

THE EQUIVALENCE BETWEEN THE DHP AND DLP FOR
ELLIPTIC CURVES USED IN PRACTICAL APPLICATIONS

A. MUZEREAU, N.P. SMART AND F. VERCAUTEREN

Abstract

We re-examine the reduction of Maurer and Wolf of the Discrete Logarithm problem to the Diffie–Hellman problem. We give a precise estimate for the number of operations required in the reduction and use this to estimate the exact security of the elliptic curve variant of the Diffie–Hellman protocol for various elliptic curves defined in standards.

1. *Introduction*

One of the oldest challenging problems in public key cryptography is to prove or disprove that the Discrete Logarithm Problem (DLP) and the Diffie–Hellman Problem (DHP) are computationally equivalent. The hard part of the equivalence is showing that we can solve the DLP using a polynomial number of group operations and calls to a function which solves the DHP.

Significant steps have already been made towards the solution and the equivalence has been proved for some groups. Intuitively, it makes sense to use such groups for the Diffie–Hellman protocol (if of course no discrete logarithm algorithm is known for them), so that breaking the Diffie–Hellman protocol is as hard as computing logarithms, that is to say secure.

For most groups in use in cryptography, it is believed that the DHP and the DLP are equivalent in a complexity-theoretic sense; i.e. there is a polynomial time reduction of one problem to the other, and vice versa. Examples of groups that have been proposed for application in the Diffie–Hellman protocol are the multiplicative group of large finite fields (prime fields or extension fields), the multiplicative group of residues modulo a composite number, elliptic curves over finite fields, and the class group of imaginary quadratic fields.

Maurer and Wolf [6, 8, 7, 10] proved that for every group G with prime order p , the equivalence holds if we are able to find an elliptic curve over \mathbb{F}_p with smooth order. The aim of this paper is to show that for various elliptic curve groups recommended by standards, such an elliptic curve exists. To this end, we will use the technique of complex multiplication to construct elliptic curves with smooth order. The implementation of this algorithm has been carried out using the software package Magma.

2. *Notation and Definitions*

Formally we define the DHP and DLP as follows:

Definition 1. Let G be a finite cyclic group generated by g . The problem of computing from $h \in G$ an integer x such that $g^x = h$ is called the discrete logarithm problem (DLP) with respect to g .

Definition 2. Let G be a finite cyclic group generated by g . The problem of computing g^{ab} from g^a and g^b is called the Diffie-Hellman problem (DHP) with respect to g .

It is easy to see that if one can solve the DLP, one can solve the DHP. Let g^a and g^b be in G . We compute a from g^a and compute $(g^b)^a = g^{ab}$. Hence $DLP \Rightarrow DHP$. This paper focuses on the reverse reduction, namely $DLP \Leftarrow DHP$.

The equivalence we are interested in is a *computational* equivalence. Suppose that one day, the DHP turns out to be easy, i.e. a given instance of this problem can be solved in a reasonable time. We want to know if this implies that the DLP is easy as well, i.e. if there exists an effective algorithm for solving a given instance of the DLP by using a “small” number of operations and of calls to a function which solves the DHP. Such a function is called a DH-oracle:

Definition 3. A DH-oracle takes as input elements g^a and g^b and returns g^{ab} .

Now, what do we mean by a “small” number of operations and a “small” number of calls to the DH-oracle? The answer is a polynomial in $\log p$, where p is the order of the group.

Definition 4. Let G be a finite cyclic group with generator g of order $|G| = p$. Given $h \in G$, the DLP and the DHP are computationally equivalent if we are able to find the unique x modulo p such that $h = g^x$, by using only:

- $O((\log p)^{O(1)})$ operations in G ,
- $O((\log p)^{O(1)})$ calls to the DH-oracle.

For given elliptic curves defined in various standards we would like to show that the number of group operations and DH-oracle calls required to reduce the DLP to the DHP is small, i.e. less than say 2^{n_1} . This would imply that if we believe that no algorithm can solve the DLP in such groups in less than 2^{n_2} operations then any future algorithm for solving the DHP, and so breaking the DHP protocol, would require $2^{n_2 - n_1}$ operations. Hence, the smaller the value of n_1 , the tighter the security reduction.

3. Algorithm Overview

We first give an overview of the method by Maurer and Wolf [6], which we shall use in our later calculations.

Let G be a cyclic group with generator g and whose order is a prime p . If a is an integer modulo p then the value of g^a is said to be the *implicit representation* of a . The idea of the algorithm is to do computations in implicit representation. For example, to compute $a + b$ (resp. $a - b$) in implicit form, we compute $g^a \cdot g^b$ (resp. $g^a \cdot (g^b)^{-1}$) which costs only one multiplication (resp. and an inversion) in G . To compute $a \cdot b$ in implicit form, one call to the DH-oracle is needed. To compute a^{-1} in implicit form, one uses the fact that $a^{p-1} = 1$, so $a^{p-2} = a^{-1}$. Hence, one can perform any algebraic algorithm on the implicit representation. The following table sums up computations in implicit representation and their average complexities.

Explicit form	Implicit form	Complexity
$a + b$	$g^{a+b} = g^a \cdot g^b$	1 multiplication in G
$a - b$	$g^{a-b} = g^a \cdot (g^b)^{-1}$	1 multiplication and 1 inversion in G
$a \cdot b$	$DH(g^a, g^b)$	1 call to the DH-oracle
a^{-1}	$g^{a^{-1}} = g^{a^{p-2}}$	2 $\log p$ calls to the DH-oracle

The following result can be found in [6] and [10].

Theorem 1. *Let G be a group. If each large prime factor p of $|G|$ is single and if for every such p a cyclic elliptic curve over \mathbb{F}_p is known with smooth order then breaking DHP and DLP are equivalent for G .*

Proof. For the sake of simplicity, let us assume that $G = \langle g \rangle$ with $|G| = p$ prime. The elliptic curve $E = E_{a,b}(p)$ with parameters a and b in \mathbb{F}_p is the set

$$\{(x, y) \in (\mathbb{F}_p)^2 \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}.$$

By a theorem of Ruck [11] we can always choose a curve E , of a given order, such that $E(\mathbb{F}_p)$ is cyclic and generated by P .

We assume we are given g^x and we are asked to compute x . First, the group element

$$g^{x^3+ax+b}$$

can be computed from g^x by $O(\log a + \log b) = O(\log p)$ group operations and two calls to the DH oracle for G . If $z = x^3 + ax + b$ is a quadratic residue mod p (which can be tested efficiently), then a group element g^y can be computed such that $y^2 \equiv z \equiv x^3 + ax + b \pmod{p}$ using an implicit version of the Tonelli-Shanks algorithm [5].

If z is not a quadratic residue, g^z can be replaced by g^{z+d} for a random offset d until z is a quadratic residue. Testing the quadratic residuosity of z modulo p can be achieved with $O(\log p)$ applications to the DH-oracle, because z is a quadratic residue modulo p if and only if $z^{(p-1)/2} \equiv 1 \pmod{p}$, that is if and only if $g^{z^{(p-1)/2}} = g$. To simplify the discussion we shall assume that z is a quadratic residue and that using the Tonelli-Shanks algorithm we have computed

$$(g^x, g^y) = \left(g^x, g^{\sqrt{x^3+ax+b}} \right)$$

with $x, y \in \mathbb{F}_p$. The point $Q = (x, y)$ is a point on the elliptic curve E . Since $|E|$ is assumed to be smooth, we can use an implicit version of the Pohlig-Hellman algorithm to compute the discrete logarithm k of Q with respect to the generator P . Computing $[k]P$ explicitly finally gives us x , as the abscissa of the point $[k]P$.

