Gen. Math. Notes, Vol. 15, No. 1, March, 2013, pp.84-91
ISSN 2219-7184; Copyright ©ICSRS Publication, 2013
www.i-csrs.org
Available free online at http://www.geman.in

Elliptic Curve Discrete Logarithm

Benamara Oualid

University of Sciences and Technology Houari Boumdienne,
Faculty of Mathematics, Departement of Algebra,
Algiers, Algeria
E-mail: benamara.oualid@gmail.com

(Received: 17-12-12 / Accepted: 18-2-13)

Abstract

In this paper we present steps to follow when designing elliptic curve discrete logarithm ECDL. Principle results from general theory are given. In order to avoid some serious attacks against ECDL, special attention are suggested on the parameters. Finally, we describe general methods to follow when designing ECDL.

Keywords: *Cryptography; Discrete logarithm problem; Elliptic curves.*

1 Introduction

In public-key cryptography, each participant possesses two keys: a public key and a private key. These are linked in a unique manner by a one way function. We will define this notion in the following section. These are functions that are easy to calculate but difficult to inverse. The security of a cryptographic protocol is related to the difficulty of inverting. So, we will have to estimate this difficulty in term of the time needed for computers to inverse this function. Complexity theory deal with this notion of complexity of algorithms. Searching for such function is an important task in cryptography and they are used as cryptographic primitive in protocols. RSA is one of the well known protocol that rely on the difficulty of factoring large numbers. There exist sub exponential time algorithm that can factor such numbers. For high level of security, RSA is not convenient and we will search for other one way function for which inverting is match harder. Discrete logarithm system over finite field is another cryptographic primitive used in security protocols. But they present

weakness and special attention is required in designing such system. On elliptic curves, we know construct discrete logarithm on the group of the point of the curve. In this paper we explain how to construct such system leading to efficient one way function and how to avoid attacks and weakness. This work is an abstract of the results obtained in [21].

2 Definitions

2.1 One Way Function

Let S be the set of binary strings and f be a function from S to S. We say that f is a one-way function if

- the function f is one-on-one and for all $x \in S$, $f(x)$ is at most polynomially longer or shorter than x
- for all $x \in S$, $f(x)$ can be computed in polynomial time
- there is no polynomial time algorithm which for all $y \in S$ returns either no if $y \notin S$ or $x \in S$ such that $y = f(x)$.

Given a one way function one can choose as the private key an $a \in S$ and obtains the public key $f(a)$. This value can be published since it is computationally unfeasible to defer a from it.Complexity theory gives a mathematical measure to define what is meant by (computationally unfeasible). We treat this notion in a next section.For some applications it will be necessary to have a special class of computational one way functions that can be inverted if one possesses additional information. These functions are called trapdoor one way functions. We will explain in this paper how to construct such function in a manner that no algorithm can invert it in less than exponential time.

2.2 Generic Discrete Logarithm Systems

Let $(G, \oplus)$ be a cyclic group of prime order l and let P be a generator of G. Let the map

$$\phi : \mathbb{Z} \to G$$

$$n \to [n]P = \underbrace{P \oplus P \oplus ... \oplus P}_{ńtimes}$$

The problem of computing the inverse of this map is called the discrete logarithm problem (DLP) to the base of P. If $[n]P = Q$ so we note $log_P Q = n$.

In this paper we consider the group of point of elliptic curve over finite field and the related DLP.

2.3 Complexity

Determining the execution time of an algorithm consist of counting the number of basic operations needed for its execution. We attach a certain function f to an algorithm that bounds the time used for the computation given the length of the input called complexity parameter. This suggest the fallowing definition: Let f and g be two real functions of s variables. The function $g(N_1, ..., N_s)$ is of order $f(N_1, ..., N_s)$ denoted by $Of(N_1, ..., N_s)$ if for a constant c one has:

$$|g(N_1, ..., N_s)| \leq cf(N_1, ..., N_s)$$

with $N_i \geqslant N$ for some constant N.

We will estimate the complexity of the DLP on elliptic curve group and so evaluate the hardness of such algorithms.

In this paper, we are searching for efficient one way functions 2.1. We construct a DLP 2.2 on elliptic curve group 3 in such a manner that no known polynomial time algorithm 2.3 can compute it.

