Separating Decision Diffie–Hellman from Diffie–Hellman in cryptographic groups

Antoine Joux[1] and Kim Nguyen *[2]

[1] DCSSI
18, rue du Dr. Zamenhoff
F-92131 Issy-les-Mx Cedex
France
Antoine.Joux@ens.fr
[2] Institut für experimentelle Mathematik
Universität GH Essen
Ellernstrasse 29
45326 Essen
Germany
nguyen@exp-math.uni-essen.de

Abstract. In many cases, the security of a cryptographic scheme based on Diffie–Hellman does in fact rely on the hardness of the Diffie–Hellman Decision problem. In this paper, we show that the hardness of Decision Diffie–Hellman is a much stronger hypothesis than the hardness of the regular Diffie–Hellman problem. Indeed, we describe a reasonably looking cryptographic group where Decision Diffie–Hellman is easy while Diffie–Hellman is equivalent to a – presumably hard – Discrete Logarithm Problem. This shows that care should be taken when dealing with Decision Diffie–Hellman, since its security cannot be taken for granted.

1 Introduction

Browsing through the cryptographic protocols based on Diffie–Hellman, we find that many of them rely on a stronger security hypothesis: the hardness of the so-called Decision Diffie–Hellman problem. This is especially true when aiming at high security levels, such as, for example, semantic security. Since the Decision Diffie–Hellman problem has been introduced in cryptography [2], its hardness has always been a concern: given a cryptographic group, is it sufficient to study the discrete logarithm problem in that group or do we need to assess the security of the Diffie–Hellman and Decision Diffie–Hellman problems in that particular group? In [7], Maurer and Wolf have given a strong heuristic argument that shows that the security of Diffie–Hellman in a given group should not be a concern, once the discrete logarithm problem is hard in that group. In this paper, we address the case of Decision Diffie–Hellman and show that it is much less hopeful.

* The second author acknowledges the financial support of SIEMENS AG, ZT IK3, Munich, Germany.

Indeed, we construct cryptographic groups based on elliptic curve cryptography where Decision Diffie–Hellman is easy, while Diffie–Hellman itself is equivalent to Discrete Logarithm. In order to construct such groups, we use ideas from [7], which we turn practical, to ensure the equivalence of Diffie–Hellman and Discrete Logarithm and from [4] to make Decision Diffie–Hellman easy. The main result of [4] was the introduction of a novel Diffie–Hellman like protocol that allows for three participants, using pairings on elliptic curves. Yet it was also explained that using the same technique, Decision Diffie–Hellman can be made easy in some special cases. However, the special cases that were given are somewhat artificial, since in order to get an efficient construction two different groups need to be pasted together. An open question was to efficiently construct a single elliptic curve where the Decision Diffie–Hellman problem becomes easy. In this paper, we show how the technique from [4] can be adapted to deal with a cryptographic group lying in a single elliptic curve.

2 Notations and background ideas

2.1 Diffie–Hellman and related assumptions

When doing cryptography using discrete logarithms in a group G, there are three related complexity assumptions on which the security usually relies. We now describe these three problems for an additive group $(G, +)$. For simplicity, we assume that G has prime order.

- **The DL problem.** The DL (discrete logarithm) problem, can be stated as follows. Given two group elements g and h, how to find an integer n, such that $h = ng$ whenever such an integer exists.
- **The DH problem.** The DH (Diffie–Hellman) problem, can be stated as follows. Given three group elements g, ag and bg, how to find an element h of G such that $h = (ab)g$.
- **The DDH problem.** The DDH (decision Diffie–Hellman) problem, can be stated as follows. Given four group elements g, ag, bg and cg, how to decide whether $c = ab$ (modulo the order of g).

Clearly, DDH is no harder than DH and DH is no harder than DL. However, in the general case, we do not know more than that about the relations between these three problems. The goal of this paper is to separate DDH from DH, i.e. to describe a group where DDH becomes easy while DH becomes equivalent to DL. Of course, we want to avoid the trivial case where DL is known to be easy, such as the additive group of a finite field.

