An efficient Algorithm using Diophantine equation and RSA Algorithm.

Thesis Submitted in partial fulfillment of the requirements for the award of

degree of

Master of Technology
in
Computer Science and Applications

Submitted by
Zakir Husain
(601103030)

Supervised By
Mrs. Sanmeet Kaur
Assistant Professor, SMCA

School of Mathematics and Computer Applications.
THARAR UNIVERSITY
PATIALA – 147004

JUNE 2013

Certificate

I hereby certify that the work which is presented in the thesis entitled, **"An efficient algorithm using Diophantine equation and RSA algorithm"**, in partial fulfillment of the requirements for the award of degree of the Master of Technology in Computer Science and Applications, submitted in School of Mathematics and Computer Applications of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Ms. Sanmeet Kaur and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other university.

Zakir Husain

Zakir Husain
601103030

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Sanmeet Kaur

Ms. Sanmeet Kaur
Assistant Professor
School of Mathematics and Computer Applications

Countersigned by

Dr. Rajesh Kumar
Head,
SMCA,
Thapar University,
Patiala.

Dr. S.K. Mohapatra
Dean Academic Affairs,
Thapar University,
Patiala.

Acknowledgement

I would like to express my deepest appreciation to Mrs. Sanmeet Kaur, my mentor and thesis supervisor for her constant support and motivation. She had been instrumental in guiding me throughout the thesis with their valuable insights, constructive criticisms and interminable encouragement. I am also thankful to Dr. Rajesh Kumar, Head, School of Mathematics and Computer Applications Department and Mr. Singara Singh, P.G. Coordinator for their constant support and encouragement.

I would like to thank all the faculty members and staff of the department who were always there at the need of the hour and provided with all the help and facilities, which I required, for the completion of this work. I express my thanks to my family for their support and affection and for believing in me always. I also want to thank my colleagues, who have given me moral support and their relentless advice throughout the completion of this work.

Zakir Husain

(601103030)

ABSTRACT

Cryptography is used for information security and communication security. In cryptography main study about various algorithms which are used for communication in the daily routine of human life. In the cryptography there are two types of algorithms which are symmetric key algorithms and asymmetric key algorithms. Symmetric key algorithms use only one key but a non symmetric key algorithms use two keys. The first key is known as public key and the second key is known as a private key in asymmetric key algorithm. In the proposed algorithm Diophantine equation with RSA algorithm is applied on the user data to provide more security and reliability.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Data communication is important for human life, and security is also very important for these data. A cryptosystem defines two types of data transformation. The first one is called encryption and the second one is called decryption. Encryption is applied on plain text i.e. the original text is known as plain text by using an encryption key .Similarly the decryption is applied on cipher text the text after applying the encryption .In cryptography main purpose is to apply the security on the plain text or on to cipher text, there are various principles of security, these principal help to identify all possible concepts of security. In this thesis the main work is how applied the Diophantine equation on RSA and noted the results. The proposed algorithm in this thesis is highly secure because the RSA algorithm is applied on two elements of plain text.

1.1 Cryptography

The art of protecting information by transforming into an unreadable format is known as cryptography [1].

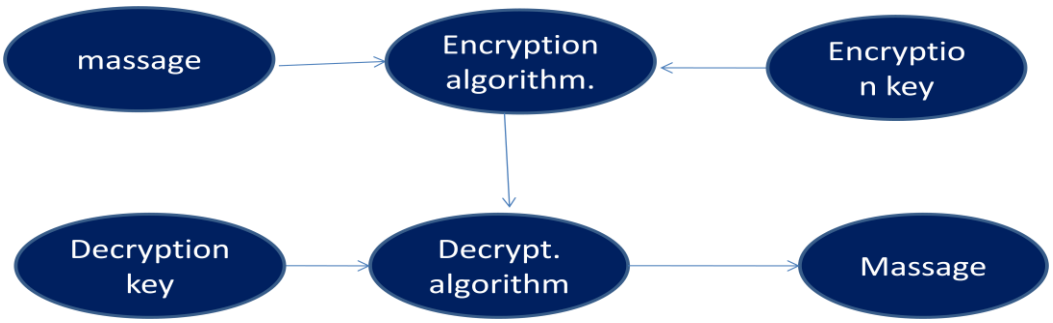


Figure1.1 Cryptography [9]

1.2 Plain text

In cryptography the plain text is ordinary readable text. Thapar University is the best university of north India is known as plain text [8].

1.3 Cipher Text

In the cryptography cipher text is an ordinary non -readable text. $%^&#@ ((&&$$$$$ is known as cipher text [8].

1.4 Encryption Algorithm

The encryption algorithm performs various transformations on plain text [6].

1.5 Secret Key

The secret key is also input to the encryption algorithm .The key is value independent of plain text [8].

1.6 Decryption algorithm

The inverse of Encryption algorithm is known as Decryption algorithm [6].

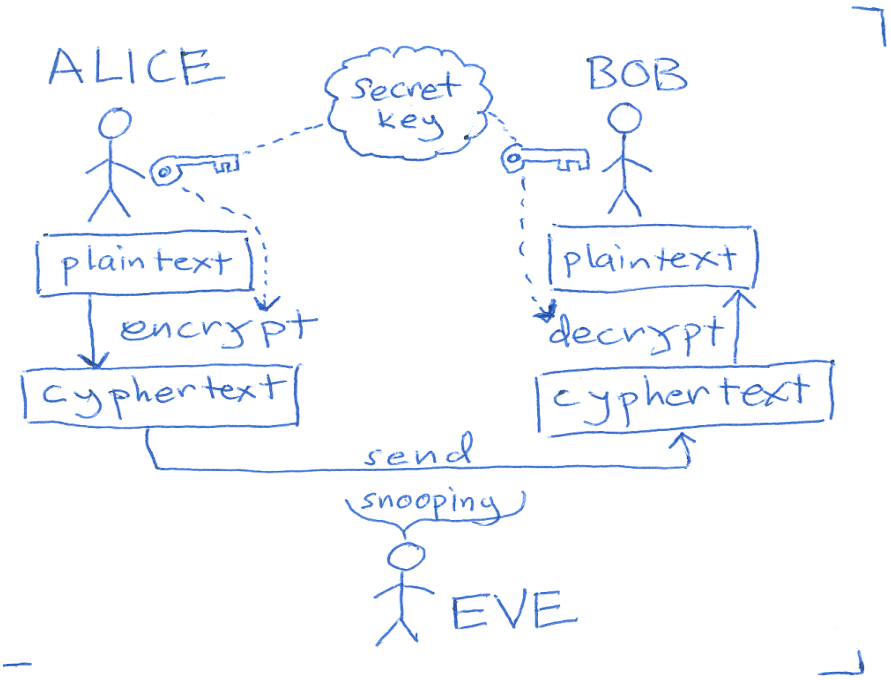


Figure 1.2 Decryption algorithm [9]

1.7 Cryptanalysis

The cryptanalysis is the process to convert the message from a non readable form to readable form without knowing the process how the message converts non readable form [7].

Figure1.3 Cryptanalysis [9]

1.8 Cryptology

The combination of cryptography and cryptanalyst is known as cryptology.

Figure1.4 Cryptology [8]

In cryptography there are various kinds of ciphers, means two kinds of ciphers the first kind is symmetric key ciphers and a second kind is non symmetric key ciphers [6].

Chapter2

LITERATURE SURVEY

2.1 Introduction

The encryption is a technique which is used to change the readable text into non readable text and decryption is used for change non readable text into readable text. There are mainly two methods for encryption and decryption. The first one is a symmetric key encryption and second is asymmetric key encryption .The symmetric key encryption is used as single key, but the asymmetric key encryption uses two keys. In symmetric key technique there are various ciphers and algorithms for encryption and decryption .These are stream and block ciphers .In stream cipher we change single bit, but in block ciphers we change one block of readable text. The stream ciphers are hill cipher, play fair cipher, Caeser cipher, and mono alphabetic ciphers. The asymmetric key encryption uses two keys. The first key is known as public key, and the second key is a private key .In nonsymmetrical key encryption algorithm sender and receiver both know about the public key but both of them does not know the private key of each other.

