Journal of Computer Science 6 (3): 244-249, 2010
ISSN 1549-3636

A Cryptosystem Using the Concepts of Algebraic Geometric Code

[1]K.V. Pramod and [2]C. Manju
[1]Department of Computer Applications,
Cochin University of Science and Technology, Cochin, India
[2]Bharathidasan Government College for Women Puducherry, India

Abstract: Problem statement: Cryptosystem using linear codes was developed in 1978 by Mc-Eliece. Later in 1985 Niederreiter and others developed a modified version of cryptosystem using concepts of linear codes. But these systems were not used frequently because of its larger key size. In this study we were designing a cryptosystem using the concepts of algebraic geometric codes with smaller key size. Error detection and correction can be done efficiently by simple decoding methods using the cryptosystem developed. **Approach:** Algebraic geometric codes are codes, generated using curves. The cryptosystem use basic concepts of elliptic curves cryptography and generator matrix. Decrypted information takes the form of a repetition code. Due to this complexity of decoding procedure is reduced. Error detection and correction can be carried out efficiently by solving a simple system of linear equations, there by imposing the concepts of security along with error detection and correction. **Results:** Implementation of the algorithm is done on MATLAB and comparative analysis is also done on various parameters of the system. Attacks are common to all cryptosystems. But by securely choosing curve, field and representation of elements in field, we can overcome the attacks and a stable system can be generated. **Conclusion:** The algorithm defined here protects the information from an intruder and also from the error in communication channel by efficient error correction methods.

Key words: Algebraic geometric code, cryptography, repetition codes, decoding

INTRODUCTION

In 1978 Mc-Eliece developed a cryptosystem using linear codes. He made use of generator matrix and other basic concepts of linear codes for the process of encryption. Niederreiter (1986) later developed or modified the system using parity check matrix. Later signature systems were also developed. Main disadvantage of this system was the key size. Key size was quite large and difficult to manage. The error detection and correction capability of an (n, k, d) can detect errors only up to (d-1)/2 errors. In this study an attempt is made to develop a cryptosystem by making use of generator matrix of an algebraic geometric code.

Algebraic geometric code: The code constructed by choosing a family of points from a curve and space of rational functions that defines vector space of curve is known as algebraic geometric code. It is the part of coding theory and is a branch of mathematics that deals with transmitting information error free (Van Lint, 1998). The most important development in the theory of error correcting code in recent years is the construction of codes from curves introduced by Goppa (1983). Cryptography is the science of transmitting messages over an insecure channel. Here we are trying to combine both, that is, error correction and security.

Key aim of algebraic geometric code is to replace polynomial over a finite field by more general constructions. Goppa used language of algebraic curves to introduce codes (Van Lint, 1998). So it can be called algebraic geometric code. Factors used for describing the algebraic geometric codes are:

Divisor: A divisor (Stichtenoth, 1993) D on a curve X is a formal sum of form:

$$D = \sum n_p P$$

where, $n_p \varepsilon Z$ and $n_p = 0$ for all but a finite number of points P on X. Divisors are often thought to be the key to understand how algebraic geometry is formed and its relationship to curve. To describe it more clearly, Let C be a non-singular projective curve in P_k^2, the projective plane is over an algebraically closed field K. For each line in L in P_k^2, we consider $L \subset C$, which is a finite set

Corresponding Author: K.V. Pramod, Department of Computer Applications, Cochin University of Science and Technology, Cochin, India

of points on C. If C is a curve of degree d and if we count points with proper multiplicity then $L \cap C$ will contain exactly d points. So we can write:

$$L \subset C = \sum n_p P$$

where, P_i ε C are the points, n_i the multiplicity and this formal sum is a divisor on C. As L varies, we obtain a family of divisors on C, parameterized by the set of all lines in P^2, which is a dual projective space $(P^2)^*$. We refer to this set of divisors as a linear system of divisors on C. If P is a point of C, we consider the set of divisors in the linear system, which contains P. They correspond to the lines L ε $(P^2)^*$ passing through P and this set of lines determines P uniquely as a point in P^2.

