Improvement in RSA Cryptosystem

Seema Verma
EIT/CSE Deptt,Faridabad, India
Email: seemaknl@gmail.com

Dr Deepak Garg
Thapar University/CSE Deptt, Patiala, India
Email: deep108@yahoo.com

***Abstract*-Boneh and Shacham gave a nice survey on four variants (Batch RSA, MultiPrime RSA, MultiPower RSA, Rebalanced RSA). Rebalanced RSA and MultiPower RSA were then combined to increase the decryption/signature generation performance. This combination theoretically improves the decryption/signature generation time about 14 times than RSA with CRT and about 56 times than the standard RSA with key size 2048bits. On the encryption side, it increased the encryption time, thus making encryption/signature verification very costly. Here in this paper we further tried to increase the encryption/signature verification performance. The proposed scheme is semantically secure also.**

Index Terms: cryptosystem, encryption, signature verification, RSA-CRT, semantic security

I. Introduction

The RSA cryptosystem [1] is still a de-facto standard in all branches of public key cryptography. However, it is rapidly losing its attractiveness. This is mainly due to the enormous computation involved. Public key algorithms are 100 times slower than private key algorithms. Many researchers are trying to improve the computational efficiency.

In 2002, Boneh and Shacham [2] gave a very nice comparison of the variants of RSA (Batch RSA [3], MPrime RSA [4], MPower RSA [5], and Rebalanced RSA [6]). All these variants are improving the decryption/signature verification performance. Their work was further extended by Caesar [7] in 2003. He combined the two variants of RSA, MPrime RSA and Rebalanced RSA and gave the performance by improving the decryption/signature generation speed by 27 times to RSA and by 4.8 times to RSA with CRT for 2048 bit moduli. It is further improved by the combined approach of MPower RSA and Rebalanced RSA (i.e. RePower RSA [8]). RePower RSA gave the performance by theoretically improving the decryption/signature generation performance by 14 times to RSA with CRT and 56times to standard RSA for 2048 bit moduli. Here in this paper we extending the work of RePower RSA, we are trying to increase the encryption speed with less compromising the decryption speed of RePower RSA.

1

[1] Footnote: Manuscript received June 6,2011; revised July 8,2011

The scheme is providing better performance; besides this it is providing semantic security, whereas RePower RSA is not having semantic security.

This is achieved with the help of DRSA [9] cryptosystem which is having the property of semantic security. The new scheme is also giving better performance as compared to DRSA cryptosystem.

In second section of the paper RSA cryptosystem and its variants are explained. In third section DRSA cryptosystem is reviewed. Then the proposed scheme is given to improve the encryption performance of RePower RSA and the security analysis is given with the comparison of proposed scheme with RePower RSA and DRSA. The paper is then concluded in the last section.

II. RSA

The RSA algorithm [1] was publicly described in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman at MIT. In cryptography, RSA is an algorithm for public-key cryptography. It is the first algorithm known to be suitable for signing as well as encryption, and one of the first great advances in public key cryptography. RSA is widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations. Public-key cryptography utilizes an asymmetric cryptography technique with two keys, one public and one private. The keys are derived from the multiple of two large prime numbers. The private key can only be deduced from the public key by factoring the large multiple. RSA's security comes from the difficulty in factoring very large numbers. Techniques for factoring numbers are improving, but the speed of all depend on the size of the number, which means they still take significant time. While the possibility exists that one day there will be an extraordinary leap in our ability to factor large numbers, it is unlikely and offers a minimal threat to RSA.

A. Working of RSA

In RSA [1] two keys are required, first is e and N (n bits) public and second is a number d that is kept secret. In order for A to send a message to B, A looks up B's public values and, if the message is M (written as a number), then A divides the message into pieces of size less than N and sends $C = M^e \bmod n$. Then B decodes by $M = C^d \bmod N$. The security of the system lies in the choices of the public and private keys.

