(x)} \approx 4$ for large values of x where $4x^2 + 1 \approx 4x^2$. The value of $\rho \approx 2$ represents the efficiency regarding the pairing in the composite prime fields.
In the case in which $r = a_p - 2$, the maximum efficiency is when $a_p \approx 2\sqrt{p}$. This case can happen when p>15 due to Hasse bound theorem. Observation: the trivial choice will be $p = x^2 + 1$ and $r = a_p \approx 2x$. In this case of k=1, the efficiency cannot be greater than the above value because the complex multiplication equation can be written as:

$$D = r(4h - a^2 r) \tag{18}$$

where h represents the cofactor $\#E(F_p) = rh$ and $ar = t - 2$. D is represented as a product of two factors: one is r and the other one is $4h - a^2 r$. As r is getting larger, the discriminant D is getting larger because the remaining factor cannot be small (the efficiency is directly proportional with the inverse of h, so h is small). In [9] the complex multiplication does not give any curve with embedding degree 1 and the presented complex multiplication discriminators does not have practical values:
($3 \times 38024920917822051230350291383772163112^2$).

From the complex multiplication theory it is known that a computationally feasible limit of the discriminant is 10^{12}. So that algorithm is interesting from the theoretical point of view because it provides an alternative to the classic Cocks-Pinch or Dupont-Enge-Morain algorithms. From the implementation point of view, due to the impractical values of the discriminant D the algorithm cannot be implemented.
The following algorithm searches for desired pairing:

```
Input n, limit, cofactor
while n < limit:
  if is_prime(n):
    p = cofactor^2*n^2 + 1
    if is_prime(p):
      E = EllipticCurve(GF(p),(-1,0))
      a =  E.order()
      if a == p - 1:
        print n
  n = n+2
```

The above algorithm represents the pseudo-code program that performs a brute-force finding of the desired curve. The searching for n with cofactor = 2 gives multiple results , but the filtering with the order curve is the most important because as it was shown above there might be fake results, giving ordinary curves with different embedding degree. A possible approach can be to keep the determinant D at low values. From the definition of D this means that $t \approx 2\sqrt{p}$, otherwise if $t = 2\sqrt{p} + z$ where z represents the free part:

$$D = t^2 - 4p = (2\sqrt{p} + z)^2 - 4p = z^2 + 4z\sqrt{p} \quad (19)$$

In this relation z is negligible and the important part is represented by $\sqrt{p}$. As the suitable values for p are in the range of $2^{250} \approx 10^{84}$ (a very rough approximation) results that any deviation of D from $2\sqrt{p}$ implies a value outside the limits. The only possibility is $t = [2\sqrt{p}]$ (the integer part).

```
qq=1000001
limitq = qq + 1000
while qq < limitq:
   if is_prime(qq):
      tq = floor(2*sqrt(qq))
      p1q = qq + 1 -tq
      p2q = qq - 1
      rq = find_max_prime_divisor(p1q,p2q)
      print qq,rq
      if rq > 10:
         if is_prime(rq):
            print qq, rq
   qq = qq+2
```

There are a number of reasons why this algorithm does not give practical results:

- It works only with numbers of the form $2^x + d$ and not $2^x - d$ because in the latter case the square root part is not so close to the main part $2^{x/2}$ and in this case the discriminant is getting larger values.
- In the simulations with $p \in [1000000, 1010000]$ the best value were 10000037 with a value of r of 37. This is not acceptable as $\rho \approx 4$ and exceeds the optimum value of 2. The majority of combinations had 1 as the maximum prime divisor (the resulting numbers are prime between them) or 2 (small multiples of 2).

5. Generation without complex multiplication

As a result, it is needed a family of curves that does not need to be computed by complex multiplication method. A family of such curves is given by theorem 4.21 from [10]:

Let p be an odd prime and let $k \neq 0 \bmod p$. Let $N_p = \#E(F_p)$ where E is the elliptic curve $y^2 = x^3 - kx$, then:

- if $p \equiv 3 \bmod 4$ then $N_p = p + 1$ and thus E represents a supersingular curve.
- if $p \equiv 1 \bmod 4$ write $p = a^2 + b^2$ where a and b are integers with b even and $a + b \equiv 1 \bmod 4$, then:

$$N_p = \begin{cases} p + 1 - 2a, & \text{if k is a fourth power mod p} \\ p + 1 + 2a, & \text{if k is a square mod p, but not a fourth power} \\ p + 1 \pm 2a, & \text{if k is not a square mod p} \end{cases}$$