Rational function and Riemann-roch theorem: Let X be a curve defined by a field F. A rational function (Stichtenoth, 1993) is defined at a point P, if there exists a representation f = A/B such that $B(P) \neq 0$. Another important thing we have to discuss before the construction and definition of algebraic geometric code is the space associated with the divisor. The space associated with the divisor may be called as linear space L(D).

Riemann-Roch theorem is one of the famous theorems in algebraic geometry (Stichtenoth, 1993; Van Lint, 1998). It deals with computation of L(D), the dimension of vector space L(D). Let X is a curve defined over a projective field and let d be the degree of curve X. The g, genus of curve is (d-1)(d-2)/2. A canonical divisor w is also defined such that deg(w) = 2g-2 and l(w) = g.

Theorem 1: Stichtenoth (1993) given a divisor D, l(D) = deg (D)+1-g+l(w-D) where w is any canonical divisor.

By making use of all the above discussed concepts of algebraic geometry, we can define an algebraic geometric code by Goppa (1983) as follows. Let X be a curve, P a set of points on the curve, D the divisor, then, algebraic geometric code associated to (X, P, D) is:

$$C(X,P,D) := \{(f(p_1),f(p_2)\ldots\ldots..f(p_n)|f \ \varepsilon \ L(D)\} \ C \ F_q^n$$

In other words, the algebraic geometric code C (X, P and D) is the image of the evaluation Map:

$$E: L(D) \rightarrow F_q^n$$
$$F \rightarrow ((f(p_1), f(p_2)\ldots\ldots..f(p_n))$$

By making use of the definition described above we can construct a Goppa code as follows. Let X be a curve, P be a set of n points on the curve $\{P_1, P_2\ldots..P_n\}$ and divisor $D = P_1+P_2\ldots.+..P_n.$ Let L(D) denote vector space for the curve. Length of the vector space l(D) as per Riemann Roch theorem is l(D) = n+g-1. For an elliptic curve d = 3 and genus g is given by Plucker's formula (Stichtenoth, 1993). g = ((d-1)(d-2))/2. So g here is 1. Then l (D) = n =# p (number of points on the curve). A code is represented by (n, k, d)) where n is the number of elements, k is the dimension and d is the distance. Here in algebraic geometric code dimension, K is degD+1-g and minimum distance d, where d>n-degD. (Thus we are mapping (X, P, D) to (n, k, d) curve). Let C = (X,P,D) be an algebraic geometric code and let $f_{1,}\ f_2\ldots\ldots.\ f_k$ be a basis for the vector space L (D) over F_q under the conditions above dim C = K and geometric matrix is defined as:

$$\begin{bmatrix} F_1(P_1) & \cdots\cdots & F_1(P_n) \\ & & \\ F_k(P_1) & \cdots\cdots & F_k(P_n) \end{bmatrix}$$

Elliptic curve cryptography: Miller (1986) and Kobltiz (1987) independently, proposed a public key cryptosystem which was analogous to the Elgammal scheme. In this group Zp* is replaced by a group of points on the elliptic curve defined over a finite field. The main attractions of elliptic curve cryptography over competing technologies such as RSA, DSA are that, various algorithms are known for solving the underlying hard mathematical problems in elliptic curve cryptography. Elliptic curve discreet logarithm problem takes fully exponential time. On other hand, the best algorithm known for solving the underlying hard mathematical problem in RSA and DSA (Integer Factorization problem and DLP problem) take sub-exponential time. This means that significant parameters used in ECC is small compared to RSA and DSA but with significant equivalent levels of security. The lack of sub exponential attack on ECC offers potential reductions in processing power, storage space, band width and electrical power. These advantages are especially important in applications on constructed devices such as smart card, pagers and cellular phones (Hanker *et al.*, 2004). The performance of ECC (Certicom, 1998) depends mainly on the efficiency of finite field computations and fast algorithms for elliptic scalar multiplication. In addition to the numerous known algorithms, the performance of ECC can be speeded up by selecting particular underlying finite field and/ or elliptic curve. ECC is used in many areas. Certicom is most prominent vendor, but there are many implementation issues for it.

