Cryptanalysis of RSA Variants and Implicit Factorization

Santanu Sarkar

August 20, 2013
Outline of the Talk

RSA Cryptosystem

Lattice based Root Finding of Polynomials

Common Prime RSA

Dual RSA

Prime Power RSA

Implicit Factorization

CRT-RSA having Low Hamming Weight Decryption Exponents

Conclusion
The RSA Public Key Cryptosystem

- Invented by Rivest, Shamir and Adleman in 1977.
- Most businesses, banks, and even governments use RSA to encrypt their private information.
RSA in a Nutshell

**Key Generation Algorithm**
- Choose primes $p, q$
- Construct modulus $N = pq$, and $\phi(N) = (p - 1)(q - 1)$
- Set $e, d$ such that $d = e^{-1} \mod \phi(N)$
- Public key: $(N, e)$ and Private key: $d$

**Encryption Algorithm:** $C = M^e \mod N$

**Decryption Algorithm:** $M = C^d \mod N$
Example

- Primes: $p = 653, q = 877$
- Then $N = pq = 572681, \phi(N) = (p - 1)(q - 1) = 571152$
- Take Public Exponent $e = 13$
- Note $13 \times 395413 \equiv 1 \pmod{571152}$
- Private exponent $d = 395413$
- Plaintext $m = 12345$
- Ciphertext $c = 12345^{13} \pmod{572681} = 536754$
Practical Example

Example

\( p = 846599862936164736402988177812099956013778770876315707836731563770 \\
5880893839981848305923857095440391598629588811166856664047346930517527 \\
891174871536167839, \\
q = 121764346862040688467973181827710403396896519724618922933494273650 \\
3033910096582171197571988374294918003138669675396892122967962313235346 \\
8174200136260738213, \\
N = 10308567936391526757875542896033316178883861174865735387244345263 \\
7137208314161521669308869345882336991188745907630491004512656603926295 \\
3518502967942206721243236328408403417100233192004322468033366480788753 \\
9303481101449158308722791555032457532325542013658355061619621556208246 \\
3591629130621212947471071208931707, \\
e = 2^{16} + 1 = 65537, \text{ and} \\
d = 101956309423526004076893177133219940094766772585504692321252302615 \\
1120238295258506352584280960487541607315458593878388760777253827593350 \\
0788233193317652234750616708162985718345962209115090210535366860135950 \\
1135207708372912478251719497009548072271475262211661830196811724409660 \\
406447291034092315494830924578345.
Factorization Methods

“The problem of distinguishing prime numbers from composites, and of resolving composite numbers into their prime factors, is one of the most important and useful in all of arithmetic.”

– Carl Friedrich Gauss

- Pollard’s $p − 1$ algorithm (1974)
- Dixon’s Random Squares Algorithm (1981)
- Williams’ $p + 1$ method (1982)
- Number Field Sieve (NFS): A. K. Lenstra et al. (1993)
Lattice

LATTICE BASED ROOT FINDING OF POLYNOMIALS
Finding roots of a polynomial

**Univariate Integer Polynomial**
- \( f(x) \in \mathbb{Z}[x] \) with root \( x_0 \in \mathbb{Z} \) efficient methods available

**Multivariate Integer Polynomial**
- \( f(x, y) \in \mathbb{Z}[x, y] \) with root \( (x_0, y_0) \in \mathbb{Z} \times \mathbb{Z} \) not efficient

**Univariate Modular Polynomial**
- \( f(x) \in \mathbb{Z}_N[x] \) with root \( x_0 \in \mathbb{Z}_N \) not efficient

**Hilbert’s tenth Problem: 1900**
Finding roots of a polynomial

**Univariate Integer Polynomial**
- \( f(x) \in \mathbb{Z}[x] \) with root \( x_0 \in \mathbb{Z} \)
  - efficient methods available

**Multivariate Integer Polynomial**
- \( f(x, y) \in \mathbb{Z}[x, y] \) with root \((x_0, y_0) \in \mathbb{Z} \times \mathbb{Z}\)
  - not efficient

**Univariate Modular Polynomial**
- \( f(x) \in \mathbb{Z}_N[x] \) with root \( x_0 \in \mathbb{Z}_N \)
  - not efficient