This theorem gives the possibility of constructing curves with embedding degree 1 if k=1 (a fourth power of 1) and a=1 for example. Such an example is represented by the curve with p=101. An quick search with the following program:

```
b = 1000000
limit = b + 1000
while b < limit:
   p = 1 + b^2
   if is_prime(p):
      print b
   b = b + 4
```

gives 18 possibilities. Among these it can be chosen 1000884. If the same program is run with the input 2^{256} only one suitable value is found:
p=11579208923731619542357098500868790785326998 4665640564039457584007913129640476
In the general case:

$$\begin{aligned} p + 1 - 2a &= 2k_1 A \\ p - 1 &= 2k_2 A \end{aligned} \quad (20)$$

where A is a prime number. Form the above equations:

$$p\left(1 - \frac{k_1}{k_2}\right) + 1 + \frac{k_1}{k_2} \leq 4\sqrt{p} \quad (21)$$

For $k_1 / k_2 = 1/2$, it results the following $p + 3 \leq 8\sqrt{p}$ (cannot be fulfilled). The conclusion here is that if $a \neq 1$ it is impossible to find a suitable value. For the rest of the cases there is the same conclusion (every case ends in a equation like $kp \leq \sqrt{p}$ which does not provide suitable values). Form the above results that the only possibility in this case it is represented by $a = \pm 1$ and $N_p = p - 1$.

If it is considered the case with $a = -1$, then $b \equiv 2 \bmod 4 \Rightarrow b = 4c + 2$.
The algorithm for discovering suitable values becomes

```
c = 1000000
limit = c + 1000
while c < limit:
   p = 1 + (4*c + 2)^2
   if is_prime(p):
      cc = maximum(p)
      print c, cc
   c = c + 1
```

c	cc	c/cc
1000041	2000083	0.4999
1000976	2001953	0.4999
1000998	2001997	0.4999

Table3. Suitable prime numbers

From the Table 3 it is easily to see that the expected ρ value will be constant among these generated curves. The value corresponds to an acceptable value of 2. For example:

- Case 1: c = 1000041, p=16001328027557, k=25

$$E(F_p): y^2 = x^3 - 25 \quad (22)$$

The generators for this curve (4253461776013, 382634269242) or (7516808453469, 9565046703970), had order 4000166 and corresponds to $\rho = 2$.

- Case 2: c=1000998, p=16031967952037, k=25

The generators for this curve (6878545080939, 4270041569676) or (14443809921043, 7687957344804), had order 4003994 and corresponds to $\rho = 2$. In this case, the pairing took 32 ms and has the value of 3419889792843. The final Tate power can be computed by addition / subtraction chains (for example in [11] is presented a Lucas chain) [12]. In the present case, however the Tate pairing value is represented by a BigInteger and thus the final power can be computed by the following algorithm:

```
A=1
B=coefficient
For(int i=0; i< power.bitLength(); i++) {
   If(power.testBit(i)) {
         A = A.multiply(B).mod(modulus);
   }
   B = B.pow(2).mod(modulus);
}
return A;
```

The final power in this case is represented by the number $4003994 = (1111010001100010011010)_2$ and thus only 22 iterations are performed. The straightforward implementation of type $a^b \bmod c$ is not computationally feasible due to the size of the coefficients and thus the reduction modulo c after each step is mandatory. Because the embedding degree is 1, there are no further optimizations possible. For the current case, k=1 implies z=1 and $t' = 2$. From this point the discriminant is not so important because $y' = 0$ and if $y = y' + k_1 r$ and $t = t' + k_2 r$:

$$q = \frac{t^2 + Dy^2}{4} = \frac{(2 + k_2 r)^2 + D(k_1 r)^2}{4} = 1 + k_2 r + r^2 \frac{k_2^2 + Dk_1^2}{4}$$

The mandatory condition here is $k_2^2 + Dk_1^2 \equiv 0 \bmod 4$. In this case, q-1 is divisible by r, and thus the embedding degree is 1. The efficiency coefficient has the minimum value at k_1=

k_2=2, thus r=101 $\rho = \frac{\log q}{\log r} \approx 2.45$ which is not convenient.

Another option in this case is k_1=0:

$$q = \frac{t^2 + Dy^2}{4} = \frac{(2 + k_2 r)^2}{4} = 1 + k_2 r + r^2 \frac{k_2^2}{4} \tag{23}$$

The maximum efficiency is when k_2=2 and for r=101 $\rho \approx 2.00$ which is optimal (q=10404). The only problem in this approach is that the group order q is not a prime number, but a composite one:

$$q = 1 + 2k_3 r + (k_3 r)^2 = (1 + k_3 r)^2 \tag{24}$$

Because of this situation the case in which k_1=0 it is not useful for generating elliptic curves over prime order fields, but is useful for composite ones, having multiple applications, especially for identity based encryption schemas [13].

