A Secure Method For Digital Signature Generation for Tamperproof Devices

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
Bachelor of Technology
In
Electronics and Communication Engineering
By

RAVI KUMAR SINGH(10609014)
And
JITEN KUMAR PATHY(10609019)

Under the guidance of
Prof G.S Rath

Department of Electronics and Communication Engineering
National Institute of Technology
Rourkela-769008

CERTIFICATE

This is to certify that the thesis entitled, **"A Secure Method For Digital Signature Generation for Tamperproof Devices"** submitted by RAVI KUMAR SINGH and JITEN KUMAR PATHY in partial fulfilments for the requirements for the award of Bachelor of Technology Degree in Electronics & Communication Engineering at National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

Prof.G. S. Rath
Dept. of Electronics & Communication Engineering
National Institute of Technology
Rourkela-769008

ACKNOWLEDGEMENT

We would like to express our deep gratitude for Prof. G.S.Rath, our project guide as well as an inspirational figure for carrying out this project. The endeavour of this nature couldn't have been attempted without reference to the the works of others whose details been given in the reference section. We acknowledge our indebtedness to all of them. We also thank our friends for their tolerance and patience during the span of this project.

RAVI KUMAR SINGH
(Roll No. 10609014)
JITEN KUMAR PATHY
(Roll No. 10609019)

CONTENTS

ABSTRACT

In the information age the security of information is one of the primary issues and any vulnerability in this regards can have devastating effects. Implementation of cryptographic algorithms to protect identification, authentication or data storage has been the prime focus in cryptographic arena specially for smaller handheld devices. This Project deals with implementation of efficient CRT-RSA algorithm for digital signature generation in smart cards and new scheme to make it secure against Bellcore attacks. Generally smartcards have very limited computational power so RSA-CRT is widely used in order to generate digital signature with a reasonably large key with reasonable speed. but despite being fairly tamperproof ,smartcards are vulnerable to side channel attacks like fault attacks, timing attacks etc. One of the simplest fault attacks is named Bellcore attack, which seriously compromises the security of the system because of it revealing the secret factorization of RSA modulus and nature of fault induced doesn’t matter. This project aims at implementing algorithm using RSA and Chinese remainder theorem which is secure against bellcore attack and alerts in case of fault.

CHAPTER 1

INTRODUCTION

Security means the protection of the information from any sort of unauthorised access or manipulation through eavesdropping or guessing, mathematical or probabilistic algorithms, ciphers and other methods. And cryptography is the science which is used to achieve these security goals.

In the modern age, it's not possible to permanently keep the information under tight container to achieve these security goals because of the most of systems are based on the free flow of these information and cost as well as efficiency issues make it not an option at all.

The main concerns of cryptography include Confidentiality, Integrity and Availability. Cryptography encrypts the static or travelling information so that unauthorised access to the information is not possible .regarding authentication ,cryptography is also used to make sure that rightful candidates from multiple parties attempting to get access to a system are authenticated properly and subsequently given access to system.

Cryptographical algorithms are used to encrypt the messages so that information is not extracted from by party other than to which it has been sent.

Integrity is achieved by cryptography by checking messages for any changes made by unauthorised entities or unauthorised methods and alerting the concerned parties regarding this. Digital signatures are used to achieve it.

Availability is also a component of information security. The information stored must be available to authorized parties

whenever required. And prevention of threat to this aspect is so achieved by using cryptographic methods.

Different types of cryptosystems are used for different purposes and in different conditions considering the speed, vulnerability or environment etc.

But cryptosystems can be widely classified into two main categories symmetrical and asymmetrical (or public key) cryptosystems. In symmetrical cryptosystems only one key is used by both the communicating parties for encryption and decryption.

In case of asymmetrical cryptosystems there is a set of two keys: public key and private key. Here public key is an open secret and can be used by anyone. Whereas private key is secret and is used by a single party.Despite being mathematically related these key are almost non obtainable from each other. In case of normal encryption the public key is used to encrypt and private key is used to decrypt the message but reverse is true for digital signatures.

