ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Issue 12, December 2016**

# A Scheme for Secure Communication Using Polynomials

Dr. M. Sampath Kumar

Associate Professor, Dept. of Computer Science & Systems Engineering, College of Engineering, Andhra University, Visakhapatnam, India

**ABSTRACT:** The encryption and decryption algorithms for secure communication using truncated polynomials have become a research interest for many researchers due to their fact that they are easily implementable. In this paper encryption and decryption algorithms based on polynomials are presented. The coefficients of polynomials used in these algorithms need to satisfy some specific conditions. Hence it is required to find the coefficients of the polynomial from a given set of points. Two efficient algorithms, one recursive and one non-recursive, are presented for generating the coefficients using Lagrangian interpolation.

**KEYWORDS:** Encryption, Decryption, truncated polynomials.

## I. INTRODUCTION

The increasing use of the Internet as a means of contacting people and exchanging information has became steadily ingrained into our lives. The growing technology has especially helped in the integration of enterprise applications and access to information from anywhere has added to the productivity of each business sector. This has also made technology solutions more complex for organizations to handle, and complexity is the anathema of security. Due to the wake of increasing security breaches, a growing awareness of information security is beginning to set in.

It is at this point cryptography comes into picture. Cryptography is starting to play a vital role in providing information security. While encryption of data is generally considered the best method of providing confidentiality and integrity for data, authentication mechanisms are used to authenticate the receiver as well the sender to prevent impersonation attacks, and also to take care of non-repudiation and access control. Proper encryption and authentication techniques are to be used to protect the information from different threats.

The encryption and decryption algorithms for secure communication using truncated polynomials are presented[3]. The organization of this paper is as follows. Section-2 describes the proposed algorithm. Section-3 describes the detailed algorithm. Section-4 describes the calculations of polynomial coefficients.

## II.RELATED WORK

In Numerical computations polynomial evaluation is most frequently occurring task. Many algorithms are there for evaluation of polynomials at large number of values. The four basic arithmetic operations are done on polynomials for evaluation purpose [1]. Factorising the polynomials with their rational coefficients is discussed by A.K.Lenstra et.al. [2].Public key authentication using with polynomial rings is addressed by J. Hoffstin et. Al [3]. Broadcasting Cryptosystems in computer network environment using interpolating polynomials is discussed by C.C. Chang and T.C Wu [5].

## III.THE PROPOSED ALGORITHM

This section presents an outline of the proposed algorithm. Each step is explained as procure in more formal way in the next section.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Issue 12, December 2016**

**1. Key Creation**

Step 1: Choose two relatively prime numbers p and q (p < q) with the condition that there exists a prime number p' such that p*p' = 1 (mod q)

Step 2: Choose a positive integer n and two polynomials f(x) and g(x) of degree n-1 with coefficients,+1, -1, or 0 only, such that g(1) = f(1)-1 = 0.

Step 3: Compute inverse polynomials $f_p$ and $f_q$ such that f * fp =1(mod p) and f*$f_q$= 1(mod q).

Step 4: Choose h = p* fp*g (mod q). Then the pair <f, $f_p$ > is the private key and h is the public key.

**2. Message Encryption**

Suppose a message m has to be sent using the public key h.

Step 1: Choose a polynomial r(x) randomly, with coefficients +1, -1 or 0 such that r(1) =0.

Step 2: Encrypt the message m as c = r*h + m (mod q)

Step 3: Transmit the cipher text c.

**3. Message Decryption**

Step 1: Upon receiving the cipher text c, computes a = f*c (mod q)

Step 2: Express the coefficients of a in the range [-q, q].

Step 3: Compute b = a (mod p)

Step 4: Compute e = $f_p$*b (mod p)

Step 5: e is the original message m.

## IV. DETAILED ALGORITHM

In this section, a detailed account of each step of the algorithm above and the procedures for key creation, encryption and decryption are discussed.

