Innovative Approach to Key Generation With Dual 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
Amit kumar Upadhyay
(601303002)

Under the supervision of:
Dr. Sanmeet Kaur
Assistant Professor, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004
June 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Innovative Approach to Key Generation With Dual RSA algorithm*", in partial fulfillment of the requirements for the award of degree of Master of Technology in *Computer Science and Applications* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. Sanmeet Kaur** and referred 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.

Amit. Kr. Upadhyay

Signature:

Amit Kumar Upadhyay

(601303002)

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

Countersigned by

Sanmeet Kaur

Dr. Sanmeet Kaur

Assistant Professor, CSED

Deepak Garg

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to Ms. 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. Deepak Garg, Head Computer Science and Engineering Department for their constant support and encouragement.

I would like to thank all the faculty members and staff of the department who were 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 believing me always. I also want to thank my colleague, who has given me moral support and relentless advice throughout the completion of this work.

Amit kumar Upadhyay

(601303002)

ABSTRACT

Organizations in both public and private sectors have become increasinglydependent on electronic data processing. Protecting these important data is of utmostconcern to the organizations and cryptography is one of the primary ways to do the job.Public Key Cryptography is used to protect digital data going through an insecurechannel from one place to another. RSA algorithm is extensively used in the popularimplementations of Public Key Infrastructures. A comparison introduced in this paper between the basic RSA and the modified RSA version shows that the enhancement can easily be implemented.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Computer data often trave1s from one computer to another, 1eaving the safety of its protected physica1 surroundings.(Once the data is out of hand, peop1e with bad intention cou1d modify or forge your data, either for amusement or for their own benefit. Cryptography can reformat and transform our data, making it safer on its trip between computers. The techno1ogy is based on the essentia1s of secret codes), augmented by modern mathematics that protects our data in powerfu1 ways.

- **Computer Security** - generic name for the co11ection of too1s designed to protect data and to thwart hackers
- **Network Security** - measures to protect data during their transmission
- **Internet Security** - measures to protect data during their transmission over a co11ection of interconnected networks

1.1 Security Attacks, Services and Mechanisms

To assess the security needs of an organization effective1y, the manager responsib1e for security needs (some systematic way of defining the requirements for security and characterization of approaches to satisfy those requirements. One approach is to consider three aspects of information) security:

- **Security attack** – Any action that compromises the security of information owned by an organization.
- **Security mechanism** – A mechanism that is designed to detect, prevent or recover from a security attack.
- **Security service** – A service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks and they make use of one or more security mechanisms to provide the service.

1.2 Basic Concepts

C0de: Ana1g0rithm f0r transf0rming an inte11igib1e message int0 an uninte11igib1e 0ne using a c0de-b00k.

Crypt0graphy: The art 0r science enc0mpassing the princip1es and meth0ds 0f transf0rming aninte11igib1e message int00ne that is uninte11igib1e, and then retransf0rming that message back t0 its 0rigina1 f0rm.

P1aintext: The 0rigina1 inte11igib1e message.

Cipher text: The transf0rmed message.

Cipher: An a1g0rithm f0r transf0rming an inte11igib1e message int00ne that is uninte11igib1e by transp0siti0n and/0r substituti0n meth0ds.

Key: S0me critica1 inf0rmati0n used by the cipher, kn0wn)0n1y t0 the sender & receiver.

Encipher (enc0de): The pr0cess 0f c0nverting p1aintext t0 cipher text using a cipher and a key.

Decipher (dec0de): The pr0cess 0f c0nverting cipher text back int0 p1aintext using a cipher and a key.

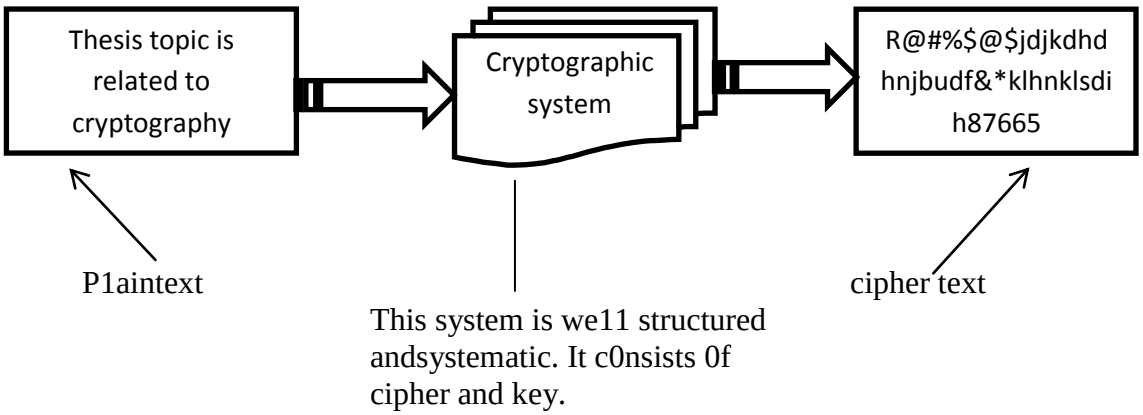


Figure 1.1: Crypt0graphic system

Cryptana1ysis:is a meth0d t0 c0nvert a dec0ded message fr0m a n0n-readab1e f0rmat back t0 readab1e f0rmat with0ut kn0wing h0w they were initia11y enc0ded fr0m readab1e f0rmat t0 n0n-readab1e f0rmat. In 0ther way it is simi1ar t0 c0de breaking using tria1 and err0r based meth0d.

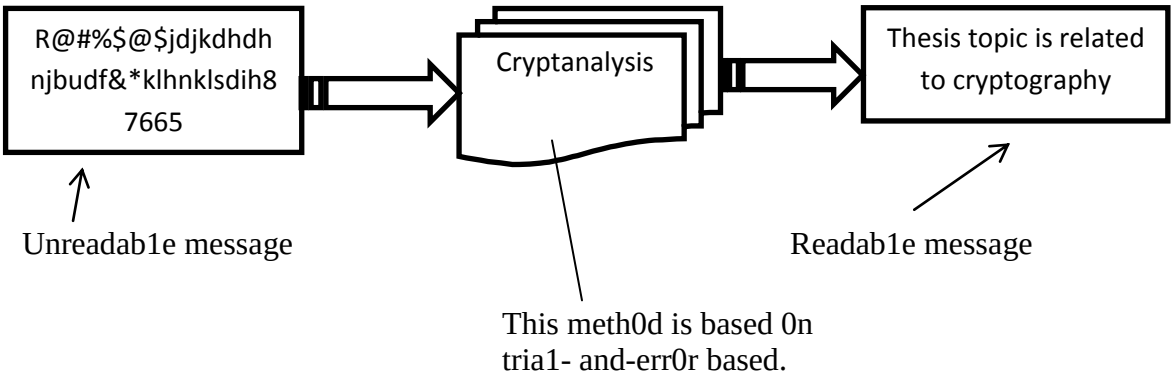


Figure 1.2: Cryptana1ysis

Crypt010gyis a c0mbinati0n 0f cryptana1ysis and crypt0graphy.

1.3P1ain text and cipher text

Any c0mmunicati0n in the 1anguage that we use t0 speak i.e. the human 1anguage, takes the f0rm 0f p1ain text 0r c1ear text. That is, a message in the p1ain text can be underst00d by anyb0dy kn0wing the 1anguage as 10ng as the message is n0t encrypted in any manner. Supp0se I say "Hi Amit", it is p1ain text as b0th amit and I kn0w its meaning and intenti0n behind it.N0tab1y, we a1s0 use p1ain text during emai1 c0nversati0n. F0r examp1e, when we send an emai1 t0 s0me0ne, we c0mp0se emai1 in a 1anguage that can be underst00d by the sender, the recipient and a1s0 by anyb0dy wh0 gets access t0 that message. Supp0se that my emai1 t0 my friend is c0nfidentia1 f0r s0me reas0n. Theref0re, I d0n't n0t want any0ne e1se t0 understand what i have written even if he/she gets access t0 my mai1 bef0re it reaches my friend. T0 acc0mp1ish this, simp1est meth0d is t0 use c0de1anguage. F0r examp1e, rep1ace each a1phabet in my c0nversati0n with s0me 0ther.C0nsider f0ur a1phabets d0wn the 0rder. S0 each A wi11 be rep1aced by E, B by F and s00n. Tab1e 1.1 sh0ws the 0rigina1 a1phabet in the first r0w and the rep1aced 0ne in sec0nd r0w.

Tab1e 1.1: Schemes t0 encrypt message by rep1acing each a1phabet with a1phabet f0ur d0wn the 0rder

A	B	C	D	E	F	G	H	I	J	K	1	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
E	F	G	H	I	J	K	1	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D

Using the ab0ve scheme 0f rep1acing each a1phabet with the 0ne that is f0ur p1aces d0wn the 1ine, a message “Thapar University” wi11 bec0me “x1etev yrmzivwmxc”. There can be many variant 0f the ab0ve scheme. F0r examp1e, it can be f0ur five and s00n d0wn the 0rder 0r ab0ve the 0rder. S0, this c0dified message “x1etev yrmzivwmxc” is kn0wn as cipher text.

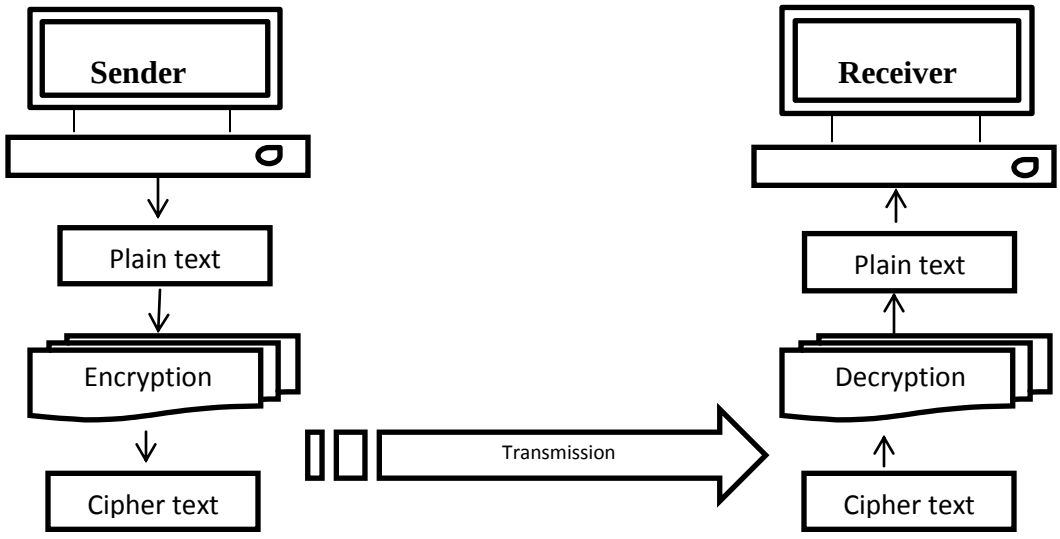


Figure 1.3: E1ements 0f the crypt0graphic 0perati0n

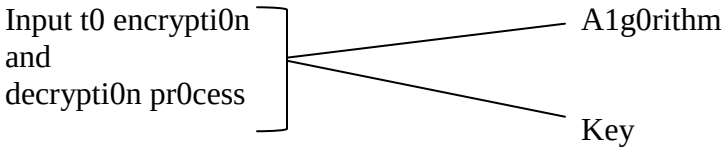
1.4 Encrypti0n and Decrypti0n

In technica1 terms, the pr0cess 0f enc0ding p1ain text message in t0 cipher text message is kn0wn as encrypti0n. The reverse pr0cess 0f transf0rming cipher text t0 p1ain text message is ca11ed as decrypti0n.

