Volume 3, Issue 8, August 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Avalanche Effect and its Statistical Analysis of NTRU & RSA Cryptosystem

P.D.N.V.V Mahesh
Computer Science and Engineering
Sri Vasavi Engineering College, India

Rakesh Nayak
Dept. of Information Technology
Sri Vasavi Engineering College, India

Abstract— Cryptography has come up as a solution which plays a vital role in information security system against malicious attacks. This security mechanism uses some algorithms to scramble data into unreadable text which can be only being decoded or decrypted by party those possesses the associated key. Asymmetric encryption techniques i.e. Rivest-Shamir-Adelman (RSA) [2, 3, and 6] and NTRU [2, 3, and 5] algorithms have been implemented. Avalanche Effect [1] was analyzed on this encryption technique. Avalanche Effect means a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the cipher texts. The proposed technique is check the which algorithm is more secured.

Keywords—: RSA, NTRU, Encryption, Decryption, cipher text, secret key, Avalanche Effect.

I. INTRODUCTION

The development in technology and networking has posed serious threats to obtain secured data communication. Cryptography is technology to converts the plain text into cipher text, it provides the security of the secrete message but cryptanalysis breaking the cipher text. Several methods have been proposed which include public key- private key algorithms such as RSA and NTRU (Nth degree truncated ring unit). For the past two decades Cryptographic techniques have become essential part of any secure digital communication. All the cryptosystems can be classified in two types Private key systems and Public key systems. Avalanche Effect is calculated on public key crypto system include RSA and NTRU.

A. Cryptography

Public key cryptosystems is generating two keys, public key and private key. Public key is used for encryption and private key is used for decryption. RSA and NTRU are examples of the public key crypto system is based on polynomial function.

1) Plain text

This is original or secret information is ready to transmit in communication channel but there is no secured for the data.

2) Encryption

Input of the process is public key and plain text. Output of the process is cipher text.

3) Decryption

It is process of converting cipher text into plain text with help of private key.

B. Avalanche

It is the techniques to check level of the security in any cryptographic method. It is that a small changes in either the plain text or the public key, should produce a significantly change in the cipher text.

C. Statistical analysis

Information is varying from one technique to other technique in cryptography. In this we apply NTRU and RSA algorithms with one bit change in plain text and public key, to identify the best technique with help of some statistical methods.

II. MATHEMATICAL GROUNDWORK

A. *RSA*

RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated the following way: Choose two distinct prime numbers p and q. For security purposes, the integer's p and q should be chosen at random, and should be of similar bit-length. Compute $n = pq$. n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1)$ Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ are co-prime. e is released as the public key exponent. Determine d as $d^{-1} = e \pmod{\varphi(n)}$, i.e., d is the multiplicative inverse of e (modulo $\varphi(n)$). This is more clearly stated as solve for d given $de = 1 \pmod{\varphi(n)}$. d is kept as the private key exponent. By construction, $d \cdot e = 1 \pmod{\varphi(n)}$. The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d, which must be kept secret. p, q, and $\varphi(n)$ must also be kept secret because they can be used to calculate d.

Sender transmits her public key (*n, e*) to receiver and keeps the private key secret. Receiver then wishes to send message *M* to sender. He first turns *M* into an integer *m*, such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the cipher text *c* corresponding to

$$C=m^e \bmod n \quad (1)$$

Sender can recover *m* from *c* by using her private key exponent *d* via computing

$$m=c^d \bmod n \quad (2)$$

Given *m*, she can recover the original message *M* by reversing the padding scheme.

In RSA algorithm after generation of the public key pair, N value is convert into binary format then change one bit value 0 to 1 or 1 to 0, convert binary to decimal, generate relative public key pair.

B. *NTRU*

The first version of the system, which was called NTRU [5], was developed in 1996 by mathematicians J. Hoffstein, J. Pipher, and Silverman. That same year, the developers of NTRU joined with D. Lieman and founded the NTRU cryptosystem. The following are part of the parameters for an implementation of NTRU [5].

N is the degree of the polynomial, p as positive integer which the coefficient of a certain product of the polynomial will reduce during the encryption and decryption process. q as a positive integer specify the coefficient of s certain product of the polynomial will be reduced during the encryption and decryption process and also used construction of the public key.

