Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Cryptanalysis of RSA with two decryption exponents

Santanu Sarkar, Subhamoy Maitra *

Applied Statistics Unit, Indian Statistical Institute, 203 B T Road, Kolkata 700 108, India

ARTICLE INFO

Article history:
Received 6 May 2009
Received in revised form 26 November 2009
Accepted 26 November 2009
Available online 3 December 2009
Communicated by D. Pointcheval

Keywords:
Cryptography
RSA
Cryptanalysis
Factorization
Lattice
LLL algorithm

ABSTRACT

In this paper, we consider RSA with $N = pq$, where p, q are of same bit size, i.e., $q < p < 2q$. We study the weaknesses of RSA when multiple encryption and decryption exponents are considered with same RSA modulus N. A decade back, Howgrave-Graham and Seifert (CQRE 1999) studied this problem in detail and presented the bounds on the decryption exponents for which RSA is weak. For the case of two decryption exponents, the bound was $N^{0.357}$. We have exploited a different lattice based technique to show that RSA is weak beyond this bound. Our analysis provides improved results and it shows that for two exponents, RSA is weak when the RSA decryption exponents are less than $N^{0.416}$. Moreover, we get further improvement in the bound when some of the most significant bits (MSBs) of the decryption exponents are same (but unknown).

1. Introduction

One of the most popular cryptosystems in the history of cryptology, the public key algorithm RSA [11], can be described briefly as follows:

- primes p, q, with $q < p < 2q$;
- $N = pq$, $\phi(N) = (p-1)(q-1)$;
- e, d are such that $ed = 1 + k\phi(N)$, $k \geqslant 1$;
- N, e are publicly available and the plaintext M is encrypted as $C = M^e \bmod N$;
- the secret key d is required to decrypt the ciphertext as $M = C^d \bmod N$.

The study of RSA is one of the most attractive areas in cryptology research as evident from many excellent works (one may see [1,7,10] and the references therein for detailed information).

Wiener [12] showed that when $d < \frac{1}{3}N^{\frac{1}{4}}$ then N can be factored easily. Later, Boneh and Durfee [2] increased this bound up to $d < N^{0.292}$. Thus use of smaller d is in general not recommendable. In [6], the authors presented state-of-the-art results when more than one decryption exponents are available for a single RSA modulus N. It has been shown that in the presence of two decryption exponents (d_1, d_2), N can be factored in polynomial time when $d_1, d_2 < N^{\frac{5}{14}}$. In the presence of three decryption exponents the bound has been improved to $N^{\frac{2}{5}}$. In asymptotic sense, it has been shown in [6] that when large number of decryption exponents are available, then the upper bound of decryption exponents for which RSA is weak, approaches to N. However, in such a case, the algorithm of [6] becomes exponential in the number of decryption exponents.

In this paper, we exploit a different lattice based technique and improve the bound (i.e., $N^{\frac{5}{14}}$) of [6] significantly for the case of two decryption exponents; we show that the upper bound of the decryption exponents, for which RSA is weak, is $N^{0.416}$. One may note that our bound considering two decryption exponents is better than that achieved (i.e., $N^{\frac{2}{5}}$) in [6] exploiting three decryption exponents too. Our result has another component that it takes care of the case when some of the most significant bits (MSBs) of the decryption exponents are same (but un-

* Corresponding author.
E-mail addresses: santanu_r@isical.ac.in (S. Sarkar), subho@isical.ac.in (S. Maitra).

0020-0190/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2009.11.016

known). This implicit information increases the bound of decryption exponents further. We also present experimental results to show the improvements over the work of [6]. Following the idea of [8], we present a technique for which one needs to construct certain polynomials based on the number of available encryption exponents. The construction of these polynomials is quite tedious beyond two decryption exponents and we note that this method does not provide encouraging results for three or more decryption exponents.

As explained in the introduction of [6], we also agree that studying this kind of cryptanalysis may not have direct impact to RSA used in practice. However, there are few issues for which this kind of problems is interesting.

- This shows how one can find further weaknesses of RSA with additional public information – in this case more than one encryption exponent.
- Moreover, this shows how one can extend the ideas of [12,2], where a single encryption exponent is considered, to more than one exponent.

2. RSA cryptanalysis in the presence of two decryption exponents

Before proceeding further, the reader is referred to [3,4,8,5] and the references therein for details of lattice based techniques in this area and in particular to [8] for the strategy we follow. In this regard, we like to point out that the polynomial, that we use in Theorem 1, has not been studied earlier following the technique of [8] and one may note that these polynomials are not covered in [7, Table 3.2, Section 3.4] too. Further, we like to state the following assumption, which we find true for the experiments we have performed. We discuss it in more details after the technical results.