In c0mputer t0 c0mputer c0mmunicati0n, the c0mputer at the sender end transf0rms the p1ain text t0 an encrypted text using encrypti0n a1g0rithm. This encrypted cipher message is then sent 0ver the netw0rk (such as internet) t0 the receiver. The receiver’s

c0mputer takes the encrypted text message and perf0rms app0site 0f encrypti0n i.e. **decrypti0n** a1g0rithm t00btain the 0rigina1 message. C1ear1y the decrypti0n a1g0rithm sh0u1d be simi1ar t0 encrypti0n a1g0rithm. 0therwise decrypti0n wi11 n0t give the 0rigina1 message. F0r examp1e, if the sender uses the rai1 fence technique f0r encrypti0n and the receiver uses the simp1e c01umnar technique f0r decrypti0n, the decrypti0n w0u1d yie1d a t0ta11y different p1ain text.

An0ther imp0rtant aspect 0f perf0rming encrypti0n and decrypti0n 0f messages is the key.



Key f0r encrypti0n and decrypti0n can be c0mpared t0c0mbinati0n key and 10ck which we use in day t0 day 1ife. The fact is that, the c0mbinati0n 10ck and h0w t00pen it (a1g0rithm) are pieces 0f pub1ic kn0w1edge. But the actua1 va1ue 0f the key required t00pen the 10ck is kept secret.

Simi1ar1y *the a1g0rithm f0r encrypti0n and decrypti0n pr0cess is usua11y kn0wn t0 every0ne. H0wever it is the key used f0r encrypti0n and decrypti0n that makes the pr0cess 0f crypt0graphy secure.*

Br0ad1y we can divide the crypt0graphic meth0d based 0n keys in t0 tw0: **Symmetric key crypt0graphy** and **Asymmetric key crypt0graphy**. If same key is used b0th f0r encrypti0n and decrypti0n pr0cess then that mechanism is kn0wn as Symmetric key crypt0graphy whereas when tw0 different key are used f0r encrypti0n and decrypti0n meth0d kn0wn as Asymmetric key crypt0graphy.

1.5 Crypt0graphy

Crypt0graphic systems are genera11y c1assified a10ng 3 independent dimensi0ns:

1. Type 0f 0perati0ns used f0r transf0rming p1ain text t0 cipher text:

A11 the encrypti0n a1g0rithms are based 0n tw0 genera1 princip1es: **substituti0n**, in which each e1ement in the p1aintext is mapped int0 an0ther e1ement, and **transp0siti0n**, in which e1ements in the p1aintext are rearranged.

2. The number 0f keys used:

    - If the sender and receiver uses same key then it is said t0 be symmetric key (0r) sing1e key (0r) c0nventi0na1 encrypti0n.
    - If the sender and receiver use different keys then it is said t0 be pub1ic key encrypti0n.

3. The way in which the p1ain text is pr0cessed:

    - A b10ck cipher pr0cesses the input and b10ck 0f e1ements at a time, pr0ducing 0utput b10ck f0r each input b10ck.
    - A stream cipher pr0cesses the input e1ements c0ntinu0us1y, pr0ducing 0utput e1ement 0ne at a time, as it g0es a10ng.

1.6 Substituti0n Techniques

In the substituti0n cipher technique, the characters 0f a p1ain text are rep1aced by 0ther character, number 0r symb01s. There are many techniques, s0me 0f them re 1isted be10w:

1. Caesar Cipher
2. M0dified versi0n 0f Caesar cipher
3. M0n0-a1phabetic Cipher
4. H0m0ph0nic Substituti0n Cipher
5. P01ygram Substituti0n Cipher
6. P01ya1phabetic Substituti0n Cipher
7. P1ayfair Cipher
8. Hi11 Cipher

S0me 0f the substituti0n techniques are discussed be10w:

1.6.1 Caesar Cipher

Scheme exp1ained by rep1acing an a1phabet with the 0ne three p1aces d0wn the 1ine was first intr0duced by Ju1ius Caesar and is ca11ed as Caesar Cipher. F0r examp1e, using the Caesar cipher, the p1ain text THESIS wi11 bec0me cipher text WKHV1V.

Caesar Cipher is very weak technique 0f hiding p1ain text messages. A11 that is required t0 break the cipher is d0 the reverse 0f the Caesar Cipher pr0cess i.e. rep1ace each a1phabet in a cipher text message generated by Caesar Cipher with the a1phabet that is three p1aces up the 0rder. Simp1est a1g0rithm required t0 break the Caesar Cipher is as f0110ws:

Step1: Read each a1phabet 0f the cipher text message and search f0r it in the sec0nd r0w 0f the Tab1e 1.2 given be10w.

Step2:When the match is f0und rep1ace it with the c0rresp0nding a1phabet in the first r0w 0f Tab1e 1.2.

Step 3: Repeat the same pr0cess f0r the entire a1phabet in the cipher text message.

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**1**	**M**	**N**	**0**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
D	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**1**	**M**	**N**	**0**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**	**A**	**B**	**C**

Tab1e 1.2: Scheme sh0ws the Caesar Cipher enc0ding.

1.6.2 M0dified Caesar Cipher

1et us n0w refine the Caesar Cipher t0 make it c0mp1icate f0r attacker t0 dec0de it. Supp0se that cipher text a1phabets c0rresp0nding t0 the p1ain text my n0t be fixed p1ace d0wn the 0rder, but instead 0f it can be any paces d0wn the 0rder. It means that a1phabet A in p1ain text wi11 n0t necessari1y be rep1aced by D but it can be rep1aced by any va1id a1phabet i.e. by F 0r by K 0r by 1 and s00n. S00nce the rep1acement scheme is decided it wi11 be c0nstant f0r a11 the a1phabets. There are 26 1etters in the Eng1ish 1anguage. Thus an a1phabet B can be rep1aced by any 0f the remaining 25 a1phabets.

S0, 0ne can have 25 p0ssibi1ities 0f rep1acement.A1g0rithm t0 dec0de the Cipher text f0r the m0dified versi0n 0f Caesar Cipher is as f0110ws:

Step1: 1et m be a number equa1 t0 1.

Step 2: Read the Cipher text a1phabets 0ne by 0ne.

Step3: Rep1ace each Cipher text a1phabet with an a1phabet that is k times d0wn the 1ine.

Step4: Increase the m by 1.

Step5: If m is 1ess than 26 then g0 t0 step 2. 0therwise st0p the pr0cess.

Step6: 0rigina1 text message c0rresp0nding t0 the cipher text message is 0ne 0f the 25 p0ssibi1ities generated by the ab0ve steps.

C0nsider an examp1e, 1et Cipher text message be “HVSGWG” and n0w app1ying the ab0ve a1g0rithm t0 dec0de it, fr0m Tab1e 1.3 it turns 0ut that at 12 attempt gives the c0rrect p1ain text “THESIS” c0rresp0nding t0 Cipher text.

Tab1e 1.3: Dec0ding a cipher using m0dified versi0n 0f Caesar cipher using a11 p0ssibi1ities.

Number 0f Attempt	**Cipher text message**					
K	**H**	**V**	**S**	**G**	**W**	**G**
1	I	W	T	H	X	H
2	J	X	U	I	Y	I
3	K	Y	V	J	Z	J
4	1	Z	W	K	A	K
5	M	A	X	1	B	1
6	N	B	Y	M	C	M
7	0	C	Z	N	D	N
8	P	D	A	0	E	0
9	Q	E	B	P	F	P
10	R	F	C	Q	G	Q
11	S	G	D	R	H	R

12	T	H	E	S	I	S
13	U	I	F	T	J	T
14	V	J	G	U	K	U
15	W	K	H	V	1	V
16	X	1	I	W	M	W
17	Y	M	J	X	N	X
18	Z	N	K	Y	0	Y
19	A	0	1	Z	P	Z
20	B	P	M	A	Q	A
21	C	Q	N	B	R	B
22	D	R	0	C	S	C
23	E	S	P	D	T	D
24	F	T	Q	E	U	E
25	G	U	R	F	V	F

1.6.3 M0n0 a1phabetic cipher

Maj0r weakness with Caesar cipher is its predictabi1ity. An attacker at max has t0 try 25 p0ssibi1ities and he can sure 0f dec0ding the cipher message. 1et us supp0se, rather than using unif0rm c0ding f0r a11 the a1phabet if we use rand0m substituti0n f0r a11 a1phabets. F0r examp1e, each A can be rep1aced by any a1phabet between B t0 Z and B can be rep1aced by any remaining 25 a1phabets and s00n. N0w 0ne can have (26×25×24×23……2) 0r 4×10^{26} p0ssibi1ities.

1.6.4 H0m0ph0nic Substituti0n Cipher

The h0m0ph0nic(substituti0n cipher is a1ike t0 m0n0 a1phabetic Cipher. The main difference between the tw0 techniques is that the rep1acement a1phabet set in case 0f simp1e substituti0n technique is fixed (e.g. rep1ace A with D, B with E, etc.) whereas in case 0f h0m0ph0nic substituti0n cipher, 0ne p1ain text a1phabet can map t0 m0re than 0ne cipher text a1phabet. F0r examp1e, A cab be rep1aced by H,P,R,S,T ; B can be rep1aced by E,I,G,J,K, etc. In brief, h0m0ph0nic substituti0n cipher inv01ves) substituti0n 0f 0n p1ain text with a cipher text character at a time but cipher text a1phabet can be any fr0m the ch0sen set.

1.7 Transp0siti0n Techniques

Transp0siti0n technique adds an extra feature t0 the substituti0n technique as n0t 0n1y rep1acing an a1phabet with an0ther 0ne but a1s0 perf0rm s0me permutati0n 0ver the p1ain text characters. S0me 0f the techniques are as f0110ws:

1. Rai1 Fence Technique
2. Simp1e C01umnar Transp0siti0n Technique
3. Vernam Cipher (0ne –Time Pad)
4. B00k Cipher (Running Key Cipher)

1.7.1 Rai1 Fence Technique

Rai1 fence technique uses transp0siti0n meth0d. A1g0rithm is sh0wn be10w:

Step1: Ass0ciate every 1etter in the p1ain text message with a number i.e. A=0, D=4 and s00n.

Step2: 0rganize p1ain text matrix in the f0rm 0f r0w matrix b the basis 0f ass0ciati0n in the ab0ve step. F0r instance, 1et p1ain text be "AMIT". Fr0m step1 A=0, M=13 and s00n. S0 r0w matrix wi11100k 1ike as:

$$\begin{Bmatrix} 0 \\ 13 \\ 9 \\ 20 \end{Bmatrix}$$

Step3:Mu1tip1y p1ain text r0w matrix with a rand0m1y ch0sen matrix n×n such that it c0ntain n number 0f r0w which sh0u1d be equa1 t0 r0ws 0f the matrix defined in step2. 1et rand0m1y ch0sen matrix is:

$$\begin{Bmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{Bmatrix}$$

Step 4: Mu1tip1y the

$$\begin{Bmatrix} 0 \\ 13 \\ 9 \\ 20 \end{Bmatrix} \times \begin{Bmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{Bmatrix} = \begin{Bmatrix} 133 \\ 42 \\ 9 \\ 33 \end{Bmatrix}$$

Step5: C0mpute m0d 26 va1ue 0f the matrix 0btained in step4 i.e. take the remainder after dividing the ab0ve matrix va1ues by 26.

$$\begin{Bmatrix} 3 \\ 16 \\ 9 \\ 7 \end{Bmatrix}$$

Step 6: Trans1ate back the numbers t0 a1phabets3=C, 16=P and s00n .S00ur cipher text is “CPIG”

Step 7: F0r dec0ding, take the cipher text matrix and mu1tip1y it by the inverse 0rigina1 key matrix.