CHAPTER 2

CRYPTOGRAPHIC ARITHMETICS

2.1 Introduction

Cryptographic algorithms are based on some very essential mathematical theory. Some of the theorems and algorithms have been discussed in this chapter. Which have been used in our planned work.

2.2 Modular Arithmetic

Modular arithmetic is a branch of arithmetic for integers, where numbers *wrap around* after they reach a certain value - the *modulus*. For any nonnegative integer a and positive integer n, if we divide a by n, we get an integer remainder r and an integer quotient q that obey the following relationship:

$a= kn+r$; $0 \leq r<n$; $q=\lfloor a/n \rfloor$

The remainder r is often referred to as a residue. If **a** is an integer and n is a positive integer, we define a mod n to be the remainder when **a** is divided by n. The integer n is called the modulus.

Thus, for any integer a, we can always write:

$a=\lfloor a/n \rfloor \times n + (a \bmod n)$

Properties of Modular Arithmetic for integers

Property	
Commutative laws	$(w + x) \bmod n = (x + w) \bmod n$ $(w \times x) \bmod n = (x \times w) \bmod n$
Associative laws	$[(w + x) + y] \bmod n = [w + (x + y)] \bmod n$ $[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n$
Distributive laws	$[w \times (x + y)] \bmod n = [(w \times x) + (w \times y)] \bmod n$ $[w + (x \times y)] \bmod n = [(w + x) \times (w + y)] \bmod n$
Identities	$(0 + w) \bmod n = w \bmod n$ $(1 + w) \bmod n = w \bmod n$

2.3 Euclidean Algorithm

Euclidean algorithm is a simple and efficient method to find the HCF two positive integers. Nonzero b is known to be a factor of a if a = nb for some n, where a, b, and n are integers. We will use the notation gcd (a, b) to mean the highest common factor of **a** and **b**. The positive integer **c** is defined to be the greatest common divisor of **a** and **b** if **c** is a divisor of **a** and of **b** and any divisor of **a** and **b** is a divisor of **c**.

An equivalent definition is the following: gcd(a, b) = max(p, such that p|a and p|b).

Because we require that the greatest common divisor be positive, gcd (a, b) = gcd (|a|, |b|).

Two integers **a** and **b** are coprime if their only common positive integer factor is 1. This is other way of saying that a and b are relatively prime if gcd (a,b)= 1.

The Euclidean algorithm's base theorem is

"For any positive integer b and any nonnegative integer a, **gcd (a,b) = gcd(b, a mod b)".**

2.4 Prime Numbers

An important aspect of Modular and cryptoarithmetic is the study of prime numbers. t. An integer p > 1 is a prime number if and only if its only divisors are ± 1 and ±p. Prime numbers play a critical role in cryptographic techniques discussed later. If P is the set of all prime numbers, then any positive integer a can be written uniquely in the following form:

$$a = \prod_{p \in P} p^{a^p} \quad where\ each\ a_p \geq 0$$

It is easy to find the highest common factor of two positive integers if we could express each integer as the product of prime factors.

The right-hand side is the multiplication of all possible prime numbers p; for any particular value of a, most of the exponents a^p be 0.

For many Cryptosystems, it is mandatory to select one or more very large prime numbers at random.
But barring some probabilistic algorithms there is no any simple yet efficient algorithm is known for primality test of a large number.

Miller Rabin primality test:-

One of the Primality testing Algorithms is Miller Rabin primality test.

Now, let n be an odd prime. Then $n-1$ is even and we can write it as $2^s \cdot d$, where s and d are positive integers (d is odd).

For each $a \in (\mathbb{Z}/n\mathbb{Z})^*$, either

$$a^d \equiv 1 \pmod{n}$$

or

$$a^{2^r \cdot d} \equiv -1 \pmod{n}$$ for some $0 \leq r \leq s-1.$

So if we can find an a such that

$$a^d \not\equiv 1 \pmod{n}$$

and

$$a^{2^r d} \not\equiv -1 \pmod{n}$$ for all $0 \leq r \leq s-1$

then a is a witness for the compositeness of n (sometimes misleadingly called a *strong witness*, although it is a certain proof of this fact). Otherwise a is called a *strong liar*, and n is a strong probable prime to base a. The term "strong liar" refers to the case where n is composite but nevertheless the equations hold as they would for a prime.