For each prime factor q of $|E|$ we proceed as follows. From (g^x, g^y) we compute (g^u, g^v) such that

$$(u, v) = \left[\frac{|E|}{q} \right] Q.$$

From the generator P of E , the points

$$(u_i, v_i) = \left[i \cdot \frac{|E|}{q} \right] P$$

are computed for $i = 0, 1, \dots, q-1$, and from (u_i, v_i) we obtain the group elements (g^{u_i}, g^{v_i}) . Since the point (u, v) has order q and $Q = [k]P$, we conclude that

$$(g^u, g^v) = (g^{u_i}, g^{v_i}) \iff k \equiv i \pmod{q}.$$

Similarly, k can be computed modulo the prime powers of the factorisation of $|E|$, and hence modulo $|E|$. From k , we compute $[k]P = Q$, and then x simply is the abscissa of the point Q .

If $|E|$ is B -smooth, then the rough complexity of this method is

- $O\left(B \cdot (\log p)^2\right)$ group operations in G and field operations in \mathbb{F}_p ,
- $O\left((\log p)^3\right)$ calls to the DH-oracle for G .

A more accurate estimate of the complexities will be given later. □

4. How long it takes to solve a given instance of the DLP

In this section, we want to find a precise estimate of how long it takes to solve a given instance of the DLP, that means how many calls to the DH-oracle and multiplications are required on average. We need to analyse precisely the method sketched in Section 3.

Let G be a cyclic group with generator g and prime order p . Given $h \in G$, we want to find the unique x modulo p such that $h = g^x$. The generalization with a composite order is possible, see Section 3, but is not necessary of practical importance, since the orders of all the groups recommended by standards are prime.

We assume that the parameters a and b of a cyclic elliptic curve $E_{a,b}(\mathbb{F}_p)$ with smooth order are given. We assume that

$$|E| = \prod_{j=1}^s q_j^{f_j}$$

with $f_j = 1$ and $q_j < B$ for $j = 1, \dots, s$. Actually, the generalization with $f_j > 1$ is possible using the analogy with the Pohlig-Hellman algorithm, but is not useful because in practice the multiple factors of $|E|$ will always be small in comparison with the largest prime factor $|E|$. Therefore, we assume that $|E| = \prod_{j=1}^s q_j$ where all q_j are not necessarily prime but are all less than the smoothness bound B .

4.1. Algorithm Overview

1. 1.1. Compute $g^{x^3+ax+b} = g^z$
- 1.2. [Test quadratic residuosity of $z \pmod{p}$
Compute $g^{z^{(p-1)/2}}$ and g and compare them. On equality, go to Step 2, else replace x by $x+d$ for a random d and go to Step 1.1.
2. [Compute g^y from $g^z = g^{y^2}$ using the algorithm of Tonelli and Shanks.]
Write $p-1 = 2^e \cdot w$ with w odd.
 - 2.1. [Initialize] Set $g^s \leftarrow g$, $r \leftarrow e$, $g^y \leftarrow g^{z^{(w-1)/2}}$, $g^b \leftarrow g^{zy^2}$, $g^y \leftarrow g^{zy}$.
 - 2.2. [Find exponent] If $g^b \equiv 1 \pmod{p}$, output g^y and go to Step 3. Otherwise, find the smallest $m \geq 1$ such that $g^{(b^{2^m})} \equiv 1 \pmod{p}$.
 - 2.3. [Reduce exponent] Set $g^t \leftarrow g^{(s^{2^r-m-1})}$, $g^s \leftarrow g^{t^2}$, $r \leftarrow m$, $g^y \leftarrow g^{yt}$, $g^b \leftarrow g^{bs}$ and go to Step 2.2.

3. Note $Q := (x, y)$ is a point on E , however we only know the implicit representation (g^x, g^y) .

For j from 1 to s , do the following:

3.1. Compute (g^{u_j}, g^{v_j}) such that $(u_j, v_j) = \frac{|E|}{q_j} \cdot Q$.

3.2. For i from 0 to $q_j - 1$, do the following:

3.2.1. Compute $(u_{ji}, v_{ji}) = i \cdot \frac{|E|}{q_j} \cdot P$, where P is a generator of E .

3.2.2. Compute $(g^{u_{ji}}, g^{v_{ji}})$

3.2.3. Compare $(g^{u_{ji}}, g^{v_{ji}})$ with (g^{u_i}, g^{v_i}) . On equality let $k_j := i$ and go to next iteration in j (or to Step 4 if $j = s$).

4. 4.1. Compute $k \bmod |E|$ such that $\forall j \in \{1, \dots, s\}, k \equiv k_j \bmod q_j$.

4.2. Compute $k \cdot P = Q$. Then $x \bmod p$ is the first abscissa of Q .

The standard binary exponentiation algorithm requires $\log_2 k$ squares and on average $\frac{1}{2} \log_2 k$ multiplications. We require this in two places:

- To compute g^{x^k} , given g^x . Then, on average $\frac{3}{2} \log_2 k$ calls to the DH-oracle are needed.
- Given a point P on an elliptic curve, to compute $kP = \sum_{i=0}^t k_i (2^i P)$. Then, on average $\frac{1}{2} \log_2 k$ additions of points and $\log_2 k$ doublings are needed.

We now expand on the second of these subprocedures.

4.2. Explicit and Implicit Point Multiplications

4.2.1. Doubling a point on an elliptic curve

Let $P = (x, y)$ and $Q = 2P = (x', y')$. Then:

$$\begin{cases} \lambda = \frac{3x^2 + a}{2y} \\ x' = \lambda^2 - 2x \\ y' = \lambda \cdot (x - x') - y \end{cases}$$

In implicit representation, we know (g^x, g^y) and we want to compute $(g^{x'}, g^{y'})$ such that $(x', y') = 2(x, y)$.

$$\begin{cases} g^\lambda = DH(DH(g^x, g^x) \cdot g^{x^2} \cdot g^{x^2} \cdot g^a, DHI(g^y \cdot g^y)) \\ g^{x'} = DH(g^\lambda, g^\lambda) \cdot g^{-x} \cdot g^{-x} \\ g^{y'} = DH(g^\lambda, g^x \cdot g^{-x'}) \cdot g^{-y} \end{cases}$$

Computing g^λ requires $4 + \frac{3}{2} \log_2 a$ multiplications, two calls to the DH-oracle and one DH-inversion¹. If g^λ is known, computing $g^{x'}$ and $g^{y'}$ requires four multiplications and three inversions in \mathbb{F}_p , and two calls to the DH-oracle.

Finally, doubling a point on an elliptic curve requires:

In explicit form: Four multiplications and one inversion in \mathbb{F}_p .

In implicit form: $8 + \frac{3}{2} \log_2 a$ multiplications and three inversions in \mathbb{F}_p , four calls to the DH-oracle and one DH-inversion.