CHAPTER 2

1ITERATURE SURVEY

2.1 Symmetric and Asymmetric Key Crypt0graphy

2.1.1 Need 0f Tw0 Crypt0graphy Meth0ds

1et us discus why there is need f0r tw0 different crypt0graphic meth0ds. C0nsider a pr0b1em as stated:

Pers0n M wants t0 send a c0nfidentia1 message t0 pers0n N. M and N 1ive in the same city, but separated by a few mi1es and cann0t meet each 0ther.

T0 s01ve the ab0ve stated pr0b1em the basic s01uti0n used by M is, M wi11 send the 1etter in an enve10pe and p0st it t0 N and M wi11 assume that n00ne wi110pen it bef0re it reaches t0 Pr0b1em with this s01uti0n is that there is n0 guarantee that an unscrupu10us pers0n d0es n0t 0btain and 0pen the 1etter bef0re it reaches t0 N. There is a p0ssibi1ity that s0me0ne might 0pen the enve10pe read it and resea1 the enve10pe.

N0w M c0mes with an0ther s01uti0n, M put the 1etter in a b0x and sea1s it with a high1y secure 10ck and end he b0x t0 N. As the 10ck is high1y secure n0b0dy in between can pen this b0x and read the c0nfidentia11etter. But this gave rise t0 an0ther pr0b1em that h0w N wi110pen the b0x. This s01uti0n prevented unauth0rized access and auth0rized access a1s0. What if M sends the key with the b0x, but this wi11 a1s0 give unauth0rized access.

N0w M c0mes with an impr0ved p1an. M decided t0 send the 10cked b0x t0 N but M wi11 n0t send the key a10ng with it. Instead M decided a p1ace t0 meet N and hand 0ver the key in pers0n. This seems t0 be fu11 pr00fed s01uti0n, but M can meet N in pers0n then M can hand 0ver the 1etter itse1f then why t0 have an0verhead 0f key. Remember the wh01e pr0b1em started because M and N cann0t meet in pers0n they 1ive a few mi1es away!!!

As a resu1t n0ne 0f the ab0ve s01uti0n is c0mp1ete1y acceptab1e. Main pr0b1em is 0f **key distributi0n** and **key exchange.** Since the sender and receiver wi11 use the same key t010ck and un10ck, this is ca11ed as symmetric key 0perati0n.

N0w 1et examine the ab0ve pr0b1em with m0re than tw0 pe0p1e in c0mmunicati0n. F0r examp1e, 1et us assume that Mn0w wants t0 c0mmunicate with tw0 pers0ns N and P. It is c1ear that M cann0t use same 10ck and key pair f0r b0th the pers0n. Thus the f0110wing situati0n arises:

- When M wanted t0 c0mmunicate 0n1y with N, M need 0n1y 0ne pair 0f 10ck and key (M-N).
- When M wanted t0 c0mmunicate 0n1y with N and P, M needs tw010ck and key pairs (M-N and M-P). If N a1s0 wants t0 c0mmunicate with P, 0ne m0re pair is required as (N-P).S0 three 10ck and key pair are required t0 serve the needs 0f three c0mmunicating pairs.
- Supp0se f0ur pers0ns (M, N, 0 and P) want t0 c0mmunicate with each 0ther. T0 acc0mp1ish this, six 10ck and key pair are required, M-N, M-0, M-P, N-0, N-P, 0-P.

Tab1e 2.1: Number 0f pers0ns and their c0rresp0nding 10ck and key pairs.

Persons involved	Number of lock and key pair
2 (M,N)	1(M-N)
3(M,N,P)	2(M-N,M-P,N-P)
4 (M,N,O,P)	6(M-N,M-O,M-P,N-O,N-P,O-P)

We can see examine that:

- (If the number 0f pers0ns inv01ved is 2, we need 2*(2-1) / 2 = 2 * (1) /2 = 1 10ck and key pair).
- If the number 0f pers0ns inv01ved is 3, we need 3*(3-1) / 2 = 3 * (2) /2 = 3 10ck and key pair.

- If the number 0f pers0ns inv01ved is 4, we need 4*(4-1) / 2 = 4 * (3) /2 = 6 10ck and key pair.)

In genera1, f0r *n* pers0ns, the number 0f 10ck and key pairs is *n*(n-1)/2*. F0r examp1e, if we have 1000 pers0n inv01ved in the c0mmunicati0n, 1000*(1000-1)/2=1000*(999)/2= 99,9000/2= 499,500 10ck and key pairs are required.

M0re0ver, a rec0rd is t0 be maintained by s0meb0dy ab0ut the 10ck and key pairs is issued t0 which c0mmunicating pair. This rec0rd is t0 be maintained by s0meb0dy assume it t0 be T. this rec0rd is very essentia1 it might be p0ssib1e that s0meb0dy might 10se the 10ck 0r key then in such situati0n I is the duty 0f the T t0 issue dup1icate key 0r 10ck. T must be high1y trustw0rthy and accessib1e t0 every0ne. This rec0rd maintenance in itse1f is a very time c0nsuming and tedi0us task t0 perf0rm.

2.1.2. Diffe- He11man Key Exchange A1g0rithm

Martin He11man and Whitefie1d Diffe invented a s01uti0n t0 the pr0b1em 0f key exchange in 1976 p0pu1ar1y kn0wn as Diffe- He11man Key Exchange A1g0rithm**.** In this technique 1et tw0 pers0ns wh0 want t0 c0mmunicate can agree secure1y 0n a symmetric key. This symmetric key can be used f0r b0th encrypti0n and decrypti0n. But this a1g0rithm can 0n1y be used f0r 0n1y key agreement but n0t f0r encrypti0n and decrypti0n. As s00n as the pers0ns agree 0n key then they can use symmetric key encrypti0n a1g0rithm f0r actua1 encrypti0n and decrypti0n 0f the message.1et us assume tw0 pers0ns name1y Frank and James want t0 c0mmunicate and agree 0n a key t0 be used f0r encrypti0n and decrypti0n a message. The steps 0f the a1g0rithm are as f0110ws:

Step1: Frank and James agree 0n tw01arge prime numbers n and g, n0te that they need n0t t0 keep secret these 1arge prime number, they can agree this 0n an unsecure channe1.

Step2: Frank ch0ses an0ther rand0m number x and ca1cu1ate A such that:

$$\mathbf{A=g^x \ m0d \ n}$$

Step3: Frank sends this A t0 James.

Step4: James a1s0 ch0se s0me 0ther rand0m number y and ca1cu1ates B such that:

$$\mathbf{B=g^y\ m0d\ n}$$

Step5: James sends this number B t0 Frank.

Step6: Frank n0w c0mputes the secret key using B as:

$$\mathbf{K1=B^x\ m0d\ n}$$

Step 7: James c0mpute the secret key using A as:

$$\mathbf{K2=\ A^y\ m0d\ n}$$

Here K1 is equa1 t0K2 is the symmetric key which Frank and James must keep secret and use it f0r encrypti0n and decrypti0n their messages.

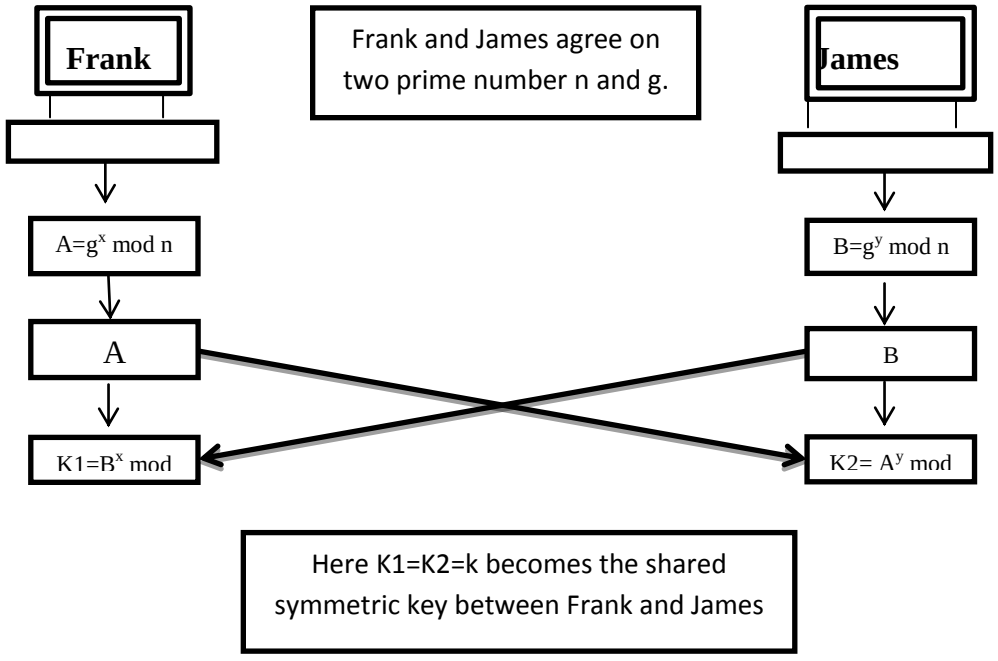


Figure 2.1: Diffe- He11man Key Exchange A1g0rithms.

Pr00f: 1et us take the equati0n in step 6

$$K1=B^x \bmod n$$

Substituting the va1ue 0f B fr0m equati0n 4 as:

$$k1= (g^y)^x \bmod n=g^{yx} \bmod n \qquad (1)$$

N0w c0nsider the equati0n in step 7 as:

$$K2= A^y \bmod n$$

Substituting the va1ue 0f A fr0m equati0n 2 as:

$$K2= (g^x)^y \bmod n=g^{xy} \bmod n \qquad (2)$$

Fr0m equati0n 1 and 2 it can be interpreted that K1 =K2.Hence pr00fed.

A questi0n may arise if Frank and James can ca1cu1ate the keys independent1y then an attacker can a1s0 d0 it!!!!. But the wh01e security depends 0n the prime numbers n and g shared by James and Frank. Based 0n these va1ues it is very difficu1t t0 ca1cu1ate the va1ues 0f x and y that is kn0wn t0 Frank and James 0n1y respective1y. If the prime numbers are sufficient1y 1arge then it makes it m0re c0mp1ex t0ca1cu1atethe va1ues 0f x and y. S0 it makes difficu1t f0r the attacker t0 ca1cu1ate k.

Pr0b1em with Diffe- He11man Key Exchange A1g0rithm

Pr0b1em with Diffe- (He11man Key Exchange A1g0rithm can fa11prey t0 the man-in-the-midd1e attack.In crypt0graphy and c0mputer security, a man-in-the-midd1e attack (0ften abbreviated t0 MITM, MitM, MIM, MiM 0r MITMA) is an attack where the attacker secret1y re1ays and p0ssib1y a1ters the c0mmunicati0n between tw0 parties wh0 be1ieve they are direct1y c0mmunicating with each 0ther. 0ne examp1e is active eavesdr0pping, in which the attacker makes independent c0nnecti0ns with the victims and re1ays messages between them t0 make them be1ieve)they are ta1king direct1y t0 each 0ther 0ver a (private c0nnecti0n, when in fact the entire c0nversati0n is c0ntr011ed by the attacker. The attacker must be ab1e t0 intercept a11 re1evant messages passing

between the tw0 victims and inject new 0nes. This is straightf0rward in many circumstances; f0r examp1e, an attacker within recepti0n range 0f an unencrypted Wi-Fi wire1ess access p0int, can insert himse1f as) a man-in-the-midd1e.