2.5 Fermat's And Euler's Theorems

Euler's theorem and Fermat's theorem play crucial role in public key cryptography. Fermat's theorem states that: *If p is prime and a is a positive integer not divisible by p*, then
$a^{p-1} \equiv 1 \pmod{p}$

Euler's totient function is also an important entity in number theory and written ϕ (n), and defined as the number of positive coprime integers less than n .By convention, ϕ (1) = 1.
if we have two prime numbers p and q, with p not equal q. Then it can be show that for n = pq, ϕ (n) = ϕ (pq) = ϕ (p) x ϕ (q) = (p-1) x (q-1)
Euler's theorem states that for *every **a** and **n** that are relatively prime*:
$a^{\varphi(n)} \equiv 1 (\text{mod } n)$.

2.6 The Chinese Remainder Theorem

The Chinese remainder theorem (CRT) states that
Integers in certain range can be reconstructed from their residues modulo a set of pair wise relatively prime moduli.
One of the mostly used features of the Chinese remainder theorem is to manipulate (potentially very large) numbers mod M in terms of tuples of smaller numbers. This can be useful when M is 150 digits or more. However, note that it is necessary to know beforehand the factorization of M.

Theorem 1. Let **m** and **n** be integers with gcd (**m, n**) =1, **M**=**m*n** and let b and c be any integers. Then the simultaneous congruences
$x \equiv b (\text{mod } m)$ and $x \equiv c(\text{ mod } n)$
have exactly one solution with **$0 \le x \le M$.**

Proof: We begin by solving the congruences **T**he solution consists of all numbers of the form **$x = my + b$**. We substitute this into second congruence, which yields
$my \equiv c - b (\text{mod } n)$
We are given that gcd (m, n) =1, so the linear congruence Theorem tells us that there is exactly one solution y_1 with $0 \le y_1 < n$. Then the solution to the original is given by $x_1 = my_1 + b$; and this will be the only solution x_1 with $0 \le x_1 < M$, since there is only one between 0 and n, and we multiplied y_1 by m to get x_1. This completes the proof.

Theorem 2. Let $m_1(x), m_2(x), ..., m_r(x)$ denote r prime polynomials of degree p ($p \geq 1$) that are relatively prime in pairs, and let b_1 , b_2 ,..., b_r denote any r prime polynomials of degrees at most p-1. Then the system of congruences

$x \equiv b_i (\mathrm{mod}\ m_i(x))$ i =1,2,3,...,r has a unique solution g(x), where