Key Generation of RSA

doi:10.4304/jait.2.3.146-151

Step 1. Two large random primes(p and q) of n/2 bits are generated such that their product N = pq is
Step2. N = pq and φ(N) = (p-1) (q-1).
Step3. Select an integer e ($1 < e < \varphi(N)$), such that gcd(e, φ(N)) = 1
Step4. Now Compute d, $1 < d < \varphi(N)$, such that $ed \equiv 1 \pmod{\varphi(N)}$.
Public key=(N, e) and
Private key=(N, d).
Encryption of RSA
C=ciphertext, M= plaintext
Step1. Represents the plaintext message as a positive integer M.
Step2. $C=M^e \bmod N$.
Decryption of RSA
Step1. $M = C^d \bmod N$.
Step2. Extracts the plaintext from the corresponding integer message M.

Here we are concerned with decryption speed of the algorithm. Factors N and d are involved in the computation. Both are large numbers(approx. n bits long). The complexity of this algorithm is $O(n^3)$.

RSA Variants

In 1982, Quisquater and Conver [10] introduced a fast deciphering algorithym to speed up the decryption process via Chinese Remainder Theorem, which was called QCRSA. It improves the standard RSA by the factor of 4. In 2002 Boneh and Shacham [2] gave the comparison of RSA variants (Batch RSA [3], MPrime RSA [4], MPower RSA [5] and Rebalanced RSA [6]) to enhance the performance of RSA cryptosystem. Here these variants are reviewed. RPrime [7] given by Caeser in 2002 is also given here. There comparisons are explained in [11]. Then the combined approach of MPower RSA and Rebalanced RSA (RePower RSA [8]) is described.

B. Batch RSA

Batch RSA variant [3] was introduced in 1989. In this variant if small public exponents are used for the same modulus N, the decryption of the two ciphertext can be done at the cost of one. Suppose C1 and C2 are the two ciphertext from message M1 and M2 respectively. There public keys are <N, 3> and <N, 5> respectively. To decrypt $C_1^{1/3} \bmod N$ and $C_2^{1/5} \bmod N$ are calculated. According to this variant both decryption processes can be merged to enhance the speed of the decryption algorithm.

Assume , $A= (C_1^5.C_2^3)^{1/15}$
Step1. $C_1^{1/3}= A^{10}/ [C_1^3.C_2^2]$
Step2. $C_2^{1/5}= A^6/ [C_1^2.C_2]$

Now we are able to decrypt both C_1 and C_2 at the cost of computing 15^{th} root. It takes the same time as a single RSA decryption and some additional arithmetic is also involved. But the condition for this technique is small values of public exponents(e1,e2). Otherwise it will not increase the decryption speed; rather it will be more expensive. Also the modulus must be the same and public exponents must be distinct for both the messages.
The generalization of this method is given for b RSA ciphertexts. For the implementation of the technique, b relatively prime public keys e1, e2,…..eb sharing common modulus N are used. This technique describes b-batch process using a binary tree. For each i one computes M_i as:
$M_i=C_i^{1/ei}=A^{\alpha i}/ [C_i^{(\alpha i-1)/ei}.\prod_{j\neq i} C_j^{\alpha i/ej}$
Where $\alpha_i=1 \bmod e_i$
And $\alpha_i=0 \bmod e_j$ (for $j\neq i$)

Here since b and e_i's are small, the exponents in this equation are also small.
Performance of Batch RSA

Batch RSA decrypts simultaneously b messages with the approximate cost of a single exponentiation (of order of N) and some small exponentiations (using public exponents). According to Fiat [3] with standard 1024-bit keys, batching improves performance significantly.

With b=4, RSA decryption is enhanced by a factor of 2.6, with b=8, by a factor of almost 3.5.

C. MultiPrime RSA

MultiPrime RSA[4] was introduced in 1998. The RSA modulus was modified so that it consists of k primes p_1, $p_2 \ldots p_k$ instead of using only two. The algorithms of Key generation, Encryption and Decryption are described as:
Key generation of MultiPrime RSA

The key pairs (public and private) are generated according to the following steps (here k is the number of primes to be used in the variant):
Step1. Compute k distinct primes p1... pk each (n/k) bits in length such that $N=\prod_{i=1}^{k} p_i$.
Step2. Compute e and d such that $d = e^{-1} \bmod \Phi(N)$, where $\gcd(e, \Phi(N)) = 1$ and $\Phi(N)=\prod_{i=1}^{k} (p_i-1)$
Step3. For $1\leq i\leq k$, compute $d_i = d \bmod (p_i - 1)$.
Public key = <N, e>,
Private key = <N, d_1, d_2 ...dk>
Encryption of MultiPrime RSA