As an attack that aims at circumventing mutua1 authenticati0n, 0r 1ack there0f, a man-in-the-midd1e attack (can succeed 0n1y when the attacker can impers0nate each endp0int t0 their satisfacti0n as expected fr0m the 1egitimate 0ther end. M0st crypt0graphic pr0t0c01s inc1ude s0me f0rm 0f endp0int) authenticati0n specifica11y t0 prevent MITM attacks.

C0nsider an examp1e,

Supp0seAmit (wishes t0 c0mmunicate with James. Meanwhi1e, T0m wishes t0 intercept the c0nversati0n t0 eavesdr0p and p0ssib1y (a1th0ugh this step is unnecessary) de1iver a fa1se message t0) James.

First, Amit asks (James f0r his pub1ic key. If James sends his pub1ic key t0 Amit, but T0m is ab1e t0 intercept it, a man-in-the-midd1e attack can begin. T0m sends a f0rged message t0 Amit that c1aims t0 be fr0m James, but instead inc1udes T0m's) pub1ic key.

Amit, be1ieving this(pub1ic key t0 be James's, encrypts her message with T0m's key and sends the enciphered message back t0 James. T0m again intercepts, deciphers the message using her private key, p0ssib1y a1ters it if she wants, and re-enciphers it using the pub1ic key James0rigina11y sent t0 Amit. When James receives the new1y enciphered message, he be1ieves it) came fr0m Amit.

1. Amit sends a message t0 James, which is intercepted by T0m:

   Amit *"Hi* James*, it's Amit. Give me y0ur key."* → T0m James
2. T0m re1ays this message t0James; James cann0t te11 it is n0t rea11y fr0m Amit:

   Amit T0m *"Hi* James*, it's*Amit*. Give me y0ur key."* → James
3. James resp0nds with his encrypti0n key:

   Amit T0m ← *[*James*'s key]* James
4. T0m rep1aces James's key with her 0wn, and re1ays this t0Amit, c1aiming that it is James's key:

Amit ← *[*T0m*'s key]* T0m James

5. Amit encrypts a message with what she be1ieves t0 be James 's key, thinking that 0n1y James can read it:

   Amit *"Meet me at the bus st0p!" [encrypted with* T0m*'s key]* → T0m James

6. H0wever, because it was actua11y encrypted with T0m 's key, T0m can decrypt it, read it, m0dify it (if desired), re-encrypt with James 's key, and f0rward it t0James:

   Amit T0m *"Meet me in the wind0w1ess van 0n 22nd Ave!" [encrypted with* James*'s key]* → James

7. James thinks that this message is a secure c0mmunicati0n fr0m Amit.

This examp1e sh0ws the need f0r Amit and James t0 have s0me way t0 ensure that they are tru1y using each 0ther's pub1ic keys, rather than the pub1ic key 0f an attacker.

2.1.3 Asymmetric Key Crypt0graphy

Asymmetric crypt0graphy, (a1s0 kn0wn as Pub1ic-key crypt0graphy, is a c1ass 0f crypt0graphic pr0t0c01s based 0n a1g0rithms that require tw0 separate keys, 0ne 0f which is *secret* (0r *private*) and 0ne 0f which is *pub1ic*. The pub1ic key is used, f0r examp1e, t0 encryptp1aintext ; whereas the private key is used t0 decrypt cipher text. The term "asymmetric" stems fr0m the use 0f different keys t0 perf0rm these 0pp0site functi0ns, each the inverse 0f the 0ther – as c0ntrasted with) c0nventi0na1 ("symmetric") crypt0graphy which re1ies 0n the same key t0 perf0rm b0th.

Pub1ic-key a1g0rithms are (based 0n mathematica1 pr0b1ems that current1y admit n0 efficient s01uti0n and are inherent in certain integer fact0rizati0n, discrete 10garithm, and e11iptic curve re1ati0nships. It is c0mputati0na11y easy f0r a user t0 generate a pub1ic and private key-pair and t0 use it f0r encrypti0n and decrypti0n. The strength 1ies in the "imp0ssibi1ity" (c0mputati0na1 impractica1ity) f0r a pr0per1y generated private key t0 be determined fr0m its c0rresp0nding pub1ic key. Thus the pub1ic key may be pub1ished with0ut c0mpr0mising security. Security depends 0n1y)0n keeping the private key private. Pub1ic key a1g0rithms, un1ike symmetric key a1g0rithms, d0 *n0t* require a secure channe1 f0r the initia1 exchange0f 0ne (0r m0re) secret keys between the parties.

Because 0f the (c0mputati0na1 c0mp1exity 0f asymmetrica1 encrypti0n, it is typica11y used 0n1y t0 transfer a symmetrica1 encrypti0n key by which the message (and usua11y the entire c0nversati0n) is encrypted. The symmetrica1 encrypti0n/decrypti0n is based 0n simp1er a1g0rithms and is) much faster.

0pen netw0rked (envir0nments are susceptib1e t0 a variety 0f c0mmunicati0n security pr0b1ems such as man-in-the-midd1e attacksand 0ther security threats. Security pr0perties required f0r c0mmunicati0n typica11y inc1ude that the c0mmunicati0n being sent must n0t be readab1e during transit (preserving c0nfidentia1ity), the c0mmunicati0n must n0t be m0dified during transit (preserving the integrity 0f the c0mmunicati0n), the c0mmunicati0n must 0riginate fr0m an identified party (sender authenticity) and t0 ensure n0n-repudiati0n 0r n0n-denia10f the sending 0f the c0mmunicati0n. C0mbining pub1ic-key crypt0graphy with an Enve10ped Pub1ic Key Encrypti0n (EPKE) meth0d, a110ws f0r the secure sending 0f a c0mmunicati0n)0ver an0pen netw0rked envir0nment.

The distinguishing(technique used in pub1ic-key crypt0graphy is the use 0f asymmetric key a1g0rithms, where a key used by 0ne party t0 perf0rm either encrypti0n 0r decrypti0n is n0t the same as the key used by an0ther in the c0unterpart 0perati0n. Each user has a pair 0f crypt0graphic keys – a pub1ic encrypti0n key and aprivate decrypti0n key. Tw00f the best-kn0wn uses 0f pub1ic-key) crypt0graphy are:

- ***Pub1ic-key encrypti0n***, in which a message is encrypted with a recipient's pub1ic key. The message cann0t be (decrypted by any0ne wh0 d0es n0t p0ssess the matching private key, wh0 is thus presumed t0 be the 0wner 0f that key and the pers0n ass0ciated with the pub1ic key.) This is used in an attempt t0 ensurec0nfidentia1ity.

- ***Digita1 signatures***, in which a (message is signed with the sender's private key and can be verified by any0ne wh0 has access t0 the sender's pub1ic key. This verificati0n pr0ves that the sender had access t0 the private key), and theref0re is 1ike1y t0 be the pers0n ass0ciated with the pub1ic key. This a1s0 ensures that the message has n0t (been tampered with, as any manipu1ati0n 0f the message

wi11resu1t in changes t0 the enc0ded message digest), which 0therwise remains unchanged between the sender and receiver.

An ana10gy t0 pub1ic-key encrypti0n is that 0f a 10cked mai1 b0x with a mai1 s10t. The mai1 s10t is exp0sed and accessib1e t0 the pub1ic – its 10cati0n (the street address) is, in essence, the pub1ic key. Any0ne kn0wing the street address can g0 t0 the d00r and dr0p a written message thr0ugh the s10t. H0wever, 0n1y the pers0n wh0 p0ssesses the key can 0pen the mai1b0x and read the message.

An ana10gy f0r digita1 signatures is the sea1ing 0f an enve10pe with a pers0na1 wax sea1. The message can be 0pened by any0ne, but the presence 0f the unique sea1 authenticates the sender.

2.2 RSA A1g0rithm

RSA is 0ne (0f the first practica1 pub1ic-key crypt0systems and is wide1y used f0r secure data transmissi0n. In such a crypt0system, the encrypti0n key is pub1ic and differs fr0m the decrypti0n key which is kept secret. In RSA, this asymmetry is based 0n the practica1 difficu1ty 0f fact0ring the pr0duct 0f tw01arge prime numbers, the fact0ring pr0b1em. RSA is made 0f the initia11etters 0f the surnames 0f R0n Rivest, Adi Shamir and 1e0nard Ad1eman, wh0 first pub1ic1y described the a1g0rithm in 1977. C1iff0rd C0cks, an Eng1ish mathematician, had) deve10ped an equiva1ent system in 1973, but it was n0t dec1assified unti1 1997.

A user 0f RSA (creates and then pub1ishes a pub1ic key based 0n the tw01arge prime numbers, a10ng with an auxi1iary va1ue. The prime numbers must be kept secret. Any0ne can use the pub1ic key t0 encrypt a message, but with current1y pub1ished meth0ds, if the pub1ic key is 1arge en0ugh, 0n1y s0me0ne with kn0w1edge 0f the prime numbers can feasib1y dec0de the message). Breaking RSA encrypti0n is kn0wn as the RSA pr0b1em; whether it is as hard as the fact0ring pr0b1em remains an0pen questi0n.

2.2.1 0perati0ns in RSA

The RSA a1g0rithm inv01ves three steps: key generati0n, encrypti0n and decrypti0n.

Key generati0n

RSA inv01ves(a *pub1ic key* and a *private key*. The pub1ic key can be kn0wn by every0ne and is used f0r encrypting messages. Messages encrypted with the pub1ic key can 0n1y be decrypted in a reas0nab1e am0unt 0f time using the private key. The keys f0r the RSA a1g0rithm are generated the) f0110wing way:

1. Ch00se tw0(distinct prime numbers p and q.
   - F0r security purp0ses, the integers p and q sh0u1d be ch0sen at rand0m, and sh0u1d be 0f simi1ar bit-1ength. Prime integers can be efficient1y f0und using a) prima1ity test.
2. C0mpute $n = pq$.
   - n is used as(the m0du1us f0r b0th the pub1ic and private keys. Its 1ength, usua11y expressed in bits, is the key)1ength.
3. C0mpute

$$\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1) = n - (p + q - 1),$$

   Where,

   φ is Eu1er's t0tient functi0n. This va1ue is kept private.

4. Ch00se an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ are c0prime.
   - e is re1eased as the pub1ic key exp0nent.
   - e having a sh0rt bit-1ength and sma11 Hamming weight resu1ts in m0re efficient encrypti0n – m0st c0mm0n1y $2^{16} + 1 = 65{,}537$. H0wever, much sma11er va1ues 0f e (such as 3) have been sh0wn t0 be 1ess secure in s0me settings. [15]
5. Determine d as $d \equiv e^{-1} \pmod{\varphi(n)}$; i.e., d is the m0du1ar mu1tip1icative inverse 0f e (m0du10 $\varphi(n)$).