**Hilbert’s tenth Problem: 1900**

Lattice based techniques help in some cases.
Lattice

Definition (Lattice)

Let \( \mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{Z}^m \) (\( m \geq n \)) be \( n \) linearly independent vectors. A lattice \( L \) spanned by \( \{ \mathbf{v}_1, \ldots, \mathbf{v}_n \} \) is the set of all integer linear combinations of \( \mathbf{v}_1, \ldots, \mathbf{v}_n \). That is,

\[
L = \left\{ \mathbf{v} \in \mathbb{Z}^m \mid \mathbf{v} = \sum_{i=1}^{n} a_i \mathbf{v}_i \text{ with } a_i \in \mathbb{Z} \right\}.
\]

The determinant of \( L \) is defined as \( \det(L) = \prod_{i=1}^{n} ||\mathbf{v}_i^*|| \).

Example

Consider two vectors \( \mathbf{v}_1 = (1, 2), \mathbf{v}_2 = (3, 4) \). The lattice \( L \) generated by \( \mathbf{v}_1, \mathbf{v}_2 \) is

\[
L = \{ \mathbf{v} \in \mathbb{Z}^2 \mid \mathbf{v} = a_1 \mathbf{v}_1 + a_2 \mathbf{v}_2 \text{ with } a_1, a_2 \in \mathbb{Z} \}.
\]
LLL Algorithm

Devised by A. Lenstra, H. Lenstra and L. Lovász (Mathematische Annalen 1982)

Main goal: Reduce a lattice basis in a certain way to produce a ‘short (bounded)’ and ‘nearly orthogonal’ basis called the \textit{LLL-reduced} basis.
Connecting LLL to Root finding

The clue was provided by Nick Howgrave-Graham in 1997.

**Theorem**

Let \( h(x) \in \mathbb{Z}[x] \) be an integer polynomial with \( n \) monomials. Let for a positive integer \( m \),

\[
h(x_0) \equiv 0 \pmod{N^m} \text{ with } |x_0| < X \quad \text{and} \quad \|h(xX)\| < \frac{N^m}{\sqrt{n}}.
\]

Then, \( h(x_0) = 0 \) holds over integers.
Connecting LLL to Root finding

**Main idea:**
We can transform a modular polynomial $h(x)$ to an integer polynomial while preserving the root $x_0$, subject to certain size constraints.

We need roughly $\det(L)^{\frac{1}{n}} < N^m$. 
RSA Variants

- Multi Prime RSA
- Twin RSA
- Common Prime RSA
- Dual RSA
- Prime Power RSA
- CRT-RSA
Common Prime RSA
Common Prime RSA

- Primes: \( p - 1 = 2ga \) and \( q - 1 = 2gb \)
- RSA modulus: \( N = pq \)
- \( ed \equiv 1 \mod 2gab \)
Common Prime RSA

- Primes: $p - 1 = 2ga$ and $q - 1 = 2gb$
- RSA modulus: $N = pq$
- $ed \equiv 1 \mod 2gab$

Existing results:
- Hinek: CT-RSA 2006
- Jochemsz and May: Asiacrypt 2006
1. Let $g \approx N^\gamma$ and $p, q$ be of same bit size
2. $e \approx N^{1-\gamma}$ and $d \approx N^\beta$

Theorem

$N$ can be factored in polynomial time if

$$\beta < \frac{1}{4} - \frac{\gamma}{2} + \frac{\gamma^2}{2}.$$
Proof

- We have $ed \equiv 1 \mod 2gab$.

- So $ed = 1 + 2kgab$.

- $ed = 1 + k\frac{(p-1)(q-1)}{2g}$.

- $2edg = 2g + k(p-1)(q-1) \Rightarrow 2edg = 2g + k(N+1-p-q)$

- Root $(x_0, y_0) = (2g + k(1-p-q), k)$ of the polynomial $f(x, y) = x + yN$ in $\mathbb{Z}_{ge}$

- Note $g$ divides $N-1$ as $p = 1 + 2ga$ and $q = 1 + 2gb$

- Let $c = N-1$
Proof

For integers $m, t \geq 0$, we define following sets of polynomials:

$$g_i(x, y) = x^j f^i(x, y) e^{m-i} c^{\max\{0, t-i\}}$$

where $i = 0, \ldots, m$, $j = m - i$.