Encryption is same as in the original RSA, thus
$C = M^e \bmod N$.
Decryption of MultiPrime RSA

The decryption is an extension of the Quisquater-Couvreur method. To decrypt a ciphertext C,
Step1. Calculate $M_i = C^{di} \bmod p_i$ for each i, $1\leq i\leq k$.
Step2. Apply the CRT to the M_i's to get $M = C^d \bmod N$.
Performance of MultiPrime RSA

The theoretical speedup[11] of this variant to the CRT RSA is given as follows:

$S_{CRT}= (2. (n/2)^3)/ (k. (n/k)^3)= k^2/4$

D. MultiPower RSA

MultiPower RSA [5] was introduced in 1998. In this variant [5], $N= p^{k-1}q$ where p and q are n/k bits. The three algorithms are described as below:
Key generation of MultiPower RSA
Step1. Generate two primes p and q of [n/k]-bits each and compute $N= p^{k-1}.q$
Step2. Compute $d=e^{-1} \bmod (p-1)(q-1)$
Step3. Compute d1=d mod (p-1) and d2=d mod (q-1).
Public Key=<N, e>
Private Key=<p, q, d1, d2>
Encryption of MultiPower RSA :

Same as in standard RSA.
*Decryption of MultiPower RSA***:**
Step1. $M_1= C^{d1} \bmod p$ and $M_2=C^{d2} \bmod q$
Step2. $M_1^e=C \bmod p$ and $M_2^e=C \bmod q$.

Step3. Using Hensel lifting[12], compute M_1' , such that $(M_1')^e$=C mod p^{k-1}
Step4. Using CRT, compute M such that M= M_1' mod p^{k-1} and M= M_2 mod q and then M=C^d mod N.

Performance of MultiPower RSA:

Decryption takes two full exponentiation modulo (n/k)-bit numbers and k-2 Hensel lifting [12]. Hensel lifting is much faster than exponentiation. So the speedup [11] over standard RSA (with CRT) is approximately:

$$S_{CRT}= (2. (n/2)^3)/ (2. (n/k)^3)= k^3/8$$

For k=3 the theoretical speedup is about 3.38

E. Rebalanced RSA

Rebalanced RSA[6] was introduced in 1990. This was designed according to specific requirements. E.g., in some applications, user would like to have the reverse behavior of the standard RSA, i.e. fast signature generation/decryption time, e.g., when a cell phone needs to generate an RSA signature that will later be verified on a fast server, one would like signing to be easier than verifying. Rebalanced RSA [6] fulfills this requirement by improving the performance of the decryption/signing algorithm by displacing the work to the encryption/verification algorithm. Because of the security reasons one can't choose small value of d. The values less than $N^{0.292}$ for d are insecure. So d is chosen as a large number, say of the order of N, but d mod p-1 and d mod q-1 are small numbers. Three algorithms of key generation, encryption and decryption are as follows:

Key generation of Rebalanced RSA

s<=n/2 bits
Step1. Generate two distinct random (n/2)-bit prime numbers p & q with gcd(p-1,q-1)=2, and calculate N=pq.
Step2. Generate two s-bits random numbers d_p and d_q, such that gcd(d_p,p-1)=1, gcd(d_q,q-1)=1 and d_p=d_q mod 2.
Step3. Calculate one d such that d= d_p mod p-1 and d=d_q mod q-1
Step4. Calculate e=d^{-1} mod Φ(N)
Public key =<N,e>
Private key=<p,q,d_p,d_q>

Encryption of Rebalanced RSA:

Same as in Standard RSA but with higher computation cost (e is large)

Decryption of Rebalanced RSA:

Same as in RSA with CRT.

Performance of Rebalanced RSA:

Speedup[11] over standard RSA with CRT is:

$$S_{CRT}=n/2s$$

For s= 160, the theoretical speedup is 6.4 times than RSA with CRT.