$$g(x) = \prod_{i=1}^{r} m_i(x).$$

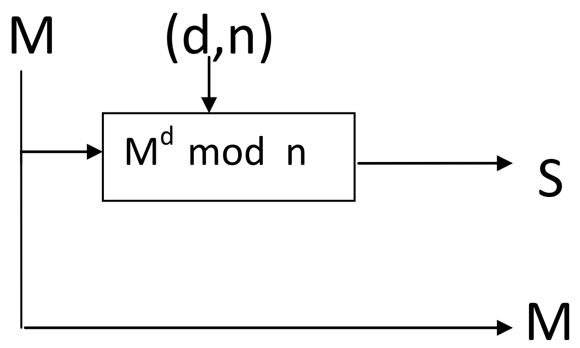
CHAPTER 3

REVIEW

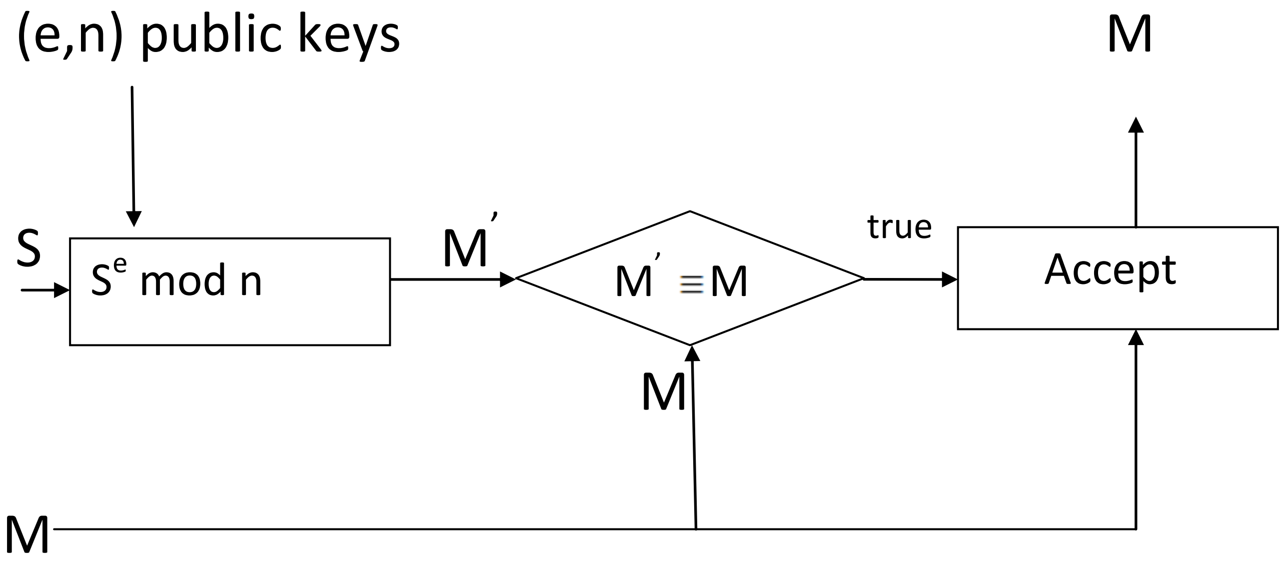
3.1 RSA Digital Signature Scheme

The RSA idea is also used in for signing and verifying a message .In this case it is called RSA Digital signature scheme. The digital signature scheme changes the roles of public and private keys. First the private and public keys of the sender not the receiver are used. Second the sender uses her own private key to sign the document; the receiver uses the senders public key to verify it.
The conventional Chinese remainder theorem (CRT) is to determine a single integer from its remainders from a set of moduli. It has tremendous applications in various areas, such as cryptography and digital signal processing.
The Chinese Remainder Theorem (CRT) allows for an efficient implementation of the RSA algorithm. The theorem is as follows. Given input, m, raise it to the e-th (or d-th) power modulo p and modulo q. The intermediate results are then combined through multiplication and addition with some predefined constants to compute the final result (the modular exponentiation to N). This approach is often used for implementing RSA in embedded systems. It requires four times less execution time and smaller amount of memory for intermediate results, since modular exponentiation is performed on half the bit size of N.

Private key



(a) Signing



(b)Verifying

Chinese Remainder Theorem has been used for hundreds of years and has been applied to many domains such as integers and polynomials as explained in last chapter. It can also be modified to design proxy signatures

3.2 **Use of Chinese remainder theorem and RSA in smartcards & it's vulnerability towards fault attacks**

Smartcards play an important role in modern cryptography. Smartcards are used to compute digital signatures,most notably digital signatures based on RSA. Since speed is still an issue with modern smartcards, enhancements have been adopted to the plain RSA signature algorithm. The
most common enhancement is the computation of an RSA signature using the Chinese Remainder Theorem (CRT).We will refer to this variant of RSA as CRT-RSA. With CRT-RSA one can expect speed up by a factor of 4 compared to plain RSA.
However, smartcards are not as tamper-resistant
as one may wish. Hence side-channel attacks like fault, power, and timing attacks, on smartcards have attracted a lot of attention.

