# Common Private Exponent Attack on Multi Prime RSA

Santosh Kumar Ravva

*(Corresponding author: Santosh Kumar Ravva)*

Department of IT, MVGR College of Engineering

Vijayaram Nagar campus, Chintalavalasa, Vizianagaram, Andhra Pradesh 535005, India

(Email: santosh@mvgrce.edu.in)

*(Received Aug, 5, 2017; revised and accepted Oct. 10, 2017)*

**Abstract**

Multiprime RSA is a variant of RSA, where the modulus is the product of three or more prime numbers. In this paper, we attack Multiprime RSA. Our attack assumes that many instances of Multiprime RSA all use different moduli, but somehow all use the same secret exponent. Our attack generalizes the existing attack on RSA by Hinek. We use lattice reduction techniques to recover the bound for secret exponent.

*Keywords: Lattices; Lattice Reduction; Multiprime RSA*

# 1 Introduction

## 1.1 Multiprime RSA

RSA cryptosystem [1–8, 11, 13, 15–17] is most popular cryptosystem from its invention Multi Prime RSA: Multi Prime RSA is a simple extension of RSA in which the modulus is the product of $r$ distinct primes. In this paper, we consider only balanced primes. If we arrange the primes in increasing order, $p_i < p_{(i+1)}$ for $i = 1, 2, \cdots, r$, then we assume that $4 < 1/2N^{(1/r)} < p_1 < N^{(1/r)} < p_r < 2N^{(1/r)}$. The key generation algorithm is same as the key generation algorithm for RSA except here we require $r$ distinct primes. As usual, the public and private key are defines as $ed \equiv 1(\bmod \phi(N))$, where $k$ is some positive integer. As in RSA, one can replace $\phi(N)$ with $N - s$. Expanding $\phi(N)$, it follows that $s$ can be written as

$$\begin{aligned} s &= N - \phi(N) \\ &= N - \prod_{(i=1)}^{r} (p_i - 1) \\ &= \sum_{(i=1)}^{r} \frac{N}{p_i} - \sum_{(i,j=1)}^{r} \frac{N}{(p_i p_j)} + \sum_{(i,j,k=1)}^{r} \frac{N}{(p_i p_j p_k)} + \cdots + (-1)^r. \end{aligned}$$

The above expression of $s$ combined with the condition for balanced primes, an upper bound on $s$ is given by $|s| < (2r - 1)N^{(1-1/r)}$.

Thus, there are $(r-1)/r$ most significant bits are common in the $\phi(N)$ and $N$, so $N$ is a good approximation for $\phi(N)$.

### 1.2 Comparison Between RSA and Multiprime RSA

The encryption algorithm for multiprime RSA is same as the encryption algorithm for RSA. Given plain text message $m$, the cipher text is calculated by $c = m^e \bmod N$. The decryption for multi prime RSA is same as the decryption for RSA, if one consider the standard decryption. If decryption uses Chinese remaindering theorem, the decryption algorithm for the multi prime RSA is the obvious extension to the decryption algorithm for CRT-RSA. The efficiency of multi prime RSA depends on two issues. First one is, the complexity of generating the $r$ distinct primes is lower than the generating two distinct primes for the original RSA. The second one is, if Chinese remaindering is used for the decryption, then the decryption costs are lower than the decryption costs for CRT-RSA.

### 1.3 Breaking Multiprime RSA

If the factorization of modulus is known, then one can break the modulus. In RSA, it is sufficient to recover the private exponent or to compute $\phi(N)$ since there are polynomial time algorithms that can factor the modulus given either of these. But there is a different issue for the multi prime RSA. There are no polynomial time algorithms that can factor the modulus given the private exponent or $\phi(N)$. But if we know the multiple of $\phi(N)$, the results of Miller can be used to probabilistically factor the modulus. Also from $ed \equiv 1 \bmod (\phi(N))$, knowing $d$ is sufficient to obtain the private exponent in order to (probabilistically) factor the modulus.

In this paper, we attack on the Multi prime RSA, if multi prime RSA is used in broadcast scenario. That is, the same message broadcasts to several people with same private exponent but different moduli. Rest of the paper is organized as follows. In Section 2, we introduce some mathematical preliminaries, In Section 3, we sketch the attack with justification. In Section 4, we provide some experimental results.

