Cryptographically Strong Elliptic Curves of Prime Order

Marcin Barański, Rafał Gliwa, and Janusz Szmidt

Abstract—The purpose of this paper is to generate cryptographically strong elliptic curves over prime fields $\mathbb{F}_p$, where $p$ is a Mersenne prime, one of the special primes or a random prime. We search for elliptic curves which orders are also prime numbers. The cryptographically strong elliptic curves are those for which the discrete logarithm problem is computationally hard. The required mathematical conditions are formulated in terms of parameters characterizing the elliptic curves. We present an algorithm to generate such curves. Examples of elliptic curves of prime order are generated with Magma.

Keywords—Mersenne primes, elliptic curves, security requirements, search algorithm, Magma.

I. INTRODUCTION

Information security is of paramount importance to many institutions of our society: governments, military, financial, businesses, etc. Many confidential information about research, products, financial status, customers, or employees, is nowadays processed and stored on computers, or transmitted to other computers. The information security is a very important area of radio communication. TRANSEC and COMSEC mechanisms require symmetric cryptography and public key cryptography.

The public key cryptography was introduced in the seminal papers of Diffie and Hellman [6] and Rivest, Shamir and Adleman [21]. The use of elliptic curves over finite fields was proposed by Miller [18] and Koblitz [17]. The cryptosystems with elliptic curves have advantage over RSA cryptosystem since we obtain the comparable security with much shorter keys, as is shown in Table 1.

The purpose of this paper is to generate cryptographically strong elliptic curves over prime fields $\mathbb{F}_p$, where $p$ is a Mersenne prime $p = 2^{521} - 1$ or $p = 2^{607} - 1$, one of the special primes or a random prime. We search for elliptic curves whose order is a prime number and the order of the twisted curve has a big prime factor. The sizes of the keys in this region are given in the Standards [26]–[29]. Our purpose is to generate independently elliptic curves with good cryptographic properties. We use the mathematical tool Magma [31] to generate the curves.

Cryptographically strong elliptic curves are those for which the Elliptic Curve Discrete Logarithm Problem (ECDLP) is resistant to known attacks. In general, ECDLP is computationally hard problem. The corresponding requirements are formulated in terms of parameters characterizing the curves. The elliptic curves with prime order have advantage over non-prime case since each non-neutral element of the curve is a generator of the group of points on the curve. The arithmetic on elliptic curves is presented in papers [5], [12] and over the field $\mathbb{F}_{2^{521} - 1}$ in [13]. We check the class number criterion and the twist security for our examples of curves. The results of the numerical experiments are given in the Appendix.

### Table 1. The sizes of cryptographic keys.

<table>
<thead>
<tr>
<th>Symmetric alg.</th>
<th>80</th>
<th>112</th>
<th>128</th>
<th>160</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECC order q</td>
<td>160</td>
<td>224</td>
<td>256</td>
<td>320</td>
<td>512</td>
</tr>
<tr>
<td>RSA modulus n</td>
<td>1024</td>
<td>2048</td>
<td>3072</td>
<td>7680</td>
<td>15360</td>
</tr>
</tbody>
</table>

II. BASIC NOTIONS

Let $p$ be a prime number. The prime field $\mathbb{F}_p$ consists of integers $\{0, 1, \ldots, p-1\}$ with arithmetic operations of addition and multiplication modulo $p$. For a prime $p > 3$ we define an elliptic curve $E$ over the field $\mathbb{F}_p$ by the equation

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

where $a, b \in \mathbb{F}_p$ and $4a^3 + 27b^2 \neq 0$. We define the set of rational points of the elliptic curve $E$ over the field $\mathbb{F}_p$ as the set $E(\mathbb{F}_p)$ of solutions $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$ of the equation (1) together with the neutral element $O$. The set $E(\mathbb{F}_p)$ has the structure of an abelian group with operations of addition and doubling of points defined according to the rules, see [4], [25].