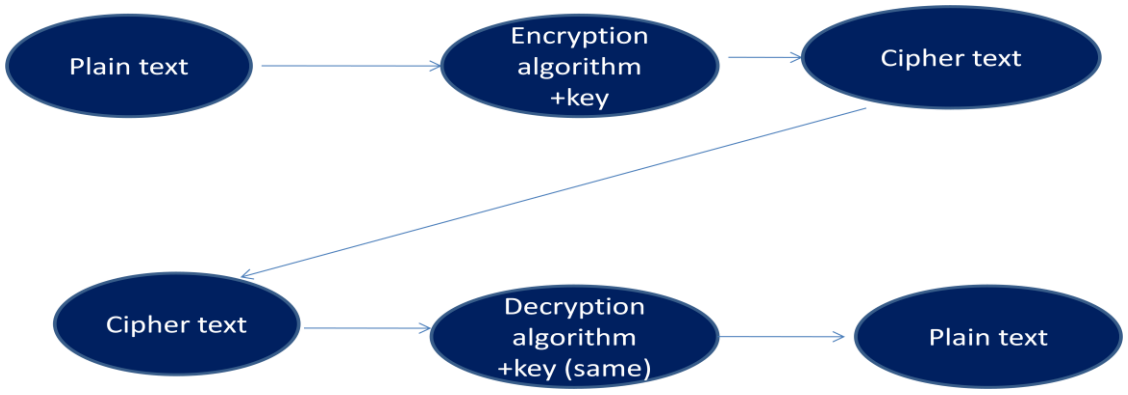


Figure2.1 Encryption or decryption process in symmetric key ciphers [10]

2.2 Caeser Cipher

The Caeser cipher involves replacing each letter of the alphabet with the latter standing three places of further down the alphabet. The algorithm can be expressed as follows. For each plain text letter p, substitute the cipher text latter c.

C =E (p) = (p + k) mod (26), where k takes the values between1 to 25.

P =D(C) = (C - k) mod (26) [9].

2.3 Hill Cipher

The hill cipher uses the system of linear equations. This encryption algorithm takes m successive plaintext letters and substitutes for them m cipher text letters. The substitution is determined by m linear equations in which each character is assigned a numerical value (a = 0, b = 1, z = 25). For, m=3 the system can be described as;

$$c1 = (k11p1 + k12p2 + k13p3) \bmod 26$$

$$c2 = (k21p1 + k22p2 + k23p3) \bmod 26$$

$$c3 = (k31p1 + k32p2 + k33p3) \bmod 26$$

This can be expressed in terms of row vectors and matrices

$$(c1\ c2\ c3) = (p\ p2\ p3) \times \text{key metrics.}$$

$$\text{Key matrix} \quad \begin{matrix} k11\ k12\ k13 \\ k21 k22\ k23 \\ k31\ k32\ k33. \end{matrix}$$

That is

$$C = PK \bmod 26$$

Where C and P are row vectors of length 3 representing the plaintext and cipher text, and K is 3x3 matrix show the encryption key. Operations are performed mod 26.In general terms, the Hill

system can be expressed as C = E (K, P) = PK mod 26 P = D (K, C) = CX inverse of K mod 26 = PK x inverse of K = P [7].

2.4 Play -fair Cipher

The Play-fair cipher uses a 5 square. This method permits the encryption of 5×5 =25 letters. This is a bit of a problem because our alphabet contains 26 letters so, two letters are combined in the same cell of the square –usually I and J. Beginning with the first row, enter the keyword from left to right skipping letters previously used and continuing on to the second (or third or fourth or fifth) row if necessary. After entering the keyword, the remaining letters of the alphabet are entered in order. Here is the square we would obtain.

G	A	L	O	I/j
S	B	C	D	E
F	H	K	M	N
P	Q	R	T	U
V	W	X	Y	Z

Table 2.1 play-fair cipher [9]

- **The rules for encryption**

1- If both letters of the diagram lie in the same row, then each letter is encrypted by the letter immediately to its right (cycling back to the first letter in the row, if needed).
2 -If both letters of the diagram lie in the same column, then each letter is encrypted by the letter immediately below it (cycling back to the first letter in the column, if needed).

3-In the remaining case, each letter is exchanged by the letter at the intersection of its row and the other letter's column. Think of drawing a rectangle with the two given letters at diagonally opposite corners, the Cipher text letters are at the other corners with the cipher text letter being in the same row as a corresponding plaintext letter [8].

2.5 Mono alphabetic Cipher

In Mono alphabetic cipher the same plaintext letters are always replaced by the same cipher text letters. Mono, meaning one, indicates that each letter has a single substitute to construct a mono alphabetic cipher to create some ordering of the alphabets [10].

Plain text alphabet: a b c d e f g h i j k l m n o p q r s t u v w x y

Cipher text alphabet: S O M E R D I N G X H B V L T U J W K Y Z F A C P

2.6 Poly alphabetic Cipher

Each plaintext character is mapped to several cipher text characters .Vigenere cipher and Beaufort cipher is the example of a poly alphabetic cipher. This cipher uses multiple one – character keys. Each of keys encrypts one plain text character .The first key encrypts the first plaintext latter , and second key encrypts the second plain text character and so on. After all keys are used, they are recycled. The main properties of poly alphabetic are (a) .It uses a set of related mono alphabetic cipher rules. (b) It uses a key that determines which rule is used for which transformation [9].

2.7 One time pad Cipher

The Verman cipher is also known as one time pad cipher .The one time pad cipher have the following properties.

(a)Treat each plain text character as a number in an increasing order, that is A=0, B=1…….Z=2.

(b) Do the same for each alphabet of the input text.

(c) Add every one number corresponding to the plain text to be corresponding each cipher text alphabet.

(d) If sum produced greater than 26, subtract 26 from it.

(e) Translate every number of sums back to the corresponding alphabet; this gives the result of cipher text [9].

2.8 Data Encryption Standard (DES) Algorithm

The DES (Data Encryption Standard) algorithm is the most used encryption algorithm in the world of cryptography. For many years, and among many people, DES has been synonymous. And nevertheless the recent coup by the Electronic Frontier Foundation in creating a $220,000 machine to crack DES-encrypted messages, DES will live on in banking for years to come through a life- extending version called "triple-DES." Since the creation of DES, many other algorithms (recipes for changing data) have emerged which are based on design principles similar to DES algorithm, to understand the basic transformations that take place in DES, we will find it easy to follow the steps involved in these more recent algorithms. But first a bit of history of how DES came about is appropriate, as well as a look toward the future. DES works on bits, the 0s and 1s common with digital computers. Each group of four bits makes up a hexadecimal, or base 16, number. Binary "0000" is equal to the hexadecimal number "0", binary "0001" is equal to the hexadecimal number "1", "0100" is equal to the hexadecimal number "8", "1010" is equal to the hexadecimal number "A", and "1110" is equal to the hexadecimal number "E". DES works by encryption groups of 64 message bits, which is the same as 16 hexadecimal numbers. To do the encryption, DES uses "keys" where are also apparently16 hexadecimal numbers long, or apparently 64 bits long. However, every 8th key bit is discarded in the DES algorithm, so that the main key size is 56 bits. But, in any case, 64 bits (16 hexadecimal digits) are the round number upon which DES is organized. Again if we take the plaintext message "8787878787878787", and encrypt it with the DES key "0E329232EA6D0D73", we end up with the cipher text "0000000000000000". If the cipher text is decrypted with the same secret DES key "0E329232EA6D0D73", the result is the original plaintext "8787878787878787". From the example this is clear that our plaintext was exactly 64 bits long. The same would be true if the plaintext happened to be a multiple of 64 bits. But most messages will not fall into this category. They will not be an exact multiple of 64 bits (that is, an exact multiple of 16 hexadecimal numbers). For example, take the message "Your lips are smoother than Vaseline". This plaintext message is 38 bytes long. So this message must be padded with more extra bytes at the tail end of the encryption. Once the encrypted message has been decrypted, these extra bytes are ignored. There are different padding schemes. Here we will just add 0s at the end, so that the total message is a multiple of 8 bytes (or 16 hexadecimal digits, or 64 bits). The plaintext message

"Your lips are smoother than Vaseline" is, in hexadecimal. "596F7572206C6970 732061726520736D 6F6F746865722074 68616E2076617365 6C696E650D0A"We then pad this message with some 0s on the end, to get a total of 80 hexadecimal digits."596F7572206C6970 732061726520736D 6F6F746865722074 68616E2076617365 6C696E650D0A0000".If we encrypt this plaintext message 64 bits (16 hexadecimal digits) at a time, using the same DES key "0E329232EA6D0D73" as before, we get the cipher text."C0999FDDE378D7ED 727DA00BCA5A84EE 47F269A4D6438190 9DD52F78F5358499828AC9B453E0E653".This is the hide code that can be transmitted. Decrypting the cipher text restores the original message "Your lips are smoother than Vaseline"

2.9 Triple-DES

Triple-DES is just DES with two 56-bit keys applied. Given a plaintext message, the first key is used to DES- encrypt the message. The second key is used to DES-decrypt, the encrypted message. The twice-scrambled message is then encrypted again with the first key to yield the final cipher text. This three-step procedure is called triple-DES .Triple-DES is just DES has done three times with two keys used in a particular order. (Triple- DES can also be done with three separate keys instead of only two. In either case the resultant key space is about power (2,112) .