¹To compute $g^{(z^{-1})}$ from g^z , we use the fact that $g^{(z^{-1})} = g^{(z^{p-2})}$ in \mathbb{F}_p . It requires on average (approximately) $3/2 \cdot \log_2 p$ calls to the DH-oracle. We call this a DH-inversion, in contrast with an inversion: computing g^{-z} from g^z .

4.2.2. Adding two points on an elliptic curve

Let $P = (x_1, y_1)$, $Q = (x_2, y_2)$ and $R = P + Q = (x_3, y_3)$. Then:

$$\begin{cases} \lambda = \frac{y_2 - y_1}{x_2 - x_1} \\ x_3 = -x_1 - x_2 + \lambda^2 \\ y_3 = \lambda \cdot (x_1 - x_3) - y_1 \end{cases}$$

In implicit representation, we know (g^{x_1}, g^{y_1}) and (g^{x_2}, g^{y_2}) , and we want to compute (g^{x_3}, g^{y_3}) such that $(x_3, y_3) = (x_1, y_1) + (x_2, y_2)$.

$$\begin{cases} g^\lambda = DH(g^{y_2} \cdot g^{-y_1}, DHI(g^{x_2} \cdot g^{-x_1})) \\ g^{x_3} = g^{-x_1} \cdot g^{-x_2} \cdot DH(g^\lambda, g^\lambda) \\ g^{y_3} = DH(g^\lambda, g^{x_1} \cdot g^{-x_3}) \cdot g^{-y_1} \end{cases}$$

Finally, adding two points on an elliptic curve requires:

In explicit form: Three multiplications and one inversion in \mathbb{F}_p .

In implicit form: Six multiplications and four inversions in \mathbb{F}_p , three calls to the DH-oracle and one DH-inversion.

Combining the above analyses we obtain that a scalar multiplication of a point on a curve requires the following number of operations.

In explicit form: Given a point P on an elliptic curve, computing explicitly the point kP requires on average:

- $\frac{11}{2} \log_2 k$ multiplications in \mathbb{F}_p .
- $\frac{3}{2} \log_2 k$ inversions in \mathbb{F}_p .

In implicit form: Given (g^x, g^y) , computing (g^u, g^v) such that $(u, v) = k \cdot (x, y)$ requires on average:

- $11 \log_2 k + \frac{3}{2} \log_2 a$ multiplications in \mathbb{F}_p (we compute g^a only once).
- $5 \log_2 k$ inversions in \mathbb{F}_p .
- $\frac{11}{2} \log_2 k$ calls to the DH-oracle.
- $\frac{3}{2} \log_2 k$ DH-inversions.

4.3. Complexity

We are now in a position to evaluate precisely the complexity of the algorithm for reducing the DLP to the DHP.

Step 1

Step 1.1 : We compute $g^{x^3+ax+b} = g^{x^3}(g^x)^a g^b$. This requires two calls to the DH-oracle and $2 + \frac{3}{2}(\log_2 a + \log_2 b)$ multiplications.

Step 1.2 : This step requires $3 \log_2 \frac{|G|}{p}$ multiplications and about $\frac{3}{2} \log_2 \left(\frac{p-1}{2}\right) \approx \frac{3}{2}(\log_2 p - 1)$ calls to the DH-oracle.

The field \mathbb{F}_p contains $\frac{p+1}{2}$ quadratic residues and $\frac{p-1}{2}$ non-quadratic residues. Let ν be the number of iterations for Step 1. The probability for having $\nu = k$ iterations is:

$$P(\nu = k) = \left(\frac{p-1}{2p}\right)^{k-1} \cdot \frac{p+1}{2p}$$

The average number $\bar{\nu}$ of iterations for Step 1 is therefore:

$$\bar{\nu} = \sum_{k=1}^{\infty} k \cdot P(\nu = k) = \sum_{k=1}^{\infty} k \cdot \left(\frac{p-1}{2p} \right)^{k-1} \cdot \frac{p+1}{2p} = \frac{2p}{p+1} \approx 2$$

Hence Step 1 requires on average about:

- $4 + 3 \log_2 a + \frac{3}{2} \log_2 b + \frac{9}{2} \log_2 \frac{|G|}{p}$ multiplications (g^b and g^h are computed only once).
- $1 + 3 \log_2 p$ calls to the DH-oracle.

Step 2

Step 2.1 requires about $\frac{3}{2}(\log_2 w - 1) + 3 = \frac{3}{2}(\log_2 w + 1)$ calls to the DH-oracle. Steps 2.2 and 2.3 require $r + 2$ calls to the DH-oracle for one iteration, and at most e iterations are expected. Since r is always smaller than e , Steps 2.2 and 2.3 need at most $e \cdot (e + 2)$ calls to the DH-oracle. We need to estimate e , the integer such that $p - 1 = 2^e \cdot w$ with w odd. It is easy to see that on average, since p is odd, that $e \approx 2$. Hence, Step 2 needs about

$$8 + \frac{3}{2}(1 + \log_2 \frac{p}{4}) = \frac{13}{2} + \frac{3}{2} \log_2 p$$

calls to the DH-oracle.

Step 3

Let j be fixed.

Step 3.1 Using results of Section 4.2, Step 3.1 requires on average, for each value of j :

- $11 \log_2 \frac{|E|}{q_j} + \frac{3}{2} \log_2 a$ multiplications in \mathbb{F}_p .
- $5 \log_2 \frac{|E|}{q_j}$ inversions in \mathbb{F}_p .
- $\frac{11}{2} \log_2 \frac{|E|}{q_j}$ calls to the DH-oracle.
- $\frac{3}{2} \log_2 \frac{|E|}{q_j}$ DH-inversions.

Step 3.2 First we compute $\frac{|E|}{q_j} \cdot P$, before entering into the loop in i . This requires on average $\frac{11}{2} \log_2 \frac{|E|}{q_j}$ multiplications and $\frac{3}{2} \log_2 \frac{|E|}{q_j}$ inversions in \mathbb{F}_p .

Step 3.2.1 We use the fact that $(u_{i+1}, v_{i+1}) = (u_i, v_i) + \frac{|E|}{q_j} \cdot P$. The cost is one addition on E , that is three multiplications and one inversion in \mathbb{F}_p .

Step 3.2.2 This step needs $3/2 \cdot (\log_2 u_i + \log_2 v_i)$ multiplications in \mathbb{F}_p . If we consider that u_i and v_i are $p/2$ on average, then $3 \log_2 p - 3$ multiplications are needed.

We can assume that on average, there are $q_j/2$ iterations in the loop in i : $k_i = q_j/2$. Thus Step 3.2 requires on average, for one j :

- $\frac{11}{2} \log_2 \frac{|E|}{q_j} + \frac{3q_j}{2} \log_2 p$ multiplications in \mathbb{F}_p .
- $\frac{3}{2} \log_2 \frac{|E|}{q_j} + \frac{q_j}{2}$ inversions in \mathbb{F}_p .