Proposed system: In this session we will see how we can develop a cryptosystem by making use of concepts of algebraic geometric code. Main thing in a cryptosystem is the generation of keys.

First, we have to choose some public key parameters. An elliptic curve having a highly secured point over a finite field is chosen along with a fixed base point. So public information include Elliptic curve and Finite field.

Once we know the elliptic curve and field on which the curve is designed we can compute the linear vector space L(D) (Stichtenoth, 1993) over the curve in finite field. From the linear vector space we can generate rational functions and can compute base point. From the theory of algebraic geometric code, by using rational functions and points on selected curve we can generate a generator matrix $G_{k\times n}$ ca be generated. Here we are converting a curve into (n, k, d) code.

Next step is the processing of generating algorithms. The process involves 4 steps they are

Key generation: There are two types of keys in a public key cryptosystems: Public key and private key.

Consider that message is transmitted between two users X and Y, They keys can be defined as follows:

- Public key information includes F_q, B, t, α Where, F_q is the finite field, B is the base point, χ is the elliptic curve and t is the dimension of linear vector space
- User Y selects a random integer α, between 0 and ord_B, where, B is the base point of the chosen elliptic curve and it is sent to user X
- User X selects a random integer β, between 0 and ord_B where, B is the base point of the chosen elliptic curve

Encryption: Encryption is the process of converting a message into a form that is not understandable by a third person. This is done by making use of keys.

Let M be the message to be transmitted. Group the message into k units i.e., m_1, m_2….m_k. Convert this message into points. Conversion of messages into points on curve is called message imbedding. Now, generation of generator matrix: From the divisor D of the curve, find a sub space A and let L (A) be the linear subspace associated with A. Let $f_{1,}$ f_2….f_t be the functions related to it. From this, we can generate a generator matrix by using L (A) rational functions and message converted points:

$E: L(A) \rightarrow F_q$
$F \rightarrow ((f\ (p_1), f\ (p_2)\ldots\ \ldots..f\ (p_{k)})$

Let G be the generator matrix created. Encryption process involves following steps:

1. Compute P = β B
2. Compute $G^1 = [G_{t*k\ +}\ \alpha\ \beta\ B], P]$
3. Send G^1 to B

Decryption: Decryption is done at receiving end to convert data into its original form. The process include following step .From receivers secret key compute α β B

Subtract αβB from G^1. By taking rational functions and solving them, we will get the points represented through generator matrix Points are then converted into messages chunks and they are in turn converted into original message.

Decoding: Decoding process includes the process of error detection and correction. Here, we are sending information as contents of generator matrix. When we analyze it, we can see that it is a repetition of the point information there by we can treat it as repetition codes. When we solve step 3 of decryption algorithm we will get a set of repeated information. A drawback of repetition code is redundancy, which means we are transmitting more data. Although the process seems to be cumbersome, it is simpler than other cryptosystems using algebraic geometric code.

MATERIALS AND METHODS

Implementation on MATLAB: Implementation process was done by using mat lab. Implementation process include algorithms for

Parameter setting: Parameter generation is an important factor in developing an algorithm based on elliptic curve and algebraic geometric code. Parameters of an elliptic curve is called domain parameters which include field size, a, b of a curve E: $y^2 = x^3 + ax + b$. Two types of finite field used in a cryptographic application are Prime field and binary field. In this study, elliptic curve over prime field is considered. F_p and domain parameters of curve include (P, a, b, B, N_B, h). P is the field size, a and b are parameters in the equation of curve, B is the base point of the curve, N_b is the order of the base point and h is an integer which is cofactor h =# $E(F_p)/n$ (Certicom, 2000a; 2000b).