## 2 Terminology

### 2.1 Lattices

Let $B = \{b_1, b_2, \cdots, b_n\}$ be set of n linearly independent vectors in $R^m$. The lattice generated by $B$ is the set $L(B) = \{\sum_{(i=1)}^{n} x_i \overrightarrow{b_i} : x_i \in Z\}$. That is, the set of all integer linear combinations of the basis vectors. The set $B$ is called basis and we can compactly represent it as an $m \times n$ matrix each column of whose is a basis vector: $B = [b_1, b_2, \cdots, b_n]$. The rank of the lattice is defined as $rank(L) = n$ while its dimension is defined as $dim(L) = m$. The volume (determinant) of a lattice denoted by $vol(L)$, is the $n$ dimensional volume of the parallelepiped spanned by any of it bases. For full dimensional lattice $vol(L) = |det(B)|$. Since lattice is discrete, there exists a smallest vector. The necessary condition for a vector $v$ to be a smallest vector in the lattice is $||v|| \leq \sqrt{n} vol(L)^{\frac{1}{n}}$, which is called Minkoswki's bound. This bound is useful as it allows for constructing the bounds on certain attacks. For good introduction of lattices and their applications refer [12, 14].

Finding the shortest vector in the lattice is a hard problem. There are some approximation algorithms to find a shortest vector in the lattice. Here we use the LLL algorithm, because it is well suited in the most of the attacks in practice.

## 2.2 Lattice Reduction

Lattice reduction is a problem to find the reduced basis of the given lattice. Reduced basis is the basis of the lattice such that the vectors are near orthogonal. So many versions exist to find reduced basis, but the one given by Lenstra, Lovasz, Lovasz is a special one, called LLL reduced. Because there exist a polynomial time algorithm for this reduction called LLL algorithm. This problem is not only solving the reduced problem, it also gives solution to the shortest vector problem in some extent.

**Definition 1** (LLL Reduced)**.** *Let $b_1, b_2, \cdots, b_n$ be a basis for a lattice and let $b_1^*, b_2^*, \cdots, b_n^*$ be its Gram-Schimdt orthogonalization. The basis $b_1, b_2, \cdots, b_n$ is said to be Lovaász-reduced or LLL-reduced, if the Gram-Schimdt coefficients satisfy $|\mu_{(i,j)}| \leq 1/2$ for $1 \leq j < i \leq n$, and $||b_i^* + \mu_{(i,i-1)} b_i^*||^2 \geq \frac{3}{4}||b_{(i-1)}^*||^2$ for $1 < i \leq n$, or equivalently $||b_i^*||^2 \geq (\frac{3}{4} - \mu_{(i,i-1)}^2)||b_{(i-1)}^*||^2$ for $1 < i \leq n$.*

A useful property of LLL reduced basis is that the bound for each vector depends on only the dimension and the volume. The property stated as in [14]. Let $L$ be a lattice spanned by linearly independent vectors $b_1, b_2, \cdots, b_n$, where $b_1, b_2, \cdots, b_n \in R^n$. By $b_1^*, b_2^*, \cdots, b_n^*$, we denote the vectors obtained by applying the Gram-Schimdt process to the vectors $b_1, b_2, \cdots, b_n$. It is known that given basis $b_1, b_2, \cdots, b_n$ of lattice $L$, LLL reduced find a new basis $b_1, b_2, \cdots, b_n$ of $L$ with the following properties:

$$\begin{aligned} ||b_i^*||^2 &\leq 2||b_{(i+1)}^*||^2; \\ ||b_1|| &\leq 2^{(n/2)} det(L)^{(1/n)}; \\ ||b_2|| &\leq 2^{(n/2)} det(L)^{(1/(n-1))}. \end{aligned}$$

The determinant of is defined as $det(L) = \prod_{(i=1)}^{w} ||b_i^*||$, where $||$ denotes the Euclidean norm on vectors. The LLL algorithm is the first algorithm to compute LLL reduced basis efficiently. For given a m dimensional lattice with n dimensional lattice vectors the LLL algorithm has run time $o(nm^5B^3)$, where $B$ is the bound on the bit length of the input basis vectors.