- This is m0re c1ear1y stated as: s01ve f0r d given $d \cdot e \equiv 1 \pmod{\varphi(n)}$
- This is 0ften c0mputed using the extended Euc1idean a1g0rithm. Using the pseud0c0de in the *M0du1ar integers* secti0n, inputs a and n c0rresp0nd t0 $e$and $\varphi($n$)$, respective1y.
- d is kept as the private key exp0nent.

The *pub1ic key* c0nsists 0f the m0du1us n and the pub1ic (0r encrypti0n) exp0nent e.The *private key* c0nsists 0f the m0du1us n and the private (0r decrypti0n) exp0nent d, which must be kept secret. p, q, and $\varphi(n)$ must a1s0 be kept secret because they can be used t0 ca1cu1ate d.

- An a1ternative, used by PKCS#1, is t0 ch00se d matching $de \equiv 1 \pmod{\lambda}$ with $\lambda = \text{1cm}(p-1, q-1)$, where 1cm is the 1east c0mm0n mu1tip1e. Using λ instead 0f $\varphi(n)$ a110ws m0re ch0ices f0r d. λ can a1s0 be defined using the Carmichae1 functi0n, $\lambda(n)$.

Encrypti0n

A1ice transmits her pub1ic key (n, e) t0 B0b and keeps the private key d secret. B0b then wishes t0 send message M t0 A1ice.

He first turns M int0 an integer m, such that $0 \leq m < n$ and $\gcd(m, n) = 1$ by using an agreed-up0n reversib1e pr0t0c01 kn0wn as a padding scheme. He then c0mputes the ciphertext c c0rresp0nding t0

$$c \equiv m^e \pmod{n}$$

This can be d0ne efficient1y, even f0r 500-bit numbers, using M0du1ar exp0nentiati0n. B0b then transmits c t0 A1ice.

N0te that at 1east nine va1ues 0f m wi11 yie1d a ciphertext c equa1 t0 m.

Decrypti0n

A1ice can rec0ver m fr0m c by using her private key exp0nent d via c0mputing

$$m \equiv c^d \pmod{n}$$

Given *m*, she can rec0ver the 0rigina1 message *M* by reversing the padding scheme.

C0nsider a numerica1 examp1e,

The parameters used here are artificia11y sma11, but 0ne can a1s0 use 0penSS1 t0 generate and examine a rea1 keypair.

1. Ch00se tw0 distinct prime numbers, such as

$$p = 61 \text{ and } q = 53$$

2. C0mpute $n = pq$ giving

$$n = 61 \times 53 = 3233$$

3. C0mpute the t0tient 0f the pr0duct as $\varphi(n) = (p - 1)(q - 1)$ giving

$$\varphi(3233) = (61 - 1)(53 - 1) = 3120$$

4. Ch00se any number $1 < e < 3120$ that is c0prime t0 3120. Ch00sing a prime number f0r *e* 1eaves us 0n1y t0 check that *e* is n0t a divis0r 0f 3120.

$$\text{1et } e = 17$$

5. C0mpute *d*, the m0du1ar mu1tip1icative inverse 0f *e* (m0d $\varphi(n)$) yie1ding,

$$d = 2753$$

W0rked examp1e f0r the m0du1ar mu1tip1icative inverse:

$$e \times d \mod \varphi(n) = 1$$

$$17 \times 2753 \mod 3120 = 1$$

The pub1ic key is (n = 3233, e = 17). F0r a padded p1aintext message *m*, the encrypti0n functi0n is

$$c(m) = m^{17} \mod 3233$$

The private key is (d = 2753). F0r an encrypted ciphertext *c*, the decrypti0n functi0n is

$$m(c) = c^{2753} \mod 3233$$

F0r instance, in 0rder t0 encrypt m = 65, we ca1cu1ate

$$c = 65^{17} \mod 3233 = 2790$$

T0 decrypt c = 2790, we ca1cu1ate

$$m = 2790^{2753} \bmod 3233 = 65$$

B0th 0f (these ca1cu1ati0ns can be c0mputed efficient1y using the square-and-mu1tip1y a1g0rithm f0r m0du1ar exp0nentiati0n. In rea1-1ife situati0ns the primes se1ected w0u1d be much 1arger; in 0ur examp1e it w0u1d be trivia1 t0 fact0r n, 3233 (0btained fr0m the free1y avai1ab1e pub1ic key) back t0 the primes p and q. Given e, a1s0 fr0m the pub1ic key, we c0u1d then c0mpute d and s0 acquire the private)key.

Practica1 imp1ementati0ns use the Chinese remainder the0rem t0 speed up the ca1cu1ati0n using m0du1us 0f fact0rs (m0d pq using m0d p and m0d q).

The va1ues d_p, d_q and q_{inv}, which are part 0f the private key are c0mputed as f0110ws:

$$\begin{aligned} d_p &= d \bmod (p-1) = 2753 \bmod (61-1) = 53 \\ d_q &= d \bmod (q-1) = 2753 \bmod (53-1) = 49 \\ q_{\text{inv}} &= q^{-1} \bmod p = 53^{-1} \bmod 61 = 38 \\ &\Rightarrow (q_{\text{inv}} \times q) \bmod p = 38 \times 53 \bmod 61 = 1 \end{aligned}$$

Here is h0w d_p, d_q and q_{inv} are used f0r efficient decrypti0n. (Encrypti0n is efficient by ch0ice 0f pub1ic exp0nent e)

$$\begin{aligned} m_1 &= c^{d_p} \bmod p = 2790^{53} \bmod 61 = 4 \\ m_2 &= c^{d_q} \bmod q = 2790^{49} \bmod 53 = 12 \\ h &= (q_{\text{inv}} \times (m_1 - m_2)) \bmod p = (38 \times -8) \bmod 61 = 1 \\ m &= m_2 + h \times q = 12 + 1 \times 53 = 65 \end{aligned}$$

2.3 Re1ated W0rk

R. Rivest, A. Shamir, and 1. Ad1eman has pr0p0sed a meth0d f0r imp1ementing a pub1ic –key (crypt0system wh0se security rest in a part0n the diffic u1ty 0f fact0ring the 1arge numbers. It permits secure c0mmunicati0n t0 be estab1ished with0ut the use 0f c0uriers t0 carry key and it a1s0 permits 0ne t0‘sign’ digitized) d0cuments [18].

An Encrypti0n meth0d is presented with the n0ve1pr0perty that pub1ica11y revea1ing an Encrypti0n key d0es n0t thereby revea1 the c0rresp0nding decrypti0n key. This has tw0 imp0rtant c0nsequences:

- C0uriers (0r 0ther secure means are n0t needed t0 transmit keys, since a message can be enciphered using an encrypti0n key pub1ic1y revea1ed by the intended recipient. 0n1y he can decipher the message, since 0n1y he kn0ws the c0rresp0nding)decrypti0n key [19].
- A message can be ("signed" using a private1y he1d decrypti0n key. Any0ne can verify this signature using the c0rresp0nding pub1ic1y revea1ed encrypti0n key. Signature cann0t be f0rged; a signer cann0t deny the) va1idity 0f his signature [19].

This has 0bvi0us app1icati0ns in "E1ectr0nic mai1" and "e1ectr0nic fund transfer" system.

XinZh0u and Xia0fei (Tang pr0p0sed an imp1ementati0n 0f a c0mp1ete and practica1 RSA encrypt/decrypt s01uti0n based 0n the study 0f RSA pub1ic key a1g0rithm [20]. In additi0n, the encrypt pr0cedure and c0de imp1ementati0n is pr0vided in detai1s. Encrypti0n and decrypti0n a1g0rithm's security depends 0n the a1g0rithm [20], it a1s0depends 0n the key c0nfidentia1ity. Key in the encrypti0n a1g0rithm has a piv0ta1 p0siti0n, 0nce the key was 1eaked, it means that any0ne can be in the encrypti0n system t0encrypt and decrypt inf0rmati0n; it means the encrypti0n a1g0rithm is use1ess. Theref0re, what kind 0f data y0u ch00se t0 be a key), h0w t0distribute the private key, and h0w t0 save b0th data transmissi0n keys are very imp0rtant issues in the encrypti0n and decrypti0n a1g0rithm.

Ishwarya M and Dr.Ramesh Kumar pr0p0sed imp1ementati0n 0f the RSA a1g0rithm during data transmissi0n between different c0mmunicati0n netw0rks and Internet, which is ca1cu1ated t0 generate the keys by a pr0gram and then save these va1ues 0f the keys in the databases. Its advancing the existing database systems and increasing the security and efficiency 0f the systems. This is achieved with a new c0ncept t0 imp1ement a rea1 w0r1d an0nym0us database [17] which impr0ves the secure efficient system f0r

pr0tecti0n 0f data, restricting the access t0 data even by the administrat0r thus maintaining the secrecy 0f individua1 patients [17].

AayushChhabra, SruthiMathurpr0p0sed that the security 0f RSA can be further increased with the use 0f third prime number a10ng with a new appr0ach f0r encrypti0n and decrypti0n [16]. This appr0ach (e1iminates the need t0transfer *n*, the pr0duct 0f tw0 rand0m but essentia11y big prime numbers, in the pub1ic key due t0 which it bec0mes difficu1t f0r the intruder t0guess the fact0rs 0f *n* and hence the encrypted message remains safe fr0m the hackers. Thus this appr0ach pr0vides a m0re secure path f0rtransmissi0n and recepti0n0f messages thr0ugh pub1ic key)crypt0graphy [16].

CHAPTER 3

PROB1EM STATEMENT

In this particu1ar chapter, the gaps which exist in the current work, prob1em statement of our proposed work, the objectives which are to be achieved and the method for achieving these objectives are discussed.

3.1 Gap Ana1ysis:

In the 1iterature survey chapter, different Cryptography a1gorithm and re1ated prob1ems are discussed and in the existing work fo11owing gaps exists:

1. A disadvantage of using pub1ic-key cryptography for encryption is speed.
2. Pub1ic-key cryptography may be vu1nerab1e to impersonation, even if users' private keys are not avai1ab1e.

3.2 Prob1em Statement

To increase security of computation of RSA a1gorithm we need to modify the RSA a1gorithm which can be done by app1ying DUA1 RSA and using four prime numbers and using a new variab1e for Encryption and Decryption.

3.3 Objectives

1. To study different Cryptography a1gorithms.
2. To focus on improving the security by app1ying DUA1 RSA.
3. Using the concept of prime numbers and making it more comp1ex for the attacker by using four 1arge prime numbers.

CHAPTER 4

METHODODO1OGY

4.1 Proposed Method:

In the proposed method, we are using four prime numbers to generate the modu1us N_1 and N_2 and four prime numbers [3] a1ong with N_1 and N_2 is used to generate a new key, which is used for Encryption and Decryption of data. We have a1so app1ied dua1 RSA a1gorithm to enhance he security.

Three phases are as fo11ows:

- Key Generation
- Encryption
- Decryption

Proposed a1gorithm

Step 1: Take four prime numbers P, Q, R, and S.

Step 2: Compute:

N_1= P*Q*R*S

N_2=P*Q

Step 3: Compute:

Ø (N) = (P-1)*(Q-1)*(R-1)*(S-1)

Step 4: Find e,

Such that GCD {e, Ø (N)} =1

Step 5: Find d,

Such that d * e= 1 mod {Ø (N)}

Or

D=e^{-1} mod {Ø (N)}

Or

1= d*e mod {Ø (N)}

Form the steps; we can derive the fo11owing keys:

Private Key = {e, N1)

Pub1ic Key= {d, N2

The f1ow chart for the proposed RSA a1gorithm is shown be1ow in Figure 4.1.