Addition of points:

1. $P + O = O + P = P$ for all $P \in E(\mathbb{F}_p)$.
2. If $P = (x, y) \in E(\mathbb{F}_p)$, then $(x, y) + (x, -y) = O$.
3. Let $P = (x_1, y_1) \in E(\mathbb{F}_p)$ and $Q = (x_2, y_2) \in E(\mathbb{F}_p)$, where $P \neq \pm Q$. Then $P + Q = (x_3, y_3)$, where

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$$

and

$$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1.$$
Let \( P = (x_1, y_1) \in E(F_p) \), where \( P \neq -P \). Then \( 2P = (x_3, y_3) \), where
\[
x_3 = \left( \frac{3x_1^2 + a}{2y_1} \right)^2 - 2x_1
\]
and
\[
y_3 = \left( \frac{3x_1^2 + a}{2y_1} \right) (x_1 - x_3) - y_1.
\]

The operations of adding and doubling points on an elliptic curve \( E(F_p) \) require performing arithmetic operations of addition and multiplication in the basic field \( F_p \). For elliptic curves over the real field \( \mathbb{R} \) the operations of addition and doubling of points on the curve have a geometric interpretation which is shown on Figures 1 and 2. An example of the elliptic curve over the finite field \( F_{23} \) is depicted in Figure 3.

Let \( E \) be an elliptic curve over the field \( F_p \). The order of the curve denoted \( q = \#E(F_p) \) is the order of the group \( E(F_p) \). In this case the Hasse Theorem says that
\[
\#E(F_p) = p + 1 - u, \quad \text{where} \quad |u| < 2\sqrt{p}.
\]
The integer \( u \) is called the trace of the curve. The group \( E(F_p) \) has the structure of an abelian group of rank 1 or 2, i.e., it is isomorphic to the group \( \mathbb{Z}_{n_1} \) or to the group \( \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \), where \( n_2|n_1 \). Here \( \mathbb{Z}_{n_1} \) and \( \mathbb{Z}_{n_2} \) are cyclic groups. If \( n_2 = 1 \), then it is a point \( P \in E(F_p) \), named the generator of the group, which satisfies
\[
E(F_p) = \{kP : 0 \leq k \leq n_1 - 1\},
\]
where \( kP = P + \cdots + P \), \( k \) times. In the special case, when the order \( q \) is a prime number, the group \( E(F_p) \) is a simple one and each non-neutral element is a generator of this group. It is important in cryptographic applications. In our investigations we consider only prime order elliptic curves over prime fields \( F_p \). In particular, we take primes of the form \( p = 2^{113} - 1 \), where \( p' \) is another prime which are called Mersenne primes and it has been found together 51 of them up to now [30].

Let \( E \) be an elliptic curve over the prime field \( F_p \) and \( P_0 \) a point in \( E(F_p) \). We denote \( \langle P_0 \rangle \) the subgroup of \( E(F_p) \) generated by the point \( P_0 \). Let \( Q \in \langle P_0 \rangle \) be an arbitrary point. Then it has the form \( Q = kP_0 \) for some integer \( k \). We call \( k \) the discrete logarithm of \( Q \) to the base \( P_0 \). In this case the Elliptic Curve Discrete Logarithm Problem (ECDLP) is to find the number \( k \). The best known algorithm to solve ECDLP, the Pollard rho method [20], finds discrete logarithms in time \( O(\sqrt{p}) \). Hence for large values of the prime \( p \) it is infeasible by today’s computers. The record of calculating discrete logarithm on elliptic curves over prime fields is for 113-bit prime [10].

III. THE CRYPTOGRAPHIC REQUIREMENTS

To resist the known attacks on ECDLP the elliptic curves have to satisfy special requirements. We follow partly the ECC Brainpool Standard [26]. The NATO [27] and the BSI [26] requirements for cryptographically strong elliptic curves are in fact the Brainpool conditions. In our process of generation we use the function CryptographicCurve from Magma [31].

1) The prime \( p \) is of the size at least \( 2^{256} \) to resist the attack with Pollard rho method [20]. We generate elliptic
curves over prime fields \( \mathbb{F}_p \), where \( p \) is a prime of the special form or a random prime.

2) Immunity to attacks using the Weil-pairing or Tate-pairing [2]. The attacks allow the embedding of the cyclic subgroup of \( E(\mathbb{F}_p) \) into the group of units of a degree-4 extension \( \mathbb{F}_{p^4} \) of \( \mathbb{F}_p \), where subexponential attacks on the Discrete Logarithm Problem in finite fields exist [14], [16], [19]. We have \( l = \min\{t : q \text{ divides } q' - 1\} \), i.e. \( l \) is the order of \( p \mod q \). The requirement is that the quotient \((q-1)/l < 100\).