## 2.3 Existing Attacks on Multiprime RSA

The most of the attacks on RSA can be generalized into Multi prime RSA. The first attack is the Wiener attack, stated in [9].

**Attack 1:** Let $N$ be an r-prime modulus with balanced primes, let $e$ be a valid public exponent and $d$ be its corresponding private exponent. Given the public key $(N, e)$, if the private exponent satisfies $d \leq \frac{N^{(1/r)}}{(2k(2r-1))}$, then the modulus can be (probabilistically) factored in time polynomial in $\log N$ for every $r \geq 2$.

The second attack is generalization Boneh-Durfee attack on RSA.

**Attack 2:** For every $\epsilon > 0$ and integer $r \geq 2$ there exists an $n_0$ such that, for every $n > n_0$, the following holds: Let $N$ be an $n$-bit $r$-prime RSA modulus with balanced primes, let $e = N^\alpha$ be a valid public exponent and let $d = N^\delta$ be its corresponding private exponent. Given the public key $(N, e)$, if the private exponent satisfies $\delta \leq \frac{1}{3r}(4r - 1 - 2\sqrt{(r-1)(r-1+3\alpha r)}) - \epsilon$, then the modulus can be (probabilistically) factored in time polynomial in n under some assumption. The above attacks are for the single instance of Multiprime RSA. There are some attacks on Multiprime RSA by considering the several instances of the same message. For example, common modulus attacks, in which same message send to the different people with the same modulus. The encryption and decryption exponents may be different. The second type is common private exponent attack, in which same message send to the different people with the same private exponent but may be different moduli and different public exponents, called common private exponent attack. In this paper, we consider the common private exponent attack on Multiprime RSA. The attack exists in the case of RSA and it is stated in [10]. We mention the same here.

**Attack 3:** For any integer $r \geq 1$, let $N_1, N_2, \cdots, N_r$ be balanced RSA moduli satisfying $N_1 < N_2 < \cdots < N_r < 2N_1$. Let $(e, N_1), \cdots, (e, N_r)$ be valid public RSA keys each with the same private exponent $d < N_r^{\delta_r}$. If $\delta_r < \frac{1}{2} - \frac{1}{2(r+1)} - \log_{N_r}(6)$, then all of the moduli can be factored in time polynomial in $\log(N_r)$ and $r$, under the some assumption. For the justification of above attack please refer [10]. In the next section, we introduce the attack on Multi prime RSA and its proof.

# 3 Attack on Multiprime RSA

## 3.1 Attack

For any integer $n \geq 1$, let $N_1, N_2, \cdots, N_n$ be balanced Multi prime RSA with $r$ primes $N_1 < N_2 < N_3 < \cdots < N_n < 2N_1$. Let $(e_1, N_1), \cdots, (e_n, N_n)$ be valid Multi prime RSA public keys each with the same private exponent $d < N_n^{\delta_n}$. If $\delta_n < \frac{n}{r(n+1)} - \log_{N_n}(4r-2)$, then all of the moduli can be factored in time polynomial in $log(N_n)$ and $n$.

## 3.2 Justification

Let $M = \lfloor N_n^{1-1/r} \rfloor$. Given the $n$ public keys $(e_1, N_1), \cdots, (e_n, N_n)$ and $d$ is a secret exponent for all instances. We begin by considering the $n$ key equations, $e_i d = 1 + k_i(N_i - s_i)$ along with the trivial equation $dM = dM$, written as

$$\begin{aligned} dM &= dM \\ e_1 d - N_1 k_2 &= 1 - k_1 s_1 \\ e_2 d - N_2 k_2 &= 1 - k_2 s_2 \\ \vdots \quad &\cdots \quad \vdots \\ e_n d - N_n k_n &= 1 - k_n s_n. \end{aligned}$$

