ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 4, April 2015

Attacks on Elliptic Curve Cryptography Discrete Logarithm Problem (EC-DLP)

Mrs.Santoshi Pote[1], Mrs. Jayashree Katti[2]

ENC, Usha Mittal Institute of Technology, Mumbai, India[1]

Information Technology, Chinchwad College of Engineering, Pune, India[2]

Abstract: This paper provides an overview of elliptic curves and their use in cryptography. The purpose of this paper is an in-depth examination of the Elliptic Curve Discrete Logarithm (ECDLP) including techniques in attacking cryptosystems dependent on the ECDLP. The paper includes properties of elliptic curve and methods for various attacks.

Keywords: ECC, Discrete log, SAGE.

I. INTRODUCTION

Introduced to cryptography in 1985, elliptic curves are quickly being adapted for cryptographic purposes. Elliptic curve cryptography is quickly becoming a leader in the industry, and is challenging other cryptosystems such as RSA and DSA to become the industrial standard; this is due to an increase in speed during implementation, the use of less memory, and smaller key sizes. Another advantage of such a cryptosystem lies in the difficulty of solving the Elliptic Curve Discrete Log Problem (ECDLP). If an elliptic curve is chosen with some care, the ECDLP is believed to be infeasible, even with today's computational power. Using elliptic curves presents a great advantage in a few areas. For instance, compared to RSA cryptosystems, elliptic curve based systems require less memory; for example, a key size of 4096 bits for RSA gives the same level of security as 313 bits in an elliptic curve system . Also, using a PalmPilot, generating a 512-bit RSA key takes around 3.4 minutes, while generating an equivalent 163-bit ECC-DSA key takes 0.597 seconds [87, 159]. Immediately we begin to see the advantages of using elliptic curves, especially on small hand held devices with little computing power. It is clear that this now gives us the advantage of setting up schemes that require smaller chip sizes, use less memory, require less resources to run, require less power consumption, etc; and can be placed in small electronic devices, such as smart cards and cell phones. Many elliptic curve cryptosystems take advantage of what is known as the ECDLP. Analogous to the Discrete Logarithm Problem (DLP) over a finite field F×p , the ECDLP has the following problem: given two points P and Q on an elliptic curve E defined over a field Fq, where q is prime or a prime power, if P = [m]Q for some m ϵ Z, determine m. Schemes and protocols such as the Deffie-Hellman key exchange, Massey-Omura encryption, El-Gamal public key encryption and El-Gamal digital signatures and even the Elliptic Curve Digital Signature Algorithm(ECDSA), all use the fact that attempting to solve the ECDLP is a difficult, if not intractable, problem.
As mentioned although the ECDLP is thought to be an intractable problem, it has not stopped people attempting to attack such a cryptosystem. Various attacks have been devised, tested and analyzed by many leading mathematicians over the years, in attempts to find weaknesses in elliptic curve cryptosystems. Some have been partially successful, while others have not. The purpose of this paper is to provide a detailed examination of the leading attacks against the ECDLP, and to use the knowledge of these attacks in an attempt to generate cryptographically strong elliptic curves.

II. ELLIPTIC CURVE CRYPTOGRAPHY

An elliptic curve over the field is the set of all solutions $(x,y) \epsilon F^2$ to the equation

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad (1)$$

where $a_1,a_2,a_3,a_4,a_6 \epsilon F$, with an extra symbolic "point at infinity" marked ∞. Using geometric properties of the "graph" of solutions, one can define an abelian group structure for the curve, with the point at infinity serving as the unit member of the group. Cryptographic systems usually use elliptic curves over prime fields F_p(for some large prime number) or binary fields (F_2^m for some integer m), since field arithmetic in these particular fields can be implemented very efficiently. In this paper we have focused on prime field curves. In curves over a prime field, the *Weierstrass equation* above can be expressed, using a variable change, as a much simpler equation of the form

$$y^2 = x^3 + ax + b . \qquad (2)$$

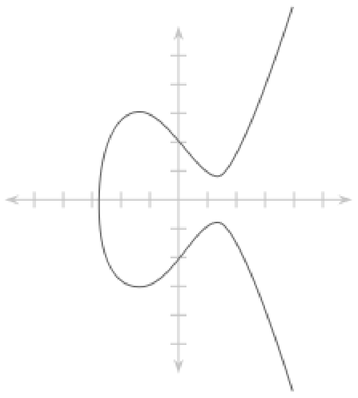

Fig. 1 Elliptic curves

An elliptic curve E gives us a set of points, we define a binary operation + on E.