IV. NEW ALGORITHMS

1. Minimal Hilbert representation

In order to simplify the curve generation, the complex multiplication theory can be used for minimal representations of the Hilbert generated polynomials.

Discriminant	Hilbert polynomial
3	x
4	x-1728
7	x+3375
8	x-8000
11	x+32768

Table4. Minimal Hilbert polynomials

The polynomials presented in Table 4 have the property that they all have degree 1, thus the root modulo q is easy to compute. The values that are skipped (1,2,5 …) from the first possible discriminant values cannot appear as discriminant values. The value of 4 is not suitable because from the elliptic curve reconstruction equation:

$$y^2 = x^3 + \frac{3j}{1728 - j} x + \frac{2j}{1728 - j} \tag{25}$$

the invariant $j \neq 1728$.

The value 7 seems to be the best choice since it provides a simple Hilbert polynomial and provides suitable values for the invariant.

The main steps of the algorithm are depicted below:

- Choose the bit length of the resulting subgroup order.
- Compute a prime number with properties (low Hamming weight, …), the desired bit length and with additional
- Compute the expected value of the group order (q).
- Compute the invariant of the elliptic curve with Hilbert polynomial
- Compute and verify the properties of the resulting curve

From the algorithm presented below it is obvious that $\rho > 2$.

- Bit length 12: the chosen value is 3389 – prime number. The order q is computed with the help of the following algorithm:

```
k3=0
r = 3389
y=2
kmax=1000
found = false
while ((found == false) & (k3 < kmax)):
   q = 1+2*k3*r+7*r^2+r^2*k3^2
   print q
   if is_prime(q) :
      print "found",k3
      found = true
   k3 = k3 + 1
```

For the above mentioned values the algorithm gives the values $k3 = 15$ and $q = 2664696143$.

$j = q - 3375 = 2664692768$, with resulting curve:

$y^2 = x^3 + 1691870565 * x + 1127913710$

This curve has order:

$N = 2664594472 = 2^3 * 29 * 3389^2$

The efficiency coefficient $\rho = \frac{\log(2664696143)}{3389} = 2.670$

- Bit length 50: the prime value found is $r = 2^{50} + 2^{23} + 1$. For these values the above algorithm gives

$q = 1561745562753003589950227984942359$

$j = 1561745562753003589950227984938984$

$y^2 = x^3 + 1016374096394811860126338847343438 * x +$
$6775827309298745734175592315622 92$
and order
$N = 1561745562753003511137233918756048 =$
$2^4 * 7 * 11 * 1125899915231233^2$
The efficiency coefficient is improved

$$\rho = \frac{log(1561745562753003589950227984942359)}{1125899915231233} = 2.205$$

For different bit values (100,200 and 600) examples of the resulted curves are in Appendix A and the pairing results are presented in Appendix B. Java implementation is performed with a modified JPBC library [12]. This algorithm is getting around $\rho \approx 2$ but somewhat greater and 2 is the theoretical bound.

2. Cyclotomic polynomial representation
The idea of cyclotomic polynomial representation is based on [6] and [14]. The representation of the cyclotomic field is performed through:

$$q = (-1)^{\frac{q-1}{2}} \prod_{j=1}^{(q-1)/2} (\zeta_q^j - \zeta_q^{-j})^2 \qquad (26)$$

and depending of the modulus (1 or 3) of the prime q:

$$\sqrt{q} = \prod_{j=1}^{(q-1)/2} (\zeta_q^j - \zeta_q^{-j}) \text{ for mod} = 1 \qquad (27)$$

$$\sqrt{-q} = \prod_{j=1}^{(q-1)/2} (\zeta_q^j - \zeta_q^{-j}) \text{ for mod} = 3 \quad (28)$$

In the current case of D=7, the current formula becomes:

$$\sqrt{-7} = \prod_{j=1}^{3}(\zeta_7^j - \zeta_7^{-j}) = (\zeta_7^1 - \zeta_7^{-1})(\zeta_7^2 - \zeta_7^{-2})(\zeta_7^3 - \zeta_7^{-3})$$

$$= \frac{(\zeta_7^2 - 1)(\zeta_7^4 - 1)(\zeta_7^6 - 1)}{\zeta_7^6} = \zeta_7(\zeta_7^2 - 1)(\zeta_7^4 - 1)(\zeta_7^6 - 1)$$