1. Key generation

f and g are two polynomial whose degree is n-1 . Next compute the inverse of the f modulo q and inverse of f modulo p. thus sender computes the polynomial f_p and f_q. With the property is satisfy that $f^* f_p = 1 \pmod p$ and $f^* f_q = 1 \pmod q$. f_p is the part of the private key and f_q is the part of the public key. f_p and f_q. If the given polynomial inverse is not possible then sender will select another polynomial and construct f_p and f_q. Now sender computes the product

$$h=p*f_q*g \pmod q. \quad (3)$$

h is the public key. f_p and f are private key polynomials. Therefore f and f_q pair is the public key, f and f_p pair is the private key polynomial.

After normal key generation change any one coefficient value 0 to 1 0r -1,1 to 0 or -1, 0 to 1 0r -1, because p value is 3 the original coefficients lies between the {-1,0,1} of the source polynomial f. generate the relative public key pair .

2. Encryption

First puts the message in the form of a polynomial m whose coefficient is range between –p/2 and p/2. Select any small random polynomial r. this is binding values which is used to secure the message. User encrypt the data with help of public key then user get the cipher text

$$e=r*h + m \pmod q \quad (4)$$

III. PROPOSED WORK

User can calculate avalanche effect using the given formula. The result shows the how much percentage of the data changed.

$$Avalanche\ \ Effect = \frac{Number\ \ of\ \ flipped\ \ bits\ \ in\ \ cipher\ \ text}{Number\ \ of\ \ bits\ \ in\ \ cipher\ \ text} * 100 \quad (5)$$

The more change in avalanche effect the more secure the system.

- Plain text, public key.
- One bit change in plain text, Public key.
- One bit change in plain text, one bit change in public key.
- Plain text, One bit change in public key.

A. *Original Plain text and Original public key*

This is normal encryption process. In this process use the original key value and original message.

B. *Original Plain text and one bit change in public key*

In this process first generate the public key. Change one bit in the public key then new key is generated, do encryption process. Compare normal encrypted file one bit change key file calculate the avalanche effect using the formula (5).

C. *One bit change in plain text and one bit change in public key*

In this process first generate the public key. Change one bit in the public key then new key is generated and change one bit in the plain text, do encryption process. Compare with normal encrypted file. Calculate the avalanche effect using the formula (5).

D. *One bit change Plain text and key*

In this process first generate the public key. Change one bit in the plain text, do encryption process. Compare with the normal encrypted file and calculate the avalanche effect using the formula (5).

The above process is applied and calculates the avalanche effect value using the equation (5) Result analysis of the above process using NTRU and RSA public key cryptosystems.

Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 or Open Type fonts are preferred. Please embed symbol fonts, as well, for math, etc.

IV. PROPOSED SYSTEM ARCHITECTURE

This evaluation method will compare Avalanche effect of Encrypting plaintext with different cryptographic algorithms. If user can select message and public key then change one bit in message and public key.

- Message and Public key.
- One bit change message, public key.
- One bit change message, one bit change public key.
- Message, one bit change public key.

In the above combinations, apply the cryptography algorithm for encryption.