3) The curves are not trace one curves. Trace one curves (or anomalous curves) are those for which \#\(E(\mathbb{F}_p) = p \). According to the works [22]–[24] for anomalous curves there are efficient algorithms to solve ECDLP.

4) The group order \( q = \#E(\mathbb{F}_p) \) must be a prime number or has a sufficiently big prime factor in order to avoid small group attack [19].

5) The class number of the maximal order of the endomorphism ring of the elliptic curve \( E \) is larger than 1000000000 [26], [27]. We search for elliptic curves over prime fields whose orders are prime numbers. This is the most consuming time of the search algorithm. In [11] it was stated a conjecture about the probability \( Q_1 \) that randomly chosen elliptic curve over a prime field \( \mathbb{F}_p \) is a prime. The conjecture states that

\[
Q_1 \sim c_p P_1 \sim \frac{c_p}{\log p} \quad \text{as} \quad p \to \infty, \tag{2}
\]

where \( c_p \) is explicitly computed constant, \( P_1 \) is the probability that a randomly chosen integer from the Hasse interval \([p + 1 - 2\sqrt{p},p + 1 + 2\sqrt{p}]\) is a prime and \( \sim \) means asymptotically equal. The numerical values of the constant satisfy \( 0.44 < c_p < 0.64 \) and usually it is closer to 0.44. The conjecture was experimentally confirmed for values of \( p \) up to \( 10^9 \). The authors of [11] put a similar conjecture for the probability \( Q_k \) that a randomly chosen elliptic curve has the order equal to \( kq \), where \( k \) is a small integer and \( q \) is a prime.

In fact, there are no attacks directly related to the class number criterion. It was stated in view of the future development of the theory of elliptic curves [8], [9]. Let us describe this condition in more details. The ring of endomorphism \( \text{End}(E) \) is isomorphic to an order in an imaginary quadratic field \( K = Q(\sqrt{-d}) \) with square-free \( d \in \mathbb{N} \). Let \#\(E(\mathbb{F}_p) = p + 1 + u \), then \( d \) can be computed as the square-free part of \( 4p - u^2 \) and then the factorization of that number is necessary to calculate the class number of the field \( K \). If \( v = \max\{a : a^2 \text{ divides } 4p - u^2\} \) then \( d = (4p - u^2)/v^2 \).

The complexity of the best known algorithm for explicitly determining the class number of \( K \) is too high in practice, hence one tries to find elements of the ideal class group of \( K \) with a large order, as the class number is not smaller than the order of an element.

We present the methods to compute class numbers and finding elements in the ideal class group with a large order. We follow the descriptions in [26] and [3]. There is a bijection between the class group of binary quadratic forms with discriminant \( d_K < 0 \) and the ideal class group of the order with discriminant \( d_K \). For this purpose we represent binary quadratic forms \( ax^2 + bxy + cy^2 \) as triples \((a, b, c)\). Then a triple \((a, b, c)\) of integers is called a positive definite primitive reduced binary quadratic form of discriminant \( d_K \) if:

- \( gcd(a, b, c) = 1 \), the form is primitive,
- \( a > 0 \) and \( |b| \leq a \leq c \) and if \( a = c \) or \( |b| = a \) than also \( b \geq 0 \), the form is reduced,
- \( b^2 - 4ac = d_K \), the form is positive definite.

We have the relation:

\[
d_K = \begin{cases} 
-4d & \text{if } -d = 1 \mod 4, \\
-d & \text{if } -d = 2 \mod 4 \text{ or } -d = 3 \mod 4.
\end{cases}
\]

The elements of the ideal class group of the number field \( K = \sqrt{-d} \) with discriminant \( d_K \) correspond bijectively to the primitive reduced quadratic forms of discriminant \( d_K \). The group law in this set is defined as follows:

- For two primitive reduced quadratic forms \((a_1, b_1, c_1)\) and \((a_2, b_2, c_2)\) their composition \((a_3, b_3, c_3)\) can be calculated by Algorithm 5.4.7 in Cohen’s book [3];
- The form \((a_3, b_3, c_3)\) is primitive and has discriminant \( d_K \) but it is not necessarily reduced. The reduction Algorithm 5.4.2 of [3] applied to \((a_3, b_3, c_3)\) outputs a primitive reduced quadratic form \((a, b, c)\) with discriminant \( d_K \);
- We denote the multiplication of quadratic forms by \( \bullet \), i.e.