The above system of equations can be written as $x_n B_n = v_n$, where $x_n = (d, k_1, k_2, \cdots, k_n)$ and

$$B_n = \begin{bmatrix} M & e_1 & e_2 & \cdots & e_n \\ 0 & -N_1 & 0 & \cdots & 0 \\ 0 & 0 & -N_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -N_n \end{bmatrix}$$

$$v_n = (dM, 1 - k_1 s_1, \cdots, 1 - k_n s_n).$$

The vector $v_r$ is an integer linear combination of the rows in the matrix $B_n$ and hence is a vector in the lattice $L$ generated by the rows in $B_n$. Since $N_i \leq N_r < 2N_1$, $k_i < d < N_n^{\delta_n}$ and $|s_i| < (2r-1)N^{1-1/r}$ for each $i = 1, 2, \cdots, n$, so the vector $v_r$ satisfies $||v_r|| < \sqrt{1 + n(2r-1)^2} N_n^{\delta_n + 1 - 1/r}$ Since

$$\begin{aligned} ||v_r||^2 &= (dM)^2 + (1 - k_1 s_1)^2 + \cdots + (1 - k_n s_n)^2 \\ &\leq (N_n^{\delta_n + 1 - 1/r})^2 + (1 - N_n^{\delta_n + 1 - 1/r}(2r-1))^2 + \cdots + (1 - N_n^{\delta_n + 1 - 1/r}(2r-1))^2 \\ &= (N_n^{\delta_n + 1 - 1/r})^2 + n(1 - N_n^{\delta_n + 1 - 1/r}(2r-1))^2 (1 + (2r-1)^2 n)(N_n^{\delta_n + 1 - 1/r})^2 \end{aligned}$$

So we have $||v_r|| < \sqrt{(1+(2r-1)^2 n)}(N_n^{\delta_n+1-1/r})$, and that the volume of the lattice $L$, given by $vol(L) = |det(B_n)|$, satisfies $vol(L) = |M \prod_{i=1}^{n}(-N_i)| = \lfloor N_n^{1-1/r} \rfloor \prod_( i = 1)^n N_i > (N_n/2)^{(n+1-1/r)}$. From Minkowski's bound, a necessary condition for the vector $v_r$ to be a smallest vector in $L$ is given by $||v_r|| < \sqrt{(n+1)} vol(L)^{1/(n+1)}$. Using the bounds on the norm of the vector and the volume of the lattice, a sufficient condition to hold is given by

$$\sqrt{(1+(2r-1)^2 n)}(N_n^{\delta_n+1-1/r}) < \sqrt{(n+1)}(\frac{N_n}{2})^{\frac{(n+1-1/r)}{n+1}}.$$

This implies, we have

$$N_n^{\delta_n+1-1/r} < c_r (N_n/2)^{(n+1-1/r)/(n+1)}$$

where

$$c_r = \sqrt{(n+1)/(1+(2r-1)^2 n)} \frac{1}{2^{\frac{n+1-1/r}{n+1}}} > (\frac{1}{2r-1})(\frac{1}{2}).$$

Compare both sides, we get $\delta_n + 1 - \frac{1}{r} < \frac{n+1-1/r}{n+1} - log_{N_r}(4r-2)$. After simplification, we get $\delta_n < \frac{n}{r(n+1)} - log_{N_n}(4r-2)$. When $r = 2$, the bound equals the bound in paper [10]. So when the secret exponent is smaller than $\delta_n$ the vector $v_n$ has satisfied the Minkowski condition, be a smallest vector in $L$. Once the vector $v_r$ is obtained we can easily factor all the moduli. From the vector $v_r$, one knows the secret exponent $d$, in turn one can compute all $k_i$'s by using the key equations $k_i = (e_i d - (1 - k_i s_i))/N_i$. From $k_i$ and $d$, one can compute $\phi(N_i) = ((e_i d - 1))/k_i$. But unfortunately, there are no deterministic algorithm to compute $N$ from $\phi(N)$, if $N$ is a product of three or more numbers. But there is probabilistic algorithm exists (MILLER-RABIN) to compute $N$ from multiplies of $\phi(N)$.