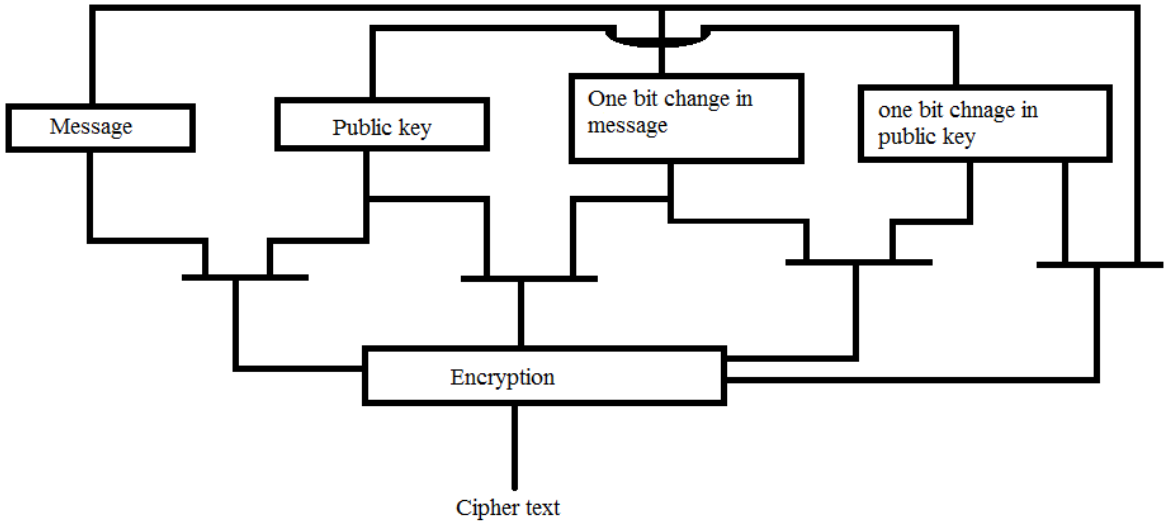


Fig. 1 Encryption process

V. EXPERIMENTAL RESULT

We first find the private/public key pair and change one bit of it so that a new private/public key pair can be formed by both NTRU and RSA.

The parameters for NTRU key generation are:

n=11, p=3, q=32

Note: We have shown one bit of change indicated by bold letters.

A. *NTRU key generation*

f = -1 + **X** + X^2 - X^4 + X^6 + X^9 - X^10

f_p= 1 + 2^X + 2X^3 + 2X^4 + X^5 + 2X^7 + X^8 + 2X^9

f_q= 5 + 9X + 6X^2 + 16X^3 + 4X^4 + 15X^5 + 16X^6 + 22X^7+ 20X^8 + 18X^9 + 30X^10

B. *One bit change in NTRU key generation*

f = -1 - **X** + X^2 - X^4 + X^6 + X^9 - X^10;

f_p=2+2 X+X^5+X^6+2 X^7+2 X^8+2 X^9+2 X^10;

f_q= 15+31 X+10 X^2+14 X^3+16 X^4+3 X^5+16 X^6+14 X^7+12 X^8+22 X^9+6 X^10

TABLE I

ORIGINAL AND ONE BIT CHANGED PUBLIC KEY AND PRIVATE KEYS WITH G=-1X^+1X^2+1X^3++1X^5-1X^8-1X^10

NTRU	***f, f_q***	***h(public key)***
Original Key	f= -1 + **X** + X^2 - X^4 + X^6 + X^9 - X^10 f_q= 5 + 9X + 6X^2 + 16X^3 + 4X^4 + 15X^5 + 16X^6 + 22X^7+ 20X^8 + 18X^9 + 30X^10	8+25X^1+22X^2+20X^3+12X^4+24X^5 +15X^6+19X^7+12X^8+19X^+16X^10
One bit change Key	f = -1 - **X** + X^2 - X^4 + X^6 + X^9 - X^10 f_q= 15+31 X+10 X^2+14 X^3+16 X^4+3 X^5+16 X^6+14 X^7+12 X^8+22 X^9+6 X^10	2+19X^1+18X^2+2X^3+10X^4+24X^5+ 27X^6+23X^7+16X^8+25X^9+26X^10

C. *RSA key generation*
Select any two random numbers p, q.
p = 23, q = 89, n = p * q=2047,
φ (*n*) = (1 - p) * (1 - q) =1936
Select e = 31;
GCD (e, φ (*n*)) =gcd (31, 1936) =1.
Public key is (n, e) = (2047, 31).
Select d, d*e=1(mod φ (*n*)), d=687;
687*31=21297 (mod 1936)
(1936*11+1)=21297
Private Key (n, d) = (2047, 687*)*.

D. *One bit change in RSA key generation*
First we convert part of the public key i.e. 31 into binary format, the value is: 00011111, keeping the value 2047 is unchanged.
Now change any one bit value. 00011101 i.e. last but one bit value is changed. New value of e=29; Now computing one bit change key in RSA.
29 are in between 1 to φ (*n*).
29 and 2047 are co primes.
Now compute relevant private key for decryption.
(d * e) % φ (*n*) = 1. One solution is d = 687.
[(1669 * 29) % 1936 = 1]
48401%1936
((1936*25) +1)%1936=1;
Avalanche key values is
Public Key :(n, e) = (2047, 29).
Private Key :(n, d) = (2047, 1669).
Original and avalanche key values are:
P=23, q=89, n=2047, φ (*n*) =1936.

TABLE III
ORIGINAL AND ONE-BIT CHANGE PUBLIC KEY

RSA	***Public key pair***	***Private key pair***
Original Key	(2047,31)	(2047,687)
One bit change Key	(2047,29)	(2047,1669)

First convert the message into ASCII values and encrypt with public key. The message is changed by one bit. (i.e., select any one character then convert into ASCII value then convert into binary value. Now change the any one bit value 0 to 1 or 1 to 0.) now encrypt the message with public key.