\[
(a, b, c) = (a_1, b_1, c_1) \bullet (a_2, b_2, c_2).
\]

The neutral element \( I \) is represented by the triple \((1, 0, -d_K/4)\) if \( d_K \equiv 0 \mod 4 \) and it is represented by \((1, 1, (1 - d_K)/4)\) if \( d_K \equiv 1 \mod 4 \). The Algorithm of [26] determines whether an element of the ideal class group of the number field \( K = Q(\sqrt{-d}) \) has an order of at least a value \( \text{MinClass} \).

**Algorithm (26)**

**Input**: A primitive reduced quadratic form \((a, b, c)\) of discriminant \( d_K \).

**Output**: “true” if the order of the corresponding element of the ideal class group is at least \( \text{MinClass} \); and “false” otherwise.

1) Set \( t = I \).
2) for \( i \) from 1 to \( \text{MinClass} - 1 \) do
   Set \( t \leftarrow t \bullet (a, b, c) \).
   If \( t = I \) then output “false” and stop.
3) Output “true”.

Code 3 is an implementation of the Algorithm in Magma.

**IV. Twist security**

Let \( E \) be an elliptic curve \( E : y^2 = x^3 + ax + b \) over the base field \( \mathbb{F}_p \). The order of the curve is \#\(E(\mathbb{F}_p) = p + 1 + u \). The twisted curves are elements of the \( \mathbb{F}_p \)-isomorphism class of curves \( E^{tw} : x^3 + at^2x + bt^3 \) with \#\(E^{tw}(\mathbb{F}_p) = p + 1 - u \), where \( t \in \mathbb{F}_p \) is square-free. The curve \( E \) is called twist-secure ([1], [15]) if both \( E \) and \( E^{tw} \) are cryptographically strong. As a minimum both \#\(E(\mathbb{F}_p) \) and \#\(E^{tw}(\mathbb{F}_p) \) have to be almost prime, i.e., have only small prime factors and one big prime factor.
The conjecture formulated in formula (2) implies that the probability $Q_{tw}^1$ that a randomly chosen elliptic curve over a prime field $\mathbb{F}_p$ is both secure and twist-secure seems to satisfy
\[
0.5 \log^2 p < Q_{tw}^1 < 5 \log^2 p.
\]

In [7] it was analysed the twist security of elliptic curves. It is believed that the use of twist-security helps to improve security in the following situations:

- If only $x$-coordinates of points are used than an expanded $x$-coordinate could lead to a point in $E^{tw}(\mathbb{F}_p)$ instead of $E(\mathbb{F}_p)$. If $E^{tw}$ is not cryptographically strong and if $Q$ has smooth order on $E^{tw}$ an attacker might take advantage of this situation by providing $x$-coordinates of points that lie on $E^{tw}$. This method is called ‘invalid $x$’.
- Let $P_0$ be a point on $E$ and $d$ an integer. The attack goes in the way that during the computation of $dP_0$ a fault is introduced that leads to computations on $E^{tw}$ instead of $E$. Then the same ideas as for ‘invalid $x$’ apply. For simplicity one can assume that the fault is injected immediately at the beginning of the computations.

Authors of [15] and [7] argued that even for twist secure curves a point validation has to be performed. Assume that a cryptographic mechanism computes $dP_0$ for a secret scalar $d$. In the applications the computation of $dP_0$ is performed by using a blinded value $d + r_iq$ with randomly chosen $r_i \in \{1, \ldots, 2^R\}$, where $R$ is a system parameter. Blinding is a widely used counter method to thwart side-channel attacks on implementations that can be accessed by an attacker. The authors of [15] state that the value of $R$ must be sufficiently big and its length depends on the structure of the underlying base field. For random $p$ 64-bit $R$ is sufficient and for special primes $p$ it has to be $\log_2(\sqrt{q})$ bits of length.

