Point Generation And Base Point Selection In ECC: An Overview

Moumita Roy[1], Nabamita Deb[2], Amar Jyoti Kumar [3]

M Tech. Student, Information Technology, GUIST,Guwahati, India [1]
Asst. Professor, Information Technology, GUIST, Guwahati, India[2]
M Tech. Student, Information Technology, GUIST, Guwahati, India [3]

Abstract: Elliptic curve cryptography (ECC) is an approach to public-key cryptography which is based on the algebraic structure of elliptic curves over finite fields. ECC, a public-key encryption technique, can be used to create faster, smaller, and more efficient cryptographic keys. Instead of the traditional method of generation, ECC generates keys through the properties of the elliptic curve equation. The technology of ECC can be used in conjunction with most public key encryption methods, viz. RSA and Diffie-Hellman. According to researchers, ECC can yield a level of security with a 164-bit key that other systems require a 1,024-bit key to achieve. As ECC can be used to establish equivalent security with lower computing power and battery resource usage, it is being widely used for mobile applications.

Keywords: ECC, Base point, RSA, Diffie-Hellman

I. Introduction

Elliptic Curve Cryptography (ECC) is a public key cryptographic technique. In public key cryptography, each user or the device taking part in the communication generally require a pair of keys, a public key and a private key, and a set of operations which remain associated with the keys to do cryptographic operations. The public key is being distributed to all the users taking part in the communication but the private key is only being known by the particular user. Some public key algorithm may require a set of predefined constants to be known by all the devices which will take part in the communication. Domain parameter of ECC is one such example. Unlike private key cryptography, public key cryptography does not require any shared secret between the communicating parties but it is much slower than private key cryptosystem.

The mathematical operation on ECC is defined over the elliptic curve equation:

$$y2 = x3 + ax + b, \qquad where\ 4a3 + 27b2 \neq 0$$

Different elliptic curves can be generated by varying the curve parameter values, that is, value of ‘a’ and ‘b’. All the (x,y) points satisfying the above equation plus a point at infinity lies on the elliptic curve. The user’s public key is a point in the curve and its private key is a randomly selected number. The base point or the generator point ‘G’ of the curve is multiplied with the private key to obtain the public key. The generator point G, the curve parameters ‘a’ and ‘b’, together with few more constants constitutes the domain parameter of ECC. The requirement of small key size in ECC is one of its main advantage. For example, a 160 bit key in ECC is considered to be as secured as 1024 bit key in RSA.

II. DETERMINATION OF POINTS IN AN ELLIPTIC CURVE OVER A FINITE FIELD

Generation of points in the elliptic curve is the basic step in elliptic curve cryptography but it is normally not shown in papers how to generate those points. So, in this example, we will determine all points on the curve over the finite field having p=17, a=1 and b=0. Therefore the equation becomes:

$$E = \{(x, y); y2 = x3 + x\}$$

Taking mod on both sides we have,

$$E = \{(x, y); y2modp = (x3 + x)mod17\}$$

For doing that, we first compute the square table over F, which tells us which points in F can have a square root.
>> For y=[0:16], the answers are-

TABLE I

0 0	4 16	8 13	12 8	16 1
1 1	5 8	9 13	13 16	
2 4	6 2	10 15	14 9	
3 9	7 15	11 2	15 4	

This generates the following square table of mod p(p=17 here).
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
0 1 4 9 16 8 2 15 13 13 15 2 8 16 9 4 1

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

Clearly, (0, 0) € E. Then, we compute x= 1,2,...., 16 to solve the equation

$$y2mod17 = (x3 + x)mod17$$

For x=1, y2=1+1 and so the square root table gives y= ±6. Hence (1, ±6) € E
For x=2, we have y2=8+2=10. The square table says that there is no solution, and so we move on to the case for x=3.
The following table computes all the needed information for x.
>> For x=[0:16]

TABLE II

0 0	4 0	8 10	12 16	16 5
1 2	5 11	9 7	13 0	
2 10	6 1	10 7	14 0	
3 13	7 10	11 16	15 7	

In this way, we have the required points on the elliptic curve as

$$E = \{(0,0), (1, \pm 6), (3, \pm 8), (4,0), (6, \pm 1), (11, \pm 4), (13,0), (14, \pm 2), (16, \pm 7)\}$$ [4].

III. BASE POINT SELECTION

Base point or generator point selection in ECC is the prime step for its security. The efficiency of the choice of base point is needed for reducing the time complexity of the algorithm thereby reducing the overall computational cost. So, the effectiveness of base point is necessary. Some chooses a random point on the curve as base point while some chooses the smallest point on the curve as the generator point. But there are some algorithms which gives optimum base point selection method. One such method is described below.
The base point of ECC on GF(p) is given below:

In ECC of GF(p), p is a prime number and Fp is a finite field of mod p. ECC uses modular arithmetic, so in modular form, the elliptic curve equation E over Fp can be defined as:

$$y2modp = (x3 + ax + b)modp, \quad where\ 4a3 + 27b2 \neq 0modp$$

Here, a and b are the curve parameters as stated earlier. Now if there is a point (x,y) which meets the above equation, then the number (x, y) is a point on the elliptic curve E. E(Fp) is used to represent set of all point which meets the curve E. The domain parameters of ECC on GF(p) are (q, a, b, p, n, h) where q is the module, a and b are the coefficient of ECC, n is the order of the base point, h is the cofactor of n, namely #(E)=nh [2].

