CHAPTER -4
4. SHARED KEY GENERATION USING TRUNCATED POLYNOMIALS

4.1 SHARED KEY CONCEPTS

The present area of research is demanding a work which is slightly diverted from the public key systems but yet based on Public Key Cryptosystems called as multiparty communication. Applications like financial transactions and storing of the servers password in a number of locations (parts to be stored), database security are looking into advancements into this area where more than a single recipient have to receive the message together. In such cases the private key of the previous two party communications needs to be shared among the different receivers of the message. The sharing has to take place in such a way that until majority of them insert their share of the private key, the message cannot be decrypted. At the same time care needs to be taken in sharing the key because a coalition of a group less than the majority should not be able to decrypt the message or calculate the remaining part of the key. These concepts are generally addressed as shared key generation [10, 18]. These concepts are explained to a certain extent in Fig 4.1

This chapter presents mechanisms that were designed, and developed as part of the work to achieve shared key concepts based on truncated polynomials. The analysis of the algorithms that were designed is also given in the last part of the chapter.
Fig 4.1 : Shared key concepts
4.2 SHARED KEY GENERATION USING TRUNCATED POLYNOMIALS

The concepts of shared key generation using truncated polynomials are based on NTRU algorithm, which was presented in section 2.3.

It is assumed that the public key $h$ is known to every body, and the private key $\{f, f_0\}$ is shared between the partners. The Trusted Third Party (TTP), will compute the inverse and distribute shares of the inverse of a polynomial, given shares of the polynomials. One more requlrement is that previous to the communication the shared recipients $A$ and $B$ already have a generated Public/Private Key pair. Further $A$ and $B$ also need to agree upon a common degree $N$ and the values of $p$ and $q$. Let $X$ is a party wishing to send a document to $A$ and $B$. The entire discussion of shared key using truncated polynomials can be put into key creation, encryption and decryption techniques.

4.2.1 Key Creation

This section deals with key creation and the algorithm for key creation.

(See Figure 4.2). The degree of the polynomial $n$ needs to be agreed upon by both the parties.

1. Let $A$ generate two polynomials $f_1$ and $g_1$ such that

$$f_1(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_{n-1} x^{n-1}$$

$$g_1(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + \ldots + c_{n-1} x^{n-1}$$

2. Similarly let $B$ generate two polynomials $f_2$ and $g_2$ such that
3. Let both A and B send their $f_1$, $f_2$ to the Trusted Third Party (TTP) by encrypting them with the public key of the TTP.

4. The TTP will calculate $f = f_1 * f_2$ and find the inverse of $f$ if it exists. Else the same is informed to A and B, in which case they have to choose fresh values of $f_1$ and $f_2$.

   The process is repeated until an inverse of $f$ exists. Let the inverse of $f$ be denoted as $f_1^{-1}$.

5. The TTP will factorize $f_p$ into two parts $f_{p1}$ and $f_{p2}$. The TTP will also calculate $f_q$ and then send the shares of $f_p$ ( $f_{p1}$ and $f_{p2}$) and $f_q$ to A and B, by encrypting them with the public keys of the A and B respectively.

6. The polynomial $g$ in NTRU algorithm will also be split into $g_1$ and $g_2$. A will have to formulate the small polynomial $g_1$ and B needs to formulate $g_2$. For exchanging the values of $g_1$ and $g_2$, A will encrypt the coefficients and degree of $g_1$ with the public key of B, and B will encrypt the coefficients $g_2$ and degree of $g_2$ with the public key of A. Then A and B will need to decrypt the messages received and compute $g = g_1 * g_2$.

7. Let each party now calculate the public key $h = p * f_q * g$ mod q.

8. The private key of A is $(f_{p1}, f_1)$ and that of B is $(f_{p2}, f_2)$.
Fig 4.2: Key Creation
4.2.2 Encryption

This section deals with the encryption process involved. The process of encryption is explained in Figure 4.3. When X wants to send a document to be viewed by both A and B, X will initially have to express the message \( m \) as a polynomial and choose a random polynomial \( r \) where \( r \) is small similar to \( f \) and \( g \). Then X needs to encrypt the message \( m \) using the formulae

\[ e = r \cdot h + m \]

![Diagram of Encryption Process](image)

Figure 4.3 : Encryption
4.2.3 Decryption

The method used to decrypt a given ciphered document by the method of sharing the private key is dealt in this section. Figure 4.2.3 gives a diagrammatic explanation of the process.