Rebalanced approach makes RSA encryption very time-consuming because the public exponent e in Rebalanced RSA-CRT will be of the same order of magnitude as Φ (N). Improvement over this was done by Galbraith [13] and Sun [14] to further decrease the public exponent e that is much shorter than Φ (N). It is further improved by Hinek in [15].

F. RPrime RSA

This variant[7] was proposed by Cesar Alison in 2002. The idea was to combine the two RSA variants Rebalanced RSA and Mprime RSA to further enhance the decryption speed. The general idea of this scheme is to use the key generation algorithm of Rebalanced RSA (modified for k primes) together with the decryption algorithm of Mprime RSA. The key generation, encryption and decryption algorithms are as follows:

Key generation of RPrime RSA

The key generation algorithm takes an integer s≤n/k and executes the following steps:
Step1.Generate k distinct random primes of n/k bits p_1, p_2..., p_k, with gcd(p_1 - 1, p_2 - 1, ..., p_k - 1) = 2; and calculate N=$p_1p_2...p_k$.
Step2. Generate k random numbers of s-bits d_{p1} , d_{p2} ..., d_{pk} , such that gcd(d_{p1} , p_1 - 1) = 1, gcd(d_{p2} , p_2 - 1) = 1, ..., gcd(d_{pk} , p_k - 1) = 1 and d_{p1} = d_{p2} = ... = d_{pk} mod 2.
Step2. Find d such that d = d_{p1} mod (p_1 – 1), d = d_{p2} mod (p_2 -1)…d = d_{pk} mod (p_k – 1)
Step3. Calculate e =d^{-1} mod Φ(N).
Public key = <N, e> and Private key = <p_1, p_2... p_k, d_{p1}, d_{p2} ...d_{pk}>

Encryption of RPrime RSA

Again, encrypting with the public key <N, e> is identical to the original RSA. However, that as in Rebalanced RSA, the public exponent e is much larger than the normally used e, and thus, the entity encrypting the message M must be prepared to use such an exponent.

Decryption of RPrime RSA

Same as in MultiPrime RSA decryption

Performance of RPrime RSA

The theoretical speedup (SQC), is therefore given by:

$$S_{CRT} = [nk]/4s$$

Comparison of all above variants is given in [11].

G. RePowerRSA (Combined Approach of MPower RSA and Rebalanced RSA)

In this approach [8] Rebalanced RSA [6] and MultiPower RSA [5] are combined to improve the decryption performance. The general idea of this scheme is to use the key generation algorithm of Rebalanced RSA with only two primes of n/k bits length and decryption algorithm of MultiPower RSA. The three algorithms for the new scheme are as follows:

Key Generation

Step1. Generate two distinct [n/k]-bit primes , p and q, with gcd(p-1,q-1)=2, and calculate N=p^{k-1}.q
Step2. Generate 2 random numbers d_p and d_q, of s-bits(s<=n/k) such that gcd(d_p,p-1)=1, gcd(d_q,q-1)=1 and d_p= $d_q$mod2.
Step3. Find d such that d= d_p mod p-1, d= $d_q$mod q-1
Step4. Calculate e=d^{-1} mod Φ(N)
Public key=<N,e> and Private Key=<p,q, d_p,d_q>

Encryption

Same as in standard RSA. But as we have shifted the cost from decryption side to encryption side, the value of public exponents will get increased. Due to the large value of e RSA encryption must be ready for the higher computational cost.

The Encryption Complexity =$(3n_e-2)(n^2+2)+o(n^2)$
Which is approx. equal to $3n_en^2-2n^2+o(n^2)$

Decryption

Step1. Compute M_1= C^{dp} mod p and M_2=C^{dq} mod q
Step2. Thus M_1^e=C mod p and M_2^e=C mod q.