Parameter generation (Certicom, 2000b) can be done by random method or Kobltiz (1987) random selection method. When we select a field number of elements on the field should be a large prime. This is to avoid attacks and to improve the security of the system. Here only 3 main domain parameters are generated; remaining parameters are generated during the encryption process. Code parameters include (n, k, d), where n is the number of points on the curve, K is the dimension and d is the distance. K is selected according to the linear vector space generated.

Algorithm 1: Parameter generation:

```
Function[p,a,b]=Domain Parameters (a, ,b)

1. Pi ← 1
2. If $4 a^3 + 27 b^3 = 0$
       exit
       else
3. P← ECC_prime (a, b)
4. N ←point _count (p , a ,b)
5. if (N = = is_ prime (N) )
         Return (p)
   Else   go to step1
5. Stop
```

Key generation procedure: This algorithm takes the domain parameters, computes the base point and generates keys. Procedure is as follows:

Algorithm 2: Key generation:

```
Function[α] = Genkey (p,a,b)
1. $[x_B,y_B]$ = Genbasept (a,b,p);
2. m = Findorder (xb,yb];
3. α = randint (1,1,m);
4. End
```

Encryption procedure: Encryption procedure includes creation of generation matrix and conversion of message into points. The procedure is an overview of the encryption process using mat lab. Process involves functions for base point generation, creation of generator matrix, elliptic curve scalar multiplication. Scalar multiplication is done by successive doubling process.

Algorithm 3: Encryption:

```
Function[CM,P] = Encryption (p, a, b, message, β)
1. $[x_p, y_p]$ = Genbasept (a,b,p);
2. $[x_p, y_p,n]$ = pcpoints (a,b,p);
3. $[x_{mp}, y_{mp}]$ = msg2points (message);
4. GM = Genmatrix ($x_{mp}$, $y_{mp}$, a, b, p);
5. s = findorder ($x_b$,$y_b$);
6. α = randint (1,1,s);
7. $\gamma = \alpha * \beta$;
8. $[x_p, y_p]$ = Succdob ($x_b$,$y_b$, α,a,p);
9. P = $[x_k, y_k]$;
10. $[x_k, y_k]$ = Succdob ($x_b$,$y_b$, γ,a,p);
11. Z = $[x_k, y_k]$;
12. CM = [GM+Z,P];
13. End
```

Decryption procedure: Decryption procedure involves: Accepting the cipher text and converting into original message. It involves converting output contents into points and then into message.

Algorithm 4: Decryption:

```
Function[Message] = Decryption (CM, a, b, p)
1. Compute Q = Succdob ($P_x$, Py, α,a,p)
2. CM = CM-Q
3. $[P_x, P_y]$ = Cipher2point (CM);
4. for I = 1:1:len
5.       Message [I] = point2msg ($P_x$[ i ], $P_y$[ j] );
6. End
```

Decoding procedure: The algorithm is as follows:

Algorithm 5: Decoding:

```
Function[ ] = Decode (CM, $P_x$, $P_y$)
1. Compute GM1 = Genrmatrix1 ($P_x$, $P_y$, a, b ,p);
2. if (CM = = GM1)
      Disp (‘No error’);
      Break;
   Else
      Disp (‘Error’);
      Go to step
   End
4. k = size (CM);
5. Count = 0;
6. for I = 1:1:k (1)
3. for j= 1:1:k (2)
       if (CM ( i , j) = = GM1 ( i , j))
           count++;
        End
4. if (count > k/2)
     Disp (‘accept the message’ +CM ( i, j ));
   else
```

```
    Disp ('Communication error');
 End
```

The received message looks like a repetition code, Here we are comparing the entries and deciding whether to accept the message.

Various functions names are given in above procedures. These function call contains code to execute corresponding functions.