V. THE NUMERICAL EXPERIMENTS

In Appendix there are examples of cryptographically strong elliptic curves of prime order over base fields $\mathbb{F}_p$, where $p$ is a special prime or a random prime of the size from 256 bits to 607 bits. The time of finding the curves increases as the size of $p$ grows. Only for the 256-bit prime it was possible to explicitly calculate the class number of the corresponding quadratic number field. In other cases we checked by running Code 3 the class number criterion. For the 607-bit prime it was not possible to check the class number criterion. The time of checking the criterion is random. We have calculated the order $l \equiv \text{Modorder}(p, q)$ of the embedding of the field $\mathbb{F}_p$ into the extended field $\mathbb{F}_{p^q}$. The factorization of the orders of the twisted curves and the factorization of $q - 1$ have been done and the sizes of the biggest factors have been given. All calculations have been done on macOS Catalina with 3.6 GHz Intel Core i7 processor. The results of checking the security criteria are shown in Tables 2 and 3.

VI. CONCLUSIONS

In classified applications we need elliptic curves with secret parameters which are independent from the examples of the curves given in the commercial standards. We have generated cryptographically strong elliptic curves over the base field $\mathbb{F}_p$, where $p$ is in a region corresponding to requirements of Suite A and Suite B. The purpose was to show how to search for such curves and it has been achieved.

REFERENCES

All numerical experiments have been done with Magma.

**Code 1: searching for cryptographic curves**

**Input:** a prime number or generation of a prime.

**Output:** a cryptographic elliptic curve and its prime order.

```plaintext
p := 2^251 - 1;
p;
K := GF(p);
repeat
E := CryptographicCurve(K);
n := #E;
until IsPrime(n); E;
n;
IsPrime(n);
```

**Code 2: The calculation of the class number**

**Input:** the prime p, q - the order of the elliptic curve over the base field GF(p).

**Output:** the exact value of the class number and the statement that it is greater than 10000000.

```plaintext
function ClassNumberField(p, q)
  u := q - p - 1;
  expr := 4 * p - u^2;
  d := SquarefreeFactorization(expr);
  K := QuadraticField(-d);
  cln := ClassNumber(K);
  printf "ClassNumber(Field) = %o \n", cln;
  if cln > 10000000 then
    wsklk := 1;
  else
    wsklk := 0;
  end if;
  return wsklk;
end function;
```

**Code 3: The estimation of the class number**

**Input:** The prime order p of the base field and the prime order q of the elliptic curve.

**Output:** The assertion that the class number criterion is satisfied.

```plaintext
function ClassNumberEstim(p, q)
  u := q - p - 1;
  expr := 4 * p - u^2;
  d := SquarefreeFactorization(expr);
  v :=.Sqrt(expr div d);
  qfmod := d mod 4;
  if qfmod eq 1 then d := d;
  else d := 4*d;
  end if;
  Q := BinaryQuadraticForms(-d);
  I := Q!1;
  t := 1;
  MinClass := 10000000;
  for iqfb in [1..127] do qfb := iqfb;
    left := qfb^2 + d;
    if ((left mod 4) eq 0) then
      qfaqc := left div 4;
      if not IsPrime(qfaqc) then
        for i in [2..2011] do qfa := i*qfa;
          f := Q!qfa;
          fred := Reduction(f);
          wsklk := 0;
        for i in [2..MinClass-1] do
          t := t*fred;
          if f eq I then wsklk := 0;
            break;
            end if;
            end for;
            end if;
            end for;
            end if;
            end for;
            end for;
            Q := BinaryQuadraticForms(-d);
            I := Q!1;
            t := 1;
            MinClass := 10000000;
            for iqfb in [1..127] do qfb := iqfb;
              left := qfb^2 + d;
              if ((left mod 4) eq 0) then
                qfaqc := left div 4;
                if not IsPrime(qfaqc) then
                  for i in [2..2011] do qfa := i*qfa;
                    f := Q!qfa;
                    fred := Reduction(f);
                    wsklk := 0;
                  for i in [2..MinClass-1] do
                    t := t*fred;
                    if f eq I then wsklk := 0;
                      break;
                      end if;
                      end for;
                      end if;
                      end for;
                      end if;
                      end for;
                      end for;
                      end if;
                      end for;
                      end for;
                      return wsklk;
                      end function;
```

**Examples of curves**

1. The prime \( p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1 \).

   Elliptic curve given by the equation (1):
   
   \[
   a = 476815057766280020269453809742175216676826826
   273317895890797437140862849937.
   \]
   \[
   b = 1131166390289526225695301750992706867567571
   68658707739244869954588904953620.
   \]