III. SELECTION OF ELLIPTIC CURVES

NIST Curves

The U.S National Institute of Standards and Technology (NIST) has endorsed ECC in its set of recommended algorithms, specifically Elliptic Curve Diffie-Hellman (ECDH) for key exchange and elliptic curve digital signature Algorithm (ECDSA) for digital signature. Nist recommended fifteen elliptic curves. Specifically, FIPS 186-3 has ten recommended finite fields. The NIST recommendation contains a total of five prime curves and ten binary curves. The curves were ostensibly chosen for optimal security and implementation efficiency.

IV. SPECIAL FIELDS

The following are some classes of finite fields that have been proposed for commercial use in elliptic curve cryptography due to their potential performance advantages.

1. Prime fields: These are finite fields of prime order. Prime fields have the advantage that their arithmetic can be efficiently implemented in software on machines which have a 32*32 bit multiply instruction.
2. NIST prime fields: These are prime fields F*p* where the prime *p* is a Mersenne prime or a Mersenne-like prime, e.g.. In particular, the finite fields F*p* have been standardized in NIST's FIPS 186-2 . Such prime fields are advantageous over random prime fields because the modular reduction operation can be performed very efficiently .
3. Binary fields: These are finite fields of order F(2^m). Binary fields have the advantage that their arithmetic can be efficiently implemented in hardware.
4. Composite binary fields: These are binary fields of order where m is a composite number. Because composite binary fields have non-trivial subfields, field arithmetic can be speed up by using lookup tables for performing subfield arithmetic.
5. Optimal extension fields: These are finite fields of order where *p* is a 32-bit or 64-bit prime and m is a small integer. Optimal extension fields were introduced by Bailey and Paar because the arithmetic in such fields is particularly efficient on 32-bit and 64-bit platforms.

V. ELLIPTIC CURVE ARITHMETIC

Now, we consider the following example. Let the two elliptic curves be $y^2 = x^3 - 4x$ and $y^2 = x^3 - 1$. The graphs are as follows:

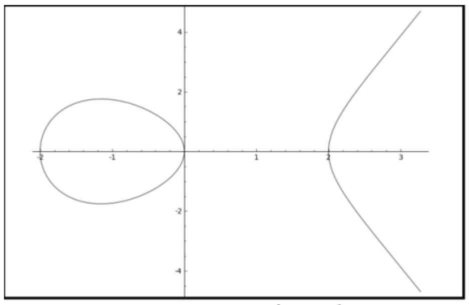

Fig. 2 Example of $y^2 = x^3 - 4x$

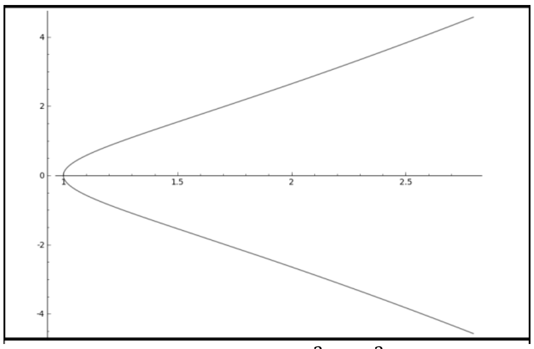

Fig. 3 Example of $y^2 = x^3 - 1$

GROUP LAW

Let E be an elliptic curve defined over the field of integers K. There is a chord-and-tangent rule for performing operations on the points of E(K) to give the third point. The operations on the group are as follows[1].

Point addition

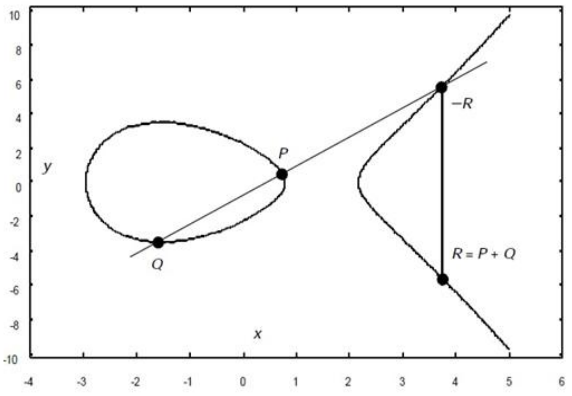

Fig.4 Point Addition

The operation is the addition of two points on the curve to obtain a third point on the curve. Let P(x_1,y_1), Q(x_2,y_2) € E_K(a, b) where P ≠ Q. Then P + Q = (x_3,y_3) where,

$$x_3 = \lambda^2 - x_1 - x_2 \quad (3)$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \quad (4)$$

Whereas λ is the slope of the line joining points P and Q,

$$\lambda = \frac{y_1 - y_2}{x_1 - x_2} \quad P \neq Q \quad (5)$$

Suppose P(x_1,y_1) € E_K(a, b) then,

$$P + (-P) = O_\infty \quad (6)$$

Where (–P) = (x_1, -y_1), and this property is called as point at infinity.