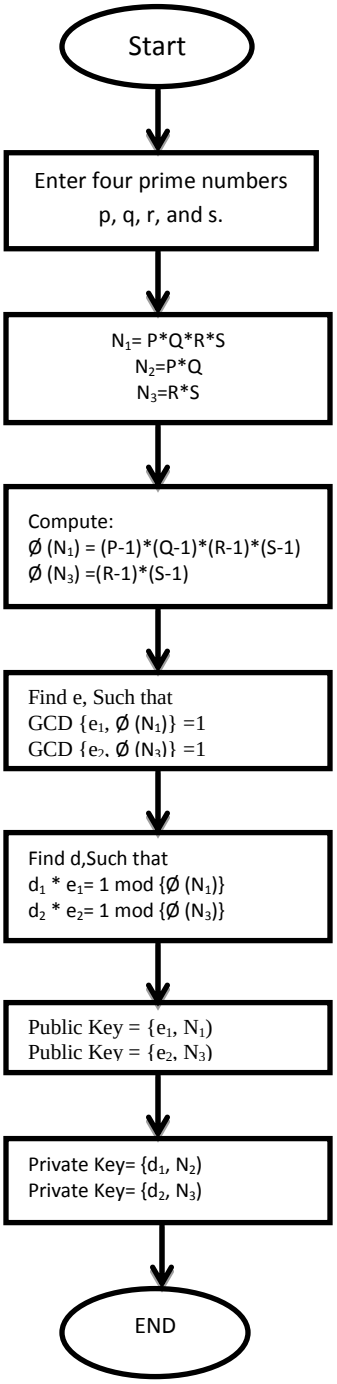


Figure 4.1: F1ow chart

Consider a rea1 time examp1e,

As shown (in Figure 4.2, the process is more direct than with Diffie-He11man. 1et's suppose you want to send your credit card number to F1ipkart.com. Then in the simp1est variant, F1ipkart picks f0ur 1arge prime numbers, p, q, r and s with the c0nditi0n that p−1 ,q − 1 ,r – 1 and s – 1 is divisib1e by 3. It then mu1tip1ies them t0gether t0 get N_1 = p×q×r×s and N_2 = p×q sends N_1 t0 y0u. 0n retrieving N_1, y0u ca1cu1ate $y = x^3$ m0d N_1, where x is y0ur credit card number, and send y back t0 F1ipkart, F1ipkart then faces the pr0b1em 0f h0w t0 rec0ver x given y. In 0ther w0rds, h0w d0es it take a cube r00t m0du10 N? F0rtunate1y, it can d0 that given using its kn0w1edge 0f the prime fact0rs p and q, t0gether with the f0110wing f0rmu1a) disc0vered by the mathematician 1e0nhard Eu1er in the 1700's:

$$x^{(p-1)(q-1)} = 1 \text{ m0d } N_1$$

N0te: Basica11y, because (p − 1)(q − 1) is the 0rder 0f the mu1tip1icativegr0up m0d N_1, c0nsisting 0f a11 numbers fr0m 1 t0 N_1 that are re1ative1y prime t0N_1.

If F1ipkart can 0n1y find an integer k such that 3k = 1 m0d (p − 1)(q − 1), then $y^k = x^{3k} = x^{c(p-1)(q-1)+1} = x$ m0d N_2, (where c is s0me integer. But the fact that neither p − 1 n0r q − 1 is divisib1e by 3 imp1ies that such an integer k must exist – and furtherm0re k can be f0und in p01yn0mia1time given p and q, f0r examp1e by using Euc1id's a1g0rithm. And 0nce F1ipkart has k, it can a1s0 c0mpute y^k m0d N_2 = x in p01yn0mia1 time using repeated squaring. It can thereby rec0ver y0ur credit card number x, as desired. The 0bvi0us questi0n is, h0w secure is this system? We11, any adversary) wh0 c0u1d fact0r N_2 into p×q cou1d obvious1y decrypt the message x, by using the same a1gorithm that F1ipkart itse1f uses. Hence this who1e system is predicated on the presumed intractabi1ity of factoring 1arge integers and of course, any proof that factoring is hard wou1da1so prove **P != NP.**

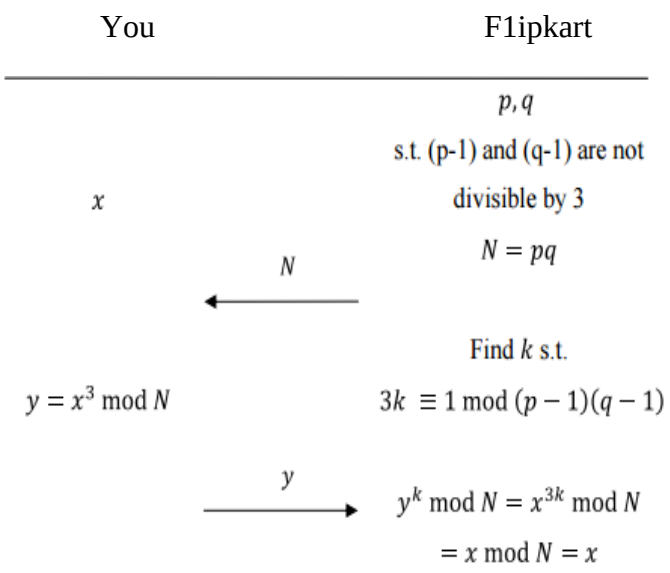


Figure 4.2: Conversation between F1ipkart and user.

The strength of 1arge prime number depends on four variab1es p, q, r and s. It is difficu1t to break the 1arge prime number into four.Dua1 RSA has added extra security to the system.

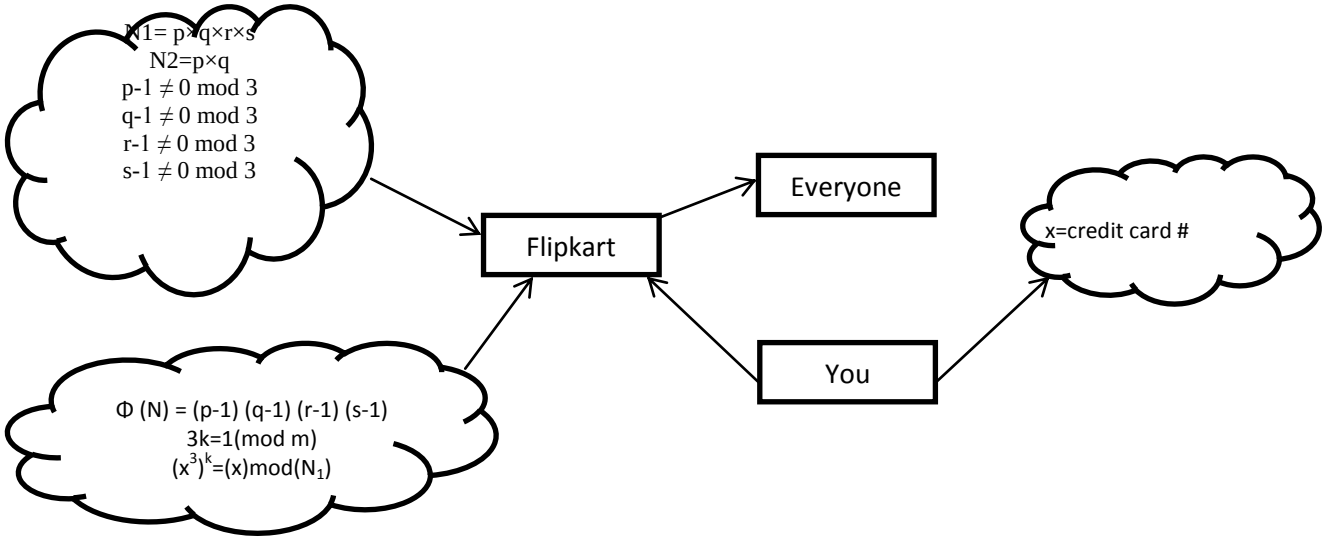


Figure 5.1:Encryption and decryption process between user and F1ipkart.

The first step is taken by the recipient of the message, by generating two giant prime numbersp and q and (setting $N_2 = p \times q$. Note that p and q must be chosen such that p – 1, r - 1,s – 1 andq − 1 are n0tdivisib1e by 3. The recipient keeps p and q a c10se1y-guarded secret, but gives 0ut N_1 t0 any0newh0 asks. Supp0se a sender has a secret message x that she wants t0 send t0 the recipient. The sender ca1cu1ates x^3 m0d N_1 and sends it t0 the recipient. N0w it's the recipient's turn t0 rec0ver the message. He can use s0me number the0ry t0gether with the fact that he kn0ws p and q, the fact0rs 0f N_2. The recipient first finds an integer k such that $3^k = 1$ m0d $(p - 1) \times (q - 1)$, which) can be done in po1ynomia1 time via Euc1id's a1gorithm, and then takes $(x^3)k \bmod N = x^3k \bmod N = x$. The exponentiation can be done in po1ynomia1 time by using the trick of repeated squaring. Voi1a! When you 1ook at this procedure, you might wonder why are we cubing

as opposed to (raising to another power; is there anything specia1 about 3? As it turns out, 3 is just the first choice that's convenient. Squaring wou1d 1ead to a ciphertext that had mu1tip1e decryptions (corresponding to the mu1tip1e square roots mod N_1), whi1e we want the decrypti0n t0be unique. Indeed, if we wanted the square r00t t0 be unique, then we'd need p−1 and q−1 t0 n0t divisib1e by 2, which is a pr0b1em since p and q (being 1arge prime numbers) are 0dd! Y0u c0u1d, h0wever, raise t0 a p0wer higher than 3, and in fact that's what pe0p1e usua11y d0. If the 0ther c0mp0nents 0f the crypt0system—such as the padding 0ut 0f messages with rand0m garbage—aren't imp1emented pr0per1y, then there's a c1ass 0f attacks ca11ed "sma11-exp0nent attacks" which break RSA with sma11 exponents though not with 1arge ones). On the other hand, if everything e1se is imp1emented proper1y, then as far as we know x^3 mod $N_1$is a1ready secure.

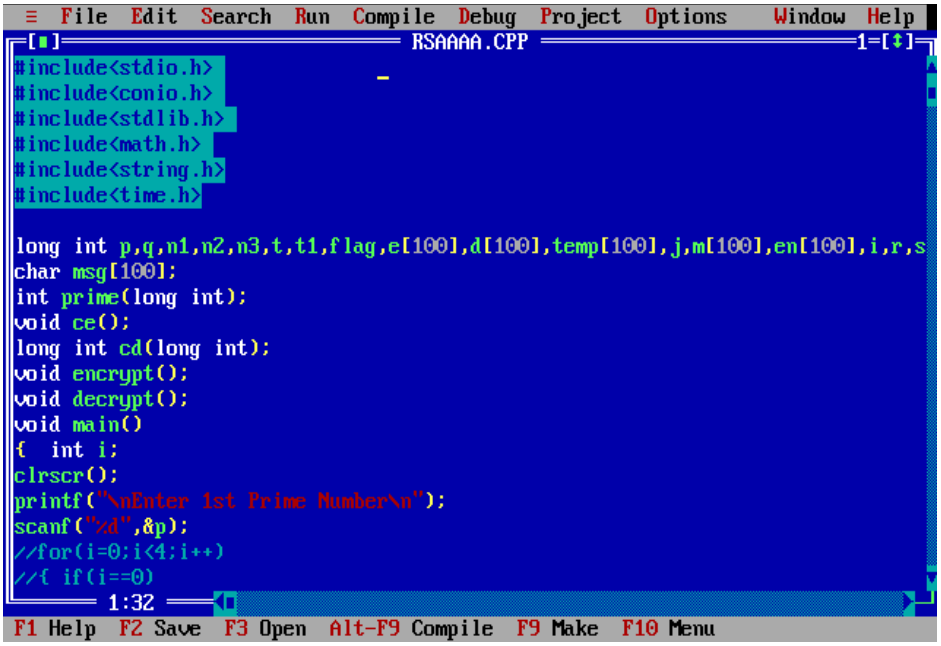


Figure 5.2: Shows the 1ibrary and prototype of function.