A possible choice for the 7th root of unity as a polynomial is x:

$$\Phi_7(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \qquad (29)$$

and thus the formula:

$$\sqrt{-7} \mapsto D(x) = -2x^4 - 2x^2 - 2x - 1 \qquad (30)$$

represents the $\sqrt{-7}$ value in the polynomial field $Q(\zeta_7)$.
For k=1, the kth root of unity is x^7:

$$t(x) = 1 + x^7$$

$$q(x) = \frac{1}{4}(t(x)^2 + \frac{(t(x)-2)^2 D(x)}{D}) = \frac{1}{4}((1+x^7)^2 + \frac{(x^7-1)^2(-2x^4 - 2x^2 - 2x - 1)}{7})$$

$$r(x) = \Phi_7(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \quad (31)$$

If there exists a value x_0 for which $q(x_0)$ and $r(x_0)$ are both primes, then the generated curve will have the desired properties. These generated curves have $\rho \approx 22/6 = 3.66$ so they are suboptimal. From [14]:

$$\begin{cases} a(x) = g(x) + 1 \\ b(x) = (a(x) - 2)h(x) \\ p(x) = (a^2(x) + b^2(x)/D)/4 \end{cases} \qquad (32)$$

it is easy to see that in order to minimize the ρ parameter it is required that the representation of g(x) (the kth root of unity) to be minimal. In the above case, the degree of this representation was too high (7). This condition is equivalent that the degree of g(x) to be lower than $\Phi_n(x)$.
In lemma 2.3 from [14], a required condition for $\sqrt{-7} \in Q/\Phi_n$ is that n=7k. The efficiency parameter is proportional with the value of $\frac{deg(x^{7k})}{deg(\Phi_{7k})} = \frac{7k}{deg(\Phi_{7k})}$.

k	7k	Φ_{7k}	deg(Φ_{7k})	ρ
1	7	$x^6 + x^5 + x^4 ...$	6	1.166
2	14	$x^6 - x^5 + x^4 ...$	6	2.333
3	21	$x^{12} - x^{11} + x^9 ...$	12	1.75
4	28	$x^{12} - x^{10} + x^8 ...$	12	2.333
5	35	$x^{24} - x^{23} + x^{19} ...$	24	1.458
6	42	$x^{12} + x^{11} - x^9 ...$	12	3.5
7	49	$x^{42} + x^{35} + x^{28} ...$	42	1.166
8	56	$x^{24} - x^{20} + x^{16} ...$	24	2.333
9	63	$x^{36} - x^{33} + x^{27} ...$	36	1.75
10	70	$x^{24} + x^{23} - x^{19} ...$	24	2.916
11	77	$x^{60} - x^{59} + x^{53} ...$	60	1.283
12	84	$x^{24} + x^{22} - x^{18} ...$	24	3.5
13	91	$x^{72} - x^{71} + x^{65} ...$	72	1.263

Table5. Efficiency coefficients

From the Table5 it is easily to see that even values of k give ρ values that are not optimized. A search for the even values ((2k+1)*7) reveals that the value 1.166 is the minimum value and it is reached in the interval (1, 5000) for 1, 7, 49, 343, 2401. For this particular case,

$$\rho_{min} \approx log(\frac{a(x)^2}{r(x)}) = 2.3 \qquad (33)$$

which is much worse than the previous results.
As a conclusion, this method does not give optimal curves from the embedding degree point of view. The polynomial x^{7k} cannot be reduced further with $\Phi_{7k}(x)$ because this will give the trivial result (1). Another possible construction is to use the relation $\zeta_k = -\zeta_{2k}$ for odd k. In order to apply this relation k=2 must be considered because it provides ζ_2 as $\zeta_2 = \zeta_{14}^7$, thus $\zeta_1 = -\zeta_{14}^7$.

As $\Phi_{14} = x^6 - x^5 + x^4 - x^3 + x^2 - x + 1$, the same problem of the representation occurs in this case too. The efficiency coefficient is $\rho \geq 2$.

All the above value satisfies $q \equiv 3 mod 4$ and thus every element has the Lagrange coefficient equal to 1:

$$y = \sqrt{x^3 + Ax + B} = \sqrt{a} = a^{\frac{p+1}{4}} mod\ p \qquad (34)$$

Since this operation presents a very large exponent, it is implemented by a 'double and add' algorithm presented in [15].