2. The pseudo Mersenne prime: \( p = 2^{384} - 317 \).

   Elliptic curve given by the equation (1):
   
   \[
   a = 2965982059611140164585089344874519177962181369
   4819913151325038371095738662951021780287684529766
   75303144969998910297.
   \]
   \[
   b = 1131166390289526225695301750992706867567571
   68658707739244869954588904953620.
   \]

3. The prime \( p = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1 \).

   Elliptic curve given by the equation (1):
   
   \[
   a = 87350709751579512343956062492337664824611521266
   162669621918133506763491913198328196882037351700
   39466627525116916.
   \]
   \[
   b = 1831389986083616554999991556610306453905550710
   4044958745047028011912149864799079798667026809496
   834343760457812568.
   \]

4. The pseudo Mersenne prime \( p = 2^{512} - 569 \).

   Elliptic curve E given by the equation (1):
   
   \[
   a = 85588533649343802088937787416639930850141397985
   45051579078816617067214962813129990627748755040261
   903974354924997700349271208555243960726915523052874
   7523.
   \]
5. The 513-bit random prime:
\[ p = 2^{513} - 1 = 2^{513} - 1 \]
Elliptic curve given by the equation (1):
\[ a = 1400939014018521886623559341318721867059603390 \]
\[ b = 24631984075740246707661438099554336356684473027 \]
\[ 7590791389505492916382420496874855258740927937865 \]
\[ 1481441148448849297943213542966556634208573839220959 \]
\[ 4141. \]

6. The 522-bit random prime:
\[ p = 88513391797632208449043070130575470195978479 \]
\[ 43363064870044038690768331393787384543670096162478 \]
\[ 96577100657348241821149971693259669179830601690596325 \]
\[ 42275504473029. \]

7. The 522-bit random prime:
\[ p = 88513391797632208449043070130575470195978479 \]
\[ 43363064870044038690768331393787384543670096162478 \]
\[ 96577100657348241821149971693259669179830601690596325 \]
\[ 42275504473029. \]

### Table 2. Properties of generated elliptic curves.

<table>
<thead>
<tr>
<th>Property</th>
<th>Curve 1</th>
<th>Curve 2</th>
<th>Curve 3</th>
<th>Curve 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search time Order q p</td>
<td>10 hours prime</td>
<td>5 hours prime</td>
<td>2 hours prime</td>
<td>27 hours prime</td>
</tr>
<tr>
<td>Order q p</td>
<td>256 bits</td>
<td>384 bits</td>
<td>384 bits</td>
<td>385 bits</td>
</tr>
<tr>
<td>Class #</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>(q-1)/l Factor q-l</td>
<td>106 bits</td>
<td>272 bits</td>
<td>382 bits</td>
<td>213 bits</td>
</tr>
</tbody>
</table>

### Table 3. Properties of generated elliptic curves.

<table>
<thead>
<tr>
<th>Property</th>
<th>Curve 5</th>
<th>Curve 6</th>
<th>Curve 7</th>
<th>Curve 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search time Order q p</td>
<td>13 hours prime</td>
<td>17 hours prime</td>
<td>19 hours prime</td>
<td>3 days prime</td>
</tr>
<tr>
<td>Order q p</td>
<td>513 bits</td>
<td>521 bits</td>
<td>522 bits</td>
<td>607 bits</td>
</tr>
<tr>
<td>Class #</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>?</td>
</tr>
<tr>
<td>(q-1)/l Factor q-l</td>
<td>427 bits</td>
<td>249 bits</td>
<td>400 bits</td>
<td>534 bits</td>
</tr>
</tbody>
</table>

**Abbreviations:**

- Search time: the time to find the elliptic curve.
- Order q: the confirmation that the order of the curve is a prime number.
- The length of the order of the base field.
- The length of the order of the elliptic curve.
- The confirmation that the class number criterion is satisfied, i.e., the class number is > 1000000.
- The length of the biggest factor in the factorization of the order of the twisted elliptic curve.
- The value of \((q-1)/l\), where \(l = \text{Modorder}(p, q)\). The confirmation that the corresponding criterion is satisfied.
- The length of the biggest factor in the factorization of \(q - 1\).