2.10 DES Weaknesses

In DES the key consists in a 56 bits provided a key space of 2^56 elements. The cryptanalysis will obtain the solution after 2^55 trails. The Data Encryption Standard (DES) is a block cipher that uses a symmetric key. DES is now considered to be insecure for many applications. This is chiefly due to the 56-bit key size being too small, in January 1999, a group of computer experts collaborated to publicly break a DES key in 22 hours and 15 minutes. There are also other analytical results which demonstrate theoretical weaknesses in this cipher, although they are infeasible to mount in practice. The algorithm is believed to be practically secure in the form of the Triple DES. Triple DES uses a key group which comprises three DES keys, K1, K2 and K3, the length of each key is 56 bits. Each triple encryption encrypts one block of 64 bits of given data. The strongest application of Triple DES is to uses 3 independent keys so that we get 3 x 56 = 168 independent key bits, but it is not uncommon to simplify the operation by just using two keys, making K1 and K3 the same but making sure K2 is independent of K1. This second

Keying option provides less security than using 3 independent keys, with 2 x 56 = 112 key bits. This second option is stronger than simply DES encrypting twice, e.g. with K1 and K2, because it protects against meet in the middle attacks. In recent years, the DES cipher has been superseded by the Advanced Encryption Standard (AES).

2.11 Asymmetric Key Cipher

The ciphers in which the use two keys for encryption and decryption is known as Asymmetric key ciphers, for example RSA algorithm.

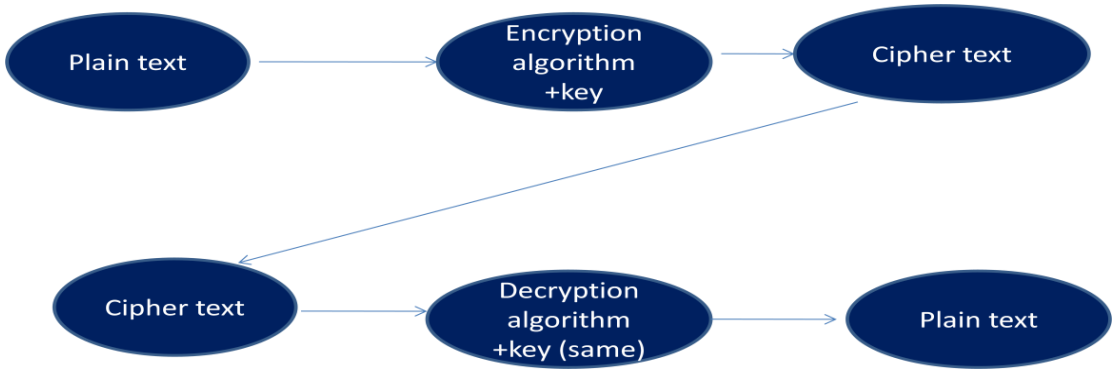


Figure 2.2 Asymmetric Key Ciphers

2.12 RSA Algorithm

RSA algorithm was developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT. Since then, the Rivest-Shamir-Adleman (RSA) scheme has become the most widely accepted and implemented general-purpose approach to public-key encryption, and RSA supports Encryption and Digital Signatures Most widely used public key algorithm, Gets its security from integer factorization Problem Relatively easy to understand and implement Patent free .

2.13 RSA Usage

RSA is used in security protocols such as, IP data security, transport data security PGP - email security SSH - terminal connection security SILC - conferencing service security.

2.14 Implementation Tools

In order to implement RSA you will need arbitrary precision arithmetic, Pseudo Random Number Generator, (PRNG) Prime number generator, Difficulty of implementation greatly depends of the target platform, application usage and how much of the tools you need to implement from scratch.

2.15 Arbitrary Precision Arithmetic

Used to handle large numbers (arbitrary in length) Provides optimized implementations of arithmetic operations, such as modular computation and exponential computation .If you need to implement these yourself the task of implementing RSA is usually large. RSA operations will use arbitrary precision of-22 arithmetic (encryption, digital signatures).

2.16 RSA Schemes

RSA Encryption or decryption scheme encryption is done always with public key. Public key must be authentic to avoid man-in-the middle attacks in protocols. Verifying the authenticity of the public key is difficult, when using certificates a trusted third party can be used. If certificates are not in use then some other means of verifying is used (fingerprints, etc). The message to been encrypted is represented as number m, $0 < m < n - l$. If the message is longer it needs to be spitted into smaller blocks .RSA encryption and decryption are not used as much as RSA digital signatures. For encryption usually symmetric algorithms are used instead since they are faster. Sometimes combination of both symmetric key encryption and public key encryption are used to make it faster (PGP). In RSA digital signatures verification scheme, digital signatures are always computed with private key. This makes them easily verifiable publicly with the public key. The raw message m is never signed directly. Instead it is usually hashed with hash function and the message digest is signed. This condition usually also means that the message m in fact is not secret to the parties so that each party can compute the message digest separately. It is also

possible to use so “called redundancy function instead of hash function”. This function is reversible which makes it possible to sign secret messages, since the message can be retrieved by the party verifying the signature. The hash function is often used Computing signature. First run the message through the hash function (or redundancy function).Since the signatures are always verified with public key the public key must be obtain and verified before the signature can be reliably verified.

2.17 Methodology of RSA Algorithm

RSA algorithm has two keys, the first key is a public key and another key is the private key. First of all, two large and distinct prime numbers p and q must be generated. The product of these numbers, we are known as n is a component of the public key. It must be large enough such that the numbers greater than 10154. We then generate the encryption key “e” for plain text which must be relatively prime number to the number z = (p - 1) (q -1). Then create the decryption key d for cipher text such that d*e mod z = 1. We now have both the public and private keys. For any given public key e we find out private key d such that

(e* d) mod (z) = 1.Where z = (p-1) * (q-1).

Then for any given plain text” PT” the cipher text can be calculated by

CT = power (PT, e) mod (n).

PT =power (CT, d) mod n.

Numerical example of RSA algorithm

Choose p = 3 and q = 11

Compute n = p * q = 3 * 11 = 33

Compute z = (p - 1) * (q - 1) = 2 * 10 = 20

Choose e such that 1 < e < z and e and n are Co prime. Let e =7

Compute a value for d such that (d * e) % z = 1. One solution is d = 3 [(3 * 7) % 20 = 1]

The public key is (e, n) => (7, 33)

Private Key is (d, n) => (3, 33)

Suppose that plain text P T =3.

The encryption is = power (27, 7) mod 33 = 3

The decryption =power (3, 3) mod 33 = 27

Example 2

Let p = 37; q = 43; n = p*q = 1591; e=575.

z= (p-1) * (q-1) =20.

Then (e*d) mod (z) =1

Then d=71

Let the plain text is Thapar and supposed that,

a=0,b=1,c=2,d=3,e=4,f=5,g=6,h=7,i=8,j=9,k=,10,l=11,m=12,n=13,o=14,p=15,q=16,r=17,s=18,t=19,u=20,v=21,

w=22, x=23, y=24, z=25.

Then

T	h	a	p	a	r
19	7	0	15	0	17

So that first plain text element is 197, and second plain text element is015, and so on……

Then

CT = power (PT, e) mod (n).

CT =power (197, 575) mod (1591) =1588.i.e

PT =power (CT, d) mod (n).

PT =power (1588, 71) mod (1591) =197.

CT =power (15, 575) mod (1591) =153.

PT =power (153, 71) mod (1591) =15.

CT =power (17, 575) mod (1591) =1171.

PT =power (1171, 71) mod (1591) =17.

So that the cipher text is15881531171. i.e. Piipdbbhb.

The RSA algorithm is highly secure only the possibility is brute force attack. But if we increase the key size the brute force attack is not possible. RSA algorithm contains two large prime numbers .At first we generate two large prime numbers with the help of the prime number theorem.

2.18 How secure is RSA?

The RSA algorithm is indeed among the strongest, but can it withstand anything, certainly nothing can withstand the test of time. In fact, no encryption technique is even perfectly secure from an attack by a realistic cryptanalyst. Methods such as brute-force are simple but lengthy and may crack a message, but not likely an entire encryption scheme. We must also consider a probabilistic approach, meaning there's always a chance someone may get the one key out of a “million". So we don't know how to prove whether an encryption scheme is unbreakable. If we cannot prove it, we will at least see if someone can break the code. This is how the NBS standard and RSA were essentially certified. Despite years of attempts, no one has been known to crack either algorithm. Such a resistance to attack makes RSA secure in practice. There is no one techniques exist which can give the proof that the encryptions scheme is completely secure. The test is any one person can think to break this scheme. The IBM spent very large time to break this method but IBM cannot break this scheme, because the approaches to break this scheme is very hard at least the factoring n .But large number n can easily break by using the Legendre factoring algorithm. [12]

2.18.1 Format factoring Algorithm

For a given polynomial f (x) has a factor (x-k) if and only if f (k) = 0, and according to Legendre theorem p is a prime number and x belongs to z (set of integers) and x is not equal to zero then power(p, m) divides x ,where m is integer. The cryptanalyst can try to find decryption key with the help of encryption key so that is clear the plain text is not completely secure .Again factoring in n is impossible because the product of two prime numbers may be a prime number or a composite number.

