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

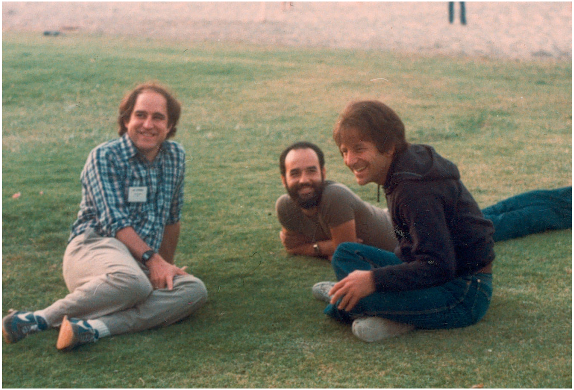
Prime Power RSA

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 $d = e^{-1} \bmod \phi(N)$
- Public key: (N, e) and Private key: d

ENCRYPTION ALGORITHM: $C = M^e \bmod N$

DECRYPTION ALGORITHM: $M = C^d \bmod N$

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 $c = 12345^{13} \bmod 572681 = 536754$

Practical Example

Example

$p =$ 846599862936164736402988177812099956013778770876315707836731563770
5880893839981848305923857095440391598629588811166856664047346930517527
891174871536167839,

$q =$ 121764346862040688467973181827710403396896519724618922933494273650
3033910096582171197571988374294918003138669675396892122967962313235346
8174200136260738213,

$N =$ 10308567936391526757875542896033316178883861174865735387244345263
7137208314161521669308869345882336991188745907630491004512656603926295
3518502967942206721243236328408403417100233192004322468033366480788753
9303481101449158308722791555032457532325542013658355061619621556208246
3591629130621212947471071208931707,

$e = 2^{16} + 1 = 65537$, 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)
- Quadratic Sieve (QS): Pomerance (1981)
- Williams' $p+1$ method (1982)
- Elliptic Curve Method (ECM): H. W. Lenstra (1987)
- 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,

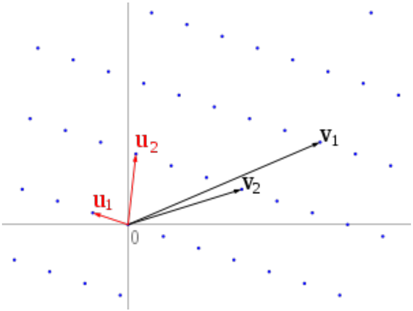
$$L = \left\{ \mathbf{v} \in \mathbb{Z}^m \mid \mathbf{v} = \sum_{i=1}^{n} a_i \mathbf{v_i} \text{ with } a_i \in \mathbb{Z} \right\}.$$

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 *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,

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

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: $N = pq$
- $ed \equiv 1 \bmod 2gab$

Common Prime RSA

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

Existing results:

- Hinek: CT-RSA 2006
- Jochemsz and May: Asiacrypt 2006

Sarkar and Maitra: DCC 2013

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 $ed \equiv 1 \bmod 2gab$.
- 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:

$$
\begin{aligned}
g_i(x,y) &= x^j f^i(x,y) e^{m-i} c^{\max\{0,t-i\}} \\
\text{where} &\quad i = 0, \ldots, m,\ j = m - i.
\end{aligned}
$$

NOTE THAT $g_i(x_0, y_0) \equiv 0 \bmod (e^m g^t)$.

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 $N_1 = p_1 q_1, N_2 = p_2 q_2$
- Same pair of keys e and d such that

$$ed \equiv 1 \bmod \phi(N_1)$$

$$ed \equiv 1 \bmod \phi(N_2)$$

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

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

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

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

$$ed_q \equiv 1 \bmod (q_2 - 1)$$

Cryptanalysis of Dual CRT-RSA

SARKAR AND MAITRA: DCC 2013

Theorem

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

$$e = N^{\alpha}, \qquad d_p, d_q < N^{\delta}.$$

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:

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

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:

$$e^2 y_1 + e y_2 + y_3 = (N_1 - 1) k_{p_1} k_{q_1}$$

$$e^2 y_1 + e y_4 + y_5 = (N_2 - 1) k_{p_2} k_{q_2}$$

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 $d < N^{(\frac{r-1}{r+1})^2}$: May (PKC 2004)
- When $r = 2$, $N^{\max\{\frac{2}{9}, \frac{1}{9}\}} = N^{\frac{2}{9}} \approx N^{0.22}$.

Sarkar: WCC 2013

Theorem

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

Proof Idea

- $ed \equiv 1 \bmod \phi(N)$ 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

$$
\begin{array}{rcl}
g_{i,j,k}(x,y,z) & = & x^j y^k z^{j+a} f_e^i(x,y,z) \\
& \text{where} & i = 0,\ldots,m,\ \ j = 1,\ldots,m-i,\ \ k = j, j+1, j+2 \text{ and} \\
g_{i,0,k}(x,y,z) & = & y^k z^a f_e^i(x,y,z) \\
& \text{where} & i = 0,\ldots,m,\ \ k = 0,\ldots,t.
\end{array}
$$

