J. Cryptology (2003) 16: 239–247
DOI: 10.1007/s00145-003-0052-4

Journal of
CRYPTOLOGY

Separating Decision Diffie–Hellman from Computational Diffie–Hellman in Cryptographic Groups*

Antoine Joux
DCSSI Crypto Lab, 51 Bd de Latour Maubourg,
F-75700 Paris 07 SP, France
Antoine.Joux@m4x.org

Kim Nguyen
Institut für experimentelle Mathematik, Universität GH Essen,
Ellernstrasse 29, 45326 Essen, Germany
nguyen@exp-math.uni-essen.de

Communicated by Cynthia Dwork

Received December 2000 and revised 17 December 2002
Online publication 10 July 2003

Abstract. In many cases the security of a cryptographic scheme based on computational Diffie–Hellman does in fact rely on the hardness of the decision Diffie–Hellman problem. In this paper we construct concrete examples of groups where the stronger hypothesis, hardness of the decision Diffie–Hellman problem, no longer holds, while the weaker hypothesis, hardness of computational Diffie–Hellman, is equivalent to the hardness of the discrete logarithm problem and still seems to be a reasonable hypothesis.

Key words. Discrete logarithm, Diffie–Hellman, Elliptic curve, Weil pairing.

1. Introduction

The discrete logarithm (DL) problem is, together with factorization, one of the main problems upon which public-key cryptosystems are built. Thus, efficiently computable groups where the DL problem is hard are very important in cryptography. However, proving the hardness of the DL problem in any group is a difficult open question. As a consequence, in order to decide if a group can be used in cryptography, one usually checks whether known algorithms can break the DL problem in that group. This is done

* The second author acknowledges the financial support of SIEMENS AG, ZT IK3, Munich, Germany. His current address is Cryptology Competence Center, Business Life Identification, Philips Semiconductors GmbH, Postfach 54 01 40, D-22502 Hamburg, Germany. kim.nguyen_3@philips.com.

by taking into account two classes of algorithms, generic algorithms which works for any group and do not take advantage of the specific representation and nongeneric algorithms whose scopes are limited to specific groups. The complexity of generic algorithms for computing discrete logarithms in a group grows as the square root of the largest prime factor of the cardinality. One well-known algorithm with such a complexity is Pollard's Rho method. Nongeneric algorithms, such as index calculus, need to be evaluated on a case by case basis. However, taking into account the hardness of the DL problem is not usually sufficient. Indeed, the proofs of security of many cryptosystems rely on either the computational Diffie–Hellman CDH problem or the decision Diffie–Hellman DDH problem. It is well known that these two problems are no harder than the DL problem itself. Moreover, there are known cases where DDH is easy while DL is still presumably hard. Indeed, whenever the group size has a small prime factor, computing discrete logarithms modulo this factor is easy. Of course, this does not help to compute the rest of the discrete logarithm, however, it suffices to decide DDH with probability better than $1/2$. A good survey about these three problems and their applications in cryptography is given in [3]. Moreover, Shoup proved in [15] that in the generic group model (i.e., in groups where no nongeneric algorithms may exist), no algorithm faster than the known square-root approach can exist for any of the three problems DL, CDH, and DDH.

In 1994 Maurer used a variation of the elliptic curve factoring method to give strong evidence that CDH and DL are probably equivalent (see [9]). This approach was formalized by Maurer and Wolf in [10] and finally appeared as a journal version in [11]. More recently, a class of groups where DDH becomes easy was used to construct cryptosystems. This class of groups is based on elliptic curves and pairings. It was used in [5] to construct a tripartite Diffie–Hellman protocol and it perfectly fits in the framework of gap problems as introduced in [14]. Our goal in this paper is to merge the two approaches and to construct a family of groups where CDH and DL are equivalent and presumably hard, while DDH is provably easy. Under the hypothesis that DL is indeed hard in our family of groups, such a construction separates CDH from DDH. Moreover, we propose to build this family in a way that does not rely on the unproven smoothness assumption from [11], but on known properties of the density of prime numbers in arithmetic series. In order to show that the construction is realistic, we give, in Section 4, a concrete example of such groups.

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, find an integer n, such that $h = ng$ whenever such an integer exists.

- **The CDH problem.** The CDH (computational Diffie–Hellman) problem can be stated as follows. Given three group elements g, ag, and bg, 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, decide whether $c = ab$ (modulo the order of g).

Clearly, DDH is no harder than CDH and CDH 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 CDH, i.e., to describe a group where DDH becomes easy while CDH becomes equivalent to DL and presumably hard. Indeed, we want to avoid the trivial cases where the three problems DL, CDH, and DDH are easy, such as the additive group of a finite field.