2.18.2 Computing z without factoring n

If attacker attacks on plain text and try to find out the value of z but practically this is impossible again if cryptanalyst factor and by using z this is also impossible .At first p + q is find out from n then z

2.19. RSA Cryptosystem modulo Power (p, k)*q

The RSA cryptosystem modulo power (p, k).q is new proposed algorithm. This proposed algorithm defined as, Generate two random number p and q and let n =power (p, k).q Compute A =LCM (p-1, q-1) and find e, d such that (e .d) = 1mod (A), and GCD (e, p) =1Then (e, n) are public keys, and d, p, q are the secret keys. Let M be the plaintext then encrypt the plaintext by the equation C=power (M, e) mod n, and for decryption M_p = M mod power (p, k), and M_q=M (mod q) using the secret key *d, p, q.* The plaintext M can be recovered by the Chinese remainder theorem. Here, M_q is computed by $M_q = C\ d$ (mod q) and M_p is computed by fast algorithm. In the given algorithm the running time is slower than that of the method using the Chinese remainder theorem for n = p*q so there are no significant advantages in using the modulus power (p, k)*q [12].

2.20 Research Issues and Challenges for Multiple Digital Signatures

In multiple digital signature schemes, Harn used a scheme on RSA, which is based on digital signature. This scheme is used for encryption and decryption of any given plain text.

2.20.1Harn's Scheme

The scheme is based on RSA algorithm. Let p and q are two large primes. Computes n = p. q and chooses e and d such that e. d mod (p - 1) (q - 1) = 1. (e, n) is a public key and d is a private key. The sender generates a digital signature S by using RSA algorithm, where S =h. power (m, d) mod n and h (.) is a public one-way hash function. Sender sends (m, S) to receiver. After receiving them, receiver can verify the correctness of the signature by checking the equation h (m) = power(S, e) mod n. If it holds, receiver can confirm that the digital signature S is signed by sender. Now, assume that sender wants to send t messages m1, m2……mt and multiple digital signatures S1, S2……. St to receiver. The multiple digital signatures are created by the above RSA algorithm. Then, sender sends (mi; Si), i = 1,2…… t, to receiver. Upon receiving the t messages m_1, m_2……..m_t and multiple digital signatures S1, S2 …… St, receiver can verify the correctness of these multiple digital signatures on messages m_1, m_2, ……m_t sing sender's public key e in the following equation

$$(\prod_{i=1}^{t} S_i)^e = \prod_{i=1}^{t} h(m_i) \bmod n;$$

If the above equation holds, receiver can confirm that the multiple digital signatures S1…….St are signed by sender. It is seen that receiver can batch verify multiple digital signatures. Therefore, Harn's scheme is more efficient to batch verifying multiple digital signatures.

2.20.2 The Weakness of Harn's Scheme

Hwang et al. proposed two attacks on Harn's scheme. They showed that a singer can forge the multiple digital signatures and pass the batch verification equation. Then, the signer can deny the digital signatures. It cannot achieve the property non-repudiation. The attacks are briefed as follows, sender sends him forged pairs (mi; Si), i = 1,2...; t to receiver, where Si = h(power (mf(i)),d) mod n, i = 1,2…….; t; f(.) is a one to one and onto function such that f(i) = j, i = 1,2…….. t and j = 1,2…… t. Upon receiving forged pairs (mi; S0 i), receiver can pass the batch verification and receiver can confirm that the multiple digital signatures S1 ……; St are signed by sender [13].

2.21 Improvements in RSA Algorithm

The most widely used RSA is based on arithmetic modulo large numbers which will lead to slow in operation in RSA decryption .The Encrypt Assistant Multi-Prime RSA(EAMRSA) will speed up the decryption process by reducing modules and private exponents in modular exponentiation.

2.21.1Encrypt Assistant Multi-Prime RSA

A new variant of RSA is called EAMRSA (Encrypt Assistant Multi-Prime RSA) will effectively combines Multi-Prime RSA and RSA-S2system. It can obtain a higher speedup than the basic RSA and the above two RSA variants. The variant also has obvious parallel characteristics and is easy to be implemented in parallel. The RSA-S2 system is The client randomly generates an integer vector D=(dl,d2, ….. ,dk) and two binary vectors f = (f 1,f 2,……..f k) and g = (gl,g2,⋯,g k) such that dp = fi dI mod(p-l) ,dq= fi dI mod(q-l) . The client sends n, D and x to the server. The server computes Z = (Z1, Z2… Z k) and sends Z back to the client, where z i= x di (mod n). The client obtains M by computing M using formula. It includes, (1) it takes security (2) parameters with additional 3 Parameters b, k and c (3) Encryption (4) Decryption (The Chinese Remainder Theorem is used).

2.22 Comparison between RSA Algorithm and DES Algorithm

1-The DES algorithm is symmetric key algorithm but RSA is non symmetric key algorithm.

2-DES algorithm depends on key size and key size is the weak point of DES algorithm.RSA is also depends on key but the key is not weak point of RSA algorithm.

3-The RSA is highly secured, because in RSA algorithm ,"n" is product of two prime numbers and with the help of "n" we get cipher text and then from cipher text we find out plain text. So in RSA it is impossible to find the value of n with the help of two prime numbers but in the DES algorithm there are 2^56 number of operations to get plain text from cipher text.

4-The triple DES algorithm contains three keys but the value of one key depends on other two keys, so that the key dependency is the weak point of triple DES algorithm .The RSA algorithm

is also key dependent algorithm but it is impossible to break RSA from keys, because the value of public key depends on private key and n(product of two prime numbers) and the factorization of ‘n” is not easy.

5-In RSA algorithm the public key is the relative prime number of (p-1)*(q-1) ,and the private key is the multiplicative inverse of public key but in DES algorithm this is not possible because single key algorithm.

6-The DES algorithm depends on rounds but RSA algorithm does not depend on rounds.

2.23 Diophantine equation of two variables

Diophantine equation is used to find the general solutions and particular solution for non homogeneous linear equation of two variables. Suppose that the non homogeneous linear equation is a*x + b*y =c. Then this equation has (n! /r! (n-r)!) Basic solutions, where n =number of variables and r is the number of equations but in Diophantine equation we do not finds the basic solutions, we find out the particular solution or general solutions. Suppose that a*x +b*y =c is an equation and d = g c d (a ,b) , if d divides c the equation have an infinite number of solutions or if d does not divides c the equation does not have any solution .

Particular solution: x0 = (c/d) s and y0 = (c/d) t.

General solutions: x = x0 + k* (b/d) and y = y0 -k* (a /d) where k = 0, 1, 2, where s and t can be calculated by from **Extended Euclidean Algorithm** .Suppose that 21x-14y=35 is a liner equation, where x, y is plain text elements and d=7 which divides the Diophantine constant. So the equation is 3x-2y=5. (a1x+b1y=c1 suppose), a1=3, b1 = -2, c1 =5.Apply Extended Euclidean Algorithm

a1*s +b1*t= g c d (a1, b1).So that s=1, t = 1, g c d (a1, b1) =1.Then particular solution is x0 = (c/d) *s, y0 = (c/d) *t.

Particular solution x0 =5*1=5, y0 =5*1=5.and **the general solution** are x0=k*b/d, where

k=0,1,2,3,4,5,6,7,8,9…….

x=5+1* (14/7) =7.

Y=y0-k* (a /d). =5+ 1*(21/7) =8.So that for the different values of we can find out an infinite number of solutions.

2.24 Diophantine equation of three Variables

When there are two Diophantine equations with three variables then at first we remove one variable from these two equations. Now the two Diophantine equations must be reduced into single equation with two variables then after applying Diophantine equation of two variables we can find out the particular solution and also general solution.

Chapter 3

PROBLEM STATEMENT

3.1Problem statement

In this research work an algorithm has been proposed to make RSA more secure .In this proposed algorithm Diophantine equation has been combined along with RSA algorithm. In this research work, a new algorithm is generated with the combination of RSA algorithm, Diophantine equation and Euclidian algorithm. First, Diophantine equation is applied on the data to be send and then its outputs is taken as inputs to the RSA algorithm. Euclidian algorithm is used in between the process of the Diophantine equation. On the sender side, Equation "ax+by=c" is used. In this equation 'a' and 'b' are constants and 'x' and 'y' are user datum. Using this equation, Diophantine constant 'c' is calculated. This 'c' is taken as an input for the RSA algorithm. Cipher text that is generated from the RSA algorithm is send to the receiver.

