On small secret key attack against RSA with high bits known prime factor

Yasufumi Hashimoto, ISIT, Japan

Partially supported by JST Strategic Japanese-Indian Cooperative Programme on multidisciplinary Research Field, which combines Information and Communications Technology with Other Fields, entitled "Analysis of cryptographic algorithms and evaluation on enhancing network security Based on Mathematical Science"

RSA

$n = pq$.
$ed \equiv 1 \bmod (p-1)(q-1)$.

Public key: n, e.
Secret key: p, q, d.

$n = pq$.
$ed \equiv 1 \bmod (p-1)(q-1)$.

Encryption: $c \equiv m^e \bmod n$.
Decryption: $m \equiv c^d \bmod n$.

Factoring $n = pq \Rightarrow$ Deciphering RSA.

There are several sub-exponential time algorithms.

Analysis with special contions

1. If the higher (or lower) half bits of p are known, RSA is broken in polynomial time
(Coppersmith, 1997, Boneh-Durfee-Frankel, 1998).

2. If d (secret key) is small enough, RSA is broken in polynomial time ($d < n^{0.25}$, Wiener, 1990;
$d < n^{0.292\cdots}$, Boneh-Durfee, 2000).

3. Others, e.g. when higher (or lower) bits of d is known, etc.

Sarkar-Maitra-Sarkar's attack (2008)

The attack when
(not more than half) higher bits of p are known,
and d is small enough *(but* $d > n^{0.292\cdots}$*)*.

Known: p_1 s.t. $|p - p_1| < n^{\alpha}$ $(0 \leq \alpha \leq 1/2)$.
$d \sim n^{\delta}$.

If

$$\delta < 1 - \sqrt{\alpha},$$

then (p, q, d) will be found in polynomial time.

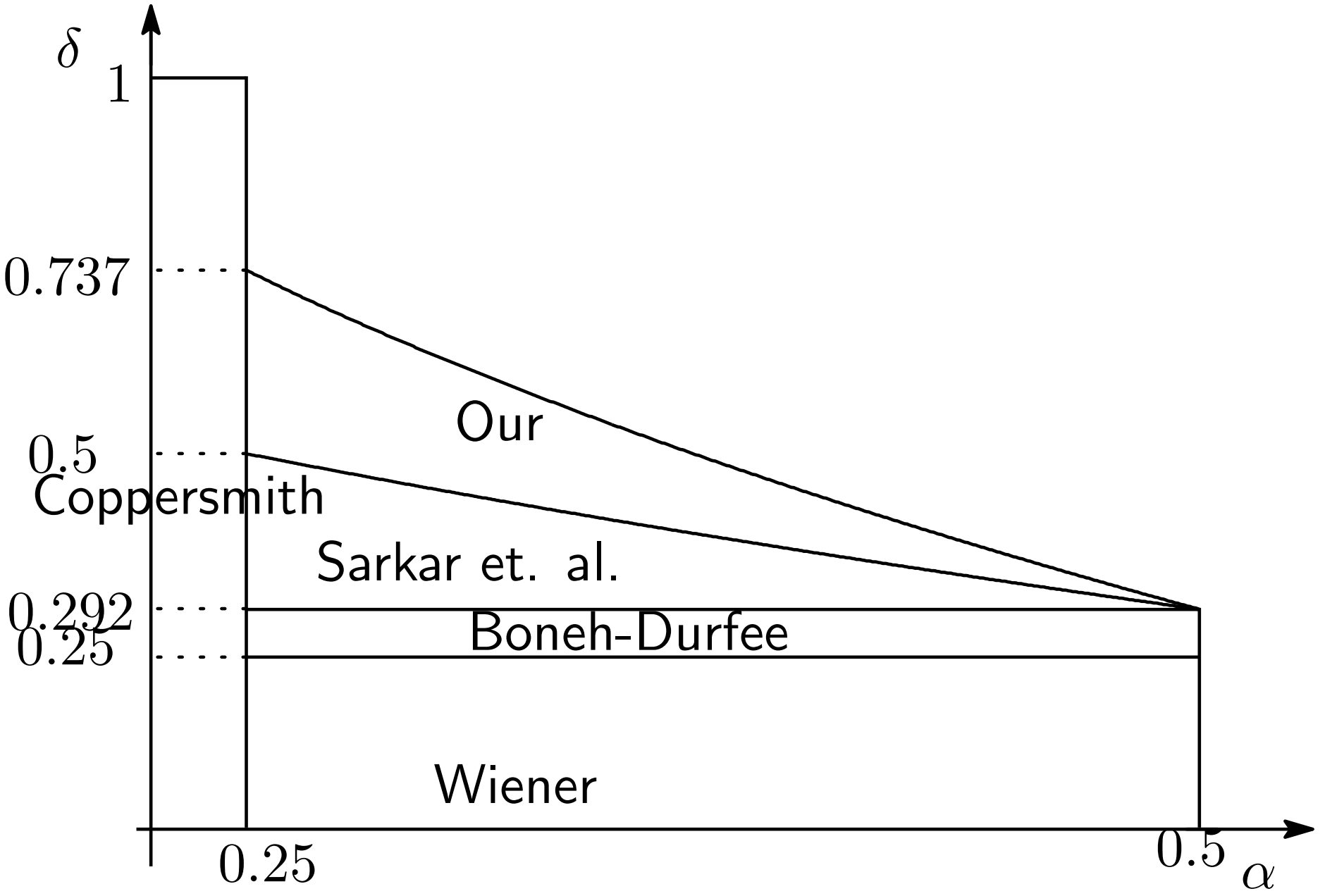
(If $\alpha = 1/2$, then $\delta < 1 - 1/\sqrt{2} = 0.292\cdots$).

