A study on application of cryptography in data encryption and decryption

Ms.J. Nirmala[1], Ms. Y. Preethi Ceon[2]
[1,2]*Assistant Professor, KG College of Arts and College*

Abstract **- This paper deals with the comparison of three algorithms namely Elliptic Curve Cryptography Algorithm, ECC Algorithm, NTRU (Theory Research Unit Algorithm) and their performance comparison is analysed using the range of data which is applicable to each algorithm. Each method uses private and public keys for data encryption and decryption.**

Index Terms **- Cipher text, Public key, Private key, Polynomial values.**

INTRODUCTION

Any network system requires the cryptosystem to be secured. Cryptography deals with keeping the information or the data away from others or in a form which is not easily understood by others. This can be achieved in many ways. One such way is using Asymmetric key cryptosystem, where the two communicating parties use two different keys for sharing message. Three types of algorithms and compared analysed in this paper.

RSA ALGORITHM

a) Key Generation

RSA uses Public Key to encrypt messages and Private Key to decrypt that message.

1. Choose two large distinct similar bit length prime numbers p and q at random.
2. Find the product of p and q and assign it to n as n=pq. The length of n is the length of the key.
3. Find $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1)$, where φ is Euler's totient function.
4. At random, choose an integer e in a way that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$. e is the exponent of Public key and should be a large value, to be secured.
5. Determine d using the formula, $de \equiv 1 \pmod{\varphi(n)}$. d is the Private Key Exponent.
6. Finally, the values of public key and private key will be like Public key=(n,e) and Private key=(n,d).

Example

Suppose we want to encrypt the message "NUMBER" to send it to the receiver,

NUMBER ⇒ 142113020518

First, we have to choose two prime numbers,

p = 59 and q = 41,

Then, n = pq = (59)(41) = 2419

Euler's Totient function $\Phi(n) = (p-1)(q-1) = (58)(40) = 2320$

Now, to choose an integer e, which satisfies the given conditions.

Therefore, e = 3

To determine the private key d, by using the formula,

$ed = 1 \pmod{\Phi(n)} \Rightarrow (3)d = 1 \pmod{2320} \Rightarrow d = 1547$

Encryption

To encrypt the plain text into a cipher text, by using the formula,

$c = M^e \pmod{n} = (142)^3 \pmod{2419} = 1611$

$= (113)^3 \pmod{2419} = 1173$

$= (020)^3 \pmod{2419} = 743$

$= (518)^3 \pmod{2419} = 930$

Therefore, the encrypted cipher text value is 1611 1173 743 930

Decryption

To decrypt the cipher text into a plain text, by using the formula,

$M = c^d \pmod{n} = (1611)^{1547} \pmod{2419} = 142$

$= (1173)^{1547} \pmod{2419} = 113$

$= (743)^{1547} \pmod{2419} = 020$

$= (930)^{1547} \pmod{2419} = 518$

Therefore, the decrypted plain text value is 142 113 020 518

Therefore, the decrypted message is 142113020518, which is displayed as NUMBER.

ECC ALGORITHM

a. Key Generation

Key generation is quite important because the process of generating public and private key takes place here.

1. Choose a number d within the range of (1 to n-1).
2. Calculate the Public key using the formula. Q= d * P. Here, Q is the Public Key and d is the Private Key.

b. Encryption

1. Let 'm' be the message we are sending. And let E be the curve. Let the message 'm' have a point on the curve and it is denoted by 'M'.
2. Choose a number k at random in the range of [1 - (n-1)].
3. Let the cipher texts to be generated be denoted as C1 and C2.
4. C1 is calculated as C1 = k*P and C2 is calculated as C2 = M + k*Q.
5. Now, this C1 and C2 will be sending, with which the other side person can decrypt the message.

c. Decryption

In order to decrypt the message 'm'; we use the formula, M=C2-d*C1.

Example

Suppose we want to encrypt the message “NUMBER” to send it to the receiver,

NUMBER ⇒ 142113020518

First, let us choose (N,P,d) is (6,59,2)

To find Q,

Q = d*P = (2)(59) = 118

Encryption

Let us choose a number k in the range (1 to N-1)

Therefore, k = 3

Now, to find C_1 and C2

C1 = k*P = (3)(59) = 177

C2 = M+k*Q = 142+(3)(118) = 496

= 113+(3)(118) = 467

= 020+(3)(118) = 374

= 518+(3)(118) = 872

Therefore, the encrypted cipher text value is 496 467 374 872

Decryption

To decrypt the cipher text into a plain text, by using the formula,

M = C2-d*C1 = 496-(2)(177) = 142

= 467-(2)(177) = 113

= 374-(2)(177) = 020

=872-(2)(177) = 518

Therefore, the decrypted plain text value is 142 113 020 518

Therefore, the decrypted message is 142113020518, which is displayed as NUMBER.