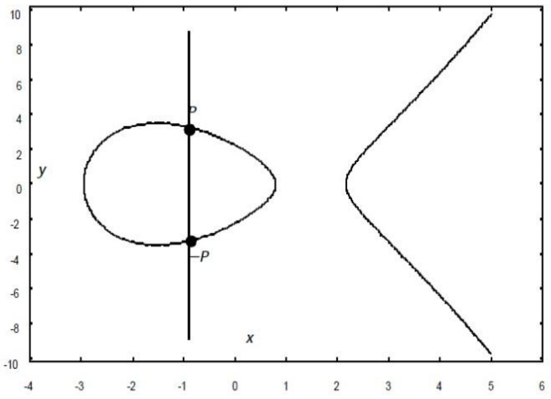

Fig. 5 Point at Infinity

Point Doubling

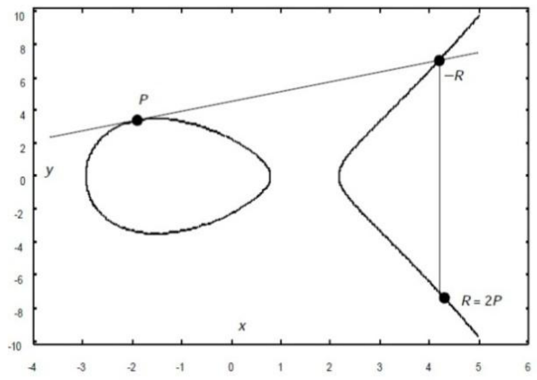

Fig. 5 Point Doubling

Let P=(x_1, y_1) € E_K(a, b) where P≠ -P then, 2P= (x_3, y_3) where,

$$x_3 = \lambda^2 - x_1 - x_2 \quad (7)$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \quad (8)$$

And λ is the slope of the line joining points P and (-P),

$$\lambda = \frac{3x_1^2+a}{2y_1} \quad (9)$$

Point Multiplication

In an arithmetic, multiplying a number by a constant *k* means adding a number to itself *k* times. The situation here is same. Multiplying a point P on an elliptic curve by a constant k means adding the point P to itself *k* times.

VI. DISCRETE LOGARITHM PROBLEM

One of the first published cryptosystems whose security depends on discrete logarithms being difficult to compute appears to be an authentication scheme. In many computer systems, users' passwords are stored in a special file, which has the disadvantage that anyone who gets access to that file is able to freely impersonate any legitimate user. Therefore that file has to be specially protected by the operating system. It has been known for a long time that one can eliminate the need for any secrecy by eliminating the storage of passwords themselves. Instead, one utilizes a function *f* that is hard to invert (i.e., such that given a *y* in the range of *f*, it is hard to find an *x* in the domain of *f* such that *f* (*x*) = *y*) and creates a file containing pairs (*i*, *f* (*pi*)), where *i* denotes a user's login name and *pi* the password of that user. This file can then be made public. The security of this scheme clearly depends on the function *f* being hard to invert.

One early candidate for such a function was discrete exponentiation; a field *GF*(*q*) and a primitive element *g* ϵ *GF*(*q*) are chosen (and made public), and for *x* an integer, one defines *f* (*x*) = *gx* .

The security of many cryptosystems depends on the intractability of the discrete logarithm problem. For instance one of the more famous public key cryptosystems, El-Gamal encryption, relies heavily on the intractability of this problem. The following is referred to as the DLP or even sometimes as the Generalized DLP.

The security of ECC depends on the difficulty of Elliptic Curve Discrete Logarithm Problem. Let P and Q be two points on an elliptic curve such that kP = Q, where k is a scalar. Given P and Q, it is computationally infeasible to obtain k, if k is sufficiently large. k is the discrete logarithm of Q to the base P. Hence the main operation involved in ECC is point multiplication. i.e. multiplication of a scalar k with any point P on the curve to obtain another point Q on the curve.