Summing up for the whole of Stage 3 we have

$$\sum_{j=1}^s \log_2 \frac{|E|}{q_j} = (s-1) \cdot \log_2 |E|.$$

Hence Step 3 requires on average :

- $\frac{3}{2} \log_2 a + \frac{33(s-1)}{2} \log_2 |E| + \frac{3 \log_2 p}{2} \cdot \sum_{j=1}^s q_j$ multiplications in \mathbb{F}_p .
- $\frac{13(s-1)}{2} \log_2 |E| + \frac{1}{2} \sum_{j=1}^s q_j$ inversions in \mathbb{F}_p .
- $\frac{11(s-1)}{2} \log_2 |E|$ calls to the DH-oracle.
- $\frac{3(s-1)}{2} \log_2 |E|$ DH-inversions.

Step 4

Step 4.1 We use the Chinese Remainder Theorem to compute $k \bmod |E|$, knowing $k \equiv k_j \bmod q_j$, for each $j \in \{1, \dots, s\}$. Using Gauss algorithm:

$$k = \sum_{j=1}^s k_j \cdot Q_j \cdot R_j \pmod{|E|}$$

with $Q_j = \frac{|E|}{q_j}$ and $R_j = Q_j^{-1} \bmod q_j$. It requires $2s$ multiplications and s inversions in $(\mathbb{F}_{q_1}, \dots, \mathbb{F}_{q_s})$.

Step 4.2 We can consider that on average, $k \bmod |E| = \frac{|E|}{2}$. Thus $\log_2 \frac{|E|}{2}$ doublings and $\frac{1}{2} \log_2 \frac{|E|}{2}$ additions on E are needed. Hence, Step 4.2 requires on average:

- $\frac{11}{2}(\log_2 |E| - 1)$ multiplications in \mathbb{F}_p .
- $\frac{3}{2}(\log_2 |E| - 1)$ inversions in \mathbb{F}_p .

4.4. Conclusion

The algorithm needs on average:

- $-\frac{3}{2} + \frac{9}{2} \log_2 a + \frac{3}{2} \log_2 b + \frac{9}{2} \log_2 |G| + \frac{3}{2} \log_2 p \cdot \left(\sum_{j=1}^s q_j - 3 \right) + 11 \left(\frac{3}{2}s - 1 \right) \cdot \log_2 |E|$ multiplications in \mathbb{F}_p .
- $-\frac{3}{2} + \frac{1}{2} \sum_{j=1}^s q_j + \left(\frac{13}{2}s - 5 \right) \cdot \log_2 |E|$ inversions in \mathbb{F}_p .
- $\frac{15}{2} + \frac{9}{2} \log_2 p + \frac{11(s-1)}{2} \log_2 |E|$ calls to the DH-oracle.
- $\frac{3(s-1)}{2} \log_2 |E|$ DH-inversions.

We supposed that $a > 0$ and $b > 0$. If actually $a < 0$, we must add three inversions in \mathbb{F}_p and if $b < 0$, one inversion must be added. In the expressions above, many

terms can be neglected. Moreover, the approximation $\log_2 |E| \approx \log_2 p$ can be done without loss of accuracy. We obtain:

- $\frac{3}{2} \log_2 p \cdot \left(\sum_{j=1}^s q_j \right)$ multiplications in \mathbb{F}_p .
- $\frac{1}{2} \sum_{j=1}^s q_j + \left(\frac{13}{2}s - 5 \right) \cdot \log_2 p$ inversions in \mathbb{F}_p .
- $\left(\frac{11s}{2} - 1 \right) \cdot \log_2 p$ calls to the DH-oracle.
- $\frac{3(s-1)}{2} \log_2 p$ DH-inversions.

Since an inversion can in general be computed in a field of large prime characteristic at a cost of at most 10 multiplications, and since a DH-inversion needs on average $\frac{3}{2} \log_2 p$ calls to the DH-oracle, we conclude

Theorem 2. *Let G a cyclic finite group of prime order p . Assume an elliptic curve E over \mathbb{F}_p has been found, whose B -smooth order is*

$$\#E = \prod_{j=1}^s q_j.$$

Solving a given instance of the DLP in G requires on average about:

$$(65s - 50) \log_2 p + \left(5 + \frac{3}{2} \log_2 p \right) \sum_{j=1}^s q_j = O \left(\frac{B \cdot \log^2 p}{\log B} \right) \text{ multiplications in } \mathbb{F}_p.$$

$$\left(\frac{11}{2}s - 1 + \frac{9}{4}(s-1) \log_2 p \right) \cdot \log_2 p = O \left(\frac{\log^3 p}{\log B} \right) \text{ calls to the DH-oracle.}$$

Using the Baby-Step/Giant-Step method to find k , as opposed to the exhaustive search method above, both complexities can be replaced by

$$O \left(\sqrt{B} \cdot (\log p)^3 \right).$$

5. Building a curve with appropriate order

We now turn to the problem of building a curve with a smooth group order over the field of p elements. According to [4], the main techniques are:

- Generate random curves and compute their group orders, until an appropriate one is found.
- Generate curves with given group order using the theory of complexity multiplication (CM).

The genesis of the efficient general point counting algorithms lies in the work of Schoof [12]. The complexity of his algorithm is $O(\log^8 p)$. To improve the computational efficiency of the basic Schoof algorithm, several techniques have evolved, owing in large part to Atkin and Elkies, see [4] for details. The improvements to the

basic Schoof algorithm are generally referred to as the Schoof-Elkies-Atkin (SEA) algorithm, whose running time is $O(\log^6 p)$.

When the order of the random elliptic curve is known, it remains to check whether or not it is smooth.

To speed up the computations we preferred the CM method, since given a prime p it is very easy to generate a large number of possible group orders. To this end, we give a quick overview of the CM method for curve construction. If E is an elliptic curve over \mathbb{F}_p of order u , then

$$Z = 4p - (p + 1 - u)^2$$

is positive by the Hasse bound. Thus there is a unique factorization:

$$Z = DV^2$$

where D is squarefree. So for each non-supersingular elliptic curve over \mathbb{F}_p of order u , there exists a unique squarefree integer D such that:

$$4p = W^2 + DV^2 \tag{1}$$

for some W and V . In this case the group order is given by

$$u = p + 1 \pm W.$$

It is said that E has *complex multiplication* by D . The value D is called a *CM discriminant* for p . To find W and V in equation (1) one uses the algorithm of Cornacchia, see [4] for details.

Once one has found values of W and V , and an associated CM discriminant D we can then build an elliptic curve with group order $p + 1 \pm W$ using the theory of complex multiplication. This last step can lead to problems, unless the value of D is sufficiently small, since for large values of D we need to construct the Hilbert class polynomial which has degree $h_D = O(\sqrt{D})$, where h_D is the class number of the order of discriminant $-D$.

Hence, we need to find a small value of D for a given prime p such that one of $p + 1 \pm W$ is smooth, where W is the solution to equation (1). The main cost is in searching for a value of D such that $p + 1 \pm W$ is smooth. Due to the size of the numbers involved a naive smoothness test is not enough, and essentially one needs to perform a full factorization using the ECM factorization method.

6. Security of the DLP

The traditional way to interpret the reduction of the DLP to the DHP is to use the result to examine the security of the discrete logarithm problem in terms of oracle calls to the Diffie–Hellman problem. In such a situation one wishes to balance the number of group operations and Diffie–Hellman oracle calls made in the reduction algorithm. As we mentioned above this can be done by the use of the Baby-Step/Giant-Step algorithm in Step 3 of the reduction above. Doing so results in a complexity of $O(\sqrt{B} \cdot (\log p)^3)$ group operations and Diffie–Hellman oracle calls.