On the receiver side, RSA algorithm is applied on the cipher text that is send from the sender. RSA algorithm will generate Diophantine constant 'c' as an output. Apply Diophantine equation and Euclidian algorithm on the Diophantine constant 'c' and then find the values of 'x' and 'y' that satisfy the Diophantine equation "ax+by=c". These 'x' and 'y' are the datum that is send by the sender.

Chapter 4

PROPOSED ALGORITHM AND METHODOLOGY

4.1 Proposed algorithm

The Diophantine equation is a*x +b*y =c (suppose).where a, b, c are constants and x, y are the two elements of given plain text, at first we find an integer d, which d divides a, b and c.

Then the equation becomes as a1*x +b1*y =c1.

Apply **Extended Euclidean Algorithm**.

a1*s +b1*t =g c d (a1, b1), where s, t are arbitrary constants.

Then a particular solution of Diophantine equation is x0 = (c /d) *s, and y0 = (c/d)*t and the **general solutions** (infinite number of solutions) → x = x0 +k*(b/d).

General solutions (infinite number of solutions) → y = y0 -k*(a /d).

Then we apply the RSA algorithm on Diophantine constant's "c", which is now the plain text for the RSA algorithm.

Step 1 for two given large prime numbers p and q.

Step 2 we calculate n, where n =p*q and then for a given public key e we find out private key d such that (d*e) mod (p-1)*(q-1). Where d is the multiplicative inverse of e.

Then cipher text =ct → power (c, e) mod (n).

The plain text =pt →power (CT, d) mod (n).

Step 3 form plain text we can find out the Diophantine constant c.

Now c is the linear combination of Diophantine constants a, b and two variables x, y so c can be written as

C=a*x+ b*y. Now for different values of x, y if linear equation is satisfied then x, y is original plain text and if not satisfied then x, y is not plain text.

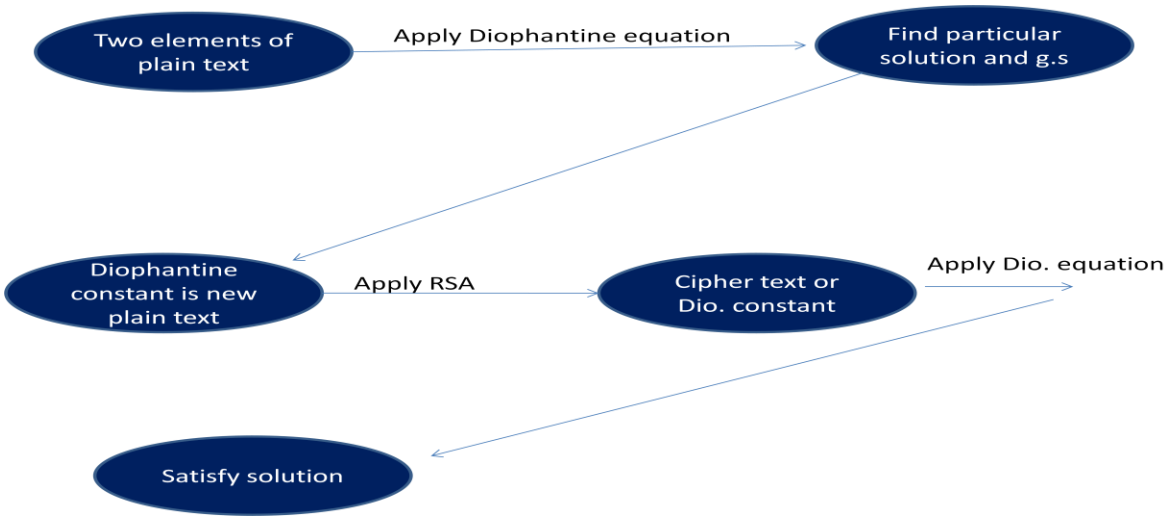


Figure 4.1 Proposed Algorithm

4.2 Methodology

Suppose that 21x-14y=35 be any give equation. Where x, y is plain text elements

d=7 which divides the Diophantine constants.

So the equation is 3x-2y=5. (a1x+b1y=c1 suppose).

a1=3, b1 = -2, c1 =5.

Apply Extended Euclidean Algorithm.

a1*s +b1*t= g c d (a1, b1).

So that s=1, t = 1, g c d (a1, b1) =1.

Then particular solution is x0 = (c/d) *s, y0 = (c/d) *t.

Particular solution: x0 =5*1=5, y0 =5*1=5. Genera solutions are

x=x0+k* (b/d),where k=0,1,2,3,4,5,6,7,8,9……

x=5+1* (14/7) =7.

y=y0-k* (a /d). =5+ 1*(21/7) =8.

So that for the different values of k we can find out an infinite number of solutions

Now my plain text is 35.

Apply RSA algorithm for given plaintext 35.

Suppose that p=7, q=17.

n=p*q=7*17=119.

Public key e= 5 (suppose) which is not the factor of 119

Find private key d such that (e*d) mod (p-1) * (q-1).

So that we find other plain text which is 35, where d is the multiplicative inverse of e.

So that d=77.Apply RSA algorithm.

Cipher text ct=power (PT, e) mod (n) = (52521875) mod 119 = 35.

Plain text pc=power (CT, d) mod (n) =35. Again c is a linear combination of Diophantine constants a, b and plain text elements, i.e c= ax +by, 35=ax +by, let us take a=7, b=14, then

35=7x +14y, and d=7, a1=1, b1=2, then a1*s+b1*t= g c d (a1, b1), s=3, t=-1,

x0= (c/d)*s =5*3=15.

y0 =5*-1=-5.

Then

35 =ax +by.

35=15*7 + (-5)*14.

75-70=5. That is the equation is not satisfied.

Again suppose 35 =21x +56y, d=7. Then 5=3x+8y, a1=3, b1 =8 and s=-5, t=2, so that the equation becomes a1*s+b1*t= g c d(a1, b1)=1.

And particular solution x0=5*-5=-25. Y0=5*2=10, then 35=21*(-25)+10*56=560 -525=35

But -25 and 56 is not possible because -25 is a negative number.

Again 35=21x +14y then apply similar manner we find out particular solution for x0=5, and particular solution for y0=5.

So that x, y is plain text.