### 3.3 Experiment

We experiment the above attack for three instances. Nowadays, the RSA modulus is 1024 bits. We have used SAGE [18] for doing all these calculations. SAGE is freely available library. Three prime numbers for first instance:

6782249115473301479860934781998946025937950413041213
356008546789157286634233311790115626404827023528453

7084744194090403239794293861446440061838524816853834
1575339547370180904074125080422385071927294881484447

8283468323162452783011818008723783652632569943569787
5085934866124166647680648543086349174694583006418 93

The modulus for first instance:

3980247950243058401243698615396728611119297559768520
6955726459272809486797422213585601085391473876721 45

5746434097489726940858155275925156644720661899031031
5226908681496694169451736542376873921000098512243 73

1954119056065582710154780722805904907071148681687475
673362851582112927479576695012270482243892047135463

Three prime numbers for second instance:

7112016950782513939301172150838079539230135679999340
432930882928796532218789384885663524476743197436939

4565359706261800235586539334646285991321427933608889
275210263010247136380569841822416543375070004144969

8316153309780533698216542357490673442538504203220461
981455565890736516986256830111465405685281007074759

The modulus for second instance:

2700164800762383548238989792096448123453203959676756
421370584710307069143336087323834665054768084857672

6636901769349674889016838942316143898411650895896357
722682461314710330128572111448604468952040247034828

5890303091007520440669270829884833458242141570600997
934491748273501275775301365373289773778866000841269

Three prime numbers for third instance:

7673016393834847941843640704678652781863359389660737
319442571552847363819488175152409712437263055468297

5377937762598075667499364718566642313937013965068651
966132665108749552319333790553717555089278142310147

5395342771000978217135825503436485980920045858078257
716039638576857698703568752771925132861090440121039

The modulus for third instance:

2226388443580189980267475930380569711871363604160786
505024146752022423699461394900237896180365281928686

8926856402008312979469194065931803571405154678409393
428008287143020782113965268621088147936991953030971

8853455860497497103437250790722599219470946672980313
102681290912136274611071861434179694093260525215701

Three Public exponents are:

3537696462947686474560092541384577100358586899102915
517881043631908384269507494307195562677701087700290

1921140746265206159324299005398887508559711938234015

3105007419987038334725462998580666261416020578531 63

5308109000893327620367972849510781232554948587141 7366
930349231085723113959443386893863404314445906898291

21197935829055974083763028541932043967055863250890 51
517217681564481829335135017685804292917335888246598

28833893582465665022776197479728184880397254999484 16
842432981714327510908996095416699707520836670108868

93041603196162462591064776047417561472027826932728 50
299066990031340778958229124744300299133384683262867

92143269622343461572326361086215626147562583797349 78
899539837473864655669157559327263924932563098153763

07173323676529429644700711031216855720683474038270 92
664257936996711466179189821591897255426658826384606

66639874186629328136670465473123321702936951734455 53
881776416131904134196664929183744841313190701469 63

The first value in the first row is

23095428212222336506037218916978757398204437199048 88
661639426983506182215086103388029979298810386972233

40542745055588530524397301568187946310400786496215 9270845
1671327366552059289826564670952931069623218348216 8639488

The required private exponent is 1.54783815979006e61 Actually this is the secret exponent, we have used in the beginning of the attack. We retrieved by using LLL algorithm. The attack uses the prime numbers of the length 1024 bits with $r = 3, 4$. If the instances are more, then one can easily break the system.

### 3.4 Practical Effectiveness

The above attack is only heuristic; the original value lies in the practice. Already we showed the successful attack as toy example. We checked the random instances of Multi prime RSA with 1024 bits moduli when a common private exponent is shared among different moduli in the range $2 \leq n \leq 10$ and $r = 3, 4$. We use the SAGE Library for experimentation. We observe that, if more instances are available, then one can easily break the system for for the values of $r = 3, 4$. We use LLL algorithm from above library. The complexity of the attack is dominated by the LLL algorithm. The complexity of the LLL algorithm is a polynomial time algorithm in size of the lattice and the exponential in size of the entries in the lattice. Most of the times, we retrieved the actual value, but some times we get the nearer value to the actual value. We have done some experiments for the size of the modulus 2048 also.

