IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 3, Issue 4, Aug-Sept, 2015
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

Polynomial Encryption Using The Subset Problem Based On Elgamal

Khushboo Thakur[1], B. P. Tripathi[2]

[1]School of Studies in Mathematics Pt. Ravishankar Shukla University
Raipur, Chhattisgarh 492001, India.

[2]Department of Mathematics, Govt. N.P.G. College of Science
Raipur, Chhattisgarh 492001, India.

Abstract

In this article we have demonstrated a polynomial message which have been encrypted through the Merkle-Hellman encryption scheme and sent by use of Elgamal. So that only the proposed receiver is efficient to decode the message. Our result also illustrated with the help of a mathematical example and algorithm..

Keywords: *Elgamal, Knapsack problem, super increasing vector, subset sum problem.*

1. Introduction

Lattices were first studied by Mathematicians Joseph Louis Lagrange and Carl Friedrich Gauss. In 1996, Mikls Ajtai showed the use of lattices as a cryptography primitive in seminal result. There are two fundamental computational problems associated with the lattice are shortest vector problem and closest vector problem. In general, CVP is to known be NP-hard and SVP is to know be NP-hard under a certain randomized reduction hypothesis [4]. The Merkle Hellman system is based on the subset sum problem [1]. The subset sum problem is a special case of the knapsack problem. The subset sum problem is hard, its decision problem was shown to be NP-complete by Karp [6]. The concept of the super increasing subset problem was coined by Merkle-Hellman in 1978. Ralph Merkle and Martin Hellman used the subset problem to create a cryptosystem for encrypt data [1]. In this system super-increasing knapsack vector s is created and the super-increasing property is hidden by creating a second vector M by modular multiplication and Permutation. Here the vector M is the public key of the cryptosystem and s is used to decrypt the message. The subset sum problem is to find a subset of a given set of positive integers $Z_1 \ldots\ldots\ldots\ldots\ldots Z_n$, such that the elements in the subset sum up to some given integer s. The subset sum problem is an NP complete problem [6] in combinatorial optimization. The knapsack problem selects the most useful items from a number of items given that the knapsack has a certain capacity. Knapsack problems are widely used to model solutions, industrial problems such as public key cryptography. Ralph Merkle and Martin Hellman used the subset problem to create a cryptosystem to encrypt data. A super-increasing knapsack vector s is created and the super-increasing property is hidden by creating a second vector M by modular multiplication and permutation. The vector M is the public key of the cryptosystem and s is used to decrypt the message [2].

ENCRYPTING MESSAGES:

This cryptosystem performs encryption in two steps.

First the polynomials are converted to their binary equivalent. These polynomials are encrypted through the Merkle-Hellman encryption scheme whose main idea is to create a subset problem which can be solved easily and then to hide the super- increasing nature by modular multiplication and permutation.

Secondly, these encrypted polynomial are further encrypted though the use of EL-GAMAL concepts.

We choose a prime number p and a primitive root g mod p and also choose a random exponent a $\in \{0.....p-2\}$ and calculate `A' such that

$$A = g^a \bmod p$$

We choose another random integer b $\in \{0.....p-2\}$ using these details we find out the value of `B' such that

$$B = g^b \bmod p$$

Here (p,g, A) is a public key and private key is a.

Thus the formula to encrypt the message using Elgamal cryptosystem is

$$c = A^b \text{ m mod p}$$

3. Mathematical Explanation:

Step 1: The first step is to choose key of length 7 bits. These are used to perform the first encryption process.

Step 2: The second step is to convert the polynomial of the message into binary. The binary sequence is represented by the variable y.

Step 3: Choose a super increasing sequence of number of positive integers. A super increasing sequence is one where every number is greater than the sum of all preceding numbers. S = $(s_1, s_2, s_3, \ldots\ldots s_n)$

Step 4: The forth step is to choose a random integer (z) such that

$$z \succ \sum_{i=0}^{n} s_i$$

and a random integer r, such that gcd (z, r) = 1 where r and z are co prime. The sequence s and the numbers z and r be the private key of the cryptosystem. All the elements $(s_1, s_2, s_3, \ldots\ldots\ldots s_n)$ of the sequence s are multiplied with the number r and the modulus of the multiple is taken by dividing with the number z. Now calculate the sequence k = $(k_1, k_2, k_3, \ldots\ldots\ldots k_n)$
Where

$$k_i = r * s_i \bmod z$$

The public key is k and private key is (r , z , s).