Step3. Using Hensel lifting [12] compute M_1', such that $(M_1')^e=C \bmod p^{k-1}$
Step4. Using CRT, compute M such that $M= M_1' \bmod p^{k-1}$ and $M= M_2 \bmod q$. Then $M=C^d \bmod N$ is the required result

Performance

For k=3, no of bits in public exponent $=n_e$, no. of bits in dp=s

Complexity of Decryption algorithm is approx. [8] and [15]=

$(9s+12n_e+10)n^2/9+o(n^2)$

The theoretical speedup of this scheme as with standard RSA with CRT is

$S_{CRT}=nk^2/8s$

For k=3 and s=160 and n=1024, the theoretical speedup over RSA with CRT is 7.20

III. DRSA Cryptosystem

The standard RSA and other variants are not semantically secure. Pointcheviel [9] gave a new Dependent RSA (DRSA) problem and proposed a cryptosystem with the property of semantic security. The three algorithms of key generation, encryption and decryption are of DRSA cryptosystem are explained below.

Key Generation

The key generation for DRSA [9] scheme is same as that for the standard RSA scheme. To generate keys in DRSA scheme, the user chooses two large primes p and q and computes N = pq. User then determines an integer e less than and relatively prime to Φ(N) and computes an integer d such that ed ≡ 1mod Φ(N). The public key and the secret key for the user R is (e, N) and d respectively. The prime p and q are also kept secret.

Encryption

To encrypt any plaintext M Є Z_N, sender S first randomly selects an integer l Є Z_N*and sends the complete ciphertext (C_1, C_2) to the receiver R. Where,

$C_1 = l^e \bmod N$

$C_2 = M (l + 1)^e \bmod N$.

Encryption complexity

Step1: As this factor can be computed in advance, this step will not contribute in the complexity.

Step2: exponentiation and multiplication

$[(3n_e-2)(n^2+n)]+[2n^2+2n]$

Here no of bits in M and I are considered approx equal to n bits

It simplifies to $3n_en^2+o(n^2)$

Decryption

To decrypt the ciphertext (C_1, C_2), receiver R first computes

1. $C_1^d \bmod n = l$
2. $M = (C_2 / (l+1)^e) \bmod N$

Decryption Complexity:

Step1: $(3n_d-2)(n^2+n)$

Step2: exponentiation+inversion+multipication =

$(3n_e-2)(n^2+n)+20(2n^2+2n)+(2n^2+2n)$

Which simplifies to $(3n_d+3n_e+38)n^2+o(n^2)$

IV. NEW SCHEME

A. Algorithm

Here RePower RSA [8] is further extended to increase the encryption/signature verification performance with a very less compromising the decryption. In this approach the DRSA [9] cryptosystem is used with RePower RSA. The Key generation, Encryption and the Decryption process are as follows.

Key Generation

The key generation algorithm describes the following steps:

- Generate two distinct [n/k]-bit primes , p and q, with gcd(p-1,q-1)=2, and calculate $N=p^{k-1}.q$
- Generate 2 random numbers d_p and d_q, of s-bits(s<=n/k) such that $\gcd(d_p,p-1)=1$, $\gcd(d_q,q-1)=1$ and $d_p= d_q \bmod 2$.
- Find d such that $d= d_p \bmod p-1$, $d= d_q \bmod q-1$
- Calculate $e=d^{-1} \bmod \Phi(N)$

Public key=<N,e> and Private Key=<p,q, d_p,d_q>

Encryption

To encrypt any message M Є Z_N, sender S chooses a random integer h Є Z_N* and computes

1. $C_1 = (h + 1)^e \bmod N$,
2. $C_2 = Mh^{-1} \bmod N$.

Ciphertext (C_1, C_2) is sent to the receiver.

Complexity of Encryption Algorithm

Step1:This can be computed in advance ,so it will not contribute in the complexity

Step2: inversion+multiplication

$20*(2n^2+2n)+ 2n^2+2n$

Which simplifies to $42n^2+o(n^2)$

Decryption

The steps in this algorithm are as follows:

Step1. Compute $h_1= C_1^{dp} \bmod p$ and $h_2=C_1^{dq} \bmod q$

Step2. Thus $h_1^e=C_1 \bmod p$ and $h_2^e=C_1 \bmod q$.

Step3. Using Hensel lifting [20] compute h_1', such that $(h_1')^e=C_1 \bmod p^{k-1}$

Step4. Using CRT, compute h such that $h= h_1' \bmod p^{k-1}$ and $h= h_2 \bmod q$. Then $h=C_1^d \bmod N$.