2.2 Where DH and DL become equivalent

In [7], it was shown that in a group G of prime order q, the DL can be solved using a DH–oracle, if some auxillary group A with nice properties can be defined over $\mathbb{F}_q$. More precisely, we need efficient algorithms that embed a large proportion of elements of $\mathbb{F}_q$ in A, that extract an embedded value from an element of A and that quickly compute discrete logarithms in A. Of course, the embedding and extracting maps should satisfy the natural property that whenever x has an embedding g, the extraction of g gives back x. One possible choice of auxillary groups is to use elliptic curves defined over $\mathbb{F}_q$ with sufficiently smooth order. In this kind of curve, thanks to the Pohlig–Hellman algorithm, the DL problem is easy. If such curves can be found, the DH and DL problems become equivalent in G. However, unless we can prove that sufficiently many smooth numbers exist around q, there exists no provable technique to find such curves. Thus, in general, [7] does not give a proof that DL and DH are equivalent, but only a heuristic argument. However it also states that in practice, finding a good auxillary curve does indeed make the two problems equivalent. We use this fact in section 3.

2.3 Pairings on elliptic curves

On elliptic curves, there exist some bilinear functions that map a pair of ℓ–torsion points (P, Q) to an ℓ–th root of unity $\langle P, Q\rangle$. The bilinearity simply means that:

$$\langle aP, bQ\rangle = \langle P, Q\rangle^{ab}.$$

These functions are called pairings. Among those, the Weil and Tate pairings are quite well known. For most elliptic curves, the pairing is defined over such a large extension of the base field that it cannot be computed and is useless for cryptographic purposes. However, when the ℓ-th roots of unity appear in a reasonably small extension, then pairings can become practical. Pairings were first used in cryptography in [8] to show that the DL problem can be transported from a super-singular curve to a finite field. In fact this construction did use the Weil pairing. In practice, the Tate pairing in Lichtenbaum's version, as described in [3], is better than the Weil pairing. Indeed, in order to be non-degenerate (i.e. non-constant) the Weil pairing need to be computed on two independent torsion points, that is two points P and Q such that neither is a multiple of the other. For the Tate pairing, the requirement is relaxed; indeed in some cases, the pairing of P with itself is different from one. As was noted in [4], whenever this happens, the decision Diffie–Hellman problem in the group generated by P becomes easy since:

$$\langle aP, bP\rangle = \langle P, P\rangle^{ab},$$
$$\langle P, cP\rangle = \langle P, P\rangle^{c}.$$

Thus deciding whether $c \equiv ab \pmod{\ell}$ can be done by testing if $\langle aP, bP\rangle = \langle P, cP\rangle$.

The case of trace 2 curves. In [4], it was mentioned that in the case of trace 2 curves, i.e. curves with $p - 1$ points, we can be sure, given a point P of order ℓ that $\langle P, P\rangle \neq 1$ as long as ℓ^2 does not divide $p - 1$. However, constructing such curves is an open problem. Indeed, the only known method to efficiently build curves of trace 2 is by complex multiplication techniques [1,5]. However, with this construction $p-1$ is necessarily equal to dn^2 where d is a small number. Thus, we cannot guarantee that $\langle P, P\rangle \neq 1$. Yet, in some reasonably frequent cases, this property still holds [11], thus it is possible to efficiently construct some trace 2 curves where DDH is easy.