Attacks on the DLP can be divided into three main categories

1. Algorithms that work in arbitrary groups, such as the exhaustive search and the Baby-Step Giant-Step algorithm,
2. Algorithms that work in arbitrary groups with special conditions present in the group, like Pollard's rho Method, and
3. Algorithms that work only in specific groups, such as the Index Calculus.

VII. ATTACKING THE DISCRETE LOGARITHM PROBLEM

1. Index Calculus Algorithm

It is the strongest family of algorithms for finding the discrete logarithms in a cyclic group. The theme is if we can find the discrete logarithms of some small and independent elements, then we should be able to determine logarithms of almost any element in the group, as most elements we can express in terms of the small independent elements whose logs are known[1].

Let p be a prime and let g be primitive root mod p , which means that g is a generator for the cyclic group F_P^X. In other words, every h≠ $0(\ mod\ p)$ can be written in the form h≡ g^k for some integer k that is uniquely determined mod P-1. Let K = L(h) denote the discrete logarithm of h with respect to g and p ,so [1]

$$g^{L(h)} \equiv h \ (\text{mod p}) \quad (10)$$

Suppose we have h_1 and h_2. Then

$g^{L(h1h2)}$ ≡ h_1 h_2 (mod p)≡ $g^{L(h_1+h_2)}$ (mod P) Which implies that

$$L(h_1 h_2) \equiv L(h_1) + L(h_2) \text{ mod (P-1)} \quad (11)$$

Therefore L changes multiplication into addition , just like the classical logarithm.

The expected running time of the indux calculus is approximately a constant times exp $\sqrt{2lnplnlnp}$, which means that it is a subexponential algorithm.

2. Baby step Giant step Algorithm

In group theory, a branch of mathematics, the baby-step giant-step is a meet-in-the-middle algorithm computing the discrete logarithm. The discrete log problem is of fundamental importance to the area of public key cryptography. Many of the most commonly used cryptography systems are based on the assumption that the discrete log is extremely difficult to compute; the more difficult it is, the more security it provides a data transfer. One way to increase the difficulty of the discrete log problem is to base the cryptosystem on a larger group. This method was developed by D. Shanks requires √ N steps and √ N storage. The algorithm is based on a space-time trade off. It is a fairly simple modification of trial multiplication, the naive method of finding discrete logarithms. This attack uses a combination of computational power and memory storage to solve the DLP. Let G be a cyclic group with generator α. Suppose that α has order n and set m = [√ n].

Observe that if $\beta = \alpha^x$, then using the euclidean algorithm we can write x as follows: x = im + j, where 0 ≤ i, j < m. Thus we have that β = αx = αim+j = αim αj , which implies that β(α−m)i = αj . To compute the discrete logarithm, we begin by computing and storing the values (j, αj) for 0 ≤ j ≤ m. We then compute β(α−m) and raise that to the exponent i for 0 ≤ i ≤ m − 1 and check these values against the stored values of αj to find a match. When a match is found, we have solved the DLP and we have x = im + j as required.

The drawbacks of this algorithm lie in the computation and formulation of the table of pairs (j, αj). At each stage we are required to compute a power of α and look in the table to see if it returns a match. If this is successful then the DLP has been solved. Unfortunately, one has to store around O(√ n) group elements, perform around O(√ n) multiplications to find the correct power of α, and in turn perform O(√ n) table look-ups [62]. As a consequence this algorithm has an expected running time of O(√ n), which makes it impractical for cryptographic purposes.
Procedure:-
1. Fix an integer m> √ N and compute mP
2. Make and store a list of iP for 0≤i<m
3. Compute the points Q-jmP for j=0,1……..m-1 until one matches an element from the stored list
4. If iP=Q-jmp, we have Q=kP with k=i+jm (mod N)

3. Pollards Rho Algorithm

This algorithm has a similar running time to the Baby-Step Giant-Step method above yet requires less memory, an immediate advantage. Let G be a cyclic group of order n, where n is prime. G is then partitioned into three subsets of roughly equal size, call these sets S1, S2 and S3. We then de ne a sequence of group elements, xi, as follows: x_0 = 1and

$$x_{i+1} = f(x_i) = \begin{cases} x_i\beta & if\ x_i \in S_1 \\ x_i^2 & if\ x_i \in S_2 \\ x_i\alpha & if\ x_i \in S_3 \end{cases}$$