Among the side-channel attacks, fault attacks seem to be easiest to realize. In particular, CRT-RSA proved to be susceptible to fault attacks. In an
extremely simple attack on CRT-RSA is described. Named the Bellcore attack, this attack reveals the secret factorization of the RSA modulus N by introducing a single fault resulting in a signature that is correct modulo one of the secret prime factors of N, but faulty modulo the other prime
factor. This attack is particularly devastating because the type of fault induced is irrelevant.

CHAPTER 4

DIGITAL SIGNATURES AND SECURITY

4.1 Digital Signature Generation using CRT-RSA Algorithm

In case of digital signature generation sender uses a one way hash function to calculate a message digest and then he signs it using private key and receiver verifies the integrity of message using the public key of the sender.

4.1.1 Basic CRT-RSA algorithm

Key generation

1. Select two distinct prime numbers p and q
2. Compute n = pq.
3. Compute euler's phi totient, $\phi = (p-1)(q-1)$
4. Select public key e < n such that gcd(e, phi)=1
5. Compute $d = e^{-1}$ mod phi.
6. Compute $dP = d^{-1}$ mod (p-1).
7. Compute $dQ = d^{-1}$ mod (q-1).
8. Compute $qInv = q^{-1}$ mod p where p>q.

Signing:-

1. Compute $mP=M^{dP}$ mod p,where M is hash of the message.
2. Compute $mQ=M^{dQ}$ mod q.
3. Compute h= qInv(mP-mQ) mod p.
4. Return: Sig=mQ+h*q.

Verification:-

1. Compute $M' = Sig^e$ mod N.
2. Compare M and M' ,where M is the hash of the received message.
3. If($M \equiv M'$) then accept.

4.1.2 Vulnerablity of CRT-RSA Against Fault Attacks

It's been showed that if an error can be introduced in mP or mQ(not in both) the the factorization of N can be found out from the faulty signature and a correct one.

Suppose error has been introduced in mP making it mP' and the resulting signature is Sig' then the value of q can be found out by

$gcd((Sig')^e - M \pmod N),N)=q$;

And p=N/q and so secret keys are recovered.

4.2 Modifications in CRT-RSA to Counter Fault Attacks

Several kinds of countermeasures against fault attacks(Bellcore attacks) have been suggested, for example to compute a signature twice and compare the two results or to verify the result with the public key before outputing. However, these two methods are too costly to be of practical interest.so we go for few other more effective and efficient countermeasures discussed ahead.

4.2.1 Shamir's Countermeasure

The idea consists in computing the two half exponentiations, mP and mQ in a redundant way.

Let t denote a random k-bit integer for some security parameter k(typically k = 32). Then the device computes

$mP^* = M^d \bmod tp$ and $mQ^* = M^d \bmod tq$

and outputs

$$\begin{cases} S = CRT(mP^*, mQ^*) \bmod N, & \text{if } mP^* \equiv mQ^* \pmod t \\ \text{Error} & \text{,otherwise} \end{cases}$$

4.2.2 A Modified Approach by BlÄomer, Otto and Seifert

They have proposed a new idea to protect every aspect of the computation of signature generation including the CRT combination.

It can be achieved by using two small integers t_1 and t_2 to compute $mP = M^d \bmod pt_1$ and $mQ = m^d \bmod qt_2$.These values are combined to S mod Nt_1t_2 via the CRT.

Precomputational Step

A precomputational step is followed at the production time in order to generate a valid RSA key with (N,e) sa public key where N= p*q and the corresponding private key d, where $e*d \equiv 1 \bmod \phi(N)$.And additionally the two small integers t_1 and t_2 are also generated whic must allow the certain conditions to make sure the scheme is completely secure:-

1. $gcd(t_1,t_2)=1$.
2. $gcd(d,\varphi(t_1))=gcd(d,\varphi(t_2))=1$.
3. t_1 and t_2 are squarefree.
4. $t_i \equiv 3 \bmod 4$ for $i \in \{1,2\}$.
5. t_2 doesn't divide $X= pt_1*((pt_1)^{-1} \bmod qt_2)$,where $pt_1=p*t_1$ and $qt_2=q*t_2$.