3 Elliptic Curves

An elliptic curve over a field K is a non singular protective algebraic curve over K with genus 1 together with a given point, I. Elliptic curve is defined as all the points on the curve (given by the Weirstrass equation)

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

with $a_1, a_2, a_3, a_4, a_5, a_6 \in K$ together with I, the point at infinity. We also require that the curve is non singular. Note that if the field we are working over has characteristic not equal to two or three then we can transform this equation to:

$$y^2 = x^3 + ax + a_4x + b$$

We define the the addition on the points of an elliptic curve E of the form

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

with the fallowing rules:

3.1 Definition

If $P_0 = I$ then $-I = I$. Otherwise let $P_0 = (x_0, y_0) \in E$. Then

$$-P_0 = (x_0, -y_0 - a_1x_0 - a_3)$$

If one of P_1 or P_2 equals I then $P + I = I + P = P$, otherwise let

$$P_1 + P_2 = P_3$$

with $P_i \in E$ If $x_1 = x_2$ and $y_1 + y_2 + a_1 x_2 + a + 3 = 0$ then $P_1 + P_2 = I$. Otherwise define

$$\alpha = \frac{y_2 - y_2}{x_2 - x_1}$$

,

$$\beta = \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}$$

if $x_1 \neq x_2$;

$$\alpha = \frac{3x_1^2 + 2a_2 x_1 + a_4 - a_1 y_1}{2y_1 + a_1 x_1 + a_3}$$

and

$$\beta = \frac{-x_1^3 + a_4 x_1 + 2a_6 - a_3 y_1}{2y_1 + a_1 x_1 + a_3}$$

if $x_1 = x_2$. Then $P_3 = (x_3, y_3)$ where

$$x_3 = \alpha^2 + a_1 \alpha - a_2 - x_1 - x_2$$

$$y_3 = -(\alpha + a_1) x_3 - \beta - a_3$$

3.2 Theorem

The addition low defined bellow has the following proprieties:

1. $P + I = P$ for all $P \in E$,
2. $P + Q = Q + P$ for all $P, Q \in E$ (the law is commutative),
3. Let $P \in E$. There is a point of E, denoted $-P$, so that

   $$P + (-P) = I$$

   (the additive inverse element),
4. the addition is associative

We obtain an associative group $(E, +)$

In designing such groups and for security reason, we want the E is of prime order or a multiplicative sufficiently large prime. The method consists of estimating a family of valid curves and use efficient algorithms (polynomial time) to compute the order of the group obtained.

4 Choice of Parameters

One has to be careful with the choice of the pair (C, F_q) if one wants to have instances in which the complexity of algorithms computing the discrete logarithm is $O(l^{1/2})$. To avoid Index Calculus Attack [4], one has to chose a curve of genus $g \leq 4$. For $g \leq 3$ we have the following results:

- l does not divide q to avoid transferring of DLP to a group in witch there is efficient attacks [1] [2] [3].
- The extension field of F_q is of sufficiently large degree k. For small k, fast algorithms for DLP can be found in [5] [6] [7].
- Let $q = p^d$,to avoid weaker DLP [8], one has to avoid the case where $\frac{d}{d_0}$ is small where d_0 is a divisor of d and the case where d is prime with the existence of a small t for which $2^t \equiv 1 mod(d)$. Also d must not be a Mersenne or a Fermat prime number

5 Point Counting

We are searching for groups of prime order or of order of large prime divisor. This is why the speed of point counting is important in designing elliptic curve DLP. The computation of the order of a group for curves C of genus g defined over fields F_q can be performed efficiently by not too complicated algorithms if

- The curve C is already defined over a small subfield F_{q_0} of F_q, [9] [10] [11] or
- The genus g is equal to 1, [12] [13] [14] or
- The characteristic of F_q is small, [15] or
- The genus of C is 1 or 2, the field F_q is a prime field, and the curve $C$is the reduction modulo q of a curve C' with complex multiplication over a given order End_C in a CM-field. Here is the algorithm [16]