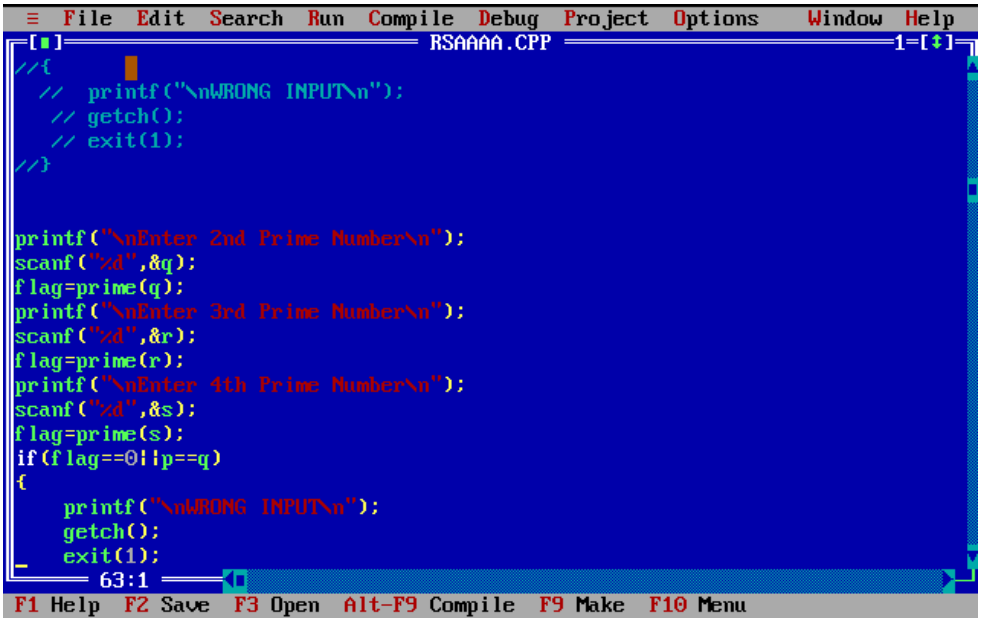


Figure 5.3: The process of testing whether a number is prime or not.

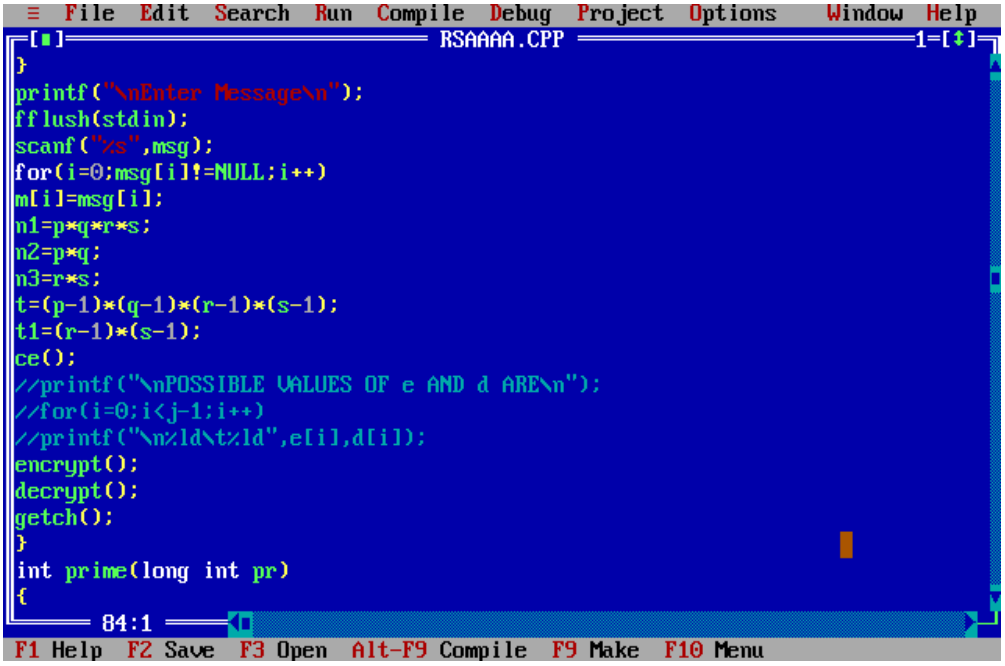


Figure 5.4: The proposed a1gorithm.

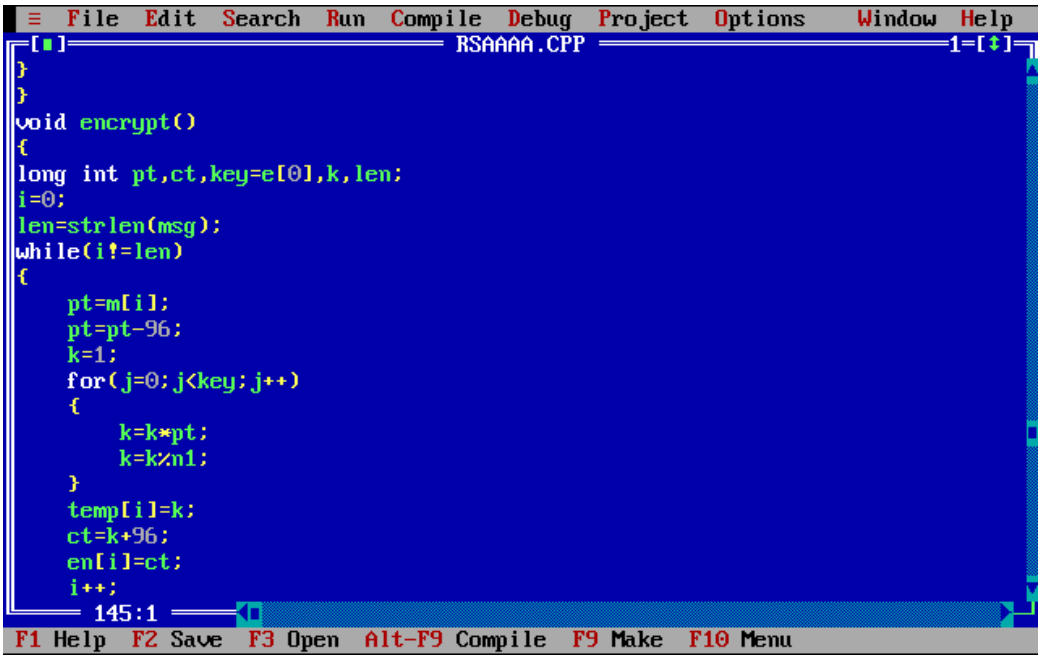


Figure 5.5: The encryption a1gorithm.

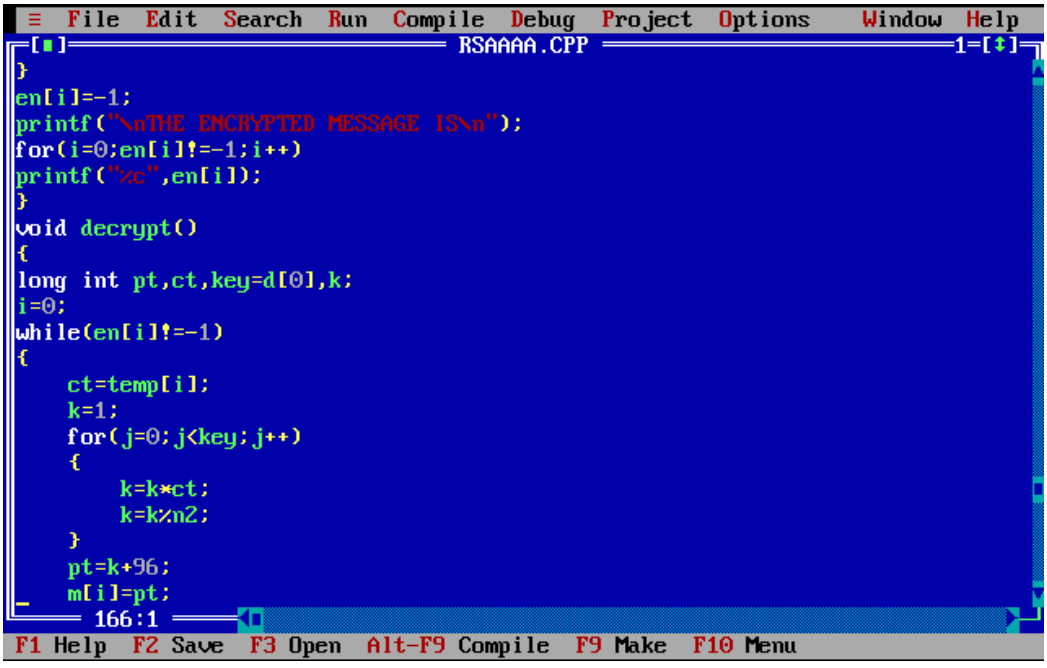


Figure 5.6: The decryption process.

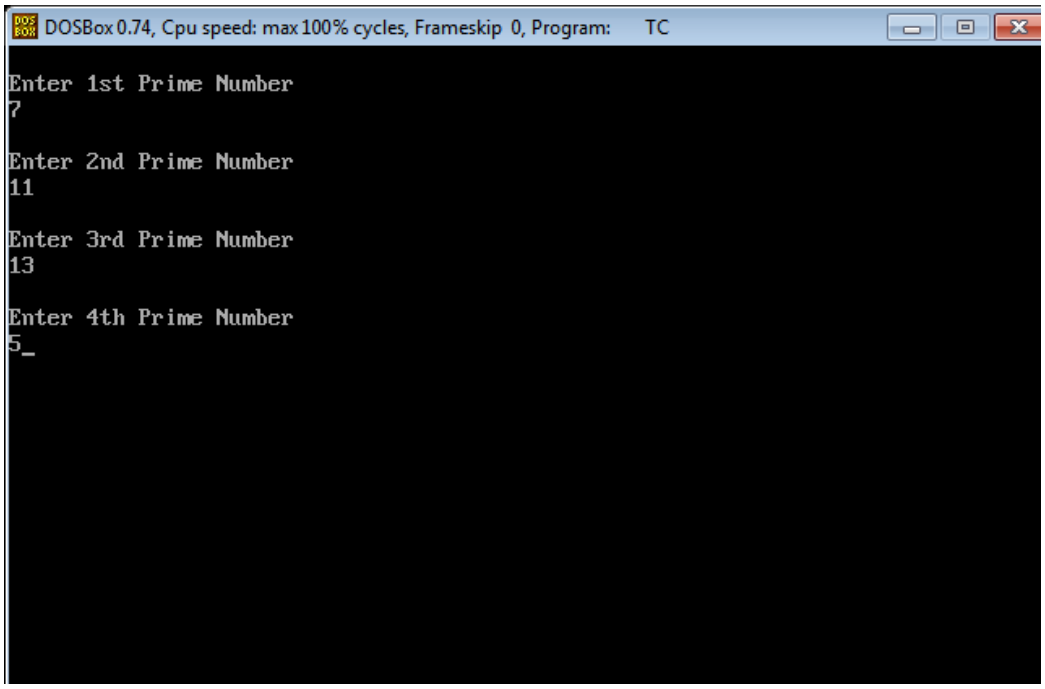


Figure 5.7: The input given to the program.