Security of the cryptosystem: The security of the systems is based on the apparent intractability of the elliptic curve discrete logarithm problem. ECDLP for a given elliptic curve $E(F_q)$, a point $P\varepsilon E(F_q)$, of order n and a point $Q\varepsilon E(F_q)$, determine an integer k, $0 \le k \le n-1$, such that Q = k P provided such an integer exist.

Various attacks that can affect the system include structural attack (Trappe and Washington, 2002), Pohlig-Hellman attack, Index calculus attack (Hanker *et al*., 2004). Pohlig-Hellman attack algorithm efficiently reduces the computation of $l = \log_p Q$ to the computation the computation of discrete logarithms in the prime order subgroups <P> (Hanker *et al*., 2004). It follows that the ECDLP in <P> is no harder than ECDLP in its prime sub-groups. Hence in order to maximize resistance to this attack the parameters of curve should be chosen so that order n of P is divisible by large prime. Pollard's Rho method makes use of iterative method and in order to avoid the attacks parameters should be chosen is such a way that ECDLP is infeasible to solve.

Structural attack: Certicom (2000a); Janwa and Moreno (1996); Niederreiter (1986) and Hanker *et al.* (2004) developed cryptosystems based on linear code during 1978. In this generator matrix is used as private key and it is scrambled using a scrambled matrix and permutation matrix to get the public key. The intruder tries to get k and n, so that generator matrix can be reproduced. This type of attack is called structural attack. In our system we are giving the parameters for constructing the generator matrix as public information, so that this type of attack doesn't have any significance.

Any attacks on this cryptosystem rely on structure or the group. So security of the system depends on size of the field q. So selection of field is important here, it should be a prime field or an optimal extension field and should be divisible by a large prime. Another important thing is number of points on curve. Let N be number of points on curve, it should be divisible by a large prime satisfying $n > 2^L$, where L is the security level which should be in range $160 \le L \le \lfloor \log_2 q \rfloor$ (Certicom, 2000b).

RESULTS

Analysis of algorithm above was done over various fields on Pentium IV processor. Time taken over various fields during the process of encryption and is as follows in Table 1 and Fig. 1.

Here we can see graphical representation of the table. we can see computing time increases with field size. Size of key also increases with field size. Disadvantage here is the size of the cipher text. This can be overcome by the advantage of decoding process which helps in the detecting of errors. An analysis has been done for the above algorithm for various message lengths and. the result of the analysis can be represented as a graph as follows in Fig. 2.

Table 1: Field size Vs time for encryption and decryption

Field F_q	127	163	192	223	321	521
Time (sec) encryption	1.16	1.46	3.14	5.44	7.18	9.12
Time (sec) decryption	2.12	2	4.32	4.523	9.12	12.186