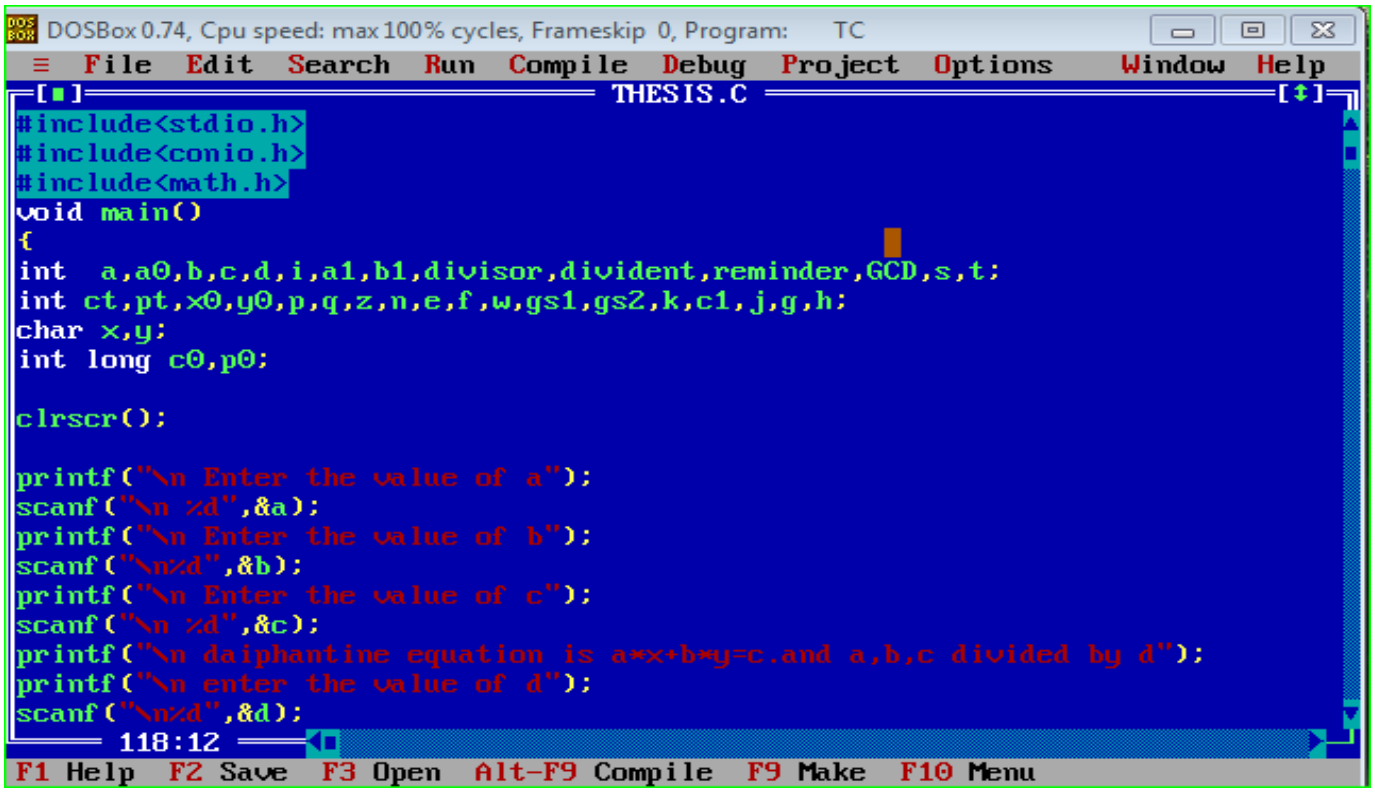
Chapter 5

IMPLEMENTATION OF ALGORITHM AND RESULTS

This is clear that RSA algorithm is a highly secure algorithm .The RSA algorithm provides the security from intruders, when one person sends the documents to other people, then we use the RSA algorithm for the security of documents from intruders, and the public key dictionary contains the pair of (e, n), where e is a public key and n is the multiplication of two prime numbers p and q. The users wishing to send a secret message to one another refer to the dictionary to find out these parameters. So that the dictionary may be arranged as follows.

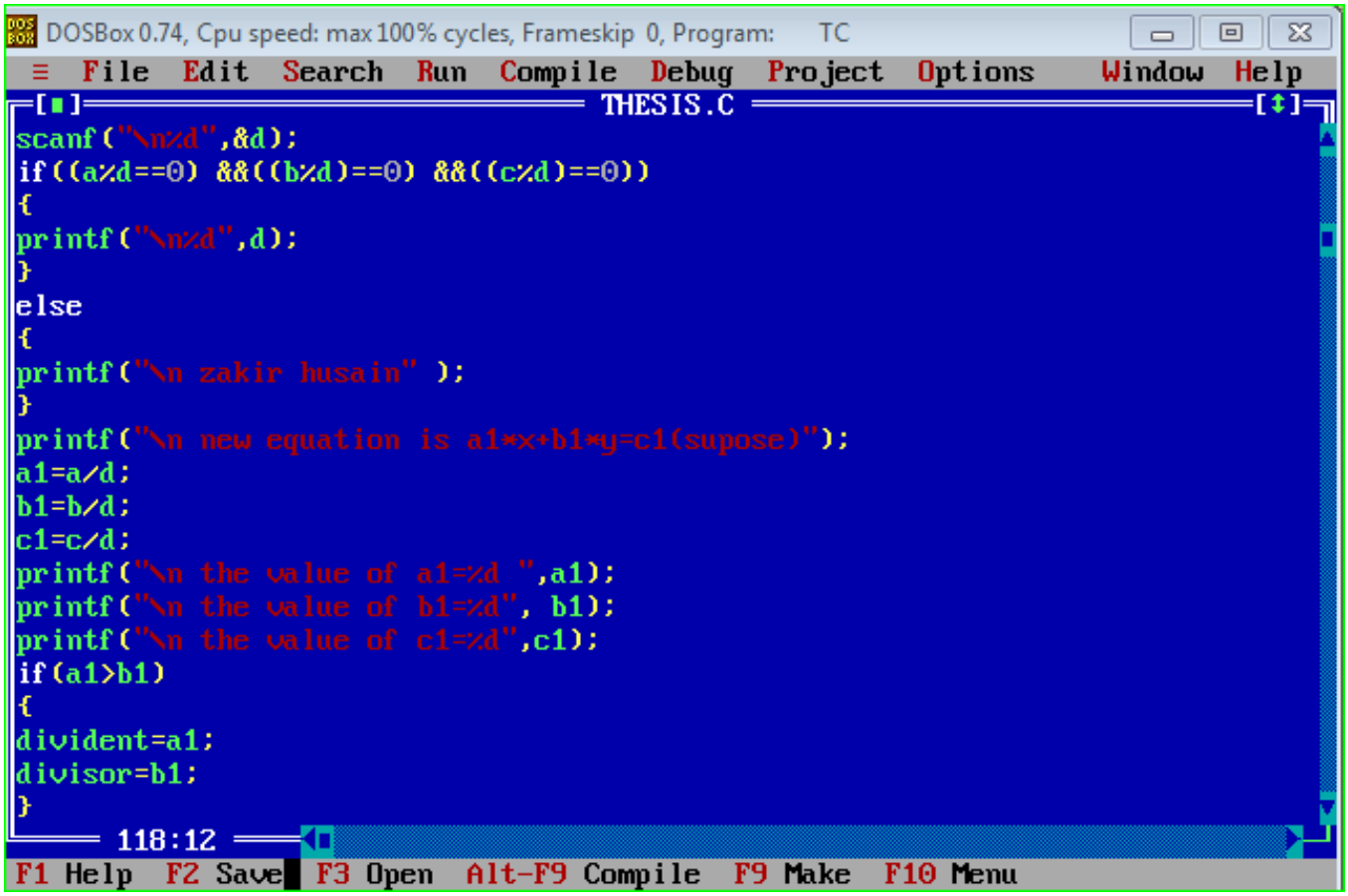
User	public key
Zakir	(e, n)
Aagoosh	(e1, n1)
Ram Kumar	(e2, n2)

5.1 Experimentation



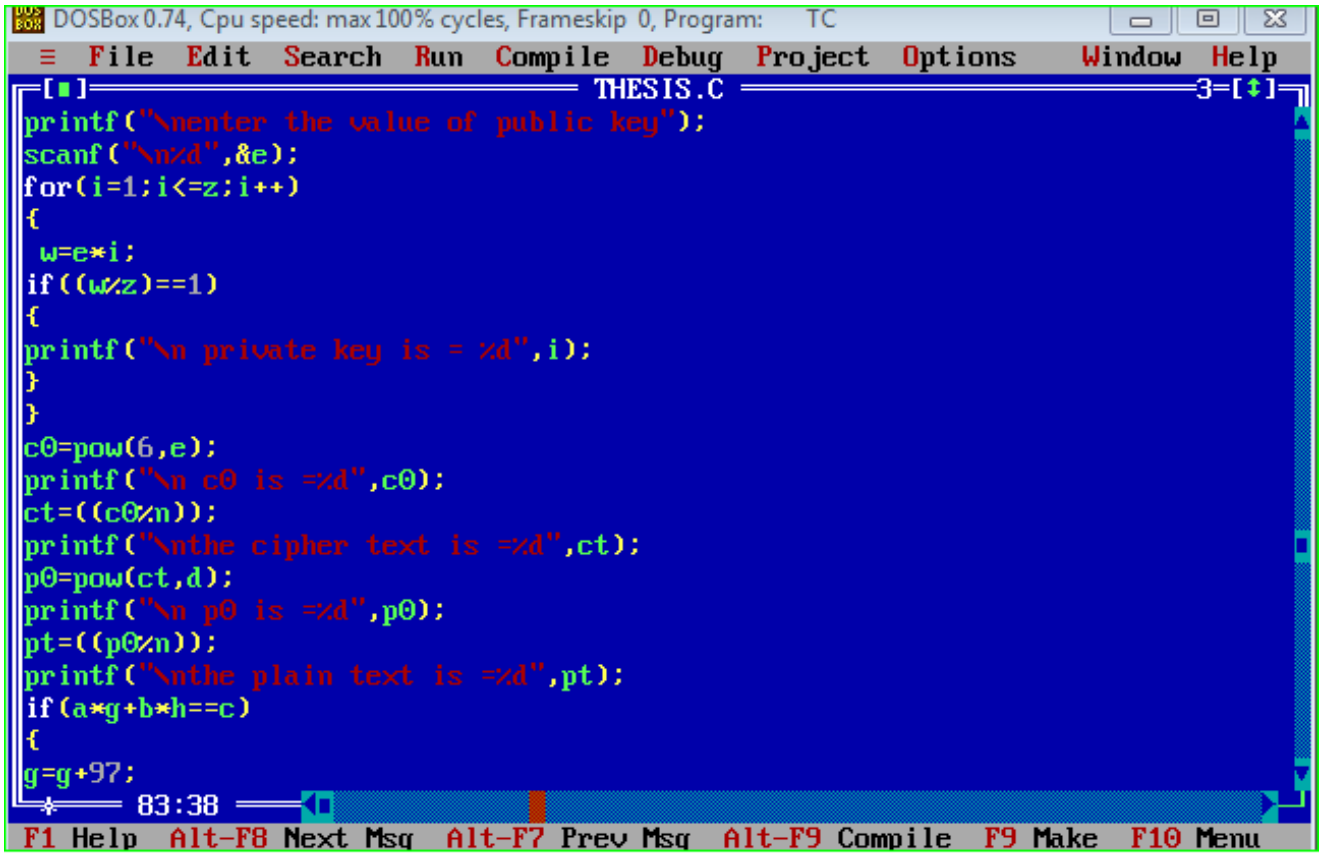
Snapshot 5.1 Programming of Algorithm (1)

The above snapshot shows that, firstly entered the value of Diophantine constants a, b, c and then calculates a Diophantine constant d, which divides a, b, c. So that the Diophantine equation must be reduced a1x+b1y=c1.Now apply Euclidian algorithm for next solution.