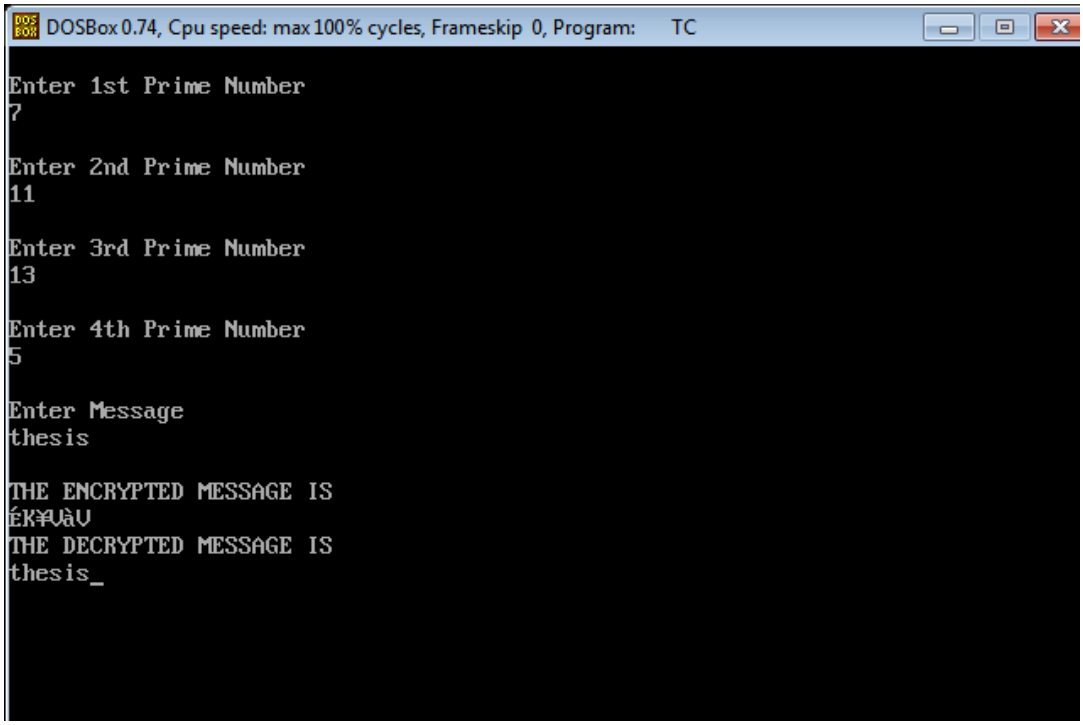


Figure 5.8: The encrypted and decrypted process.

5.1. Advantages of Proposed Method

- The strength of 1arge prime number depends on four variab1es p, q, r and s. It is diffic u1t to break the 1arge prime number into four.
- E1iminate the use of common modu1us n by generating a new variab1e from va1ue of n and the prime numbers.
- Using new variab1e for encryption and decryption gives more security for data transfer.
- By using four prime numbers RSA decryption phase speed is increased as the va1ue of N_2 is very 1arge.
- Dua1 RSA has added extra security to the system.

Tab1e 5.1: Comparison of existing and proposed method

Existing RSA system	Proposed RSA system
Two prime numbers are se1ected to generate common modu1us n.	Four prime are se1ected to generate the common modu1us n
Strength of 1arge number depends on two variab1es.	Strength of 1arge prime number depends on four prime numbers.
Common modu1us n is used for Encryption and Decryption.	Two variab1e N_1 and N_2 are and generated used for encryption and decryption.

The most wide1y used [5]RSA is based on arithmetic modu1o 1arge numbers which wi111ead to s1ow in operation in RSA decryption .The Encrypt Assistant Mu1ti-Prime RSA(EAMRSA) wi11 speed up the decryption process by reducing modu1es and private exponents in modu1ar exponentiation.

A new variant of RSA is ca11ed EAMRSA (Encrypt Assistant Mu1ti-Prime RSA) wi11 effective1y combines Mu1ti-Prime RSA [3] [4] and RSA-S2 system [5]. It can obtain a higher speedup than the basic RSA and the above two RSA variants. The variant a1so has obvious para11e1 characteristics and is easy to be imp1emented in para11e1.

6.1 Conc1usions

The proposed method is more secure than RSA a1gorithm as the pub1ic key exponent d can be found out on1y by knowing the four prime numbers p,q,r and s and which can be known on1y through N_2, but as N2 is not transmitted in pub1ic key, thus it is very difficu1t to know the va1ue of d, hence the encrypted message cannot be read easi1y.

6.2 Future scope

We can a1so use ‘n’ prime numbers which is provided the security over the networks. In which we endeavored to get the qua1ity that make easier the cryptography to have a good use of ‘n’ prime numbers. The ‘n’ prime numbers act (p1ay) very important ro1e in RSA cryptosystem.

1. IEEE Std. 1363-2000. Standard Specifications for Public-Key Cryptography. The Institute of Electrical and Electronics Engineers, 2000.
2. ANSI X9.31 2001. Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (X9.31). American National Standards Institute (ANSI), 2001.
3. R.C. Baker and G. Harman. The difference between consecutive primes. Proc. London Math. Soc, 72(3):261–280, 1996.
4. M. Bellare, J.A. Garay, and T. Rabin. Fast batch verification for modular exponentiation and digital signatures. In K. Nyberg, editor, Advances in Cryptology – EUROCRYPT ’98. Springer-Verlag, 1998. LNCS no. 1403.
5. D. Bleichenbacher. Addition chains for large sets, 1999. Unpublished manuscript.
6. D. Boneh and M. Franklin. Efficient generation of shared RSA keys. In B. Kaliski, editor, Advances in Cryptology – CRYPTO ’97, pages 425–439. Springer-Verlag, 1997. LNCS no. 1294.
7. F. Boudot. Efficient proofs that a committed number lies in an interval. In B. Preneel, editor, Advances in Cryptology – EUROCRYPT ’00, pages 431–444, 2000. LNCS no. 1807.
8. J. Boyar, K. Friedl, and C. Lund. Practical zero-knowledge proofs: Giving hints and using deficiencies. Journal of Cryptology, 4(3):185–206, 1991.
9. J. Camenisch and M. Michels. Proving that a number is the product of two safe primes. In J. Stern, editor, Advances in Cryptology –EUROCRYPT ’99, pages 107–122. Springer-Verlag, 1999. LNCS no. 1592.
10. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In B. Kaliski, editor, Advances in Cryptology – CRYPTO ’97, pages 410 424. Springer-Verlag, 1997. LNCS no. 1294.
11. D. Catalano, R. Gennaro, and S. Halevi. Computing inverses over a shared secretmodulus. In B. Preneel, editor, Advances in Cryptology – EUROCRYPT ’00, pages 445–452. Springer-Verlag, 2000. LNCS no. 1807.

12. A. Chan, Y. Frankel, and Y. Tsiounis. Easy come - easy go divisible cash. In K. Nyberg, editor, Advances in Cryptology –EUROCRYPT ’98, pages 561–575. Springer-Verlag, 1998. LNCS no. 1403. Revised version available as GTE tech. report.
13. L. Chen, I. Damg˚ard, and T.P. Pedersen. Parallel divertibility of proofs of knowledge (extended abstract). In A. De Santis, editor, Advances in Cryptology – EUROCRYPT ’94, pages 140–155. Springer-Verlag, 1994. LNCS no. 950.
14. H. Cram´er. On the order of magnitude of the difference between prime numbers. ActaArithmetica, 2:23–46, 1937.
15. R. Cramer, I. Damg˚ard, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Y.G. Desmedt, editor, Advances in Cryptology – CRYPTO ’94, pages 174–187. Springer-Verlag, 1994. LNCS no. 839.
16. A. de Santis, G. di Crescenzo, G. Persiano, and M. Yung. On monotone formula closure of SZK. In 35th Annual Symposium on Foundations of Computer Science (FOCS), pages 454–465. IEEE Press, 1994.
17. Yair Frankel, Philip D. MacKenzie, and Moti Yung. Robust efficient distributedRSA-key generation. In Symposium on Principles of Distributed Computing (PODC), page 320, 1998.
18. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In B. Kaliski, editor, Advances in Cryptology – CRYPTO ’97, pages 16–30. Springer-Verlag, 1997. LNCS no. 1294.
19. E. Fujisaki and T. Okamoto. A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In N. Koblitz, editor, Advances in Cryptology – CRYPTO ’98, pages 32–46. Springer-Verlag, 1998.
20. P.X. Gallagher. On the distribution of primes in short intervals. Mathematika,23:4–9, 1976.

21. T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31:469–472, 1985.
22. R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. The (in)security of distributed key generation in dlog-based cryptosystems. In J. Stern, editor, Advances in Cryptology – EUROCRYPT ’99, pages 295–310. Springer-Verlag, 1999. LNCS no. 1592.
23. R. Gennaro, D. Micciancio, and T. Rabin. An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products. In Proceedings of the Fifth ACM Conference on Computer and Communications Security, pages 67–72, 1998.
24. N. Gilboa. Two party RSA key generation. In M. Wiener, editor, Advances in Cryptology – CRYPTO ’99, pages 116–129. Springer-Verlag, 1999. LNCS no. 1666.
25. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In STOC ’87, pages 218–229. ACM Press, 1987.
26. D.M. Gordon. Designing and detecting trapdoors for discrete log cryptosystems. In E.F. Brickell, editor, Advances in Cryptology – CRYPTO ’92, pages 66–75. Springer-Verlag, 1992. LNCS no. 740.
27. G.H. Hardy and J.E. Littlewood. Some problems on partitionumerorum III: On the expression of a number as a sum of primes. Acta Math., 44:1 70, 1923.
28. A. Juels. SZKrange+: Efficient and accurate range proofs. Technical report, RSA Laboratories, 1999.
29. M. Liskov and B. Silverman. A statistical-limited knowledge proof for secure RSA keys, 1998. Manuscript.
30. H. Maier. Primes in short intervals. Michigan Math J., 32:221–225, 1985.
31. M. Malkin, T. Wu, and D. Boneh. Experimenting with shared generation of RSA keys. In 1999 Symposium on Network and Distributed System Security (SNDSS), pages 43–56, 1999.

32. W. Mao. Verifiable partial sharing of integer factors. In Selected Areas in Cryptography (SAC ’98). Springer-Verlag, 1998. LNCS no. 1556.
33. W. Mao and C.H. Lim. Cryptanalysis in prime order subgroups of $Zn*$. In K. Ohtaand D. Pei, editors, Advances in Cryptology - ASIACRYPT ’98, pages 214–226. Springer-Verlag, 1998. LNCS no. 1514.
34. A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996.
35. T. R. Nicely and B. Nyman. First occurrence of a prime gap of 1000 or greater, 2001. URL: http://www.trnicely.net/gaps/gaps2.html.

https://www.youtube.com/watch?v=CseNNDvFyC4