2.2. *The Case of Elliptic Curves*

Among the groups that populate cryptosystems, elliptic curves are frequently encountered. An elliptic curve defined over a finite field $\mathbb{F}_{p^s}$ forms a group G. In general, no nongeneric algorithm is known for solving the DL problem on elliptic curves. However, in some specific cases, there exists a fast algorithm that moves the DL problem from the curve to the multiplicative group of an extension $\mathbb{F}_{p^{rs}}$ of its field of definition. This algorithm makes use of bilinear functions called pairings.

These functions map pairs of ℓ-torsion points (P, Q) to the ℓ-th roots of unity $\langle P, Q\rangle$. The bilinearity simply means that

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

These pairings take their values in an extension $\mathbb{F}_{p^{rs}}$ of the field of definition of the elliptic curve. Two constructions of pairing are frequently encountered. The first construction, called the Weil pairing, was introduced in cryptography in [12] to show that the DL problem can be transported from a supersingular curve over a finite field to a small extension of this field. The second construction, called the Tate pairing, was proposed in [4] as a more efficient alternative to the Weil pairing.

The pairings can be defined for all elliptic curves. However, they can be efficiently computed only when r is small enough, by using an algorithm which was originally proposed by Miller in [13]. It is known that a pairing can be defined in an extension of degree r, where r is the smallest integer such that ℓ divides $p^{rs} - 1$. The easiest case, often encountered in pairing-based applications, is the case of supersingular curves, where $r = 2$ in large characteristic and can go up to 6 in small characteristic (the maximum occurs for curves in characteristic 3). In recent years pairing-based cryptography has been a fast growing new field; a survey of this growth can be found in [6].

2.3. *Where CDH and DL Become Equivalent*

In [9] the relation between DL and CDH in groups of known order was studied. The main result gives conditions for the equivalence of the two problems when the factorization

of the group order is known. For a group with prime order, the theorem can be restated as follows:

Theorem 1. *Let G be a (cyclic) group of prime order q. If E is an elliptic curve defined over $\mathbb{F}_q$ whose order $\mathcal{O}$ is B-smooth for some smoothness bound B, then discrete logarithms in G can be computed using $O(\log^2 q)$ calls to a DH-oracle and $O((B/\log(B))\log^2 q)$ group operations.*

We recall that a number is said to be B-smooth when all its prime factor are no larger than B.

In [9] and in the related papers [10] and [11] the existence of a good elliptic curve satisfying the condition of the theorem was considered. However, all the general statements are conjectural and depend on the existence of sufficiently smooth integers near q. Moreover, even if such a number $\mathcal{O}$ exists near q, constructing a curve over $\mathbb{F}_q$ with cardinality $\mathcal{O}$ is a hard problem and thus the theorem does not lead to an effective algorithm proving the equivalence of DL and CDH. However, if a family of groups is given together with their auxiliary curves (as in Theorem 1), we can use the result of Maurer and Wolf to prove polynomial-time equivalence between DL and CDH; assuming that the smoothness bound B is polynomial in $\log q$. We use this fact in Section 3.

2.4. *Where DDH Becomes Easy*

Let G be the group of prime order ℓ generated by an ℓ-torsion point P of an elliptic curve. When considering the DDH problem in this group, it is natural to view pairings as tools to solve this problem. Indeed, whenever (P, aP, bP, cP) is a decision Diffie–Hellman instance, we have

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

We known that $\langle P, P\rangle$ is either 1 or a primitive ℓ-th root of unity. In the first case the pairing does not help to solve DDH. However, in the second case we know that $c = ab$ if and only if $\langle aP, bP\rangle = \langle P, cP\rangle$. Thus, whenever $\langle P, P\rangle \neq 1$ computing DDH is easy. However, the known properties of pairings and more precisely the nondegeneracy properties are not sufficient to prove $\langle P, P\rangle \neq 1$. With the Weil pairing, the situation is even worse, since the definition of this pairing implies that $\langle P, P\rangle$ is always equal to 1. How can we construct a pairing with a strong nondegeneracy property $\langle P, P\rangle \neq 1$?

The case of supersingular curves. A first construction of curves with a pairing and a point P such that $\langle P, P\rangle \neq 1$ involves supersingular curves. With a supersingular curve defined over $\mathbb{F}_p$, the properties of the usual Weil and Tate pairing imply that any ℓ-torsion point P with coordinates in $\mathbb{F}_p$ satisfies $\langle P, P\rangle = 1$. However, Verheul proposed in [16] to use some special endomorphisms that he calls distortions to build modified pairings. These distortions map points defined over the ground field to points defined over an extension field. For such an endomorphism Φ we get the nice property $\langle P, \Phi(P)\rangle \neq 1$. As a consequence, we can use the modified pairing $\langle \cdot, \Phi(\cdot)\rangle$ to solve

Table 1. Distortions in some supersingular curves ($p > 3$).