The case of supersingular curves. With supersingular curves defined over $\mathbb{F}_p$, the properties of the pairing imply that a ℓ–torsion point P with coordinates in $\mathbb{F}_p$ satisfies $\langle P, P\rangle = 1$. This means that we need a point that is independent from P to get a useful result. Luckily, the fact that the curve is supersingular means that its endomorphism ring has a very special structure. Namely, it contains extra endomorphisms which cannot be written as a combination of the usual endomorphisms (Frobenius and multiplication by integers). These extra endomorphisms map points defined over the ground field to points defined over an extension field. For such an endomorphism Φ we might get the nice property, $\langle P, \Phi(P)\rangle \neq 1$. In that case, we can solve DDH by comparing $\langle aP, \Phi(bP)\rangle$ and $\langle P, \Phi(cP)\rangle$. Table 1 describes some possible extra endomorphism Φ for frequently encountered supersingular curves over prime fields $\mathbb{F}_p$. The first two cases presented in the table are the well known supersingular curves with $p+1$ points. The third case is a supersingular curve defined over $\mathbb{F}_{p^2}$ with p^2-p+1 points. The Weil pairing maps this curve to the group used in the XTR cryptosystem (see [6]), as was first pointed out at the rump session of Crypto'00 [9]. Since then, it has been looked at in deeper details [12] and it turns out that DDH is easy in the elliptic curve and is presumably hard in the XTR group.

Field	Curve	Morphism	Conditions	Group order
$\mathbb{F}_p$	$y^2 = x^3 + ax$	$(x, y) \mapsto (-x, iy)$ $i^2 = -1$	$p \equiv 3 \pmod 4$	p+1
$\mathbb{F}_p$	$y^2 = x^3 + a$	$(x, y) \mapsto (\zeta x, y)$ $\zeta^3 = 1$	$p \equiv 2 \pmod 3$	p+1
$\mathbb{F}_{p^2}$	$y^2 = x^3 + a$	$(x, y) \mapsto (\omega \frac{x^p}{r^{(2p-1)/3}}, \frac{y^p}{r^{p-1}})$ $r^2 = a, r \in \mathbb{F}_{p^2}$ $\omega^3 = r, \omega \in \mathbb{F}_{p^6}$	$p \equiv 2 \pmod 3$	$p^2 - p + 1$

Table 1. Extra endomorphism in some supersingular curves

3 Constructing groups that separate DH and DDH

Using the ideas from section 2.2 and 2.3, constructing groups where DDH is easy and where DH and DL are provably equivalent is now a simple matter. Indeed, we use one of the elliptic curves proposed in section 2.3, together with a large prime divisor q of its order. We also need an auxillary curve defined over $\mathbb{F}_q$ of sufficiently smooth order. In order to construct these parameters, we start by choosing q and the auxillary curve. Several different methods can be used at that point. A first solution is to choose q and a random curve, to count the number of points and to test whether it is smooth. When q is a 160–bit number, this can be done reasonably quickly using an early abort strategy as shown in section 4. Alternatively, we can decide to work with a special curve whose number of points is easy to compute, e.g. a supersingular curve or a curve with complex multiplication. Assume that we decided to use the super-singular curve $y^2 = x^3 + x \pmod{q}$, then when $q \equiv 3 \pmod{4}$ is a prime, the order of the curve is $q+1 \equiv 0 \pmod{4}$. Now choose a smooth number $m \equiv 0 \pmod{4}$; if $m-1$ is prime, we are done. For 160–bit numbers, this procedure is very efficient. Alternatively, we can use complex multiplication, some possible curves of that kind are shown in table 2. Note that all the imaginary quadratic fields in table 2 have class number 1, however the construction method works quite easily for class numbers up to several hundreds.

Complex multiplication by main order of	Curve	Conditions	Group order
$\mathbb{Q}(\sqrt{-2})$	$y^2 = x^3 + 4x^2 + 2x$	$p = 2n^2q^2 + 1$	$2n^2q^2$
$\mathbb{Q}(\sqrt{-11})$	$y^2 + y = x^3 - x^2 - 7x + 10$	$p = 11n^2q^2 + 1$	$11n^2q^2$
$\mathbb{Q}(\sqrt{-43})$	$y^2 + y = x^3 - 860x + 9707$	$p = 43n^2q^2 + 1$	$43n^2q^2$

Table 2. Some complex multiplication curves

Once q is known, we need to choose p and an elliptic curve such that DDH is easy. For all the curves we have considered in section 2.3, p can be expressed as a simple function of the number of points which is a multiple of q (or even q^2 in some cases). For example, when using the supersingular curve $y^2 = x^3 + x$, p should be of the form $\lambda q - 1$ and when using complex multiplication, p should be of the form $dn^2q^2 + 1$ with d small. Finding prime values of that form is easy when dealing with the sizes usually encountered in cryptography such as 1024–bit numbers.