V. CONCLUSIONS

The generation of elliptic curves with embedding degree 1 is not hard and a new algorithm was developed. The running times become larger as the bit size of the underlying field grows. The solutions density for this new algorithm is much higher than the optimal ones. The efficiency of this algorithm grows as the bit size of the underlying field grows. It is computed (Appendix A) that for 600 bits, the computed efficiency is $\rho = 2.030$.

REFERENCES

[1] D. Freeman, M. Scott, E. Teske, "A taxonomy of pairing friendly curves", *Journal of Cryptology*, vol. 23, pp. 224-280, 2010

[2] N. Koblitz, A. Menezes, "Pairing based cryptography at high security levels", *Proceedings of Cryptography and Coding 2005*, LNCS 3796, pp. 13-36, 2005

[3] S. Kwon, "Efficient Tate pairing computation for supersingular elliptic curves over binary fields", *Information Security and privacy*, LNCS vol. 3574, pp. 134-145, 2005

[4] T. Kerins, W.P. Marnane, E.M. Popovici, P.S.L.M. Baretto, "Efficient hardware for the Tate pairing calculation in characteristic three", *Cryptographic Hardware and Embedded Systems (CHES)*, pp.412-426, 2005

[5] Y. Kawahara, T. Takagi, E. Okamoto, "Efficient implementation of Tate pairing on a mobile phone using Java", *Computational Intelligence and Security*, Springer-Verlag Berlin, Heidelberg, pp. 396-405, 2007

[6] F. Brezing, A. Weng, "Elliptic curves suitable for pairing based cryptography", *Cryptology ePrint Archive*, Report 2003/143

[7] Y. Nogami, E. Yanagi, T. Izuta, Y. Morikawa, "Ordinary pairing friendly curve of embedding degree 1 whose order has two large prime factors", *Memoirs of the Faculty of Engineering*, Okayama University, vol. 45, pp. 46-53, 2005

[8] E. Kostantinou, A. Kontogeorgis, Y. Stamatiou, C. Zaroliagis, "On the efficient generation of prime order elliptic curves", *Journal of cryptology*, vol. 23, pp. 477-503, 2010

[9] P. Duan, S. Cui, C.W. Chan, "Special polynomial families for generating more suitable pairing-friendly elliptic curves", *The 5th WSEAS International Conference on Electronics, Hardware, Wireless Optimal Communications*, 2005

[10] L. Washington, "Elliptic curves: number theory and cryptography",*Chapman & Hall/CRC*, New York, 2003

[11] A. Caro, "JPBC" – Java Pairing Based Cryptography at [Online]: http://gas.dia.unisa.it/projects/jpbc/index.html

[12] B. Tudor, "Hidden Vector Encryption in Prime Order Fields", *IFAC, WICS 2010 Preprints*, pp. 7-14, 2010

[13] B. Waters, "Efficient IBE without random oracles", *Proceedings of Eurocrypt 2005*, vol. 3494, pp. 114-127, 2005

[14] A. Murphy, N. Fitzpatrick, "Elliptic curves for pairing applications", *Cryptology ePrint Archive*, Report 2005/302

[15] D. Hankerson, A. Menezes, S. Vanstone, "Guide to elliptic curve cryptography", *Springer-Verlag*, 2004

Appendix A

- Bit length 100.

$r = 2^{100} + 2^{49} + 1$

q=2211146748900372583038565002466254529674768770744071721563719083

$\rho = 2.104$

- Bit length 200.

$r = 2^{200} + 2^{57} + 1$

q=15328235276323889388197536205009878486021136335601637622998557402731672589194707067000336498758517703246544101942857870350283

$\rho = 2.062$

- Bit length 600.

$r = 2^{600} + 2^{25} + 1$

$\rho = 2.030$

Appendix B

For some of the intended subgroup sizes the Tate pairing was computed. The results are presented along with the paired points.

- Tate pairing: 1009540039 took 16 ms

q=1561745562753003589950227984942359
a=1016374096394811860126338847343438
b=677582730929874573417559231562292
P_x=362643955287941103437571966411523
P_y=916115057470821650797876771657217
Q_x=1085175173804325577041166051867696
Q_y=449838472319049454517271473610379

- Tate pairing: 507565679276123973313698906530379 took 62 ms

q=2211146748900372583038565002466254529674768770744071721563719083
a=1088024590728754763082468493277045879681235426874067037594845896
b=2199447559752751564080688997162200272904002798412092506105709986
P_x=1471379789741545005048748290249427995344539480843895353908073109
P_y=338839908341084521055252672835916935601566260563516145456245240
Q_x=694542554370750504780198244526226138705933204869813803436632119
Q_y=1606608703232757748737737981482420487566980128115209770057645995