Plain text indicates PT, public indicate PK, one bit change public key is OCPK and one bit change in plain text is OPT.

TABLE IIIII
RESULT ANALYSIS USING NTRU & RSA IN PERCENTAGE

File size	**1 KB**		**2KB**		**5KB**		**10KB**	
	NTRU	**RSA**	NTRU	**RSA**	NTRU	**RSA**	NTRU	**RSA**
PT&OCPK	99.98	99.81	99.99	99.81	99.99	99.97	99.999	99.98
OPT&OCPK	99.98	99.81	99.99	99.81	99.99	99.97	99.999	99.98
OPT&PK	0.0171	0.188	0.008	0.188	0.001	0.022	0.001	0.010

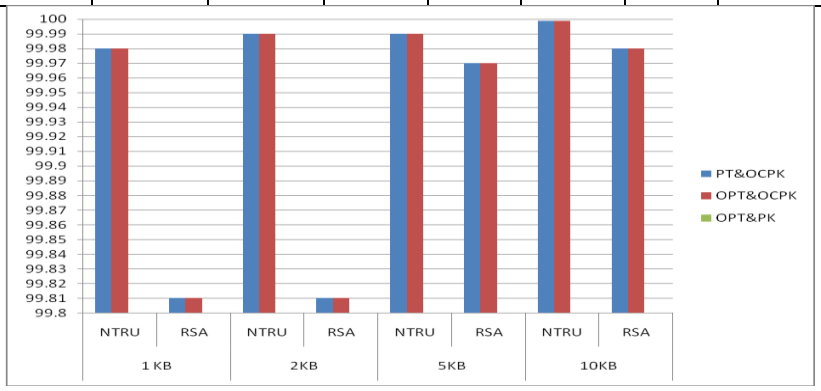


Fig. 2 Graphical Representation Of RSA and NTRU with One-Bit Change for Different File Sizes.

E. *Statistical Result Analysis*

1) Mean

Mean or Average is defined as the sum of all the given elements divided by the total number of elements.

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N}$$

N is the no. of characters of the given file. x_i is the each character value**.**

2) Standard deviation

The S.D shows how much variation or dispersion there is from the average.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N}}$$

N is the no. of characters of the given file. x_i is the each character value and μ is the mean value

TABLE IVV
ANALYSIS OF NTRU USING MEAN AND S.D

Techniques	***MEAN***	***S.D***
PT&PK	17.74	7.70
PT&OCPK	20.65	7.20
OPT&OCPK	20.65	7.20