4 Examples

4.1 Finding q and an auxillary curve

Using the point counting algorithm by Schoof–Atkin–Elkies a curve over a prime field $\mathbb{F}_q$ where q is a 160–bit prime can be counted in about a minute, hence it is reasonable to choose random curves defined over $\mathbb{F}_q$ and search for one with a smooth group order. Indeed, 160–bit primes have about 48 digits and we know that the elliptic curve factoring method can find 50–digit factors (see [10]), using a smoothness bound $B = 56 \cdot 10^6$ by trying 5300 curves on average. Hence, with such a smoothness bound, we can expect to find a curve in less than 4 days on a single machine. In fact, taking into account the fact that each point counting execution gives the number of points on two curves (the original one and its twist) this estimation can be lowered to two days. We did search for a good curve of the form $y^2 = x^3 + x + b$ for the prime $q = 2^{160} + 7$. For example, we found that for $b = 93$, the number of points on the twist is:

$$3 \cdot 7 \cdot 17 \cdot 827 \cdot 1811 \cdot 117427 \cdot 519797 \cdot 1377931 \cdot 2160461 \cdot 65938193 \cdot 228136331.$$

Writing the equation of the twist as $y^2 = x^3 + 25x + 125 \cdot 93$, we get a good curve with smoothness bound 228136331.

However we can reverse this process, meaning that we choose a smooth number first and then construct a curve which has this number of points. One way to do this is to find a smooth number n such that $q = n - 1$ is prime and $q \equiv 3 \pmod 4$. Then the supersingular curve $y^2 = x^3 + x$ defined over $\mathbb{F}_q$ will have group order $q + 1$ which is smooth by construction. To find such a number, we can start from the 149–bit number:

$$n = 2^2 \cdot 3 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 \cdot 53 \cdot 59 \cdot 61 \cdot 67 \cdot 71 \cdot 73 \cdot 79 \cdot 83 \cdot 89 \cdot 97 \cdot 101 \cdot 103 \cdot 107 \cdot 109.$$

Then we look for a prime of the form $\lambda n - 1$, with λ in the range $[1307 \ldots 2612]$. Many choices of λ result in a good candidate, for example $\lambda = 2^4 \cdot 163 = 2608$ is the largest possible such value. We let $q_1 = 2608 \cdot n - 1$, it is a prime congruent to $3 \pmod 4$, hence $y^2 = x^3 + x$ is a desired curve of smooth order over $\mathbb{F}_{q_1}$.

Alternatively, we can make use of complex multiplication in order to produce curves with smooth order. Consider for example the prime

$$q_2 = 2 \cdot 3^2 \cdot 5^2 \cdot 7^4 \cdot 11^4 \cdot 13^2 \cdot 17^2 \cdot 19^4 \cdot 23^4 \cdot 29^4 \cdot 41^4 \cdot 43^2 \cdot 47^2 \cdot 53^4 \cdot 59^2 \cdot 71^2 + 1.$$

By construction this prime splits in $\mathbb{Q}(\sqrt{-2})$ and hence gives rise to an elliptic curve of trace 2: the quadratic twist of the curve $y^2 = x^3 + 4x^2 + 2x$ over $\mathbb{F}_{q_2}$. This curve has smooth group order $q_2 - 1$.