NTRU Algorithm

a. Key Generation

1. In order to generate the key pair, we consider two polynomials f and g, with coefficients much smaller than q, with degree at most N-1 and with coefficients in {-1, 0, 1} are required.
2. The value of f should be chosen in such a way that, f. fp=1 (mod p) and f.fq=1 (mod q) should exist.
3. Her f and fp are the Private keys.
4. The public key can be calculated using the formula, h=pfq.g(mod q).

b. Encryption

1. Choose a polynomial 'm' with co-efficient {-1,0,1}, where 'm' is the Message.
2. Choose a polynomial 'r' randomly with small co-efficient.
3. The message can then be finally encrypted using the receivers Public key as; e=r.h+m (mod q).
4. Now, e is the cipher text and this is sent safely to the receiver.

c. Decryption

1. To decrypt the message, the receiver should first multiply the cipher text and the part of private key f. This is done by a=f.e (mod q).
2. Now calculate the value of a mod p, using the formula; b=a(mod p)=f.m (mod p).
3. With the value of b, the receiver can multiply b with the other part of private key fp, in order to recover the original message.
4. This is done by c=fp.b = fp.f.m (mod p), i.e. c=m (mod p). Here, c is the original message sent by sender.

Example

Suppose we want to encrypt the message "NUMBER" to send it to the receiver,

NUMBER ⇒ 1 0 1 0 1 1 1

Let N = 11, p = 3, q = 32

First, let us consider two polynomials f and g which satisfies the conditions,

f = -1 1 1 0 -1 0 1 0 0 0 1 -1 and

g = -1 0 1 1 0 1 0 0 -1 0 -1

CONCLUSION

Thus, this paper deals with the comparison and analysis performance of three algorithms namely Elliptic Curve Cryptography Algorithm, ECC Algorithm, NTRU (Number Theory Research Unit) Algorithm in a simple mathematical way.

REFERENCES

[1] R. Nara, N. Togawa, M. Yanagisawa, T. Ohtsuki, Scan-Based Attack against Elliptic Curve Cryptosystems, Proc. of Design Automation Conference (ASP-DAC'10), pp. 407-412, 2010.

[2] Sriram VSS, Dinesh S, Sahoo G (2010). Multiplication Based Elliptic Curve Encryption Scheme with Optimized Scalar Multiplication (MECES). Int. J. Comput. Appl., 1(11): 65-69.

[3] Zhu J H, Cui G H, Zheng M H, Zhou S Y.Trusted Verifiable Multisecret Sharing Scheme Based on (t, n) Threshold. Journal of Chinese Computer Systems, 2008,29(4): 635-638.

[4] Abomhara M, Khalifa OO, Zakaria O, Zaidan AA, Zaidan BB, Alanazi HO (2010). "Suitability of Using Symmetric Key to Secure Multimedia.

[5] Barkan E, Biham E, Keller N (2008). Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication. J. Cryptol., 21(3):392-429

[6] Collen Marie O'Rourke "Efficient NTRU Implementation" A thesis For Master of Science at Worcester Polytechnic Institute, Apr 2002.

[7] Challa N, Pradhan J (2007). Performance Analysis of Public key Cryptographic Systems RSA and NTRU. IJCSNS Int. J. Comput. Sci. Netw. Security, 7: 87-96.

[8] Hoffstein, J., Pipher, J. and Silverman, J.H., 1998. "NTRU: A Ring Based Public Key Cryptosystem", Proceedings of ANTS III, Portland, Oregon, USA, volume 1423 of Lecture Note in Computer Science, pp. 267-288, Springer-Verlag.

[9] J. Hermans, F. Vercauteren, and B. Preneel, "Speed Records for NTRU", in proc. of CT-RSA, LNCS vol. 5985, pp. 73-88, Springer, Heidelberg, 2010.

[10] A. Kamal, A. Youssef, Fault Analysis of the NTRU Encrypt Cryptosystem, IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, vol. E94-A, issue 4, pp. 1156-1158, 2011.

[11] Kute VB, Paradhi PR., Bamnote GR (2009). A Software Comparison of RSA and ECC. Int. J. Comput. Sci. Appl., 2(1): 43-59.

[12] Kapoor V, Sonny V, Abraham Singh R (2008). "Elliptic Curve Cryptography." ACM Ubiquity, 9(20): 20-26.

[13] LI Feng, LI Da-xing. Improved Multisecret Sharing Threshold Scheme[j].Computer Engineering, 2008,34(5):11-13

[14] R. Nara, N. Togawa, M. Yanagisawa, T. Ohtsuki, Scan-Based Attack against Elliptic Curve Cryptosystems, Proc. of Design Automation Conference (ASP-DAC'10), pp. 407-412, 2010.

[15] Sriram VSS, Dinesh S, Sahoo G (2010). Multiplication Based Elliptic Curve Encryption Scheme with Optimized Scalar Multiplication (MECES). Int. J. Comput. Appl., 1(11): 65-69.

[16] Zhu J H, Cui G H, Zheng M H, Zhou S Y.Trusted Verifiable Multisecret Sharing Scheme Based on (t, n) Threshold. Journal of Chinese Computer Systems, 2008,29(4): 635-638.