**1. Key-Generation:**

This procedure creates the public key h and private key pair (f, $f_p$) and assumes the following functions:

1. The procedure *polyf*(f, df) generates a random polynomial of degree n-1 with coefficients df number of 1's and df-l number of -1's and rest zeroes.
2. The procedure *polyg*(g, dg) generates a random polynomial of degree n-l with coefficients dg number of 1's and dg number of -1's and rest zeroes.
3. The procedure *inverspoly*(f, p) generates an inverse polynomial of f(mod p) of degree n-l.
4. The procedure *convolution*(a, b) returns the convolution of the polynomials a(x) and b(x).

```
Create key (int p, q, n, polynomial f, fp, h)
 {
  back:   polyf (f, df);
          polyg (g, dg);
          fp= inverspoly(f, p);
          fq= inversepoly(f, q);
          if(not (inverspoly(f, p) or inverspoly(f, q)) then goto back;
          pfq= p*fq(mod q);
          h=convolution(pfq, g):
          h= h(mod q);
  }
```

**2. Encryption:**

This function encrypts the message in polynomial form using the public key h. This uses the procedure *polyr*(r, dr, n) to generate a random polynomial r(x) with dr number of coefficients +1 and dr number of coefficients -1 and the rest are all zeroes.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 12, December 2016

```
Polynomial Encode(int q, polynomial m, h)
    {
        e = m;
        polyr(r, dr, n);
        e += convolution(r, h);
        e = e( mod q);
        Encode = e;
}
```

**3. Decryption:**

The following procedure decrypts the encoded message e in polynomial form of using private key <f, fp>

```
Polynomial Decode (e, h: polynomial)
{
        a = convolution (f, e);
        a= a (mod q);
        b =a( mod p );
        c = convolution (fp, b);
        c = c (mod p);
        Decode = c;
}
```

The convolution function used in the above algorithms is given below

```
Polynomial convolution(a, b: polynomial);
    {
        for( k=1; k<=n; k++)
       {
         c[k]=0;
         j=k-l;
         for( k=1; k<=n; k++)
           {
             if (j == 0) j = n;
             c[k] = c[k] + a[i] * b[j];
              j = j-1;
           }
     }
                Convolution = C;
}
```

The following algorithm finds the inverse of a polynomial a(x) in the ring of polynomials of degree n-l with respect to a prime number p. Here $f_0^{-1}$ is chosen such that $f_0 * f_0^{-1} = 1$(modulo p)

Step 1: Initialization

f(x):= a(x);
g(x):= $x^n$-1;
b(x):=1;
c(x):=0;
k:=0;

Step 2: $f_0$:=f(0);

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 12, December 2016

```
        while ( f0 ≠ 0)
         {
           f(x)=f(x)/x;
           c(x)=c(x)*x;
           k=k+1;
         }
Step3: if(deg(f(x) ≠ 0)
         {
           b(x)= f0^-1*b(x)(mod p);
           return (x^n-k*b(x)(mod x^n-l);
         }
Step 4: if(deg(f(x) < deg(g(x))
         {
            Swap( f(x), g(x));
            Swap(b(x), c(x));
         }
Step 5:g0=g(0);
          u=f0*g0^-1(mod p);
          f(x)=(f(x)-u*g(x))(mod p);
          b(x)=(b(x)-u*c(x))(mod p);
        go to Step 2;
Step 7: End;
```

## V. CALCULATIONS OF COEFFICIENTS

The polynomial arithmetic is basically involved in the manipulation of its coefficients. It is known that n+1 distinct point on it uniquely determine an nth degree polynomial. The converse of the statement is also true. That is, given n+1 distinct points $(x_i, y_i)$ there exists a unique nth degree polynomial a(x) passing through these points [2]. We now give a procedure to produce the coefficients of A(x).

Here *padd*(a, b) and *pmult*(a,b) are two functions used for adding and multiplying two polynomials a(x) and b(x), x and y are two arrays containing n points $(x_i, y_i)$ $(1 \leq i \leq n)$ and ans is the interpolating polynomial of these points.