6 Primality

In the applications we envision we must be sure that a given integer p is prime. The most obvious way is to try for all integers $n \leq \sqrt{N}$ whenever n devise N. This is not realizable in practice.In fact, it is too time-consuming to prove primality, or for that matter compositeness, of a given integer by failing or succeeding to find a proper factor of it. The best primality test algorithms

will only prove p to be prime and will not output any divisor in case it is not. Most algorithms will be probabilistic in nature in the sense that one output is always true while the other is only true with a certain probability. Iterating this algorithm allows us to enlarge the probability that the answer that was given only with a certain probability actually holds true. These techniques offer quite good performance. To prove primality using probabilistic algorithms one usually starts with some iterations of an algorithm whose output non prime is always correct while the output prime is true only with a certain probability (Trial division, Fermat tests, Rabin Miller test, Lucas pseudoprime tests, BPSW tests). After passing some rounds one uses an algorithm that is correct when it outputs prime (Atkin Morain ECPP test [18], APRCL Jacobi sum test [17]). The reason for this order is that usually the algorithms of the first type have a shorter running time and allow us to detect composite integers very efficiently. Factoring algorithms (Pollard s rho method, Pollard s p-1 method, Factoring with elliptic curves, Fermat Morrison Brillhart approach [19] [20]) on the other hand, are usually much slower.

7 Conclusion

In this paper we present a general method to use when designing elliptic curve diffie-Hellmann logarithm. We review the important results obtained in this field and present them in a methodical approach. Even if the security of such system repose on the fact that there is no polynomial time algorithm witch solve DLP, it still there is no proof of the non existence of such algorithm.

References

[1] T. Satoh and K. Araki, Fermat quotients and the polynomail time discrete log algorithms for anomalous elliptic curves, *Comm. Math. Univ. Sancti Pauli*, 47(1998), 92-91.

[2] I. Samaev, Evaluation of discrete logarithms in a group pf p-torsion points of an elliptic curve in charachteristic p, *Math. Compu.*, 67(1998), 353-356.

[3] N.P. Smart, The discrete logarithm problem on elliptic curve of trace one, *J. Cryptology*, 12(1999), 193-196.

[4] L.M. Adleman, J. Demarrais and M.D. Huang, A subexponential algorithm for discret logarithms over hyperelliptic curves of large genus over GF(q), *Theoret. Comput. Sci.*, 226(1999), 512-516.

[5] S.D. Galbraith, K. Harrison and D. Soldera, Implementing the tate pairing, *Algorithmic Number Theory Symposium*, 2369(2002), 324-337.

[6] P.S.L.M. Barreto, H.Y. Kim, B. Lynn and M. Scott, Efficient algorithms for pairing-based cryptosystems, *Advenced in Cryptology*, 2442(354) (2002), 380-589.

[7] P.S.L.M. Barreto, H.Y. Kim, B. Lynn and M. Scott, On the selection of pairing-freindly groups, *Lecture Notes in Comput. Sci.*, 3006(2004), 17-25.

[8] P. Gaudrey, F. Hess and N.P. Smart, Constructive and destructive facets of Weil descent on elliptic curves, *J. Cryptology*, 15(2002), 531-534.

[9] V. Shoup, Lower bounds for discrete logarithms and related problems, *Advences in Crryptology*, 1233(1997), 256-266.

[10] S. Pohlig and M. Hellmann, An improvement algorithm for computing logarithms over GF(p) and its cryptographic signifiance, *IEEE Tran. Inform. Theory*, 24(1978), 106-110.

[11] D. Shanks, Class number: A theory of factorization and generalization, *Proc. Symp. Math.*, 20(1971), 415-440.

[12] A.O.L. Atkin, The number of points on an elliptic curve modulo a prime, *E-mail on the Number Theory Mailing List*, (1988).

[13] A.O.L. Atkin, The number of points on an elliptic curve modulo a prime, *E-mail on the Number Theory Mailing List*, (1991), 414-415.

[14] N.D. Elkies, Explicite isogenies, *Draft*, (1991).

[15] J.F. Mestre, Lettre adressée à Gaudry et Harley, www.math.jussieu.fr/ mestre, (2000).

[16] A.M. Spallek, Kurven vom Geschlecht 2 und ihre Anwendung in Public-key Kryptosystemen, Universitat Gesamthchhschule Essen, *PhD. Thesis*, (1994).

[17] H. Cohen and A.K. Lenstra, Implementation of a new primality test, *Math. Comp.*, 48(48) (1987), 103-121.

[18] J. Franke, T. Kleinjung, F. Morain and T. Wirth, Proving the primality of very large numbers with fast ECPP, *Algorithmic Number Theory Symposium*, 3076(2004), 194-207.

[19] C. Pomerance, Analysis and comparison of some integer factoring algorithms, *Math. Center Tracts*, 154(1983), 89-139.

[20] C. Pomerance, The quadratic sieve factoring algorithm, *Advances in Cryptology*, 209(1985), 169-182.

[21] R. Avanzi, C. Doche, T. Lange, K. Nguyen and F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, Chapman, (2006).