Main result

If

$$\delta < \begin{cases} \frac{1}{4}\left(\alpha + \frac{9}{2} - \sqrt{\left(2\alpha - \frac{1}{3}\right)(10\alpha + 17)}\right), & (\alpha \leq 29/62), \\ \frac{1}{4}\left(5 - 2\alpha - \sqrt{(5-2\alpha)(6\alpha - 1)}\right), & (\alpha \geq 29/62), \end{cases}$$

then (p, q, d) will be found in polynomial time.


δ
1
0.737
0.5
0.292
0.25
Our
Coppersmith
Sarkar et. al.
Boneh-Durfee
Wiener
0.25
0.5
α

Outline of attacks using LLL

1. Describe the problem by *polynomial equations*.
2. Generate a *lattice* by the coeff. of the polynomials.
3. Find *small* vectors in the lattice.
4. Solve the equations derived from the small vectors.

Q. Why are *small* vectores required?

A. Equations/$(\mathbb{Z}/N\mathbb{Z})$ will be equations/$\mathbb{Z}$.
(e.g. $x^3 \equiv 8 \bmod N \Rightarrow x = ?$)

Howgrave-Graham's lemma.
$h(x_1, \cdots, x_n)$: a polynomial with w monomials.
$m, N, (x'_1, \cdots, x'_n), (X_1, \cdots, X_n)$: integers such that
1. $x'_1 < X_1, \cdots, x'_n < X_n$,
2. $h(x'_1, \cdots, x'_n) \equiv 0 \bmod N^m$,
3. $||h(x_1X_1, \cdots, x_nX_n)|| < N^m/\sqrt{w}$.
Then $h(x'_1, \cdots, x'_n) = 0$ holds over $\mathbb{Z}$.