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

DDH in the group generated by P. Table 1 describes some distortions Φ for frequently encountered supersingular curves over finite fields.

Trace 2 curves. Another construction of curves with a pairing and a point P such that $\langle P, P\rangle \neq 1$ was mentioned without proof in [5]. It involves trace 2 curves, i.e., curves with $p - 1$ points. Assume that we are given a trace 2 curve defined over $\mathbb{F}_p$, with a large prime ℓ dividing $p - 1$ such that ℓ^2 does not divide $p - 1$. In that case the ℓ-torsion contains exactly ℓ points that are defined over $\mathbb{F}_p$, the rest of the ℓ-torsion being in some extension of $\mathbb{F}_p$. As a consequence, there exists an ℓ-torsion point P that generates the subgroup of the ℓ-torsion defined over $\mathbb{F}_p$. According to the main theorem in [4], the Tate pairing is a surjective map. This implies that there exist two ℓ-torsion points Q and R such that $\langle Q, R\rangle \neq 1$. Since P generates the subgroup of ℓ-torsion points defined over $\mathbb{F}_p$, we can write $Q = aP$ and $R = bP$. As a consequence, we conclude that $\langle P, P\rangle \neq 1$.

However, constructing such curves is an open problem. Indeed, the only known method to build curves of trace 2 efficiently is by complex multiplication techniques [1], [7]. Yet, with this construction $p - 1$ is necessarily equal to dn^2 where d is a small number. Thus, ℓ^2 divides $p - 1$ and we cannot guarantee that the ℓ-torsion has exactly ℓ points. When ℓ^2 points of ℓ-torsion are present, the relation $\langle P, P\rangle \neq 1$ will not hold for all (nonzero) ℓ-torsion points. However, since the Tate pairing is nondegenerate, it can be shown that when the Tate really differs from the Weil pairing, then among the $\ell + 1$ subgroups of order ℓ, at most two will contain points which are self-degenerate. First, note that the Tate pairing can be written as a product of a symmetric and an antisymmetric pairing. The symmetric pairing S is

$$S(P, Q) = (\langle P, Q\rangle \cdot \langle Q, P\rangle)^{1/2}$$

and the antisymmetric pairing A is

$$A(P, Q) = \left(\frac{\langle P, Q\rangle}{\langle Q, P\rangle}\right)^{1/2}.$$

Note that since ℓ is an odd prime, the above square roots are well defined in the group of ℓ-th roots of unity. Moreover, the antisymmetric pairing is the Weil pairing raised to

some constant power. Assume that the Tate pairing itself is not antisymmetric, then S is nondegenerate and symmetric. Moreover, for any point P, we have $\langle P, P\rangle = S(P, P)$. Thus, without loss of generality, to analyze the behavior of the Tate pairing on single point evaluation, we can assume that the Tate pairing is symmetric.

With this hypothesis in mind, assume that P and Q are two linearly independent ℓ-torsion points. Then all ℓ-torsion points can be expressed as $R = aP + bQ$. By bilinearity and symmetry of the pairing, we see that

$$\log(\langle R, R\rangle) = a^2 \log(\langle P, P\rangle) + 2ab \log(\langle P, Q\rangle) + b^2 \log(\langle Q, Q\rangle),$$

where log is the discrete logarithm in the finite field. Thus, $\log(\langle R, R\rangle)$ can be expressed as a homogeneous polynomial in a and b of degree at most two. Thanks to the nondegeneracy of the pairing, this polynomial is nonzero. Therefore, the polynomial has at most two homogeneous roots. Thus, at most two of the $\ell + 1$ possible subgroups can be self-degenerate.

3. A Family of Groups that Separate CDH and DDH

In this section we merge the ideas from Sections 2.3 and 2.4, in order to construct groups where the DDH problem is easy and where CDH and DL are provably equivalent. Moreover, the groups are such that DL is presumably hard. The trick is to use a family of elliptic curves that have a single point pairing as in Section 2.4, in a way that allows us to exhibit a good auxiliary curve as explained in Section 2.3.