**1. Direct Algorithm:**

An algorithm for computing the coefficients is given in this section with a time complexity of of $O(n^3)$. A more efficient algorithm can be stated if we can observe the recursive nature of its coefficients which is discussed in the next section.

```
Interpolate(a,y, n, ans)
    {
     int  denom, n;
     polynomial poly, ans;
     read x[n], y[n];
        ans = 0;
        for( i=1; i <= n; i++)
         {
            poly =1;
            denom = 1;
              for(j=1; j <= n; j++)
               {
                    if( i!= j)
```

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Issue 12, December 2016**

```
                {
                  poly= pmult(poly, x – x(j));
                  denom* = (x(i) – x(j));
                }
              }
          }
          ans= padd(ans, pmult(y(i)/denom, poly);
    }
```

**2. Recursive Algorithm:**

If the interpolation polynomial for n points $(x_i, y_i)$, i = 1,2...,n, is already known and another point $(x_{n+1}, y_{n+1})$ is given then the interpolating polynomial for all these n+1 points can be calculated using the following formula using recursion.

If $a_{n-1}(x)$ is the interpolation polynomial for the points n-1 points $(x_k, y_k)$, k=1, 2,...,n-1 then the interpolation polynomial $a_n(x)$ for the points $(x_k, y_k)$, k=1, 2,...,n is given recursively as

$$a_n(x) = (a_n – a_{n-1}(x_n)) (d_{n-1}(x) / d_{n-1}(x_n)) + a_{n-1}(x) \text{ where } d_{n-1}(x) = (x - x_1) \ldots (x – x_{n-1})$$

The following computes the interpolating polynomial based on this recursive formula. Horner's rule is being used for evaluating the polynomials for evaluating $a_{n-1}(x)$ at $x_n$ and the $d_{n-1}(x)$ at $x_n$. Here x[n], y[n] are arrays of the n pairs of points and a is the unique interpolating polynomial of them. The coefficients of the unique interpolating polynomial of degree n are returned in a.

```
coeff(x, y, n, a)
 {
    float x[n], y[n], num, denom;
    polynomial d, g;
       g = y(1);
       d = x –x(1);
       for( i=2; i<=n; i++)
          {
               denom = horner(d, i-1, x(i));
               num = horner(g, i-2, x(i));
               g= padd(pmult((y(i) – num) / denom, d),g);
               d = pmult(d, x- y(i));
           }
    }
```

These algorithms are useful in generating the coefficients of the polynomials in the algorithms proposed for encryption and decryption.

## VI. CONCLUSION

The polynomial based crypto-systems for encryption and decryption are gaining importance in view of their implementation. The basic limitation in these algorithms is the coefficient calculations. In this paper a polynomial based crypto-system is proposed. The coefficient calculation is done using the Lagrangian interpolation method [1,4]. These algorithms can be used in distributed environment also[5].

## REFERENCES

[1] A.B.Borodin and I. Munro," Evaluating Polynomials at Many Points", *Information Processing Letters*, 1:2, pp 66-68 1971.
[2] A.K. Lenstra, H.W. Lenstra Jr., L. Lovasz, "Factoring Polynomials with Rational Coefficients", *Mathematische Annalen*. Vol: 261, pp 513-534, 1982.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Issue 12, December 2016**

[3] J. Hoffstein, D. Lieman, J. Silverman, "Polynomial Rings and Efficient Public Key Authentication", *Proceeding of the International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99),* M. Blum and CH. Lee, eds., City University of Hong Kong Press.1999.
[4] J. Hoffstein, J. Silverman, "Polynomial Rings and Efficient Public Key Authentication II", *Proceedings of a Conference on Cryptography and Computational Number Theory (CCNT '99),* I. Shparlinski et.al. eds. Birkhauser, pp 269-286., 1999.
[5] C. C. Chang and T. C. Wu, '"Broadcasting Cryptosystem in Computer Networks using Interpolating Polynomials," J. Computer System Science and Engineering, Vol. 6, pp. 185-188, 1991.