# 4 Conclusion

We showed that Lattice methods can recover the secret exponent in a certain kind of "Multi prime RSA" setting. Our attack assumes that many instances of RSA all use different multi-prime moduli, but somehow all use same secret exponent. In this scenario, we investigate about the smallness of the secret exponent. If it is less than the above bound, then one can break the system. We also observe that if the number of instances is increasing, then the breaking the system becomes easy. We use LLL algorithm to attack this system. LLL algorithm has so many applications in the fields like cryptology, Communications and Number theory.

# References

[1] K. Banarjee, S. N. Mandal, S. K. Das, "Improved trail division technique for primality checking in RSA algorithm," *International Journal of Computer Network and Information Security*, vol. 5, no. 9, July 2013.

[2] D. Boneh, "Twenty years of attacks on the RSA cryptosystem," *Notices of the American Mathematical Society*, vol. 46, no. 2, pp. 203-213, 1999.

[3] D. Boneh, G. Durfee, "Cryptanalysis of RSA with private key d less than N0:292," in *Advances in Cryptology (Eurocrypt'99)*, Lecture Notes in Computer Science 1952, pp. 1-11, 1999.

[4] D. Coppersmith, "Finding a small root of a bivaraite integer equation: Factoring with high Bits Known," in *Lecture Notes in Computer Science*, vol. 1070, pp. 178-189, Springer, 1996.

[5] M. Ernst, E. Jochemsz, A. May, B. de Weger, "Partial key exposure attacks on RSA up to full size exponents," in *Advanced in Cryptology (EUROCRYPT'05)*, pp. 1-11, 2000.

[6] H. Graham, "Finding small roots of univariate modular equations revisited," in *Lecture Notes in Computer Science*, vol. 1355, pp. 131-142, Springer, 1997.

[7] J. Hastad, "Solving simultaneous modular equations of low degree," *SIAM Journal of Computing*, vol. 17, no. 2, pp. 336-341, Apr. 1988.

[8] M. Hermann and A. May, "Solving linear equations modulo divisors:on factoring given any bits," in *Lecture Notes in Computer Science*, pp. 406-424, 2008.

[9] M. J. Hinek, "On the security of multi-prime RSA," *Journal of Mathematical Cryptology*, vol. 2, no. 2, pp. 117-147, July 2008.

[10] M. J. Hinek, *Small Private Exponent Partial Key-Exposure Attacks On Multi Prime RSA*, Centre for Applied Cryptographic Research, University of Waterloo, 2004.

[11] E. Jochemsz, A. May, "A strategy of finding roots of multivariate polynomials with new applications in attacking RSA variants," in *Lecture Notes in Computer Science*, pp. 267-282, 2006.

[12] R. S. Kumar, C. Narasimham, S. P. Setty, "Lattice based tools for cryptanalysis in various applications," in *International Conference on Computer Science and Information Technology*, pp. 530-537, 2012.

[13] R. S. Kumar, C. Narasimham, S. P. Settee, "Generalization of Boneh-Duree's attack on arbitrary public exponent RSA," *Interantional Journal of Computer applications*, vol. 49, no. 19, 2012.

[14] A. Lenstra, H. Lenstra, L. Lovasz, "Factoring polynomials with rational coeffiecients," *Mathematiche Annalen*, vol. 261, pp. 515-534, 1982.

[15] Y. Lu, R. Zhang, and D. Lin, "Factoring multi-power RSA modulus $N = p^r q$ with partial known bits," in *Lecture Notes in Computer Science*, vol. 7959, pp. 57-71, 2013.

[16] Y. Lu, R. Zhang, and D. Lin, "Factoring RSA modulus with known bits from both p and q:a lattice method," in *Lecture Notes in Computer Science*, vol. 7873, pp. 393-404, 2013.

[17] R. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[18] W. A. Stein, et al., *Sage Mathematical Software*, The Sage Development Team, 2011. (`http://ww.sagemath.org`)

# Biography

**Dr. Santosh Kumar Ravva** working as Sr.Asst.prof in the department of Information Technology in MVGR College of Engineering, Vizianagaram, India. He published more than 10 publications in various journals in the area of cryptology. His research interests are Cryptanalysis of RSA, Access control in Wireless sensor networks, Attribute based encryption schemes for cloud computing. He is member in IEI, IAENG, and life member in CRSI.