**Note that** $g_i(x_0, y_0) \equiv 0 \mod (e^m g^t)$.

Dimension of the lattice $L$ is $\omega = m + 1$
Proof

- Condition: $\det(L) < e^{m\omega} g^{t\omega}$

- Here $\det(L) = (XYe)^{\frac{m^2 + m}{2}} c^{\frac{t^2 + t}{2}}$
Dual RSA
Dual RSA

Proposed by H.-M. Sun, M.-E. Wu, W.-C. Ting, and M.J. Hinek
[IEEE-IT, August 2007]

- Two different RSA moduli $N_1 = p_1q_1$, $N_2 = p_2q_2$
- Same pair of keys $e$ and $d$ such that

$$ed \equiv 1 \text{ mod } \phi(N_1)$$

$$ed \equiv 1 \text{ mod } \phi(N_2)$$

Applications: blind signatures, authentication/secrecy etc.
Dual CRT-RSA

Motivation: CRT-RSA is faster than RSA

Sun et al. proposed a CRT variant of Dual RSA.

Dual CRT-RSA:
- Two different RSA moduli $N_1 = p_1 q_1, N_2 = p_2 q_2$
- Same set of keys $e$ and $d_p, d_q$ such that

$$
ed_p \equiv 1 \mod (p_1 - 1)$$

$$
ed_p \equiv 1 \mod (p_2 - 1)$$

$$
ed_q \equiv 1 \mod (q_1 - 1)$$

$$
ed_q \equiv 1 \mod (q_2 - 1)$$
Theorem

Let $N_1, N_2$ be the public moduli of Dual CRT-RSA and suppose

$$e = N^\alpha, \quad d_p, d_q < N^\delta.$$

Then, for $\alpha > \frac{1}{4}$, one can factor $N_1, N_2$ in poly($\log N$) time when

$$\delta < \frac{1 - \alpha}{2} - \epsilon$$

for some arbitrarily small positive number $\epsilon > 0$. 
Sketch of the proof

Note the following:

- $ed_p \equiv 1 \mod (p_1 - 1) \iff ed_p - 1 + k_{p_1} = k_{p_1} p_1$
- $ed_q \equiv 1 \mod (q_1 - 1) \iff ed_q - 1 + k_{q_1} = k_{q_1} q_1$

Combining these two relations:

$$(ed_p - 1 + k_{p_1})(ed_q - 1 + k_{q_1}) = k_{p_1} k_{q_1} N_1$$
Sketch of the proof

This in turn gives us:

\[ e^2 y_1 + ey_2 + y_3 = (N_1 - 1)k_{p_1}k_{q_1} \]
\[ e^2 y_1 + ey_4 + y_5 = (N_2 - 1)k_{p_2}k_{q_2} \]

where we have

\[ y_1 = d_p d_q, \]
\[ y_2 = d_p (k_{p_1} - 1) + d_q (k_{q_1} - 1), \]
\[ y_3 = 1 - k_{p_1} - k_{q_1}, \]
\[ y_4 = d_p (k_{p_2} - 1) + d_q (k_{q_2} - 1), \]
\[ y_5 = 1 - k_{p_2} - k_{q_2}. \]
Sketch of the proof

Consider the polynomial

\[ f(X, Y, Z) = e^2X + eY + Z \]

to obtain:

\[ f(y_1, y_2, y_3) \equiv 0 \pmod{N_1 - 1} \]
\[ f(y_1, y_4, y_5) \equiv 0 \pmod{N_2 - 1} \]
Sketch of the proof

Combine the two modular equations to obtain $G$ such that

$$G(y_1, y_2, y_3, y_4, y_5) \equiv 0 \pmod{(N_1 - 1)(N_2 - 1)}$$

where $G(x_1, x_2, x_3, x_4, x_5) = x_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_5x_5$

We prove that one can find the root $(y_1, y_2, y_3, y_4, y_5)$ of $G$ if

$$\delta < \frac{1 - \alpha}{2} - \epsilon$$
Prime Power RSA
Prime Power RSA

- RSA modulus $N$ is of the form $N = p^r q$ where $r \geq 2$

- An electronic cash scheme using the modulus $N = p^2 q$:
  Fujioka, Okamoto and Miyaguchi (Eurocrypt 1991).