Step 5: The message is encrypted by multiplying all the elements of sequence k_i with the corresponding elements of sequence y_i where y_i is the i- th bit of the message and $y_i \in \{0,1\}$. The numbers are then added to create the encrypted message M_i forms the cipher text of the cryptosystem.

Therefore,

$$M_i = \sum_{i=0}^{n} k_i y_i$$

4. Example:

Encrypting the message $x^6 + x^5 + x^4$

Step 1: The first step is to convert the Companion matrix in binary equivalent- A= 1 1 1 0 0 0 0

Step 2: The second step is to choose a super-increasing sequence s is created s = (1, 3, 11, 23, 72, 114, 258)
This problem is easy because s is a super-increasing sequence.

Step 3: The binary sequence is y = $(y_1, y_2, y_3, \ldots\ldots\ldots y_n)$, choose a number z that is greater than the sum of all super increasing sequence then z = 457 and choose a number r that is in the range [1; z) where r = 15. The private key consists of z, s and r. To calculate a public key, generate the sequence k by multiplying each element in s by r mod z

k = (15, 45, 165, 345, 166, 339, 214)

because

k1 = 1 * 15 mod 457 = 15
k2 = 3 * 15 mod 457 = 45
k3 = 11 * 15 mod 457 = 165
k4 = 23 * 15 mod 457 =345
k5 = 72 * 15 mod 457 = 166
k6 = 114 * 15 mod 457 = 339
k7 = 258 * 15 mod 457 = 214
where the sequence k is a public key.

Step 4: The message is encrypted by multiplying all the elements of sequence k
with the corresponding elements of sequence y and adding the resulting sum.
Therefore, the encrypted message is

$$M = \sum_{i=0}^{n} k_i y_i$$

Encrypting the message $x^6 + x^5 + x.^4$ Its binary equivalent is 1 1 1 0 0 0 0, Where k = (15, 45, 165, 345, 166, 339, 214) and y = (1 1 1 0 0 0 0)

Then, M = 15 + 45 + 165 = 225

Step 5: This is encrypted using elgamal concepts.
Here we choose a prime number p=283, g=179 and choose random number b=113.
The value of `a' is choosen is 106. Therefore,

A = 179^{106} mod 283

A = 74
Thus the message is encrypted according to the formula

$$C = A^{b} M \bmod p$$

Thus the encrypted code for the first character becomes

C = 74^{113} * 225 mod 283 = 81

These codes are sent to the receiver. Thus the message to be transmitted to the receiver is 0081.

4. DECRYPTING MESSAGES

During the decryption process, the blocks of encrypted code are separated. On these blocks decryption is performed using the ELGAMAL concepts.
The formula used is

$$M = B^{p-1-a} C \bmod p$$

The output of it is decrypted using Merkle-Hellman Knapsack cryptosystem decryption process.

To decrypt the message M, the recipient of the message would have to find the bit stream which satisfies the Equation [1]

$$M = \sum_{i=0}^{n} k_i y_i$$

The first step is to calculate the modular multiplicative inverse of `r' in r mod z [3]. This is calculated using the Extended Euclidean algorithm. This is denoted by r^{-1}.

The second step is to multiply each element of the encrypted message (M) with r^{-1} mod z . Since r was chosen such that gcd (r, z) =1. The largest number in the set which is smaller than the resulting number is subtracted from the number. This continues untill the number is reduced to zero [5].

5. Example.

Step 1: Decrypting the message: C= 0081

Step 2: Decrypting encrypted code. Perform the decryption process on C=0081
using the concepts of ELGAMAL.
Here we chooses g=179 and random number b=113. Therefore,

`B' = 179^{113} mod 283

`B' = 237
Thus the message is encrypted according to the formula, M= B^{p-1-a} C mod p

Thus M = 237^{176} * 81 mod 283 =225

Step 3: The modular inverse of 15 in 15 mod 457 is calculated using the extended Euclidean algorithms and was found out to be 61.