Snapshot 5.2 Programming of Algorithm (2)

The above programming portion of proposed algorithm shows that how can be calculated the g c d of any two given numbers with the help of Euclidian algorithm. The Euclidian algorithm gives the gcd of two numbers, and also helpful to calculate the multiplicative inverse of any two numbers.

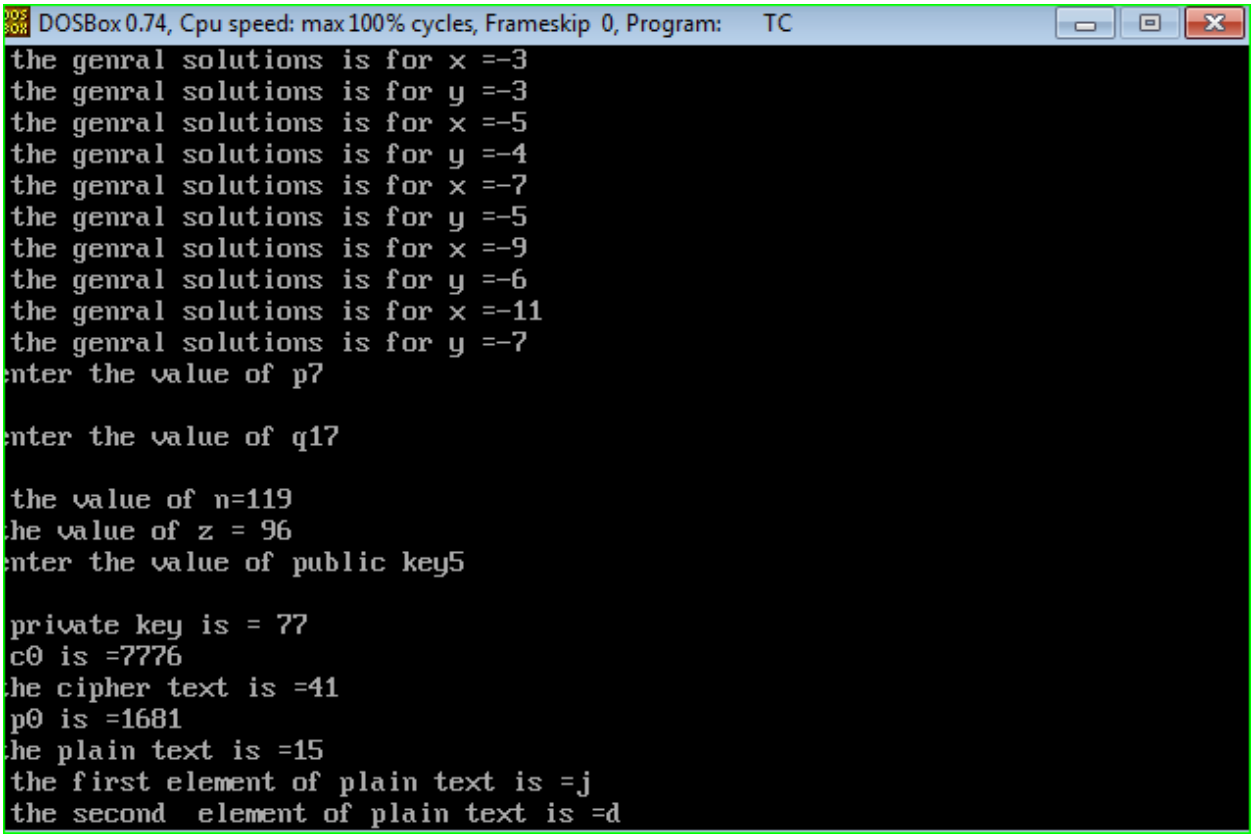


Snapshot 5.3 Programming of Algorithm (3)

This snap shows that, how can be calculated the private key with the help of public key .In RSA algorithm the public key is multiplicative inverse of private key. The programming portion also shows how a plain text element can be converted into integer element?

5.2 Results

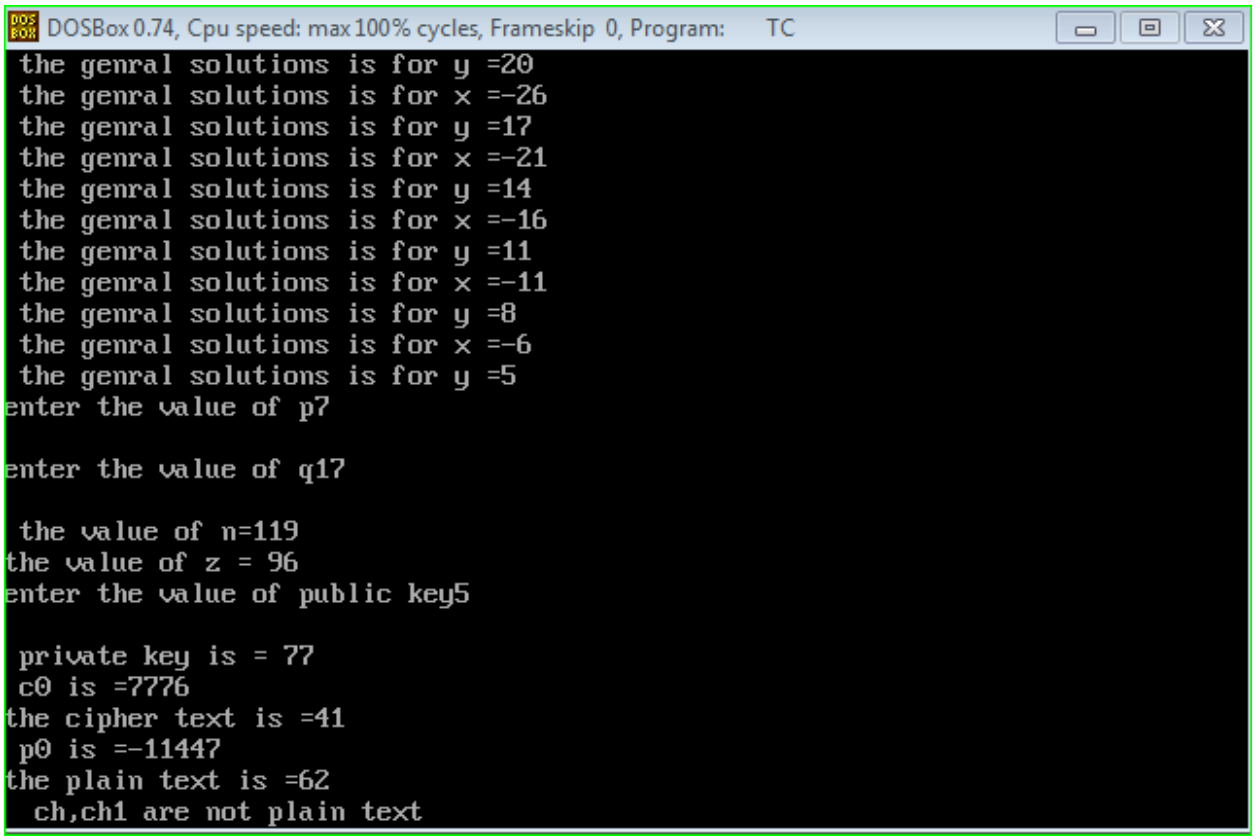
The proposed algorithm which uses the Diophantine equation and RSA algorithm is highly secure. Because in the proposed algorithm the RSA algorithm uses on Diophantine equation's variables, that is in the first step used of RSA algorithm for only two words of given plain text, and similarly for next two words of given plain text and so on.