Q. How to get small vectors?

A. Using the LLL algorithm.

LLL algorithm. (Lenstra-Lenstra-Lovasz, 1982)
$n_1 \geq n_2 \geq 1$.
$u_1, \cdots, u_{n_2} \in \mathbb{R}^{n_1}$: linearly independent vectors.
L: the lattice generated by $\{u_1, \cdots, u_{n_2}\}$.
$\det(L)$: the volume of the unit lattice.
The vectors with the following norms will be found in polynomial time.

$$||b_1|| \leq 2^{\frac{n_2-1}{4}} (\det L)^{\frac{1}{n_2}}, \quad ||b_2|| \leq 2^{\frac{n_2}{4}} (\det L)^{\frac{1}{n_2-1}}, \cdots$$

Aim.

1. Generate "good" polynomials and a lattice.
2. Estimate the size of unknowns with

$$2^{\frac{n_2-1}{4}} (\det L)^{\frac{1}{n_2}} < N^m / \sqrt{w}.$$

Tools.

1. $n = pq$.
2. $ed + k(n - p - q + 1) = 1$.

Basic Polynomial

1. For Boneh-Durfee's $\delta < 0.292\cdots$.

$$f(x, y) = x(n + 1 - y) - 1.$$

$(x, y) = (k, p + q)$ is a solution of $f(x, y) \equiv 0 \bmod e$.

2. For Sarkar-Maitra-Sarkar's $\delta < 1 - \sqrt{\alpha}$.

$p_2 := p - p_1, q := q - q_1 \ll n^{\alpha}$.

$$f(x, y) = x(n + 1 - p_1 - q_1 - y) - 1.$$

$(x, y) = (k, p_2 + q_2)$ is a solution of $f(x, y) \equiv 0 \bmod e$.

3. Our version.

$a, b \sim n^{\gamma}$: $\left|\dfrac{q_1}{p_1} - \dfrac{b}{a}\right| < n^{\alpha - 1/2}$.

$\Delta := aq_2 + bp_2 - \lfloor a(n - p_1 q_1)/p_1 \rfloor \sim n^{2\alpha - 1/2 + \gamma}$.

$k\,(a(n - p_1 - q_1 + 1) - ap_2 - aq_2) - a \equiv 0 \bmod ae$.
$k\,(\Delta + (b - a)p_2 + M) - a \equiv 0 \bmod ae$.

$f(x, y, z) := x(z - (a - b)y + M) - a \equiv 0 \bmod ae$
の解は， $x = k$, $y = p_2$, $z = \Delta$.

方程式の生成

$$F^{(0)}_{l,i,j}(x,y,z) := (ae)^{m-l}x^i z^j f^l(x,y,z),$$
$$F^{(1)}_{l,i,j}(x,y,z) := (ae)^{m-l}x^i y z^j f^l(x,y,z).$$

パラメーター.

$$\begin{cases} 0 \le l \le m, 0 \le i \le m-l, j = 0, \\ 0 \le l \le m, i = 0, 1 \le j \le t. \end{cases}$$

注：$n = pq \Rightarrow bp_2^2 = p_2\Delta + M_1 p_2 + p_1\Delta + M_2$
なので，F**の**y**次数は高々**1.

$m = 1,\ t = 1$**のときの格子** L.

	1	y	x	xy	xz	xyz	z	yz	xz^2	xyz^2
$F^{(0)}_{0,0,0}$	*									
$F^{(1)}_{0,0,0}$		*								
$F^{(0)}_{0,1,0}$			*							
$F^{(1)}_{0,1,0}$				*						
$F^{(0)}_{1,0,0}$	*		*	*	*					
$F^{(1)}_{1,0,0}$		*	*	*	*	*				
$F^{(0)}_{0,0,1}$							*			
$F^{(1)}_{0,0,1}$								*		
$F^{(0)}_{1,0,1}$					*	*	*		*	
$F^{(1)}_{1,0,1}$					*	*		*	*	*

行列式は対角成分の積.