Waterhouse [13] determined the possible values of $\#E(\mathbb{F}_p)$ and showed that for all integers $d \in [p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p}]$, there exists an elliptic curve over \mathbb{F}_p of order d . Furthermore, a theorem by Rück [11] implies that the group structure can

be assumed cyclic. This implies the following non-uniform reduction of the DLP to the DHP. For a number p , we define $\nu(p)$ to be the minimum of the set of the largest prime factors of the integers d in the interval $[p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p}]$. This leads to the following theorem:

Theorem 3. *For every finite cyclic group G of order $|G| = \prod p_i^{e_i}$ and such that all multiple prime factors p_i of $|G|$ are smaller than a polynomial in $\log |G|$, there exists an algorithm that makes calls to a DH oracle for G and computes discrete logarithms of elements of G in*

$$\sqrt{\max\{\nu(p_i)\}} \cdot (\log |G|)^{O(1)}$$

group operations and calls to the Diffie–Hellman oracle.

A plausible smoothness assumption, see [9, 10], is that:

$$\nu(n) \text{ is of order } (\log n)^{O(1)} \quad (2)$$

This assumption implies the existence of a $(\log n)^{O(1)}$ -smooth cyclic elliptic curve over \mathbb{F}_p for each prime number p . Therefore for every cyclic group G there exists a small piece of information, which depends only on the order of G , that makes breaking the Diffie-Hellman protocol and computing discrete logarithms polynomial-time equivalent in G . This information is a string S consisting of the prime factors p_i of $|G|$ and appropriate elliptic curve parameters a_i and b_i for all p_i .

Corollary 1. *If the smoothness assumption (2) is true, then for every cyclic group $G = \langle g \rangle$ whose order contains no multiple prime factors greater than a polynomial in $\log |G|$, there exists a string S of length at most $3 \log |G|$ such that when given S , solving the DHP is polynomial-time equivalent to solving the DLP.*

Using the specific properties of the elliptic curve groups defined in the various standards, we now show the existence of an auxiliary elliptic curve which has very smooth order, i.e. the order simply is a power of two.

Suppose first that the elliptic curve E is defined over a finite field \mathbb{F}_{2^n} , then the theorem of Hasse implies that $\#E \in [2^n + 1 - 2^{n/2}, 2^n + 1 + 2^{n/2}]$. Furthermore, all the elliptic curve groups in the standard have an order of the form $\#E = h \cdot p$ with p a large prime and the cofactor h either 2 or 4. This implies that the prime p itself is contained in the interval

$$[2^{n-\delta} + 1/h - 2^{n/2-\delta}, 2^{n-\delta} + 1/h + 2^{n/2-\delta}] \quad (3)$$

with $h = 2^\delta$, i.e. $\delta = 1, 2$. The theorem by Waterhouse showed that for each $d \in [p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p}]$ there exists an elliptic curve over \mathbb{F}_p with group order d . Since p is contained in the interval (3), an easy calculation shows that it is highly likely that $d = 2^{n-\delta}$ is contained in the Hasse-interval $[p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p}]$. This implies that there exists an auxiliary elliptic curve over \mathbb{F}_p with group order $2^{n-\delta}$. For all characteristic two curves in the SECG list [3] we find that the value of d is in the required interval.

For elliptic curve groups defined over a large prime finite field \mathbb{F}_q , this reasoning no longer holds. However, to speed up computations, the primes q in use in the standards are of a special form, in particular, most q are very close to a power of 2. Since the co-factor h is either 1 or 4, we conclude that $\#E = h \cdot p$ with p close

to 2^n for some n . For all the curves over large primes fields to be found in [3] we see that all of those of bit length greater than (or equal to) 160, bar `secp256r1` have the property that p is sufficiently close to 2^n for the reasoning to hold.

Knowing the existence of an auxiliary elliptic curve group with very smooth order is however not useful in practice since it might require exponential time to construct this elliptic curve. Hence, it is of interest for a given group, proposed for use in a Diffie–Hellman protocol, to also present the best known string S which produces the tightest possible reduction between the DHP and the DLP.

7. Security of the DHP

We now examine what the reduction means for the security of the Diffie–Hellman protocol in the elliptic curve setting. We want to estimate the number of operations an adversary to the Diffie–Hellman protocol would require, under the assumption that the best algorithm for solving the elliptic curve discrete logarithm problem will take \sqrt{q} operations.

In this case we wish to minimize in the reduction the number of calls to the Diffie–Hellman oracle, at the expense of increasing the number of group operations. Hence, one uses the naive version of Step 3 in the reduction rather than the Baby-Step/Giant-Step algorithm. This allows us to obtain a tighter security reduction.

For each elliptic curve in the SECG standards [3], which includes all the curves in the NIST [2] and the most used ones in the ANSI [1] standards, we searched for the best values for

- The discriminant D ,
- The factorized order of the auxiliary elliptic curve (supposed to be smooth),
- The smoothness bound B ,
- The parameters a and b of the elliptic curve,
- The number of group operations and the number of calls to the DH-oracle required, using Theorem 2.

The various values for each curve are presented in the Appendix.

The Tables 1 and 2 summarize the results. The value of B is the size of the largest prime factor of the order of the auxiliary curve, M represents the number of field multiplications required by the reduction algorithm and DH is the number of Diffie–Hellman oracle calls. The value T represents the tightness of the security reduction. We do not give any values for the larger curves, since we were unable to find a suitable D due to the difficulty of factoring integers of this size.

To interpret what these tables mean, we illustrate with an example. Consider the curve `secp256r1`: with current knowledge it is believed that to solve the DLP on this curve requires on average 2^{128} computational steps. This would imply, given our auxiliary curve, that the DHP could not be solved in 2^{108} steps and thus solving the DHP on this curve is infeasible with today's computing technology. Thus we can conclude that protocols which depend on the DHP for their security can be safely deployed when using the curve `secp256r1`.

To obtain a tightness of the security reduction we need to look at two values. There is the cost of field multiplications, represented by $\log_2 M$ in Tables 1 and 2.

Table 1: Summary of results for curves of large prime characteristic

SECP curve	D	$\log_2 B$	$\log_2 M$	$\log_2 DH$	$\log_2 T_{DH}$
SECP112R1	49271	24	32	18	38
SECP112R2	232	24	31	18	38
SECP128R1	1147	34	41	18	46
SECP128R2	1099	32	40	18	46
SECP160K1	615	29	36	20	60
SECP160R1	1687	33	41	18	62
SECP160R2	2947	46	53	19	61
SECP192K1	391443	37	44	20	76
SECP192R1	334852	38	46	19	77
SECP224K1	58531	53	62	19	93
SECP224R1	41187	42	51	20	92
SECP256K1	56296	56	65	20	108
SECP256R1	41752	53	62	20	108
SECP384R1	22312	83	91	22	170
SECP521R1	-	-	-	-	-

Furthermore, we also need to look at

$$T_{DH} \approx \frac{\sqrt{\#E}}{DH}.$$

If we assume the existence of an algorithm to solve the DLP on E would take roughly $\sqrt{\#E}$ steps then the value of T_{DH} gives the minimum number of operations which an algorithm to break the DHP would take, assuming $M < T_{DH}$. Hence, it is the value of T_{DH} which gives the exact security result, given the witness curve we have found. If one could find a better witness elliptic curve then one would obtain a tighter security reduction, and hence a larger value of T_{DH} .