Since we use supersingular curves in this construction, the technique from [12] that transports the discrete logarithm from the main elliptic curve to $\mathbb{F}_{p^2}$ can be applied. This means that the best known algorithm for the computation of discrete logarithms on the curve will be subexponential in size p^2. With this in mind, we can either choose to use a relatively small q to balance the runtime of generic and nongeneric algorithms or use a value of q near p. For simplicity of exposition, we choose the latter possibility. However, the reader should keep in mind that in practical applications it is probably better to choose q much smaller than p.

The first step is to choose a size parameter B. Then we randomly pick prime numbers smaller than B and multiply them together until their product ω_0 becomes larger than 2^B and thus lies in the interval $[2^B, B \cdot 2^B]$. Let $\omega = 3\omega_0$ and search for the smallest prime q in the arithmetic sequences $k\omega - 1$. Once q is found, search for the smallest prime p in the arithmeric sequence $4lq - 1$. Assume for the moment that k and l are smaller than B. Then, since $p \equiv 3 \pmod 4$, the elliptic curve defined by the equation $y^2 = x^3 + x$ over $\mathbb{F}_p$ is supersingular. It defines a group of cardinality $p + 1 = 4lq$. In this group the DL problem reduces to the DL problem in the subgroup $\mathbb{G}_q$ of order q. Indeed, computing DL in the subgroup of order $4l$ can be done in polynomial time in p, since $4l$ is of the order of the logarithm of p. Since q^2 does not divide $p + 1$, we know from Section 2.4 that the DDH problem is easy in $\mathbb{G}_q$.

Moreover, since $q \equiv 2 \pmod 3$, the elliptic curve $y^2 = x^3 + 1$ over $\mathbb{F}_q$ is supersingular and has order $q + 1 = k\omega$. We can check that $q + 1$ is B-smooth, with B of the order of $\log(q)$. As a consequence, we can use the result of Maurer and Wolf to get a polynomial time reduction between the DL and the CDH problems in $\mathbb{G}_q$.

Note that as it is stated this construction is heuristic, since we did not prove that k and l are small enough. However, the repartition of prime numbers in arithmetic sequences and, according to Dirichlet's theorem, when a and b are coprime, means there are infinitely many primes in the sequence $a+b\cdot k$. Moreover, the density of primes in these sequences has been well studied and the above sequence asymptotically contains $n/(\varphi(b)\log(n))$ primes. Thanks to this result, we expect k and l to be polynomial in the size of the numbers we are generating. Thus, our generation algorithm is efficient.

4. Examples

In this section we give concrete examples of the construction from the previous section, in the range of interest of cryptographic groups, i.e., with a prime p of more than 1024 bits. With this choice, the discrete logarithm algorithm of Menezes et al. from [12] is in term of performance roughly equivalent to the factorization of a 2048-bit number. While much faster than generic algorithms, this is clearly out of reach.

Let ω be the following 1025-bit number:

$$\begin{aligned}\omega \;=\;& 28614152364758670754979783444264950724447325424621817219022377\\ & 98118285353898731383055715293731110481311224301068506150248137\\ & 70577172500713174603375895936535479551488189551415389394380573\\ & 34010503210028199274963755950022335547173968127510867339122042\\ & 46467407071834626059446823238131719891516369377491294936843 24\\ =\;& 2^2\cdot 3^3\cdot 7\cdot 11^3\cdot 17^2\cdot 19\cdot 23\cdot 31\cdot 37^2\cdot 43\cdot 53\cdot 67\cdot 71\cdot 73^2\cdot 83^2\cdot 107^3\cdot 113\\ & \cdot 131\cdot 137\cdot 139\cdot 151\cdot 157^2\cdot 163\cdot 167\cdot 173\cdot 179^2\cdot 191\cdot 193\cdot 227^3\cdot 229^2\\ & \cdot 239\cdot 241^2\cdot 251\cdot 257\cdot 269\cdot 281^2\cdot 293\cdot 307\cdot 311\cdot 313\cdot 317\cdot 337^3\cdot 347\cdot 353\\ & \cdot 367\cdot 401\cdot 409\cdot 431^3\cdot 433^2\cdot 443\cdot 457^2\cdot 461^2\cdot 463\cdot 467\cdot 487\cdot 503^2\cdot 521\\ & \cdot 523\cdot 541\cdot 547\cdot 557\cdot 571\cdot 587\cdot 593\cdot 599\cdot 607^4\cdot 613^2\cdot 641\cdot 653^2\cdot 659\cdot 673\\ & \cdot 677^2\cdot 733\cdot 739\cdot 743\cdot 769\cdot 787^2\cdot 809^2\cdot 821\cdot 829\cdot 839\cdot 853\cdot 857\cdot 863\cdot 877\\ & \cdot 881^2\cdot 883\cdot 911\cdot 919\cdot 937^2\cdot 947\cdot 953^2\cdot 997\end{aligned}$$