Theorem 1:
Let assume a and p are integers, p>0. The basic idea of the algorithm is that at first we have to select an effective random point on the curve and then scalar multiplication is done using the random point. Finally, the scalar multiplication value is being used to judge the base point of the elliptic curve.

Algorithm 1: the base point choice algorithm of ECC on GF(p).
Input: a, b, p, n, h.
Output: Effective base point G on curve having order n.

Steps:

```
S1. Randomly choose x (0 ≤ x < p);
S2. a =(x3 + ax + b) mod p;
S3. Judging whether a belongs to quadratic residue of mod p, if so y is gotten, marked G = (x, y) go to S4, if not, go to S1.
S4. According to point G to compute G = hG, then judging whether G meets y2 = x3 + ax + b and G is not
infinite point. If so, G is the solved base point, then go S5, if not, go to s1.
S5. Return G
```

Algorithm 2: To judge whether a is quadratic residue and get the coordinate y.
Input: a, p, among them, a is gotten from S2 of Algorithm 1, p is mod;
Output: a, y, among them, a is gotten from S2 of Algorithm 1, y is y-coordinate;

Steps:

```
S1. If a=0, return (0, 0), or go to S2
S2. sum←0, y←1, i←1
S3. for i←1 to p do
sum← (sum+i) mod p,
if a=sum then
return (a,y),
else y←y+1, i←i+2,
if  i=p
return (-1,-1)[3].
```

IV. SECURITY AND EFFICIENCY OF ELLIPTIC CURVE

1024-bit parameters of RSA and Diffie-Hellman are normally being used in majority of the public systems. But these 1024-bit systems were sufficient for use until 2010 as stated by the US National Institute for Standards and Technology. Beyond that, NIST recommends that they were required to be upgraded to something which provides more security. But the question was what should these systems be changed to? One idea was to increase the public key parameter size to a level appropriate for which it can be used for another decade. The second option was to take advantage of the past 30 years of public key research and analysis and to move from first generation public key algorithms to elliptic curves. The judgments are made about the correct key size for a public key system in order to look at the strength of the conventional (symmetric) encryption algorithms that the public key algorithm will be used to key or authenticate. For a conventional encryption algorithm, the length of a key in bits is a common measure of security. In order to attack an algorithm with a k-bit key, roughly 2k-1 operations will be required. Hence, a public key system would be secure when one would use parameters that require at least 2k-1 operations to attack. In order to protect 128-bit AES keys one should use 3072-bit parameters of RSA or Diffie-Hellman which is basically three times the size in use throughout the Internet today. For elliptic curves, the equivalent key size is only 256 bits. Now it can be noticed that as symmetric key sizes increase the required key sizes for RSA and Diffie-Hellman increase at a much faster rate than the required key sizes for elliptic curve cryptosystems. Therefore, it can be said that elliptic curve systems offer much more security per bit increase in key size than either RSA or Diffie-Hellman public key systems.

The attractive feature of elliptic curve cryptography is not limited to security only. Elliptic curve cryptosystems also are more computationally efficient than the first generation public key systems, RSA and Diffie-Hellman. Although its arithmetic is slightly more complex per bit than that of RSA or DH arithmetic, the added strength per bit more than makes up for any extra compute time. Elliptic curves offer much better solution than first generation public key systems like Diffie-Hellman in channel-constrained environments. The National Security Agency has decided to move to elliptic curve based public key cryptography for protecting both classified and unclassified National Security information. The Cryptographic Modernization Initiative in the US Department of Defense aims at replacing almost 1.3 million existing equipments over the next 10 years.[1].

V. Conclusion

Elliptic curve cryptography provides better performance in efficiency and greater security than most of the first generation public key techniques viz. RSA and Diffie-Hellman which are in use. Point generation and base point selection is the key feature for security in ECC. It is also found that a considerably smaller key size can be used for ECC as compared to RSA. Therefore, we can say that there is computational advantage of using ECC with a shorter key length than comparably secure RSA. Even in terms of data files and encrypted files, ECC is more efficient. So, because of this, ECC can be efficiently used for wireless communication having low data rate transmission and for constrained devices due to low power requirements.

References

[1] http://www.nsa.gov/business/programs/elliptic_curve.shtml
[2] Elliptic Curve Cryptography- An Implementation Guide by Anoop MS.
[3] Yah HU, Yan CUI, Tong Li, “An Optimization Base Point Choice Algorithm of ECC on GF(p)”.
[4] http://www.math.wvu.edu/~hjlai/Teaching/Math373-578/Matlab-Example_2009.pdf