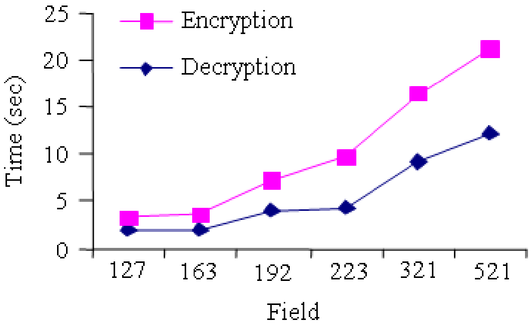

Fig. 1: Field size VS execution time

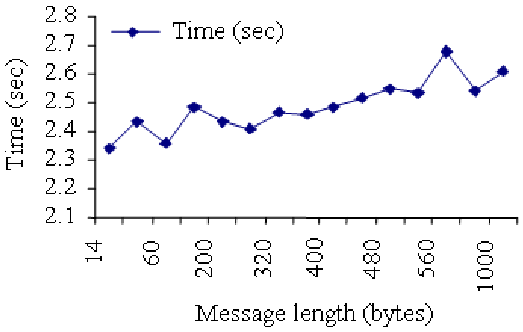

Fig. 2: Message length Vs time

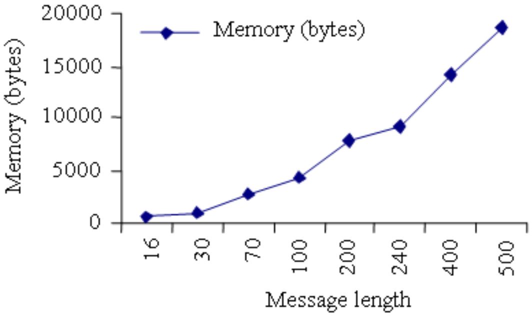

Fig. 3: Message space Vs message length

The results shows that increase message size will not effect the computation time. The program was executed for a field 163 over various message size fro 14-1000 bytes and result show that there is not much variation in time, when the message size is increased.

The system was tested for the size of the output message. As indicated earlier output represent a repetition code here. So as the size of the message increases the output size also increases. Figure 3 shows length of cipher text Vs message size in the above mentioned system.

DISCUSSION

From the graph we can see that the memory requirements increase as the message size increase.

Here we can see computing time increases with field size and also increasing the size of the message does not effect the processing time, Size of key also increases with field size. Disadvantage here is the size of the cipher text. This can be overcome by the advantage of decoding process which helps in the detecting of errors.

CONCLUSION

Here we have developed a public key cryptosystem by combining concepts of algebraic geometric code and elliptic curve cryptography. Mc-Eliece and Niederreiter developed a cryptosystem based on binary linear code. Main disadvantage of their system was its large key size. Key size used here is very small compared to Mc-Eliece and Niederreiter cryptosystems. Decoding process is simple since we are treating received information as repetition codes.

Attacks are common to all cryptosystems. But by securely choosing curve, field and representation of elements in field, we can overcome the attacks and a stable system can be generated. Main advantage of this system is, apart from decryption, error detection as well as correction can be done. However, this comes at a cost in the form of increased bandwidth.

REFERENCES

Certicom, 1998. The ECC-remarks on security of ECC. http://eref.uqu.edu.sa./The elliptic curve.pdf

Certicom, 2000a. SEC1 elliptic curve cryptography. http://www.secg.org/collateral/Sec1_final.pdf

Certicom, 2000b. SEC2 Elliptic curve domain parameter. http://www.secg.org/collateral/Sec2_final.pdf

Goppa, V.D., 1983 Codes on algebraic curves soviet. Math. Dokl, Translat., 259: 207-214.

Hanker, D., A.J. Menzes and S. Vanstone, 2004. Guide to Elliptic Curve Cryptography. 1st Edn., Springer, Germany, ISBN: 0-387-95273-x, pp: 311.

Janwa, H. and O. Moreno, 1996. Mc-Eliece public-key cryptosystem based on algebraic geometric codes designs. Codes Cryptogr., 8: 293-307.

Kobltiz, N., 1987. Elliptic curve crypto systems. Math. Comput., 48: 203-209.

Miller, V., 1986. Uses of elliptic curve in C cryptology. Lecturer Notes Comput. Sci., 218: 417-426.

Niederreiter, 1986. Knapsack type cryptosystems and algebraic coding theory. Probl. Control Inform. Theor., 15: 19-31.

Stichtenoth, H., 1993. Algebraic Function Fields and Codes. 1st Edn., Springer-Verlag, Berlin, ISBN: 13: 978-35407678784, pp: 260.

Trappe, W. and L.C. Washington, 2002. Introduction to Cryptography with Coding Theory. 2nd Edn., Prentice Hall, New Jersey, USA., ISBN: 10: 013186239.

Van Lint, J.H., 1998. Introduction to Coding Theory. 3rd Rev. Edn., Springer-Verlag, New York-Heidelberg-Berlin, ISBN: 3540641335, pp: 227.