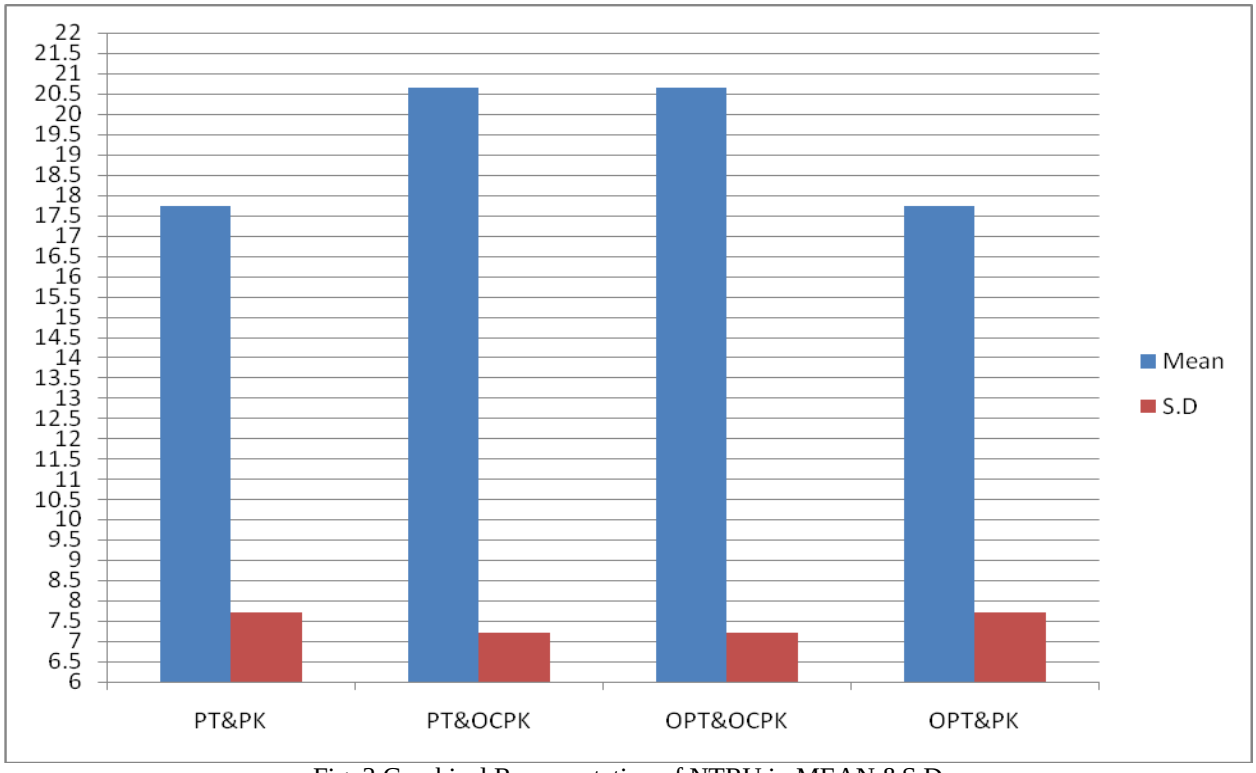
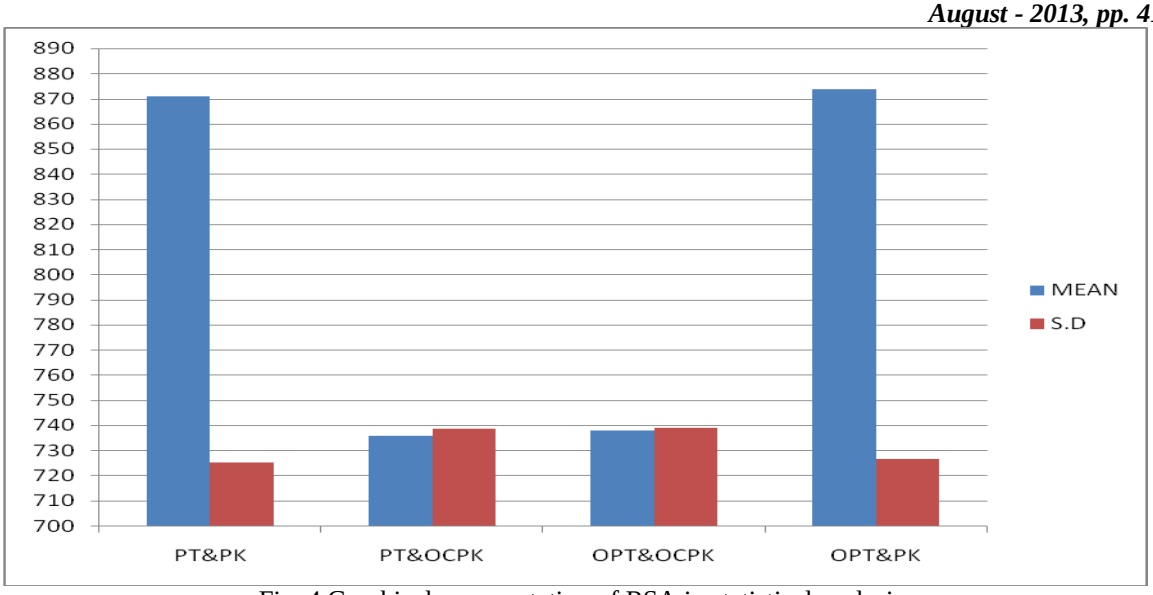


Fig. 3 Graphical Representation of NTRU in MEAN &S.D

TABLE V
ANALYSIS OF RSA USING MEAN & S.D

Techniques	***MEAN***	***S.D***
PT&PK	871.06	725.19
PT&OCPK	735.85	738.67
OPT&OCPK	738.03	739.20
OPT&PK	873.61	726.58



. Fig. 4 Graphical representation of RSA in statistical analysis

3) Correlation Coefficient

The correlation coefficient, denoted by r, is measure strength of the relationship between two variables.

$$r = \frac{1}{(n-1)} \sum \frac{(X-\mu_X)}{\sigma_X} \frac{(Y-\mu_Y)}{\sigma_Y}$$

n is the no. of characters of the given file, μ is the mean value and σ is the standard deviation. If the coefficient result is "+1" then two variables are absolutely identical. If result is "Zero" then two variables are completely uncorrelated. If result is "-1" then two variables are anti correlated.