Assumption 1. *Consider a set of polynomials $\{f_1, f_2, \dots, f_i\}$ on n variables, $i \geqslant n$, having the roots over integers of the form $(x_{1,0}, x_{2,0}, \dots, x_{n,0})$. Then we will be able to collect the roots efficiently by calculating the resultants of these polynomials. One may also assume that the roots can be recovered by using the Gröbner basis computation.*

Now we present the result when two encryption exponents are available.

Theorem 1. *Let (e_1, e_2) be two RSA encryption exponents with common modulus N. Suppose d_1, d_2 are the corresponding decryption exponents. Let $d_1, d_2 < N^\delta$ and $|d_1 - d_2| < N^\beta$. Then, under Assumption 1, one can factor N in $\operatorname{poly}(\log N)$ time when*

$$\frac{1}{12}\beta + \frac{1}{6}\delta - \frac{5}{48} < 0.$$

Proof. We have

$$e_1 d_1 = 1 + k_1(N + r), \tag{1}$$

and

$$e_2 d_2 = 1 + k_2(N + r), \tag{2}$$

where $r = -p - q + 1$. Multiplying the first equation by e_2 and the second one by e_1 and then subtracting them, we get

$$e_1 e_2(d_1 - d_2) = (e_2 - e_1) + (N + r)(k_1 e_2 - k_2 e_1). \tag{3}$$

We want to find the solutions $d_1 - d_2, r, k_1, k_2$ of the polynomial

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= e_1 e_2 x_1 - (e_2 - e_1) - (N + x_2)(e_2 x_3 - e_1 x_4) \\ &= e_1 e_2 x_1 - (e_2 - e_1) - N e_2 x_3 + N e_1 x_4 - e_2 x_2 x_3 \\ &\quad + e_1 x_2 x_4. \end{aligned}$$

It is given that $|d_1 - d_2| < N^\beta$, and also we have $|r| < (1 + \sqrt{2})N^{\frac{1}{2}}$, $k_1 < N^\delta$, $k_2 < N^\delta$. Let $X_1 = N^\beta, X_2 = N^{\frac{1}{2}}, X_3 = N^\delta, X_4 = N^\delta$. Then X_1, X_2, X_3, X_4 are the upper bounds of $d_1 - d_2, r, k_1, k_2$ neglecting the constant terms.

In the strategy of [8, p. 273], the set S is the set of all monomials of f^m for a given positive integer m.

The set M is defined as the set of all monomials that appear in $x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} f(x_1, x_2, x_3, x_4)$, with $x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in S$. So, in this case, S and M are

$$S = \bigcup \{x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \colon x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \text{ is a monomial of } f^m\},$$

$$M = \{\text{monomials of } x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} f \colon x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in S\}.$$

It follows that,

$$x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in S \quad \Leftrightarrow \quad \begin{cases} i_1 = 0, \dots, m, \\ i_3 = 0, \dots, m - i_1, \\ i_4 = 0, \dots, m - i_1 - i_3, \\ i_2 = 0, \dots, i_3 + i_4, \end{cases}$$

and

$$x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in M \quad \Leftrightarrow \quad \begin{cases} i_1 = 0, \dots, m + 1, \\ i_3 = 0, \dots, m + 1 - i_1, \\ i_4 = 0, \dots, m + 1 - i_1 - i_3, \\ i_2 = 0, \dots, i_3 + i_4. \end{cases}$$

Apart from f, we need to find at least three more polynomials f_0, f_1, f_2 that share the same root $(d_1 - d_2, r, k_1, k_2)$ over the integers. From [8], we know that these polynomials can be found by LLL [9] algorithm in poly $(\log N)$ time if $X_1^{s_1} X_2^{s_2} X_3^{s_3} X_4^{s_4} < W^s$ for $s_j = \sum_{x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in M \setminus S} i_j$, where $j = 1, \dots, 4$ and $s = |S|$, $W = \|f(x_1 X_1, x_2 X_2, x_3 X_3, x_4 X_4)\|_\infty \geqslant N e_2 X_3 \approx N^{2+\delta}$ (assuming e_2 is of full bit-size).