Note that the value of DH is not really affected that much by the smoothness value. The smoothness value mainly affects the number of group operations M . We now argue that it is highly likely for auxiliary elliptic curves to exist which would imply a tight reduction for all elliptic curve Diffie–Hellman problems.

Firstly note that since we are assuming an exponential algorithm for the discrete logarithm problem and we are trying to reduce the number of oracle calls, we do not mind if the number of group operations is exponential, as long as it is less than the eventual estimated number of operations in the Diffie–Hellman algorithm. Hence, if $\#E$ factors as a product of three primes of roughly the same order then we would have that the reduction of Theorem 2 would require on average

$$145 \log_2 p + 3 \left(5 + \frac{3}{2} \log_2 p \right) p^{1/3}$$

group operations and

$$\frac{1}{2} (31 + 9 \log_2 p) \log_2 p$$

Diffie–Hellman oracle calls. In particular this would imply

Table 2: Summary of results for curves of even characteristic

SECT curve	D	$\log_2 B$	$\log_2 M$	$\log_2 DH$	$\log_2 T_{DH}$
SECT113R1	36883	27	34	18	38
SECT113R2	78859	22	30	18	38
SECT131R1	1348	40	47	18	47
SECT131R2	410107	32	40	18	47
SECT163K1	7	47	55	19	62
SECT163R1	384591	38	46	19	62
SECT163R2	6107	34	42	19	62
SECT193R1	7	48	57	19	77
SECT193R2	11	47	55	19	77
SECT233K1	7	41	50	20	96
SECT233R1	2263	69	77	20	96
SECT239K1	7	38	47	21	98
SECT283K1	7	30	38	22	119
SECT283R1	11768	61	69	20	121
SECT409K1	7	81	90	22	182
SECT409R1	-	-	-	-	-
SECT571K1	-	-	-	-	-
SECT571R1	-	-	-	-	-

Theorem 4. *Assuming in the interval $[p+1-\sqrt{p}, p+1+\sqrt{p}]$ there is an integer which is the product of three primes of roughly equal size, then there exists a string S which implies that the best algorithm to solve the EC-DHP for an elliptic curve of order p takes time at least*

$$O(\sqrt{p}/(\log_2 p)^2)$$

operations.

All that remains is to estimate the probability that a number of size around p is a product of three primes of roughly the same size. The number of primes of size around $p^{1/3}$ is roughly, by the prime number theorem, $3p^{1/3}/\log p$. Hence, the number of integers of size about p which are the product of three primes of roughly the same size is about

$$\frac{27p}{(\log p)^3}.$$

Hence, the probability is roughly $27/(\log p)^3$. Since this is a polynomial sized probability on an exponentially sized interval one can conclude that a string as in the above theorem must exist.

References

1. ANSI. *X9.62 – Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*. 1999.
2. NIST. *FIPS 186.2 Digital Signature Standard (DSS)*. 2000.

3. SECG. *SEC2 : Recommended Elliptic Curve Domain Parameters*. See <http://www.secg.org/>, 2000.
4. I.F. Blake, G. Seroussi and N.P. Smart. *Elliptic curves in cryptography*, Cambridge University Press, 1999.
5. H. Cohen. *A Course In Computational Algebraic Number Theory*. Springer-Verlag, GTM 138, 1993.
6. U. M. Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms. In *Advances in Cryptology – CRYPTO '94*, Springer LNCS 839, 271-281, 1994.
7. U. M. Maurer and S. Wolf. On the difficulty of breaking the DH protocol. *Technical Report #24*, Department of Computer Science, ETH Zurich, 1996.
8. U. M. Maurer and S. Wolf. Diffie-Hellman Oracles. In *Advances in Cryptology – CRYPTO '96*, Springer LNCS 1109, 268-282, 1996.
9. U. M. Maurer and S. Wolf. The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms. *SIAM Journal on Computing*, **28**, 1689–1721, 1999.
10. U. M. Maurer and S. Wolf. The Diffie-Hellman protocol. *Designs, Codes, and Cryptography*, **19**, 147–171, 2000.
11. H.-G. Rück. A note on elliptic curves over finite fields. *Math. Comp.*, **49**, 301–304, 1987.
12. R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p . *Math. Comp.*, **44**, 483–494, 1985.
13. W.C. Waterhouse. Abelian varieties over finite fields. *Ann. Sci. École Norm. Sup.*, **2**, 521–560, 1969.

Appendix A. *Elliptic curve domain parameters over \mathbb{F}_p*

Appendix A.1. *SECP112R1*

Found an elliptic curve for $D = 49271$ of order r , for which the smoothness bound B satisfies $B < 2^{24}$.

p	4451685225093714776491891542548933
a	3004369490124403223448210599048220
b	3673105177820870473395479313142990
r	4451685225093714803294692780292748

$$r = 2^2 \cdot 3 \cdot 23 \cdot 163 \cdot 1063 \cdot 1226387 \cdot 1356227 \cdot 6294503 \cdot 8891461.$$

Appendix A.2. *SECP112R2*

Found an elliptic curve for $D = 232$ of order r , for which the smoothness bound B satisfies $B < 2^{24}$.

p	1112921306273428674967732714786891
a	359905074524213046491509591844468
b	242752696076267039534173226322926
r	1112921306273428740027674877345678

$$r = 2 \cdot 3^2 \cdot 23^2 \cdot 29311 \cdot 140263 \cdot 1231487 \cdot 2081407 \cdot 11091127.$$

Appendix A.3. *SECP128R1*

Found an elliptic curve for $D = 1147$ of order r , for which the smoothness bound B satisfies $B < 2^{41}$.

p	340282366762482138443322565580356624661
a	172790473223220633077385689780905158119
b	95197664303165298255467477011327989561
r	340282366762482138439330622080962487075

$$r = 5^2 \cdot 37 \cdot 89^2 \cdot 937 \cdot 116341 \cdot 237781 \cdot 182865533 \cdot 9797974619.$$

Appendix A.4. *SECP128R2*

Found an elliptic curve for $D = 1099$ of order r , for which the smoothness bound B satisfies $B < 2^{32}$.

p	85070591690620534603955721926813660579
a	19079756378658212268578415722353658703
b	17286679956707251736290994162128614641
r	85070591690620534605542893917033437500

$$r = 2^2 \cdot 5^7 \cdot 103 \cdot 325541 \cdot 1901551 \cdot 1497538799 \cdot 2851021241.$$

Appendix A.5. *SECP160K1*

Found an elliptic curve for $D = 615$ of order r , for which the smoothness bound B satisfies $B < 2^{29}$.

p	1461501637330902918203687197606826779884643492439
a	1461501637330902918203686915170869725397159163571
b	17903465558938225297050987194894647156975772976
r	1461501637330902918203684599257432351992987979840

$$r = 2^6 \cdot 3 \cdot 5 \cdot 7 \cdot 29 \cdot 313 \cdot 859 \cdot 1693 \cdot 1861 \cdot 44371 \cdot 227089 \cdot 403681 \cdot 7954649 \cdot 273612893.$$