Algorithm(Infective CRT-RSA)

1. Compute $dP= d \bmod \varphi(pt_1)$.
2. Compute $dQ= d \bmod \varphi(qt_2)$.
3. Compute $et1 = dP^{-1} \bmod \varphi(t_1)$.
4. Compute $et2 = dQ^{-1} \bmod \varphi(t_2)$.
5. Compute $mP= M^{dP} \bmod pt_1$.
6. Compute $mQ= M^{dQ} \bmod qt_2$.
7. Compute $qt_2Inv = qt_2^{-1} \bmod pt_1$.
8. Compute $h = (qt_2Inv * (mP-mQ)) \bmod pt_1$.
9. Compute $s= mQ+ h* qt_2$.
10. Compute $c_1=(M-(s\hat{}et_1)+1) \bmod t_2$.
11. Compute $c_2=(M-(s\hat{}et_2)+1) \bmod t_1$.
12. Return:

$$\begin{cases} Sig = (s\hat{}(c_1*c_2)) \bmod N & ,\text{if } c_1=c_2=1; \\ Error & ,\text{otherwise} \end{cases}$$

4.3 SOFTWARE USED : MATLAB with VPI(Variable Precision Integers):-

When symbolic toolbox in MATLAB is not available VPI is best alternative to play with large integers written by John D'Errico.This is well documented.

To use the vpi (Variable Precision Integer) arithmetic tools, we will need to have the top level directory (VariablePrecisionIntegers) on our MATLAB search path.

Main Functions used from VPI in this project are described below with examples:-

- vpi - Creator function for a variable precision integer

>>vpi('4658675395636539563983939649349221012460124720234835364873682145')

ans =
4658675395636539563983939649349221012460124720234835364873682145

- randint - Generate random integers from the set [1:n], uniform, with replacement

>>randint(vpi('465867539563653956794893219045658694021648353648736 82145'))

ans =
615110424709012000317824023311826394372187653396159898 7

- nextprime -Returns the smallest prime larger than an integer input.

>>nextprime(vpi('4582891463982692165197896195015872326354327853419723654298182834'))

ans =
4582891463982692165197896195015872326354327853419723654298183219

- powermod - Compute mod(a^d,n)

>>powermod(vpi('4582891463982692165197896195015872326354327853419723654298182834'),vpi('4658675395'),vpi('4658675395636539563983939649349221012460124720234835364873682145'))

ans =
4639583415682935546058548380065444993081994590468820544499717424

- totient - Compute euler’s totient function

>>totient(vpi(‘2100289417064542343925053224693817826511802246632094
31069478460157826527761769759358877591139568786214009976311617’)
)

ans=

210028941706454234392505322469381782651180224663204848178014477
419832415225747821834572274861556643027083379945556

- minv - Compute multiplicative inverse modulo

>>minv(65537,vpi('458289146398269216519789619501587232635432785341
9723654298183219'))

ans =
405255518275461127377990567450073773485530863617854410390378888
4

CHAPTER 5

IMPLEMENTATION RESULTS

5.1 Digital Signature Scheme using RSA with CRT

5.1.1 RSA with CRT for Signature Generation By sender

M =
342784643950386265423498759365173387345894536257824864538258238
435735632856467589734632524945654689436325598674569022519042876
64

p =
104915396960119927201630972583681496129259704522616388700181806
740437

q =
106201486708719353119399266190219638979391534377188310570290153
797219

d =
953095371886047153145368193398489390567899643644764532863576928
650047669134244282697664126635440777673298940137494620328270193
50042097

dp =
518885936443286571003626152304021901221858614628616660350487953
65637

dq =
107113817712839300565974815679288312503437453992891760817495142
6827

qinv =
878585031551241112293783371923448378096472363129827559969160205
08914

Sig =
202037638221536373981256290516149217930873579606432767140255312
397839776319492297092888274576594724429327437395360847938950902
9516144744

5.1.2 Signature Verification at reciever

Sig =
20203763822153637398125629051614921793087357960643276714025531239783977631949229709288827457659472442932743739536084793895090295161447 44

e = 65537

N =
111421711358001912728662139298531571868798537218256677343967221044051018427497224881414843172046476479487902894799091684604878 04665444703

M1 =
3427846439503862654234987593651733873458945362578248645382582384357356328564675897346325249456546894363255986745690225190428766 4

As we see Here M1= M .So Integrity of message is intact.