General Case

Recall

- $N = p^r q$
- $ed \equiv 1 \bmod p^{r-1}(p-1)(q-1)$

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

$$\begin{aligned} g_{i,j,k}(x,y,z) &= x^j y^k z^{j+a} f_e^i(x,y,z) \\ \text{where} \quad & i = 0,\ldots,m,\ j = 1,\ldots,m-i,\ k = j, j+1,\ldots,j+2r-2 \text{ and} \\ g_{i,0,k}(x,y,z) &= y^k z^a f_e^i(x,y,z) \\ \text{where} \quad & i = 0,\ldots,m,\ k = 0,\ldots,t. \end{aligned}$$

General Case

r	δ	$\max\left\{\frac{r}{(r+1)^2}, \left(\frac{r-1}{r+1}\right)^2\right\}$
2	0.395	0.222
3	0.410	0.250
4	0.437	0.360
5	0.464	0.444
6	0.489	0.510
7	0.512	0.562
8	0.532	0.605
9	0.549	0.640
10	0.565	0.669

Table: Numerical upper bound of δ 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

$$1001010100\,\overbrace{10100100101010010011}$$

Coppersmith (Eurocrypt 1996)

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

$$\overbrace{100101010010100}\,100101010010011$$

Boneh et al. (Asiacrypt 1998)

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

$$100101010010100\,\overbrace{100101010010011}$$

Herrmann and May (Asiacrypt 2008)

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

$$100\,\overbrace{1010100}\,10100\,\overbrace{1001010100}\,10011$$

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?

Sarkar and Maitra: IEEE-IT 2011

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* $\mathrm{poly}\{\log \mathrm{N}, \exp(\mathrm{k})\}$ *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

k	Bitsize of p_i, q_i $(1-\alpha)\log_2 N, \alpha \log_2 N$	No. of shared LSBs May et al. in p_i				No. of shared LSBs (our) in p_i			
		Theory	Expt.	LD	Time	Theory	Expt.	LD	Time
3	750, 250	375	378	3	< 1	352	367	56	41.92
* 3	700, 300	450	452	3	< 1	416	431	56	59.58
* 3	650, 350	525	527	3	< 1	478	499	56	74.54
# 3	600, 400	600	-	-	-	539	562	56	106.87
* 4	750, 250	334	336	4	< 1	320	334	65	32.87
* 4	700, 300	400	402	4	< 1	380	400	65	38.17
* 4	650, 350	467	469	4	< 1	439	471	65	39.18
* 4	600, 400	534	535	4	< 1	497	528	65	65.15

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

$$d_p \equiv d \bmod (p-1) \text{ and } d_q \equiv d \bmod (q-1).$$

- To decrypt the ciphertext C, one needs

$$C_p \equiv C^{d_p} \bmod p \text{ and } C_q \equiv C^{d_q} \bmod q.$$

Calculating x^y:

- $\ell_y = \lceil \log_2 y \rceil$ many squares
- $w_y = wt(bin(y))$ 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: ℓ_e, ℓ_N, ℓ_k
Output: p, d_p

Choose an ℓ_e bit odd integer e;
Choose random ℓ_k bit integer k_p coprime to e;
Find odd integer d_p such that $d_p \equiv e^{-1} \bmod k_p$;
$p = 1 + \frac{ed_p - 1}{k_p}$;

$$(\ell_e, \ell_N, \ell_d, \ell_k) = (176, 1024, 338, 2) \text{ WITH } w_{d_p} = w_{d_q} = 38$$

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:
  $N = pq, ed_p = 1 + k_p(p-1), ed_q = 1 + k_q(q-1)$
- We have:
  1. $q = p^{-1} N \bmod 2^a$
  2. $d_p = (1 + k_p(p-1))\, e^{-1} \bmod 2^a$
  3. $d_q = (1 + k_q(q-1))\, e^{-1} \bmod 2^a$

The Tool for Cryptanalysis

- w_{d_p}, w_{d_q} 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: τ.
- 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

δ	0.08	0.09	0.10	0.11	0.12	0.13
Suc. prob.	0.59	0.27	0.14	0.04	-	-
Time (sec.)	307.00	294.81	272.72	265.66	-	-
Suc. prob.	0.68	0.49	0.25	0.18	0.08	0.02
Time (sec.)	87.41	84.47	80.18	74.57	79.33	76.04

Lim et al (SAC 1996)

- $\ell_N = 768, \ell_{d_p} = 384, w_{d_p} = 30, e = 257; \Rightarrow \delta \approx \frac{30}{384} = 0.078$
- $\ell_N = 768, \ell_{d_p} = 377, w_{d_p} = 45, e = 257; \Rightarrow \delta = \frac{w_{d_p}}{\ell_{d_p}} \approx 0.12$

Galbraith et al (ACISP 2005)

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

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

Santanu Sarkar and Subhamoy Maitra. Cryptanalytic Results on Dual CRT and Common Prime RSA. Designs, Codes and Cryptography. Volume 66, Number (1-3), pp. 157-174, 2013.

Santanu Sarkar. Small Secret Exponent Attack on RSA Variant with Modulus $N = p^2q$. International Workshop on Coding and Cryptography, 2013. April 15-19, 2013, Bergen.

Santanu Sarkar and Subhamoy Maitra. Approximate integer common divisor problem relates to implicit factorization. IEEE Transactions on Information Theory, Volume 57, Number 6, pp. 4002-4013, 2011.

Santanu Sarkar and Subhamoy Maitra. Side Channel Attack to Actual Cryptanalysis: Breaking CRT-RSA with Low Weight Decryption Exponents. CHES 2012, LNCS 7428, pp. 476-493, 2012.


Thank You!