Step5. From h we can now finally calculate

$M=C_2h \bmod N$

Decryption Complexity:

Complexity of Repower algorithm+Step 5

$(9s+12n_e+10)n^2/9+(2n^2+2n)$

Which simplifies to $(9s+12n_e+18)n^2/9+o(n^2)$

B. Analysis of the New Scheme

Semantic Security

Semantic security is a widely-used definition for security in an asymmetric key encryption algorithm. For a cryptosystem to be semantically secure, it must be infeasible for a computationally-bounded adversary to derive significant information about a message (plaintext) when given only its cipher text and the corresponding public encryption key. The proposed cryptosystem is semantically secure against chosen plaintext attack. This we can say on the basis that in order to determine any information about the plaintext M from the cipher text (C_1, C_2), the adversary needs to have information about h−1 (mod N), where this h is a randomly chosen element in Z_N*. We cannot calculate the value of h without knowing the value of the secret key d. Even the value of h-1(mod N) can't be calculated even if we know partial information about h. Thus the given new scheme is semantically secure.

C. Comparison with RePower RSA

Computational efficiency

In the proposed scheme, the random values h can be taken in advance, thus the user has the values $(h+1)^e$ and h^{-1} computed well in advance. Thus time consumed during the encryption phase in the new scheme will be lowered. Because the value of e is very large, so it saves a large amount of time during encryption. In this case, the encryption process requires only one multiplication modulo N and that is quite affordable. Whereas in RePower RSA [8] scheme, having large value of the encryption exponent e, the encryption cost is very high. As compared to only one multiplication modulo N in new scheme, RePower RSA requires one exponentiation to the power e modulo N, resulting in poor performance of encryption side. Thus it can be concluded that the new scheme encryption phase has better performance than RePower RSA and of course than any other variants of RSA. This encryption performance enhancement is at the cost of slight increase in decryption cost. Besides other computation the proposed scheme requires to compute one extra multiplication modulo N at the decryption process. Thus the decryption speed is computationally as expensive as RePower RSA.

In Table I ne=no of bits in public exponent
nd=no of bits in private exponent
s=private factor less than n/3

In RePower RSA ne≈n, so it cost very high in Encryption process. But this factor is not involved in the new scheme, that's why the encryption is very low as compared to RePower approach. This result in a small increase in cost of decryption side as shown in the Table I.

Semantic Security

The proposed scheme is semantically secure due to randomness added in the comutation, but RePower RSA is not semantically secure.

D. Comparison with DRSA scheme

Decryption Phase

Our proposed scheme is computationally less expensive than DRSA cryptosystem. In our proposed scheme, Hensel lifting is used, that is less expensive than the exponentiation. On average it is required to compute less than one exponent to the power d modulo N and one multiplication modulo N. Whereas in DRSA, decryption process requires to compute one exponentiation to the power e and to the power d modulo N, one inversion and one multiplication under modulo N. Hence our proposed scheme is more efficient than that of the D-RSA scheme.

Encryption Phase

The efficiency of encryption process of D-RSA and our proposed scheme both is same.

In Table II ne=no of bits in public exponent
nd=no of bits in private exponent
s=private factor less than n/3

As shown in the Table II, in new scheme the encryption complexity is almost same (in DRSA there is no constraint of n_e), whereas decryption cost is lower in new scheme as compared to DRSA.

Table I

Comparison	RePower	New Scheme
Encryption Complexity	$3n_en^2-2n^2+o(n^2)$	$42n^2+o(n^2)$
Decryption Complexity	$(9s+12n_e+10)n^2/9+o(n^2)$	$(9s+12n_e+18)n^2/9+o(n^2)$

Table II

Comparison	DRSA	New Scheme
Encryption Complexity	$3n_en^2+o(n^2)$	$42n^2+o(n^2)$
Decryption Complexity	$(3n_d+3n_e+38)n^2+o(n^2)$	$(9s+12n_e+18)n^2/9+o(n^2)$