Snapshot 5.4 Results of Algorithm (1)

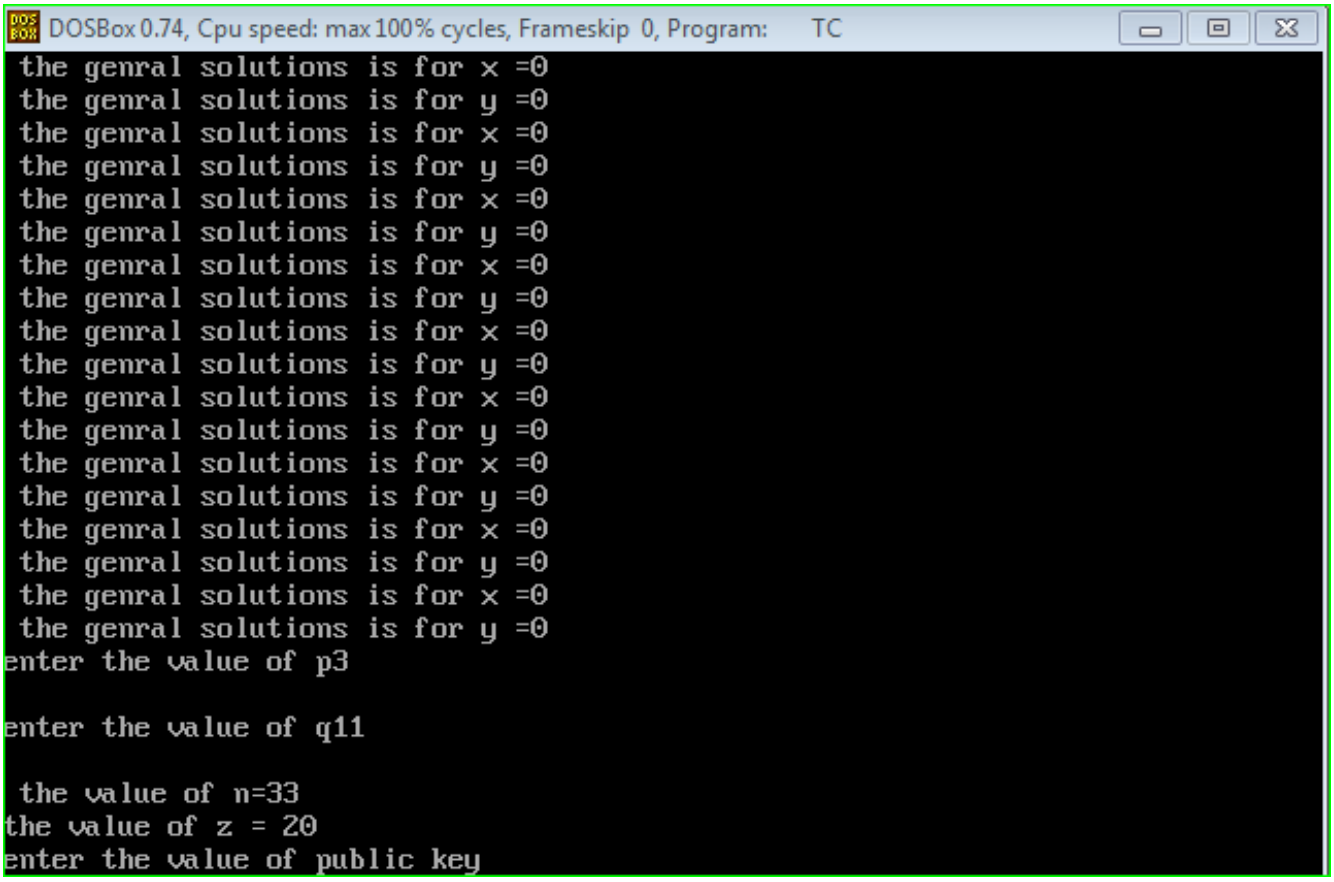
In this snap the entering values are Diophantine constant a, b, c, and two plain text elements *j*, *d*.

At first, find out Diophantine constant d, which is gcd of (a, b, c) that is d divides a, b and c. In next step, apply 'Euclidian algorithm', this algorithm helps to find gcd of any two numbers. After applying 'Euclidian algorithm' calculates the value of c, which is input for RSA algorithm and then apply RSA algorithm on the given plain text (c) to produce the cipher text. Finally, from RSA algorithm, again calculate the Diophantine constant 'c'. Now, c=ax+by, where x, y are plain text elements and a, b are constants. Apply Diophantine equation on the above equation. If the equation is satisfied then x, y are original plain text elements, otherwise they are not original plain text elements. In the above screen shot j and d are plain text elements.



Snapshot 5.5 Results of Algorithm(2)

In the above snap shot, Diophantine equation is used and with the help of this equation, particular solution and general solutions (Infinite number solutions) are evaluated. In the above screen shot, j and d are not plain text elements.



Snapshot 5.6 Results of Algorithm (3)

This snapshot shows only particular and general solutions of Diophantine equation and also shows the results of Euclidian algorithm and RSA algorithm.

5.3 Comparison between RSA Algorithm and Proposed Algorithm

1-For a given plain text, the RSA algorithm applies on whole plain text but the proposed algorithm can apply on two elements of plain text so that proposed algorithm is highly secure.

2-In proposed algorithm the Diophantine equation gives particular solutions and also infinite number of solutions and we can select the plain text either particular solution or infinite number of solutions, that is plain text depends on Diophantine equation but in RSA the plain text independent form keys (public or private).

3-The Diophantine calculator can use in proposed algorithm.

5.4 Comparison with Other Algorithms

The results from new algorithm are more efficient compare than RSA algorithm because, in this algorithm the RSA algorithm uses for each two plain text elements. We are comparing the proposed algorithm from the following algorithm which is given below.

Let b, k, m be given integer such that g c d (b, m) =1, (gcd (k, n) =1.Then for given plain text x

Power (x, k) = b mod (m) for cipher text

x=power (b, u) mod (m). For plain text

U =inverse of k mod (n).

Where n= (p-1) * (q-1).

From above algorithm it is easily find cipher text from given plain text. So this algorithm is easy to break but proposed algorithm is more secure rather than this algorithm .The proposed algorithm is highly secure because we use RSA algorithm in proposed algorithm .and RSA algorithm is highly secure .

Chapter 6

CONCLUSION AND FUTURE SCOPE

6.1Conclusion

In this thesis report, various encryption algorithms mainly RSA algorithm and DES algorithm, are studied. A new algorithm is proposed by applying the Diophantine equation on RSA algorithm with the name of “An efficient algorithm using Diophantine equation and RSA algorithm”. The proposed algorithm is highly secure because in this algorithm the RSA algorithm must be applied only two plain text elements but the RSA algorithm uses for all plain text at a time. But the complexity of the proposed algorithm is very high because on receiver side, it is not easy to generate plain text. It is necessary that receiver must apply the Diophantine equation to generate the original plain text.

6.2 Future Scope

In future this proposed algorithm will become highly secure if its complexity is reduced and will be implemented on all fields of cryptography.

Reference

[1]. "Cryptographic Algorithms for Protection of Computer Data During Transmission and Dormant Storage", Federal Register 38, No. 93 (May 15, 1973).

[2]. "Data Encryption Standard", Federal Information Processing Standard (FIPS) Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington D.C. (January 1977).

[3]. Carl H. Meyer and Stephen M. Matyas, Cryptography: "A New Dimension in Computer Data Security", John Wiley & Sons, New York, 1982.

[4]. Dorthy Elizabeth Robling Denning: "Cryptography and Data Security", Addison Wesley Publishing Company, Reading, Massachusetts, 1982.

[5]. D.W. Davies and W.L. Price: "Security for Computer Networks: An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer", Second Edition, John Wiley& Sons, New York, 1984, 1989.

[6]. Miles E. Smith and Dennis K. Branstad: "The Data Encryption Standard: Past and Future," in Gustavus J. Simmons, Ed., Contemporary Cryptography: The Science of Information Integrity, IEEE Press, 1992.

[7]. Douglas R. Stinson, Cryptography: "Theory and Practice", CRC Press, Boca Raton, 1995.

[8]. Bruce Schneier: "Applied Cryptography", Second Edition, John Wiley & Sons, New York,

[9]. Comer Douglas The internet book ,prentice Hall, India, 1999.

[10]. Frouuzan Behrouz: "Data communication and Networking", Tata McGraw Hill, 2002.

[11]. Scott,Charlie et al: "Virtual Private Networks O", Reilly,2000.

[12]. Smith Richard: "Internet cryptography", Pearson Education Asia, 2002.

[13]. Harn, L.: "Batch verifying multiple RSA digital signatures", Electronics Letters, 32(12), 1219–1220, 1988.

[14]. Rivest, R.L., A. Shamir, and L. Adelman: "A method for obtaining digital signatures and public key cryptosystem", ACM, 21(2), 120–126, 1978.

[15]. S. Wesley Changchien and Min-Shiang Hwang: “A batch verifying and detecting multiple RSA digital signatures”, International Journal of Computational and Numerical Analysis and Applications, vol. 2, no. 3, pp. 303–307, 2002.

[16]. Dorothy E. R. Denning: “Cryptography and Data Security”, Massachusetts: Addison-Wesley, 1982.

[17]. M. S. Hwang, I. C. Lin, and K. F. Hwang: “Cryptanalysis of the batch verifying multiple RSA digital signatures”, Informatica, vol. 11, no. 1, pp. 15–19, 2000.