5.2 Signature Scheme using BlÄomer, Otto and Seifert Algorithm immune against Bellcore fault attack

5.2.1 Signature generation

p =

2139282821159865272539411662776441773409368002071637769806979418
49920059

q =

2199836613845719643752249882500044737014200690401035597379534884
7836989

t1 =

11

t2 =

375

dP =
1881290607519950244875000810617815574463455297169469585766720594
750016429

dQ =
3196509586879883076971100378420152895154752378574577346623099172
153029809

qt2inv =
2187331573318026287676711877682844027881606671247725376700642508
35051662

M=

7538495656467234356459723646648625423498067054286109374563940024
5656774502536246250967485629198045364530962352769754734906756923
46236569032

mP =
21128369662656767202252889805004049190316775311952534045226897020047916 92

mQ =
4621245794400843824104575697080704840364108931450282295285557842583325582

h =
2264046227225286429440510410525678701027301755880411667574830241449958617

s =
18676994197835440708585764504622923905870199160401946650939727250775146754917792001767558963719864790473214638101074964054725466856666847039905457

et1 =

9

et2 =

89

N =
47060726773586362580760506163268276519914587154248570092622753153582713221891709941907835760432399709921145622633977300200667626188447132623 51

Sig =
32978140763720365399956590380717827680909775961140381870187994335261484711614968185297339802885855765040550399127713351005527850910248148966 89

5.2.2 Verification Of Signature

Sig =
32978140763720365399956590380717827680909775961140381870187994335261484711614968185297339802885855765040550399127713351005527850910248148966 89

e =65537

N =
4706072677358636258076050616326827651991458715424857009262275315358271322189170994190783576043239970992114562263397730020066762618844713262351

M1 =
75384956564672343564597236466486254234980670542861093745639400245656774502536246250967485629198045364530962352769754734906756923462365690 32

Here again we find that M1=M .Thus message or Signature is not tampered.

CONCLUSION

The above project was under taken in order to implement efficient and safer algorithms for Digital Signature of RSA system in Cryptography. Various types of Symmetrical and Asymmetrical Cryptography methods were studied along with their mathematical backgrounds. First RSA with CRT was used to speedily generate digital signature and verification and Finally a modified form of CRT-RSA secure against Fault attack was implemented with MATLAB.

REFERENCES

[1] W.Stallings; “Cryptography and Network Security” 2nd Edition, Prentice Hall, 1999

[2] Bruce Schneir: Applied Cryptography, 2nd edition, John Wiley & Sons, 1996

[3] Cryptography and Network Security – Behrouz Forouzan

[4] A. Shamir. Method and apparatus for protecting public key schemes from timing and fault attacks, 1999. US Patent No. 5,991,415, Nov. 23, 1999.

[5] Practical Fault Countermeasures for Chinese Remaindering Based RSA, By
Mathieu Ciet and Marc Joye.

[6] A New CRT-RSA Algorithm Secure Against Bellcore Attacks, Johannes Bl¨omer, Martin Otto, Jean-Pierre Seifert. CCS’03, October 27–30, 2003, Washington, DC, USA.

[7] S.-M. Yen, S. Kim, S. Lim, and S. Moon. RSA speedup with Chinese remainder theorem immune against hardware fault cryptanalysis. IEEE Transactions on Computers 52(4):461{472, 2003.

[8] http://www.di-mgt.com.au/rsa_alg.html

[9]
http://www.mathworks.com/matlabcentral/fileexchange/22725-variable-precision-integer-arithmetic

[10] A novel method of encryption using modified rsa algorithm and chinese remainder theorem, Sangeeta Patel and Partha Prittam Nayak,Nit Rourkela