V. CONCLUSION

In this paper RSA public cryptosystem is explained with its combined variants of MultiPower RSA and Rebalanced RSA, i.e, RePower. Repower RSA gave the maximum decryption/signature generation performance of all the variants of RSA (Batch RSA, Multiprime RSA, MultiPower RSA, Rbalanced RSA, RPrime RSA). But this performance is at the cost of very high encryption/signature verification cost. The devices with constrained resources, like PDAs, are not always used for decryption/signature generation. There is the requirement that these constraint devices are used for encryption/signature verification as well. In that case the use of RePower RSA will not be giving good performance. DRSA is explained to provide the semantic security to the system. The proposed approach in the paper gives better encryption performance at the cost of a small decrease in decryption side. Besides it provides the semantic security to the system which is not provided by RePower RSA. The scheme is proven to be better than RePower RSA as well as to DRSA

REFERENCES

[1] R. Rivest, A. Shamir, and L. Adleman, "A Method for Ob taining Digital Signatures and Public Key Cryptosystems", Communications of the ACM Vol. 21, No. 2, pp. 120 - 126, 1978.
[2] Dan Boneh, "Twenty Years of Attacks on the RSA Crypto system", Notices of the AMS, Vol. 46, No. 2, Feb. 1999, pp. 203-213.
[3] A. Fiat. Batch RSA. "Advances in Cryptology", Crypto '89, Vol. 435, 1989, pp.175–185.
[4] T. Collins, D. Hopkins, S. Langford, and M. Sabin, "Public Key Cryptographic Apparatus and Method", US Patent #5,848,159. Jan. 1997.
[5] T. Takagi, "Fast RSA-Type Cryptosystem Modulo p^kq", Crypto '98, 1462 of LNCS. 1998, pp. 318–326
[6] M. Wiener, "Cryptanalysis of Short RSA Secret Expo nents", IEEE Transactions on Information Theory, Vol. 36, No. 3, 1990, pp. 553–558.
[7] Cesar Alison Monteiro Paixao, "An efficient variant of the RSA cryptosystem", preprints (2003).
[8] Garg D,Verma S, "Improvement over Public Key cryptographic Algorithm" Advance Computing Conference,2009, IACC 2009, IEEE International Conference,March 2009, pp. 734-739
[9] David Pointcheval, "New public key cryptosystem based on the dependent RSA problem", Eurocrypt'99 LNCS Springer-Verlag,,1999, Vol. 1592, pp.239-254.
[10] J-J. Quisquater and C. Couvreur, "Fast Decipherment Al gorithm for RSA Public-Key Cryptosystem", Eletronic

Letters, Vol. 18, 1982, pp. 905–907.
[11] A.A. Mamun, M. M. Islam, S.M. M. Romman, A.H.S.U Ahmad, "Performance Evaluation of Several Efficient RSA Variants", IJCSNS VOL.8 No.7, July 2008, pp. 7-11.
[12] H. Cohen, "A Course in Computational Algebraic Number Theory", Graduate Texts in Mathematics, Vol. 138, p. 137.
[13] S. D. Galbraith, C. Heneghan, and J. F. McKee, "Tunable balancing of RSA," Proceedings Information Security and Privacy, 2005, Vol. 3574, pp. 280–292.
[14] H.-M. Sun and Mu-En. Wu, "Design of Rebalanced RSA-CRT for Fast Encryption," Information Security Confer ence 2005. pp. 16-27.
[15] M. J. Hinek, "Another look at small RSA exponents," Top ics in Cryptology, CT-RSA 2006, 2006, Vol. 3860, pp. 82–98.
[16] Camille Vuillaume. Efficiency Comparison of Several RSA Variants Master Thesis, Fachbereich Informatik der TUDarmstadt, 2003

Seema Verma received the B.Tech degree in Computer Science from JMIT Radaur, Kurukshetra University, India in 2001, and M.Tech. in Computer Science from AAIDU, Allahabad, India in 2007. She is now pursuing Ph.D in Computer Science from Thapar University, India. She is now working as Asst Prof. in Computer Science & IT department in EIT, Haryana, India.

Dr Deepak Garg, Professor, Department of Computer Science and Engineering, Thapar University, Patiala (Punjab), India, Senior Member of IEEE, Secretary of IEEE Computer Society (Delhi Section), Life Member of Computer Society of India (CSI), Indian Society of Technical Education (ISTE), Indian Science Congress Association Kolkata