This in turn defines two sequences of integers ai and bi. The sequences ai and bi are defined as follows: set $a_0 = 0 = b_0$andfori > 0;

$$a_{i+1} = \begin{cases} a_i & if\ x_i \in S_1 \\ 2\alpha_i\ mod\ n & if\ x_i \in S_2 \\ a_i + 1\ mod\ n & if\ x_i \in S_3 \end{cases}$$

$$b_{i+1} = \begin{cases} b_i + 1\ mod\ n & if\ x_i \in S_1 \\ 2b_i\ mod\ n & if\ x_i \in S_2 \\ b_i & if\ x_i \in S_3 \end{cases}$$

We then begin with a pair (x1, x2) and iteratively compute pairs (X_i, X_{2i}) until we nd a pair of group elements such that X_i = X_{2i} for some i. When such a pair is found we then have the following relation:

$$\alpha^{a_i}\beta^{b_i} = \alpha^{a_{2i}}\beta^{b_{2i}}$$

$$\beta^{b_i - b_{2i}} = \alpha^{a_i - a_{2i}}$$

$$(b_i - b_{2i})\ log_\alpha\beta = (a_i - a_{2i})\ mod\ n$$

Thus obtain a solution for

$$x = log_\alpha\beta = (b_i - b_{2i})^{-1}(a_i - a_{2i})\ mod\ n$$

4. Pohlig Hellman Algorithm

The algorithm was first discovered by Roland Silver, but first published by Stephen Pohlig and Martin Hellman (Independent of Silver). Thus it is sometimes called as Silver – Pohlig Hellman Algorithm. It is a special purpose algorithm used for computing discrete logarithms in a multiplicative group whose order is a smooth integer.

The Pohlig Hellman method:
P, Q are elements in a group G and we want to find an integer k with Q=kP. We also know the order N of P and we know the prime factorization

$$N = \prod_i q_i^{e^i}$$

The idea of Pohlig Hellman is to find k (mod $q_i^{e^i}$) for each i , then use the Chinese Remainder theorem to combine these and obtain k (mod N).

Let q be a prime, and let q^e be the exact power of q dividing N. Write k in its base q expansion as

$$k = k_0 + k_1 q + k_2 q^2 + \ldots..$$

with $0 \le k_i < q$. We will evaluate k (mod q^e) by successively determining k_0 , k_1 , k_2 , …. , k_{e-1} .

The procedure is as follows[1]:
1. Compute T = {j ($\frac{N}{q}$.P) 0 ≤ j ≤ q-1
2. Compute $\frac{N}{q}$ Q . This will be an element of $k_0\left(\frac{N}{q}.\mathrm{P}\right)$ of T.
3. If e = 1, stop. Otherwise, continue.
4. Let $Q_1 = Q - k_0P$.
5. Compute $\frac{N}{q^2}$Q . This will be an element of $k_1\left(\frac{N}{q}.\mathrm{P}\right)$ of T
6. if e=2 stop otherwise continue.Suppose we have computed k_0, k_1, …. , k_{r-1}, and Q_1, …. , Q_{r-1}.
7. Let $Q_r = Q_{r-1} - k_{r-1}q^{r-1}P$.
8. Determine k_r such that $\frac{N}{q^{r+1}}.Q_r = K_r\ (\frac{N}{q}\ \mathrm{P})$
9. If r = e -1 , stop. Otherwise, return to step (7).

Then K$\equiv k_0 + k_{1q} \ldots . + k^{e-1}q^{e-1}$ (mod q^e) Therefore we find k_1. similarly, the method produces k_2 , k_3,….. We have to stop after r = e-1

VIII. CONCLUSION

The cryptographic strength of elliptic curve encryption lies in the difficulty for a cryptanalyst to determine the secret random number *k* from *kP* and *P* itself. The following algorithm implemented in System for Algebra and Geometry Experimentation (SAGE) software and verify the expected time of the attacks on DLP.

Attacks	Expected Time
Baby step Giant step method	$\sqrt[0]{n}$
Pollard Method	$\sqrt[0]{\frac{\pi n}{2}}$
Index calculus Method	$L_q\,[\frac{1}{2}\,, C]$
Pohlig Hellman Method	$\sqrt[0]{n}$

REFERENCES

[1]. Lawrence C. Washington university of Maryland "Elliptic Curves Number Theory and Cryptography.
[2]. Jason S. Howell,'' The indux calculus algorithm for discrete logarithm,'' March 31,1998.
[3]. Joppe W.BOS, Alina Dudeanu, Dimitar Jetchev, "Collision bounds for the additive pollard rho algorithm for solving discrete logarithms."
[4]. J.H. silverman. The Arithmetic of elliptic curves, volume 106 of graduate Texts in mathematics. Springer-Verlag,New Yoark,1986.