TABLE VV

CORRELATION COEFFICIENT COMPARISONS IN NTRU AND RSA

Technology	***NTRU***	***RSA***
corr(PT&PK, PT&OCPK)	0.311	0.59
corr (PT&PK, OPT&OCPK)	0.311	0.59
corr (PT&PK, OPT&PK)	0.999	0.989

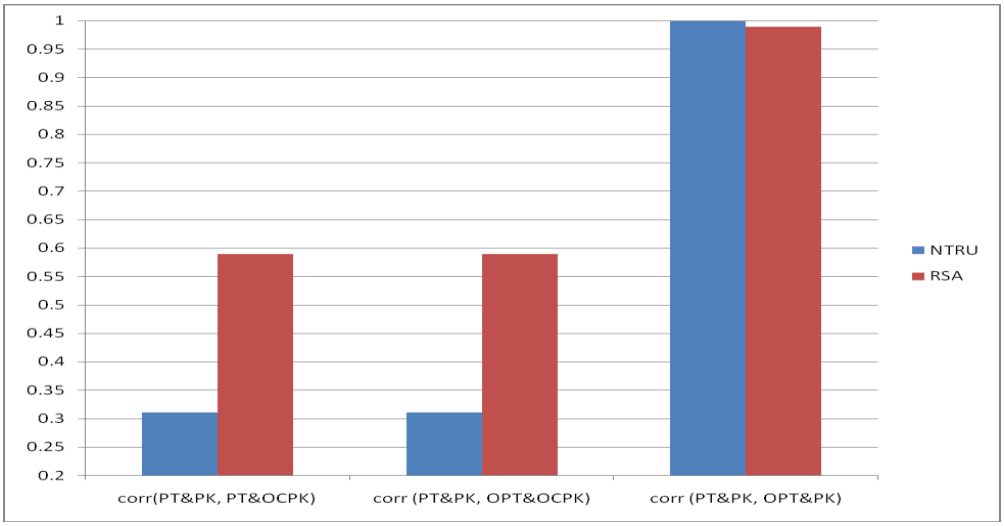


Fig. 5 Analysis of NTRU technique using statistical methods

VI. CONCLUSION

In this paper we propose and perform the analysis to compare the two well known public key cryptosystem is NTRU&RSA by finding avalanche effect and confirmed the result by statistical analysis. We observed that when one bit change in public key is made or both one bit changes in plain text as well as in public key, is made, NTRU seems to be more secure than RSA.As the correlation coefficient in NTRU is close to "0" as compare to RSA. So we can confirm the result.

Acknowledgment

This paper is a part of our M.Tech Project. I am grateful to my friend Ch.VeerendraKumar for giving valuable suggestions, comments and contribution.

REFERENCES

[1] Akash Kumar Mandal1, Mrs. Archana Tiwari "Analysis of Avalanche Effect in Plaintext of DES using Binary Codes" International journal of Emerging trends & Technology in computer science Volume 1, Issue 3, September – October 2012

[2] Rakesh Nayak, Jayaram Pradhan. C.V Sastry "Dependent Private Key Generation in NTRU Cryptosystems" IJCA Special issue on "Network security and cryptography" NSC 2011.

[3] A.Nadeem, "A performance comparison of data encryption algorithms", IEEE information and communication technologies, pp.84-89, 2006.

[4] NeetuSettia. "Cryptanalysis of modern Cryptography Algorithms". International Journal of Computer Science and Technology. December 2010.

[5] www.ntru.com

[6] Cryptography and Network Security: principles and practices', William Stallings, Pearson education first Indian reprint 2003.

P.D.N.V.V Mahesh received his B.Tech degree in computer science and engineering from Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, in 2010 and pursuing his post graduation from Jawaharlal Nehru Technological University, Kakinada in computer science and Engineering. His areas of interest Information Security.

Rakesh Nayak received degree M.Sc, M.Phil in Mathematics MCA from IGNOO. M.Tech in computer science and Engineering From Acharya Nagarjuna university in 2010. He is pursuing Ph.D (submitted), from Berhampur university, Odisha. He is currently working as Associate Professor in Department of Information Technology, Sri Vasavi engineering college. His areas of interest Information Security and Cryptography, Data Mining.