- $\frac{1}{r+1}$ fraction of MSBs of $p \Rightarrow$ polynomial time factorization:
  Boneh, Durfee and Howgrave-Graham (Crypto 1999)
Prime Power RSA

- $d \leq N^{\frac{1}{2(r+1)}}$: Takagi (Crypto 1998)
- $d < N^{\frac{r}{(r+1)^2}}$ or $d < N^{\left(\frac{r-1}{r+1}\right)^2}$: May (PKC 2004)
- When $r = 2$, $N^{\max\left\{\frac{2}{9},\frac{1}{9}\right\}} = N^{\frac{2}{9}} \approx N^{0.22}$. 
Theorem

Let \( N = p^2 q \) be an RSA modulus. Let the public exponent \( e \) and private exponent \( d \) satisfies \( ed \equiv 1 \mod \phi(N) \). Then \( N \) can be factored in polynomial time if \( d \leq N^{0.395} \).
Proof Idea

- $ed \equiv 1 \mod \phi(N)$ where $N = p^2 q$.
- So we can write $ed = 1 + k(N - p^2 - pq + p)$.
- We want to find the root $(x_0, y_0, z_0) = (k, p, q)$ of the polynomial $f_e(x, y, z) = 1 + x(N - y^2 - yz + y)$.
- Note $y_0^2 z_0 = N$
Proof Idea

For integers $m, a, t \geq 0$, we define following polynomials

$$g_{i,j,k}(x, y, z) = x^j y^k z^{j+a} f_i(x, y, z)$$

where $i = 0, \ldots, m$, $j = 1, \ldots, m - i$, $k = j, j + 1, j + 2$ and

$$g_{i,0,k}(x, y, z) = y^k z^a f_i(x, y, z)$$

where $i = 0, \ldots, m$, $k = 0, \ldots, t$. 
General Case

Recall

$N = p^r q$

$ed \equiv 1 \mod p^{r-1}(p-1)(q-1)$

For integers $m, a, t \geq 0$, we define following polynomials

$g_{i,j,k}(x, y, z) = x^i y^j z^k + a f_e^i(x, y, z)$

where $i = 0, \ldots, m$, $j = 1, \ldots, m - i$, $k = j, j+1, \ldots, j+2r-2$ and

$g_{i,0,k}(x, y, z) = y^k z^a f_e^i(x, y, z)$

where $i = 0, \ldots, m$, $k = 0, \ldots, t$. 
General Case

<table>
<thead>
<tr>
<th>$r$</th>
<th>$\delta$</th>
<th>$\max \left{ \frac{r}{(r+1)^2}, \left( \frac{r-1}{r+1} \right)^2 \right}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.395</td>
<td>0.222</td>
</tr>
<tr>
<td>3</td>
<td>0.410</td>
<td>0.250</td>
</tr>
<tr>
<td>4</td>
<td>0.437</td>
<td>0.360</td>
</tr>
<tr>
<td>5</td>
<td>0.464</td>
<td>0.444</td>
</tr>
<tr>
<td>6</td>
<td>0.489</td>
<td>0.510</td>
</tr>
<tr>
<td>7</td>
<td>0.512</td>
<td>0.562</td>
</tr>
<tr>
<td>8</td>
<td>0.532</td>
<td>0.605</td>
</tr>
<tr>
<td>9</td>
<td>0.549</td>
<td>0.640</td>
</tr>
<tr>
<td>10</td>
<td>0.565</td>
<td>0.669</td>
</tr>
</tbody>
</table>

*Table:* Numerical upper bound of $\delta$ for different values of $r$
Implicit Factorization
Explicit factorization

**Rivest and Shamir** (Eurocrypt 1985)

$N$ can be factored given $2/3$ of the LSBs of a prime

\[
\begin{array}{c}
1001010100 \\
\underline{101001001010010011}
\end{array}
\]

**Coppersmith** (Eurocrypt 1996)

$N$ can be factored given $1/2$ of the MSBs of a prime

\[
\begin{array}{c}
100101010010010010011 \\
\underline{10010101001010010010011}
\end{array}
\]