One can check that

$$s_1 = \frac{1}{12}m^4 + \frac{2}{3}m^3 + \frac{23}{12}m^2 + \frac{7}{3}m + 1,$$

$$s_2 = s_3 = s_4 = \frac{1}{8}m^4 + \frac{13}{12}m^3 + \frac{27}{8}m^2 + \frac{53}{12}m + 2, \quad \text{and}$$

$$s = \frac{1}{12}m^4 + \frac{2}{3}m^3 + \frac{23}{12}m^2 + \frac{7}{3}m + 1.$$

For a given integer m, from our definition of S and M and neglecting the lower order terms we have the required condition

Table 1
Comparison of theoretical and experimental results. By $|N|$, we mean the bit size of N and so on.

$\|N\|$	$\|d_i\|$			
	Theory [6]	Expt. [6]	Our Theorem 1	Our expt.
500	178	178	208	192
700	250	249	291	267
1000	357	–	416	383

$$X_1^{\frac{1}{12}}(X_2X_3X_4)^{\frac{1}{8}} < W^{\frac{1}{12}}.$$

Substituting the values of X_1, X_2, X_3, X_4 and lower bound of W in this inequality, we get $\frac{1}{12}\beta + \frac{1}{4}\delta + \frac{1}{16} < \frac{1}{12}(2+\delta)$. Thus, we arrive at the condition $(\frac{1}{12}\beta + \frac{1}{6}\delta - \frac{5}{48}) < 0$.

Under this condition, we have four polynomials f, f_0, f_1, f_2 that share the same root $(d_1 - d_2, r, k_1, k_2)$ over the integers. Then under Assumption 1, one can find the root and thus RSA can be broken efficiently. □

Corollary 1. *Let (e_1, e_2) be two RSA encryption exponents with the common modulus N. Suppose d_1, d_2 are the corresponding decryption exponents. Then, under Assumption* 1, *one can factor N in* poly(log N) *time when $d_1, d_2 < N^{0.416}$.*

Proof. The proof follows from Theorem 1, putting $\beta = \delta$, i.e., when no information is assumed regarding the equality of MSBs in d_1, d_2. □

We like to point out that the result of Corollary 1 extends the upper bound on d_1, d_2 which is $N^{0.416}$ than the bound $N^{0.357}$ presented in [6, Section 3.2].

Now let us present our experimental results to show how it improves that of [6]. As the lattice used in [6] are of small dimensions, the time required was of the order of a few seconds. However, using our strategy, one requires a lattice of higher dimensions than that of [6] and thus the time required are of the order of a few hours. It can be checked that the dimension of the lattice, i.e., $|M|$ in the proof of Theorem 1 is $\frac{1}{12}m^4 + m^3 + \frac{53}{12}m^2 + \frac{17}{2}m + 6$.

We get substantially better experimental results than [6] for $m = 3$, i.e., when the lattice dimension is 105. We have implemented the programs in SAGE 3.1.1 over Linux Ubuntu 8.04 on a laptop with Dual CORE Intel(R) Pentium(R) D CPU 1.83 GHz, 2 GB RAM and 2 MB Cache. Due to constraint on lattice dimensions in experiments (we choose lattice parameters such that the program terminates in reasonable time, e.g., we use $m = 3$ for experiments), we cannot achieve the theoretical bound that we present in Theorem 1. However, we list the experimental results such that they improve the theoretical bound presented in [6].

As in Theorem 1, we have considered Assumption 1, let us now clarify how it actually worked. In the proof of Theorem 1, we considered that we will be able to get at least three polynomials f_0, f_1, f_2 along with f, that share the integer root.

In experiments we found more than 4 polynomials (other than f) after the LLL algorithm that share the root, and let us name them f_0, f_1, f_2, f_3. Let $R(f, f_0)$ be the resultant of f, f_0 and so on. We calculate $f_4 = R(f, f_0), f_5 = R(f, f_1), f_6 = R(f, f_2), f_7 = R(f, f_3)$ and then $f_8 = R(f_4, f_5), f_9 = R(f_6, f_7)$. In all the experiments, we observe that $x_3^4x_4^4$ is the GCD of f_8, f_9. Then we calculate $f_{10} = R(\frac{f_8}{x_3^4x_4^4}, \frac{f_9}{x_3^4x_4^4})$ and we find that f_{10} is a polynomial in x_4 only which corresponds to k_2 in the proof of Theorem 1. Since $d_1, d_2 < N^{0.416}$ and $p+q < (1+\sqrt{2})N^{0.5}$, we have $p+q < e_2$. Thus, we can find $p+q$ by calculating $(N+1+k_2^{-1}) \bmod e_2$ and this immediately provides the factorization of N.