From ω, we can now compute $q = 90\cdot\omega - 1$ and $p = 4\cdot 10\cdot q - 1$. We can check that p and q are primes and verify the conditions of the construction proposed in Section 3. These numbers were generated using the following GP-PARI code (see [2]):

```
w=3;while(w<2^1024,w=w*prime(random(168)+1))
k=1;while((isprime(k*w-1))==0,k=k+1);print("k=",k);q=k*w-1
l=1;while(((isprime(4*l*q-1))==0),l=l+1);
print("l=",l);p=4*l*q-1
```

Experiments show that this code usually runs in a minute or less (more precisely 1000 experiments on a Pentium II at 450 Mhz took 760 minutes) and frequently gives values

of k and l under 1000. Among our 1000 experiments, 12 values of k and 60 values of l were above 1000. No value of k was greater than 2000 and no value of l was greater than 3000.

Therefore, the theoretical construction proposed in Section 3 leads to practical instances of groups that separate DDH and CDH (assuming that DL is hard in our groups).

5. Conclusion

The above construction of reasonably looking cryptographic groups where DDH is easy, while CDH is known to be as hard as DL has both positive and negative consequences in cryptography. On the negative side, we have shown that real groups may behave quite differently than generic groups. As a consequence, when proposing new systems based on "exotic" groups, some care needs to be taken. On the positive side, we can exhibit groups with a gap between DDH and CDH and thus use pairing-based cryptosystems with a good level of confidence. At present, the abundance of new systems relying on the properties pairings and even on the easiness of DDH (e.g., an application to verifiable random functions was proposed in [8]) is clearly in favor of the positive side.

References

[1] A.O.L. Atkin and F. Morain. Elliptic curves and primality proving. *Mathematics of Computation*, 61:29–68, 1993.

[2] C. Batut, K. Belabas, D. Bernardi, H. Cohen, and M. Olivier. Users's guide to PARI-GP. http://www.parigp-home.de.

[3] D. Boneh. The decision Diffie–Hellman problem. In *Algorithmic Number Theory*, volume 1423 of Lecture Notes in Computer Science, pages 48–63. Springer-Verlag, Berlin, 1998.

[4] 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.

[5] A. Joux. A one round protocol for tripartite Diffie–Hellman. In W. Bosma, editor, *Proceedings of the ANTS-IV Conference*, volume 1838 of Lecture Notes in Computer Science, pages 385–394. Springer-Verlag, Berlin, 2000.

[6] A. Joux. The Weil and Tate pairings as building blocks for public key cryptosystems. In *Algorithmic Number Theory*, volume 2369 of Lecture Notes in Computer Science, pages 20–32. Springer-Verlag, Berlin, 2002.

[7] 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 Computer Science, pages 250–263. Springer-Verlag, Berlin, 1994.

[8] A. Lysyanskaya. Unique signatures and verifiable random functions. In *Advances in Cryptology - CRYPTO* 2002, volume 2442 of Lecture Notes in Computer Science, pages 597–612. Springer-Verlag, Berlin, 2002.

[9] U. Maurer. Towards the equivalence of breaking the Diffie–Hellman protocol and computing discrete logarithms. In *Advances in Cryptology - CRYPTO* '94, volume 839 of Lecture Notes in Computer Science, pages 271–281. Springer-Verlag, Berlin, 1994.

[10] U. Maurer and S. Wolf. Diffie–Hellman oracles. In N. Koblitz, editor, *Advances in Cryptology - Crypto* '96, Volume 1109 of Lecture Notes in Computer Science, pages 268–282. Springer-Verlag, Berlin, 1996.

[11] U. Maurer and S. Wolf. The relationship between breaking the Diffie–Hellman protocol and computing discrete logarithms. *SIAM Journal on Computing*, 28(5):1689–1721, 1999.

[12] 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.

[13] V. Miller. Short programs for functions on curves. Unpublished manuscript, 1986.
[14] T. Okamoto and D. Pointcheval. The gap problems: a new class of problems for the security of cryptographic primitives. In *Public Key Cryptography*, *PKC* 2001, volume 1992 of Lecture Notes in Computer Science, pages 104–118. Springer-Verlag, Berlin, 2001.
[15] V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *Advances in Cryptology - Eurocrypt '*97, volume 1233 of Lecture Notes in Computer Science, pages 256–266. Springer-Verlag, Berlin, 1997.
[16] E. Verheul. XTR is more secure than supersingular elliptic curve crypto systems. Preprint.