**Boneh et al.** (Asiacrypt 1998)

$N$ can be factored given $1/2$ of the LSBs of a prime

\[
\begin{array}{c}
10010101001010010010100 \\
\underline{100101010010010011}
\end{array}
\]

**Herrmann and May** (Asiacrypt 2008)

$N$ can be factored given a random subset of the bits (small contiguous blocks) in one of the primes

\[
\begin{array}{c}
100 \\
\underline{1010100} \\
10100 \\
\underline{1001010100} \\
10011
\end{array}
\]
Implicit Factorization

In PKC 2009, May and Ritzenhofen introduced Implicit Factorization

**Scenario:**

- Consider two integers $N_1, N_2$ such that $N_1 = p_1 q_1$ and $N_2 = p_2 q_2$ where $p_1, q_1, p_2, q_2$ are primes.
- Suppose we know that $p_1, p_2$ share a few bits from LSB side, but we do not know the shared bits.

**Question:**
How many bits do $p_1, p_2$ need to share for efficiently factoring $N_1, N_2$?
Theorem

Let \( q_1, q_2, \ldots, q_k \approx N^\alpha \), and consider that \( \gamma_1 \log_2 N \) many MSBs and \( \gamma_2 \log_2 N \) many LSBs of \( p_1, \ldots, p_k \) are the same. Also define \( \beta = 1 - \alpha - \gamma_1 - \gamma_2 \).

Then, one can factor \( N_1, N_2, \ldots, N_k \) in \( \text{poly}\{\log N, \exp(k)\} \) if

\[
\beta < \begin{cases} 
C(\alpha, k), & \text{for } k > 2, \\
1 - 3\alpha + \alpha^2, & \text{for } k = 2,
\end{cases}
\]

with the constraint \( 2\alpha + \beta \leq 1 \), where

\[
C(\alpha, k) = \frac{k^2(1 - 2\alpha) + k(5\alpha - 2) - 2\alpha + 1 - \sqrt{k^2(1 - \alpha^2) + 2k(\alpha^2 - 1) + 1}}{k^2 - 3k + 2}.
\]
## Comparison with the existing works

<table>
<thead>
<tr>
<th>$k$</th>
<th>Bitsize of $p_i, q_i$</th>
<th>No. of shared LSBs May et al. in $p_i$</th>
<th>No. of shared LSBs (our) in $p_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$(1 - \alpha) \log_2 N, \alpha \log_2 N$</td>
<td>Theory</td>
<td>Expt.</td>
</tr>
<tr>
<td>3</td>
<td>750, 250</td>
<td>375</td>
<td>378</td>
</tr>
<tr>
<td>* 3</td>
<td>700, 300</td>
<td>450</td>
<td>452</td>
</tr>
<tr>
<td>* 3</td>
<td>650, 350</td>
<td>525</td>
<td>527</td>
</tr>
<tr>
<td># 3</td>
<td>600, 400</td>
<td>600</td>
<td>-</td>
</tr>
<tr>
<td>* 4</td>
<td>750, 250</td>
<td>334</td>
<td>336</td>
</tr>
<tr>
<td>* 4</td>
<td>700, 300</td>
<td>400</td>
<td>402</td>
</tr>
<tr>
<td>* 4</td>
<td>650, 350</td>
<td>467</td>
<td>469</td>
</tr>
<tr>
<td>* 4</td>
<td>600, 400</td>
<td>534</td>
<td>535</td>
</tr>
</tbody>
</table>

**Table:** For 1000 bit $N$, theoretical and experimental data of the number of shared LSBs in May et al. and shared LSBs in our case. (Time in seconds)
CRT-RSA
The CRT-RSA Cryptosystem

- Improves the decryption efficiency of RSA, 4 folds!
- Invented by Quisquater and Couvreur in 1982.
- The most used variant of RSA in practice.
CRT-RSA: Faster approach for decryption

- Two decryption exponents \((d_p, d_q)\) where
  \[ d_p \equiv d \mod (p - 1) \text{ and } d_q \equiv d \mod (q - 1). \]

- To decrypt the ciphertext \(C\), one needs
  \[ C_p \equiv C^{d_p} \mod p \text{ and } C_q \equiv C^{d_q} \mod q. \]

Calculating \(x^y\):

- \(\ell_y = \lceil \log_2 y \rceil\) many squares

- \(w_y = \text{wt}(\text{bin}(y))\) many multiplications
CRT-RSA: Faster through low Hamming weight

- Lim and Lee (SAC 1996) and later Galbraith, Heneghan and McKee (ACISP 2005): $d_p, d_q$ with low Hamming weight.