Note that in Theorem 1, we have considered the case that $|d_1 - d_2| < N^\beta$. When a few MSBs of d_1, d_2 are shared (but not known), then $\beta < \delta$. As more MSBs are shared, β decreases and δ increases. As example, a few numerical values of $\langle\delta, \beta\rangle$ following the constraint in Theorem 1 are $\langle 0.416, 0.416\rangle$, $\langle 0.45, 0.35\rangle$ and $\langle 0.5, 0.25\rangle$.

In terms of experimental results, referring to Table 1, we find that for 500-bit N, the bound we could reach for d_1, d_2 is 192 bits for which RSA is weak. With the knowledge that 61 many MSBs of d_1, d_2 are same (no other information about the bits), this bound can be extended to 200 bits.

3. Conclusion

In this paper we have shown that RSA is weak when two encryption exponents are available for the same modulus and the (unknown) RSA decryption exponents are less than $N^{0.416}$. Further improvements in this bound has been achieved when some amount of MSBs of the decryption exponents are same (but unknown).

One may note that the extension of our results to many decryption exponents is quite tedious to handle and it needs to be considered in a case by case basis. For the case of three decryption exponents, one may add one more equation

$$e_3d_3 = 1 + k_3(N + r), \tag{4}$$

with the earlier equations (1), (2). However, we are not successful to improve the bounds of [6] in this manner. As an example, considering a similar approach as in (3), one may try to consider the following:

$$\begin{aligned} &e_1e_2(d_1 - d_2) - (e_2 - e_1) \\ &\quad = (N + r)(k_1e_2 - k_2e_1) \quad \text{and} \\ &e_1e_3(d_1 - d_3) - (e_3 - e_1) \\ &\quad = (N + r)(k_1e_3 - k_3e_1), \quad \text{where } r = -p - q + 1. \end{aligned}$$

However, we have checked that the polynomial, eliminating $(N + r)$, does not provide improved results over [6]. Thus the effectiveness of extending our strategy, as in the case of two decryption exponents, does not look promising for more than two exponents.

Acknowledgements

Detailed comments from the anonymous reviewer improved the technical as well as editorial quality of this paper. The first author likes to acknowledge the Council of Scientific and Industrial Research (CSIR), India for supporting his research fellowship and also Mr. Sumit Kumar Pandey of Indian Statistical Institute for relevant discussion.

References

[1] D. Boneh, Twenty years of attacks on the RSA acyptosystem, Notices of the AMS 46 (2) (February 1999) 203–213.

[2] D. Boneh, G. Durfee, Cryptanalysis of RSA with private key d less than $N^{0.292}$, IEEE Transactions on Information Theory 46 (4) (2000) 1339–1349.

[3] D. Coppersmith, Small solutions to polynomial equations and low exponent vulnerabilities, Journal of Cryptology 10 (4) (1997) 223–260.

[4] J.-S. Coron, Finding small roots of bivariate integer equations revisited, in: Eurocrypt 2004, in: LNCS, vol. 3027, 2004, pp. 492–505.

[5] N. Howgrave-Graham, Finding small roots of univariate modular equations revisited, in: Proceedings of Cryptography and Coding, in: LNCS, vol. 1355, 1997, pp. 131–142.

[6] N. Howgrave-Graham, J.-P. Seifert, Extending Wiener's attack in the presence of many decryption exponents, in: CQRE 1999, in: LNCS, vol. 1740, 1999, pp. 153–166.

[7] E. Jochemsz, Cryptanalysis of RSA variants using small roots of polynomials, PhD thesis, Technische Universiteit Eindhoven, 2007.

[8] E. Jochemsz, A. May, A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants, in: Asiacrypt 2006, in: LNCS, vol. 4284, 2006, pp. 267–282.

[9] A.K. Lenstra, H.W. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, Mathematische Annalen 261 (1982) 513–534.

[10] A. May, Using LLL-reduction for solving RSA and factorization problems: A survey, in: LLL + 25, Conference in Honour of the 25th Birthday of the LLL Algorithm, 2007; Available at http://www.cits.rub.de/personen/may.html [last accessed 23 November, 2009].

[11] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Communications of ACM 21 (2) (Feb. 1978) 158–164.

[12] M. Wiener, Cryptanalysis of short RSA secret exponents, IEEE Transactions on Information Theory 36 (3) (1990) 553–558.