Step 4: The encrypted message M is 225 and s = (1, 3, 11, 23, 72, 114, 258).
Again, 225 * 61 mod 457 = 15
Now decompose 15 by selecting the largest element in s which is less than or equal to 15. Then selecting the next largest element less than or equal to the difference, until the difference is 0.
15-11 = 4
4-3 = 1
1 - 1= 0
Thus, the binary sequence becomes 1 1 1 0 0 0 0.
The polynomial equivalent to this binary sequence is $x^6 + x^5 + x^4$.

5.1 Encryption Process Algorithms

In this section, we propose algorithms for above encryption and decryption mathematical example.

A.. Algorithm for iterative multiplication of binary number and super increasing number:
Input : Binary number a(m) and super-increasing array b[m].
Input : Size of array n.
Initialize : mul = 0, i = 0
while(i <= n)
{
mul = mul + a[i]*b[i];
i++
}
return (mul)

B. Algorithm:

Initialize : r = 15 and n = 457
Input : Size of array element:
i=0
while(I <= size)
{
input : array element of k
i++

}

```
i=0
while(i <= size)

{
c=r*s[i]
k(i)=c mod n
return (k(i))

  }
```

C. An algorithms for encrypted message:

```
function myfun (x,y, n)
{
A = 1
j = 1
while(j <= y)
A=(A * x) mod n
return (A)

  }
```

Input : g = 179
Input : b = 113
Input : p = 283
Now we create a function pt = elgamal mod (g, b,n)

Output : Plain text

Thus the algorithms for the scheme C = a^b m mod p is :

```
function encrypt(x, y, m, n)
{

k=1

j=1

while(j <= y)
{

k=(k*x)
  }

C=(k*m) mod n

return (C)
  }
```

5.2 Decryption Process Algorithms:

A. Algorithm:

Initialize : g= 179, b = 113

```
Input : g
Input : b
input : p
i=1
while(i <= b)
{
B = 1
B = (B * g) modp
return (B)
  }
```

B. Algorithm for Inverse modulo :

```
Function inv (a; b)
i=1
while (i < b)
{
num=(a * i) mod n
if(num=1)
{
return(i)
  }
  }
```

C. Algorithm:

```
Input : Size of array
Input :  Enter check value (say 15)
Input :  Enter array (s) elements
i=1
while(i <= size)

{
Input : array elements
i++

  }
while(i <= size)

{
j=1 while(j <= size)

{
if(s(i)> s(j))
temp=a(i)
a(i) = a(j)
a(j) = temp
i++

  }
i=1
while(check value ! = 0)

{
if(check value > = s(i))
check value = check value - s(i)
i++
return (check value)

  }
  }
```

4. Conclusions

This paper explain how to encrypt and decrypt data by the working of subset sum problem through the use of Elgamal concepts. The whole cryptosystem was demonstrated by encrypted a polynomial $x^6 + x^5 + x^4$ and then decrypting it.

References

[1] M. Hellman and R. Merkle, "Hiding information and signatures in trapdoor knapsacks", IEEE Trans. Inform. Theory, Vol. 24, 1978, 525-530.

[2] A. Menezes, P.vanOorschot and S.Vanstone, Handbook of Applied Cryptography, CRC Press 1996.

[3] W. Diffie and M. Hellman, "New directions incryptography", IEEE Trans. Inform. Theory, Vol. 22, 1976, 644-654.

[4] J. Hoffstein, J. Pipher and J. H. Silverman, An Introduction to Mathematical Cryptography, Undergraduate Texts in Mathematics, Springer 2008.

[5] Ashish Agarwal, "Encrypting Message using the Merkle Hellman Knapsack Cryptosystem", International Journal of Computer Science and Network Security Vol. 11 2011,12-14.

[6] Richard M. Karp, Reducibility among combinatorial problems, in Complexity of Computer Computations, Raymond E. Miller and James W. Thatcher (eds.) Plenum Press, NY, 1972.