The ciphered text $e$ is sent to both $A$ and $B$. Let each participant (i.e., $A$ and $B$) choose random small polynomials $r_1$ and $r_2$.

1. Let $A$ calculate $z_{a1} = f_1^*(e + r_1)$, let $B$ calculate $z_{b1} = f_2^* (e + r_2)$

2. The values of $z_{a1}$ and $z_{b1}$ are to be exchanged using the previously agreed upon public keys $pub_1$ and $pub_2$. After decryption $z_{a1}$ will be known by $B$ and $z_{b1}$ by $A$.

3. Now $A$ will calculate $z_{a2} = f_1^* z_{a1}$ mod $q$ and $B$ will calculate $z_{b2} = f_2^* z_{a1}$ mod $q$. $A$ will then need to adjust the coefficients of $z_{a2}$ in the range of $[q/2, -q/2]$. $B$ will also need to repeat the same with $z_{b2}$. $A$ will then calculate $z_{a3} = z_{a2}$ mod $p$ and $B$ will calculate $z_{b3} = z_{a2}$ mod $p$.

4. $A$ will calculate $z_{a4} = z_{a3} * f_{p1}$ and $B$ will calculate $z_{b4} = z_{b3} * f_{p2}$. Then these values of $z_{a4}$ and $z_{b4}$ will need to be exchanged between $A$ and $B$.

5. $A$ will retrieve the message $m$ by computing $m = f_{p1} * z_{b4} - r_1$ and $B$ will retrieve the message $m$ by computing $f_{p1} * z_{a4} - r_2$. 

39
Fig 4.4: Decryption

4.3 ANALYSIS OF THE SHARED KEY USING NTRU ALGORITHM

The strength of the NTRU algorithm lies in keeping f and g secret [5]. Though f is to be maintained secret, it is f, that is commonly used hence this paper has concentrated in keeping both f, f, by dividing f into shares and each party involved in the communication have to keep their shares of f, secret. The f value is calculated by the TTP after receiving the shares of f, from each participant as f, for i=1,2, . . n. The TTP after calculating f and then its f, w.r.t p will factorize it into n parts and distribute it to all the n parties where f,=f, . It will not be possible for any individual party to decrypt
all by themselves as they have to obtain the shares of \( f_p \) from the remaining parties. While obtaining the values the algorithm suggested in this paper has taken care that the value of \( f_p \) of any party is not revealed to the others. The information necessary for decoding the message is only revealed.

Initially each user will need to compute \( z_{a1} \) (\( z_{b1} \) for user B). The Value of \( z_{a1} \) may be computed as follows

\[
z_{a1} = f_1(e + r_1)
\]

\[
= f_1(r * h + m + r_1)
\]

\[
= f_1(r (p * f_q * g) + m + r_1)
\]

\[
= f_1(r * p * f_q * g + m + r_1)
\]

{similarly \( z_{b1} \) may be computed as \( z_{b1} = f_2(r * p * f_q * g + m + r_2) \)}

These values are exchanged by the two parties

Hence A knows both \( z_{a1} \) and \( z_{b1} \)

Once \( z_{a1} \) and \( z_{b1} \) are exchanged the message can be retrieved as follows

When A calculates \( z_{a2} \) where \( z_{a2} \) can be calculated as

\[
z_{a2} = f_1 * z_{b1} \mod q
\]

\[
= f_1 * f_2(r * p * f_q * g + m + r_2) \mod q
\]

\[
= f_2(r * p * f_q * g + m + r_2) \mod q
\]

\[
= p * r * g + f(m + r_2) \mod q
\]
As the values of $r,g,m$ and $r_2$ are small compared to $q$

$$z_{a2} = p \times r \times g + f(m + r_2)$$

$$z_{a3} = z_{a2} \mod p$$

$$z_{a4} = z_{a3} \times f_{p1}$$

$$= z_{a2} \times f_{p1} \mod p$$

$$= f_{p1} (p \times r \times g + f(m + r_2) \mod p$$

$$= f \times f_{p1} (m + r_2) \mod p$$

Similarly $z_{b4} = f \times f_{p2} (m + r_1) \mod p$

These values of $z_{a4}$ and $z_{b4}$ are exchanged. Hence $A$ knows $z_{b4}$

Once $z_{b4}$ is known the message $m$ is retrieved by the operation

$$f_{p1} \times z_{b4} - r_1 = f_{p1} \times f \times f_{p2} (m + r_1) \mod p - r_1$$

$$= m + r_1 - r_1$$

$$= m.$$