LLL アルゴリズムで，HG's lemma の条件をみたすようなベクトルを作れるのは，

$$2^{\frac{n_1-1}{4}}(\det L)^{1/n_2} < (ae)^m/\sqrt{w},$$

つまり，(m を大きくとったとき)，

$$\begin{aligned}\delta < & \frac{5}{6} + \frac{2}{3}\alpha + \frac{4}{3}\gamma \\ & - \frac{1}{3}\sqrt{(4\alpha - 1 + 2\gamma)(4\alpha + 5 + 8\gamma)}\end{aligned}$$

のときであることがわかる．

改良

t: 「l に依存せず一定」 → 「l に比例する」

とすれば，次のようなよりよい評価を得る．

$$\delta < \begin{cases} \dfrac{1}{3}\Big(2+4\alpha+5\gamma \\ \quad -2\sqrt{(2\alpha-1/2+\gamma)(1+8\alpha+4\gamma)}\Big), & (8\alpha+3\gamma \le 3), \\ \\ 1+\gamma-\sqrt{(1+\gamma)\,(2\alpha-1/2+\gamma)}, & (8\alpha+3\gamma \ge 3). \end{cases}$$

($\alpha = 1/2,\ \gamma = 0$ のとき，$\delta < 0.292\ldots$．「主結果」よりよい).

まとめ

・秘密鍵が小さいとき，RSA**は多項式時間で破られる（**[Wiener], [BD], etc**）．**

・既知ビットが増えると，解析可能なd**は大きくなる（**[SMS]**）．**

・$p : q \sim$ **(小さい)** : **(小さい)** **のとき，解析可能な**d**は大きくなる（主結果）．**

(ii-2) Our cases.

The "geometrically progressive matrix" is not (directly) used, since the structure of the lattice is different.
Taking eliminations carefully, we see that, if

$$t_{l-1,0} \leq t_{l,0} \leq t_{l-1,1} + 1, \quad t_{l-1,1} \leq t_{l,1} \leq t_{l-1,0},$$

then F's are given by the linear combinations of

$$\{x^{i_1} y^{i_2}, x^{i_1} y^{i_2} z (i_1 \geq i_2),$$
$$y^j (xy + a)^l, y^j z (xy + b)^l (j \geq l)\}.$$

The condition

$t_{l-1,0} \leq t_{l,0} \leq t_{l-1,1} + 1, t_{l-1,1} \leq t_{l,1} \leq t_{l-1,0}$;

$$\{t_{l,0}\} = \{0, 1, 1, 2, 2, 3, \cdots\},$$
$$\{t_{l,1}\} = \{0, 0, 1, 1, 2, 2, \cdots\} : \text{good.}$$
$$\{t_{l,0}\} = \{0, 0, 1, 1, 1, 2, \cdots\},$$
$$\{t_{l,1}\} = \{0, 0, 0, 1, 1, 1, \cdots\} : \text{good.}$$
$$\{t_{l,0}\} = \{0, 1, 2, 3, 4, 5, \cdots\},$$
$$\{t_{l,1}\} = \{0, 0, 1, 2, 3, 4, \cdots\} : \text{bad.}$$

In such "good" cases, the lattice L is given by
This gives our bound
with

$$t_{l,0}, t_{l,1} \sim$$

Special cases: $p : q \sim a : b$ with small a, b.

If a, b with

$$\left| \frac{p}{q} - \frac{a}{b} \right| < \frac{1}{n^{1/2-\alpha}}$$

are smaller, larger d can be recovered.

Especially, a, b are too small, e.g. $(1, 1), (3, 2), (4, 5)$, etc, the upper bound is given by

Wegar(2001). If $|p-q| < n^{\alpha}$ and

$$\delta <$$

then p, q, d can be recovered in polynomial time.

This is just the case of $(a, b) = (1, 1)$ and the bound is almost same.
Our approach is also used to other cases.

Conclusion.

We improve the (theoretical) upper bound of the small secret key attack when the upper bits of p is known.

The size of the lattice of our lattice is about twice of SMS's lattice for the same m.

If p/q is approximated by a/b with small a, b, the upper bound is larger.