Appendix A.6. *SECP160R1*

Found an elliptic curve for $D = 1687$ of order r , for which the smoothness bound B satisfies $B < 2^{33}$.

p	1461501637330902918203687197606826779884643492439
a	718377688256110771217022131053884288216489138828
b	1238971813496228540776451419136332561991357802220
r	1461501637330902918203688424922129493127811783056

$$r = 2^4 \cdot 321203 \cdot 8923427 \cdot 29516021 \cdot 42625897 \cdot 3481179073 \cdot 7276295861.$$

Appendix A.7. *SECP160R2*

Found an elliptic curve for $D = 2047$ of order r , for which the smoothness bound B satisfies $B < 2^{46}$.

p	1461501637330902918203685083571792140653176136043
a	655062771545891081498390154252772734026497232152
b	1357472896926393615526429969353994153112175717483
r	1461501637330902918203687083877010067007102538697

$$r = 13 \cdot 59 \cdot 593 \cdot 216259 \cdot 33288527 \cdot 61258009 \cdot 177100211 \cdot 41143243334041.$$

Appendix A.8. *SECP192K1*

Found an elliptic curve for $D = 391443$ of order r , for which the smoothness bound B satisfies $B < 2^{37}$.

p	6277101735386680763835789423061264271957123915200845512077
a	2891688927942385717662330176188476121706509164722098684885
b	3631279080780438745071960993655785924653694921708037763767
r	6277101735386680763835789422941247413016760975163307189017

$$r = 7 \cdot 11 \cdot 13 \cdot 1051 \cdot 6793 \cdot 37549 \cdot 43133 \cdot 2271419 \cdot 5200957 \cdot 11660993 \cdot 47366447 \cdot 83112406499.$$

Appendix A.9. *SECP192R1*

Found an elliptic curve for $D = 334852$ of order r , for which the smoothness bound B satisfies $B < 2^{38}$.

p	6277101735386680763835789423176059013767194773182842284081
a	1481433377960133065432076431108995661271110021392435544043
b	2902118254460179386083821192003072328009361155076752304741
r	6277101735386680763835789423017727995705162516013110575168

$$r = 2^6 \cdot 2131 \cdot 2184989 \cdot 18476453 \cdot 33606343 \cdot 4164787607 \cdot 54362974597 \cdot 149834064623.$$

Appendix A.10. *SECP224R1*

Found an elliptic curve for $D = 41187$ of order r , for which the smoothness bound B satisfies $B < 2^{43}$.

p	26959946667150639794667015087019625940457807714424391721682722368061
a	23460081865821181303624713958223357609917948711549193721638896695929
b	3273417389998776218705144285430973820638637246093177637445286651529
r	26959946667150639794667015087019635299668807630150493702817664059123

$$r = 3^3 \cdot 149 \cdot 599 \cdot 857 \cdot 38299 \cdot 83101 \cdot 3691603 \cdot 7802849 \cdot 7620458239 \cdot 3019441906903 \cdot 6188589965407.$$

Appendix A.11. *SECP256K1*

Found an elliptic curve for $D = 56296$ of order r , for which the smoothness bound B satisfies $B < 2^{56}$.

p	115792089237316195423570985008687907852837564279074904382605163141518161494337
a	101536210918317263530844315620356466572420515739950505111806790686414677761558
b	95232046388132560069665168540287371714637458669069034950173986030825553082665
r	115792089237316195423570985008687907852197792184195191368021646146255312579200

$$r = 2^7 \cdot 5^2 \cdot 59 \cdot 46499 \cdot 93151 \cdot 94204592001827 \cdot 4214180265645761 \cdot 5538146513558221 \cdot 64401523664207893.$$

Appendix A.12. *SECP256R1*

Found an elliptic curve for $D = 41752$ of order r , for which the smoothness bound B satisfies $B < 2^{53}$.

p	1157920892103562487626974469494075735299969552241357603424222590 61068512044369
a	9570593245439767400553787783714695185846841060234040301136864782 9387908402942
b	3716676022000329468510218458731106244470537668936013453596789992 4412649482072
r	1157920892103562487626974469494075735293337467972563527979292961 62221692322308

$$r = 2^2 \cdot 3^2 \cdot 12256103 \cdot 15612089137 \cdot 7289979571159 \cdot 149179734594697 \cdot 1983840344370161 \cdot 7791482602842641.$$

Appendix A.13. *SECP384R1*

Found an elliptic curve for $D = 22312$ of order r , for which the smoothness bound B satisfies $B < 2^{83}$.

p	3940200619639447921227904010014361380507973927046544666794690527 9627659399113263569398956308152294913554433653942643
a	3495422476893511507764170933655197793772564054511972818307355261 2441088315258988909149521992373592603029088704011143
b	3268661575522837661036589866475303171469696409068184552089407303 3851417407607049334078498717401678595869535800502600
r	3940200619639447921227904010014361380507973927046544666793595219 3702934349926631776730637420557407670112513135662474

$$r = 2 \cdot 139 \cdot 19553 \cdot 1717730921 \cdot 9562711553 \cdot 10066439953 \cdot 298186652651 \cdot 2192234732221 \cdot 8854959912191 \cdot 1266378047297295563 \cdot 5980297858075074334711093.$$

Appendix B. *Elliptic curve domain parameters over \mathbb{F}_{2^m}*

Appendix B.1. *SECT113R1*

Found an elliptic curve for $D = 36883$ of order r , for which the smoothness bound B satisfies $B < 2^{27}$.

p	5192296858534827689835882578830703
a	3474938539161152927043580292550591
b	1558057946307173711043091072669181
r	5192296858534827604473796497656972

$$r = 2^2 \cdot 223 \cdot 263 \cdot 701 \cdot 101161 \cdot 205651 \cdot 16978771 \cdot 89386247.$$

Appendix B.2. *SECT113R2*

Found an elliptic curve for $D = 78859$ of order r , for which the smoothness bound B satisfies $B < 2^{22}$.

p	5192296858534827702972497909952403
a	3041840610520282906862898200203228
b	3069042394742773008284442116800394
r	5192296858534827760603176992948165

$$r = 5 \cdot 23 \cdot 379 \cdot 1193 \cdot 4691 \cdot 6317 \cdot 455237 \cdot 2263879 \cdot 3269753.$$

Appendix B.3. *SECT131R1*

Found an elliptic curve for $D = 1348$ of order r , for which the smoothness bound B satisfies $B < 2^{40}$.

p	1361129467683753853893932755685365560653
a	45239632266554388252161689599402116416
b	706868321905334292241779518902735015058
r	1361129467683753853832674756442350922482

$$r = 2 \cdot 11 \cdot 95327 \cdot 2175549221 \cdot 43465878091 \cdot 623951414393.$$

Appendix B.4. *SECT131R2*

Found an elliptic curve for $D = 410107$ of order r , for which the smoothness bound B satisfies $B < 2^{32}$.

p	1361129467683753853879535043412812867983
a	1140179019215365634862578195345768387972
b	67023528779017902874068420857795220826
r	1361129467683753853807176701419194246589

$$r = 7 \cdot 13 \cdot 17 \cdot 124904441 \cdot 674468357 \cdot 3166793159 \cdot 3297994789.$$