4.2 Choosing p and the main curve

Once q is chosen, p is easily constructed. For example, we can now find a supersingular curve whose order is a multiple of q_1. We simply look for a prime of the form $4\lambda q_1 - 1$. Here is a possible value:

$$\begin{aligned} p_1 = {} & 17976931348623159077293051907890247336179769789423065727343008115773267580550096313270847732 \\ & 24075360211201138798713933576587897688144166224928474306394741243777678934248654852763022196 \\ & 01246094119453082952085005768838150682342462881473913110540827237163350508117621323351641252 \\ & 319835054989875442662233317, \end{aligned}$$

Now the supersingular curve $y^2 = x^3 + x$ defined over $\mathbb{F}_{p_1}$ has a subgroup of order q_1.

As explained before, we can also work with trace 2 curves. For example, the prime

$$\begin{aligned} p_2 = {} & 10670060186383776944160772179366270641259594083276237150022019567758593793429649789360814336 \\ & 76897328420496223452607179142772034762933054027679044991310709383278097774649981692256846811 \\ & 97721746477548968118598258710445553784811268805673258859628640140181305129265780039630649196 \\ & 38999286909139705330098359729606070445812557282252233 \end{aligned}$$

is of the form

$$p_2 = 2 \cdot q_2^2 \cdot n^2 + 1.$$

Hence the quadratic twist of $y^2 = x^3 + 4x^2 + 2x$ over $\mathbb{F}_{p_2}$ has order $p_2 - 1$. On that curve of equation $y^2 = x^3 + 4 \cdot 5x^2 + 2 \cdot 25x$, there exists a point P of order q_2 with $\langle P, P \rangle \neq 1$.

5 Conclusion

The above construction of reasonably looking cryptographic groups where Decision Diffie–Hellman is easy, while Diffie–Hellman is known to be as hard as Discrete Logarithm gives an eery feeling about all the cryptographic protocols that use the DDH assumption, especially when dealing with elliptic curve cryptography. We feel that this issue needs to be addressed in the near future. This could be done either by devising protocols that avoid DDH altogether or by proving that in certain cases, this problem also becomes provably hard.

References

1. A.O.L. Atkin and F. Morain. Elliptic curves and primality proving. *Mathematics of Computation*, 61:29–68, 1993.
2. S. Brands. An efficient off–line electronic cash system based on the representation problem. Technical Report CS–R9323, CWI, Amsterdam, 1993.
3. G. Frey, M. Müller, and H.-G. Rück. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Transactions on Information Theory*, 45(5):1717–1718, 1999.
4. A. Joux. A one round protocol for tripartite Diffie–Hellman. In Wieb Bosma, editor, *Proceedings of the ANTS-IV conference*, volume 1838 of *Lecture Notes in Comput. Sci.*, pages 385–394. Springer, 2000.
5. G.-J. Lay and H. Zimmer. Constructing elliptic curves with given group order over large finite fields. In L. Adleman, editor, *Algorithmic Number Theory*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 250–263. Springer, 1994.
6. A. Lentra and E. Verheul. The XTR public key system. In Mihir Bellare, editor, *Proceedings of CRYPTO'2000*, volume 1880 of *Lecture Notes in Comput. Sci.*, pages 1–19. Springer, 2000.
7. U. Maurer and S. Wolf. The relationship between breaking the Diffie–Hellman protocol and computing discrete logarithms. *SIAM J. Comput.*, 28(5):1689–1721, 1999.
8. A. Menezes, T. Okamoto, and S. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transaction on Information Theory*, 39:1639–1646, 1993.
9. A. Menezes and S. Vanstone. ECSTR(XTR): Elliptic curve singular trace representation. Rump session of Crypto'00, August 2000.
10. P. Montgomery. *An FFT Extension of the elliptic curve method of factorization.* PhD thesis, University of California, Los Angeles, 1992.
11. H. G. Rück and K. Nguyen. A comparison of the Weil and Tate pairing. preprint.
12. E. Verheul. XTR is more secure than supersingular elliptic curve crypto systems. Preprint.