- Maitra and Sarkar (CT-RSA 2010): large low weight factors in $d_p, d_q$. 
Galbraith, Heneghan and McKee (ACISP 2005)

Input: $\ell_e, \ell_N, \ell_k$
Output: $p, d_p$

1. Choose an $\ell_e$ bit odd integer $e$;
2. Choose random $\ell_k$ bit integer $k_p$ coprime to $e$;
3. Find odd integer $d_p$ such that $d_p \equiv e^{-1} \mod k_p$;
4. $p = 1 + \frac{ed_p - 1}{k_p}$;

$(\ell_e, \ell_N, \ell_d, \ell_k) = (176, 1024, 338, 2)$ with $w_{dp} = w_{dq} = 38$

Comparison in decryption: 26% Faster
Sarkar and Maitra (CHES 2012)

The Tool for Cryptanalysis:

- Henecka, May and Meurer: Correcting Errors in RSA Private Keys (Crypto 2010).

- Three equations:
  \[ N = pq, \quad ed_p = 1 + k_p(p - 1), \quad ed_q = 1 + k_q(q - 1) \]

- We have:
  1. \[ q = p^{-1}N \mod 2^a \]
  2. \[ d_p = (1 + k_p(p - 1)) e^{-1} \mod 2^a \]
  3. \[ d_q = (1 + k_q(q - 1)) e^{-1} \mod 2^a \]
The Tool for Cryptanalysis

- $w_{dp}, w_{dq}$ are taken significantly smaller than the random case.
- Take the all zero bit string as error-incorporated (noisy) presentation of $d_p, d_q$.
- If the error rate is significantly small ($< 8\%$), one can apply the error correcting algorithm of Henecka et al to recover the secret key.
- Time complexity of the error-correction heuristic: $\tau$.
- The strategy attacks the schemes of SAC 1996 and ACISP 2005 in $\tau O(e)$ time. For our scheme in CT-RSA 2010, it is $\tau O(e^3)$.
Experimental results: parameters $d_p, d_q$

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>0.08</th>
<th>0.09</th>
<th>0.10</th>
<th>0.11</th>
<th>0.12</th>
<th>0.13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suc. prob.</td>
<td>0.59</td>
<td>0.27</td>
<td>0.14</td>
<td>0.04</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Time (sec.)</td>
<td>307.00</td>
<td>294.81</td>
<td>272.72</td>
<td>265.66</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Suc. prob.</td>
<td>0.68</td>
<td>0.49</td>
<td>0.25</td>
<td>0.18</td>
<td>0.08</td>
<td>0.02</td>
</tr>
<tr>
<td>Time (sec.)</td>
<td>87.41</td>
<td>84.47</td>
<td>80.18</td>
<td>74.57</td>
<td>79.33</td>
<td>76.04</td>
</tr>
</tbody>
</table>

**Lim et al (SAC 1996)**

$\ell_N = 768, \ell_{dp} = 384, w_{dp} = 30, e = 257; \Rightarrow \delta \approx \frac{30}{384} = 0.078$

$\ell_N = 768, \ell_{dp} = 377, w_{dp} = 45, e = 257; \Rightarrow \delta = \frac{w_{dp}}{\ell_{dp}} \approx 0.12$

**Galbraith et al (ACISP 2005)**

$(\ell_e, \ell_{dp}, \ell_{kp}) = (176, 338, 2), w_{dp} = 38 \Rightarrow \delta \approx \frac{38}{338} \approx 0.11$

**Maitra et al (CT-RSA 2010)** $\delta \approx 0.08$
Summary of the talk

In this talk, we have

- RSA Cryptosystem
- Studied Lattice based techniques for finding root(s) of polynomials
- Common Prime RSA
- Dual RSA
- Prime Powe RSA
- Implicit Factorization
- CRT-RSA
Reference


Thank You!