Appendix B.5. *SECT163K1*

Found an elliptic curve for $D = 7$ of order r , for which the smoothness bound B satisfies $B < 2^{47}$.

p	5846006549323611672814741753598448348329118574063
a	436666434877668258823803719348499160373586072970
b	2537677374898501979843125107803433313073823501906
r	5846006549323611672814737350185634603092193196584

$$r = 2^3 \cdot 7 \cdot 37 \cdot 109 \cdot 127 \cdot 163^2 \cdot 1621 \cdot 2377 \cdot 108217 \cdot 166456142911 \cdot 110524002744079.$$

Appendix B.6. *SECT163R1*

Found an elliptic curve for $D = 384591$ of order r , for which the smoothness bound B satisfies $B < 2^{38}$.

p	5846006549323611672814738465098798981304420411291
a	3598501456017062622843366105464281020697517386731
b	3794366797806473208539416266526735711914428597879
r	5846006549323611672814736357645947545594295623552

$$r = 2^7 \cdot 3 \cdot 11 \cdot 73 \cdot 83 \cdot 11987 \cdot 2436653 \cdot 96400099 \cdot 39366362213 \cdot 187341845711.$$

Appendix B.7. *SECT163R2*

Found an elliptic curve for $D = 6107$ of order r , for which the smoothness bound B satisfies $B < 2^{34}$.

p	5846006549323611672814742442876390689256843201587
a	1971779764808594439938298597120083718882412855992
b	2919208698799468165458356085216747700266610486436
r	5846006549323611672814741176549674399533963167327

$$r = 3 \cdot 7 \cdot 11 \cdot 73 \cdot 2969 \cdot 5253529 \cdot 31696801 \cdot 45160931 \cdot 1142969071 \cdot 13584708629.$$

Appendix B.8. *SECT193R1*

Found an elliptic curve for $D = 7$ of order r , for which the smoothness bound B satisfies $B < 2^{57}$.

p	6277101735386680763835789423269548053691575186051040197193
a	4214314953695281606317765669538470787451698106413109773246
b	5081199397163309482341515630009800876618472202695636568683
r	6277101735386680763835789423141002752577886153547101234964

$$r = 2^2 \cdot 16493 \cdot 20357 \cdot 307267 \cdot 507697 \cdot 2708335079 \cdot 32535336276871 \cdot 340022400028151.$$

Appendix B.9. *SECT193R2*

Found an elliptic curve for $D = 11$ of order r , for which the smoothness bound B satisfies $B < 2^{47}$.

p	6277101735386680763835789423314955362437298222279840143829
a	1682147688126917153631050709488859741134328492640033528607
b	264754754554262586709664461739324417205425201301983141038
r	6277101735386680763835789423274585688137975497386122403859

$$r = 3 \cdot 991 \cdot 4261 \cdot 81349 \cdot 2948267501 \cdot 3798403579 \cdot 5157510886093 \cdot 105461115430301.$$

Appendix B.10. *SECT233K1*

Found an elliptic curve for $D = 7$ of order r , for which the smoothness bound B satisfies $B < 2^{42}$.

p	3450873173395281893717377931138512760570940988862252126328087024 741343
a	2960715072569869853979448287313485194143356659529431958732372206 621038
b	3326135022600898528397930600960785714488324439782127540000056560 176002
r	3450873173395281893717377931138512772098228224648806713791596044 678224

$$r = 2^4 \cdot 3^2 \cdot 233^2 \cdot 6323 \cdot 8353 \cdot 11369 \cdot 16067 \cdot 42457 \cdot 8282401 \cdot 57838579969 \cdot 847443689801 \cdot 2654592796909.$$

Appendix B.11. *SECT233R1*

Found an elliptic curve for $D = 2263$ of order r , for which the smoothness bound B satisfies $B < 2^{69}$.

p	6901746346790563787434755862277025555839812737345013555379383634 485463
a	1820580472413438819478596485548429166239797309078298630715158034 355885
b	5566028978147099146744423773839461465717298343414823787585395847 618317
r	6901746346790563787434755862277025656535329344094001426198225867 440096

$$r = 2^5 \cdot 41 \cdot 179 \cdot 4421 \cdot 12113 \cdot 372709 \cdot 5690131288087 \cdot 684905249387674699 \cdot 377813292995836757497.$$

Appendix B.12. *SECT283K1*

Found an elliptic curve for $D = 7$ of order r , for which the smoothness bound B satisfies $B < 2^{30}$.

p	38853377844514581418389238136470378132848117337930613242958749975 29815829704422603873
a	70520703283217932184170826184835244812710662939658625745142527614 2497082742704087423
b	30473359959625905609761638006968804673610266633999617501689723576 43806457717015761693
r	38853377844514581418389238136470378132848092459699421970610175876 57671234911570097856

$$r = 2^6 \cdot 3^2 \cdot 11^2 \cdot 29^2 \cdot 71^2 \cdot 281^2 \cdot 491 \cdot 541 \cdot 631 \cdot 1051 \cdot 2017 \cdot 7393 \cdot 13721 \cdot 25621 \cdot 58321 \cdot 263201 \cdot 8160041 \cdot 34727701 \cdot 70155401 \cdot 590927681.$$

Appendix B.13. *SECT283R1*

Found an elliptic curve for $D = 11768$ of order r , for which the smoothness bound B satisfies $B < 2^{61}$.

p	7770675568902916283677847627294075626569625924376904889109196526 770044277787378692871
a	3422981077976423247239195216319872557284569707504901532042197958 683089247371980579520
b	6276574248286091477128707738969025423084065186293300973268585960 583153939014879333073
r	7770675568902916283677847627294075626569620569022047168760060447 800769935356398335566

$$r = 2 \cdot 3^4 \cdot 77017 \cdot 20644333 \cdot 18467891557 \cdot 5900455542323 \cdot 16487178989053 \cdot 11798339334074779 \cdot 1423266460791026459.$$

Appendix B.14. *SECT409K1*

Found an elliptic curve for $D = 7$ of order r , for which the smoothness bound B satisfies $B < 2^{81}$.

p	3305279843951242994759576540163855199142023414821406096423243950 22880711289249191050673258457777458014096366590617731358671
a	1033680698937310341345940257123806820112490512095292504499979695 10593783321670819331640037071502449026324693899776335435165
b	1285642659106471269597807458659735928564548877028381746666103128 58466841394491249918527543345304621504772105000387535535964
r	3305279843951242994759576540163855199142023414821406096423243673 00908325422789012626692102789067380459577492473485499336944

$$r = 2^4 \cdot 29^2 \cdot 71 \cdot 2843 \cdot 5279 \cdot 6323 \cdot 8353 \cdot 16067 \cdot 42457 \cdot 181281031 \cdot 1159018351 \cdot 8896753517 \cdot 37852407181 \cdot 4860847115041 \cdot 853621649145207671 \cdot 2179267320551430510798251.$$

A. Muzereau jimoid@club-internet.fr

Ecole Nationale Supérieure d'Informatique et de
Mathématiques Appliquées de Grenoble,
Rue de la Passerelle, 481,
Domaine Universitaire,
B.P.72, 38402 St Martin,
France.

N.P. Smart nigel@cs.bris.ac.uk

F. Vercauteren frederik@cs.bris.ac.uk

Dept. Computer Science,
University of Bristol,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB,
United Kingdom.