Bull. Korean Math. Soc. **46** (2009), No. 4, pp. 721–741
DOI 10.4134/BKMS.2009.46.4.721

PARTIAL KEY EXPOSURE ATTACKS ON RSA AND ITS VARIANT BY GUESSING A FEW BITS OF ONE OF THE PRIME FACTORS

Santanu Sarkar and Subhamoy Maitra

Abstract. Consider RSA with $N = pq$, $q < p < 2q$, public encryption exponent e and private decryption exponent d. We first study cryptanalysis of RSA when certain amount of the Most Significant Bits (MSBs) or Least Significant Bits (LSBs) of d is known. The basic lattice based technique is similar to that of Ernst et al. in Eurocrypt 2005. However, our idea of guessing a few MSBs of the secret prime p substantially reduces the requirement of MSBs or LSBs of d for the key exposure attack. Further, we consider the RSA variant proposed by Sun and Yang in PKC 2005 and show that the partial key exposure attack works significantly on this variant.

1. Introduction

RSA [20] is one of the most popular cryptosystems in the history of cryptology. Here, we use the standard notations in RSA as follows:

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

Though RSA is quite safe till date if applied with proper cryptographic practices, the literature related to its cryptanalysis is quite rich. RSA is found to be weak when the prime factors of any one of $p \pm 1$, $q \pm 1$ is small [19, 28]. In [11], it has been pointed out that short public exponents may cause weakness if the same message is broadcast to many parties. One very important result regarding RSA weak keys has been presented in [27], where it has been shown

Received November 14, 2008; Revised December 23, 2008.
2000 *Mathematics Subject Classification.* Primary 11Y05; Secondary 94A60.
Key words and phrases. cryptanalysis, factorization, lattice, LLL algorithm, RSA, side channel attacks, weak keys.
The preliminary version of this paper appeared in the proceedings of the International Conference on Information Security and Cryptology (ICISC) 2008.

that N can be factored from the knowledge of N, e if $d < \frac{1}{3}N^{\frac{1}{4}}$. Though it has been shown [21] that the idea of [27] cannot be substantially extended further than the bound of d as $O(N^{\frac{1}{4}})$, many papers [3, 9, 25, 26] used the idea of Continued Fraction (CF) expression to get different kinds of weak keys in RSA (one may follow the material from [22, Chapter 5] for basics of CF expression and related results). The seminal idea of [8] using lattice based techniques has also been exploited in great detail [5, 1] to find weak keys of RSA when $d < N^{0.292}$. An outstanding survey on the attacks on RSA before the year 2000 is available in [4]. For very recent results on RSA, one may refer to [13, 17] and the references therein.

One important model of cryptanalysis is the side channel attack such as fault attacks, timing attacks, power analysis etc. [6, 7, 14, 15], by which an adversary may obtain some bits of the private key d. In [6], it has been studied how many bits of d need to be known to mount an attack on RSA. The constraint in the work of [6] was the upper bound on e which is $\sqrt{N}$. The study attracted interest and the idea of [6] has been improved in [2] where the bound of e was increased upto $N^{0.725}$. Then the work of [10] improved the result for full size public exponent e. We present further improvement over the work of [10] noting that if one guesses a few MSBs of p, then the requirement on the number of bits in d gets substantially reduced.

As an example (see Example 1 later) with practical parameters, for a specific 1024 bit N and 309 bit d, the idea of [10] requires 112 many MSBs of d to be exposed, whereas, our idea requires only 80 MSBs of d with a guess of 21 bits of MSBs in p. First of all, the total requirement of bits to be known in our case is $80 + 21 = 101$, which is 11 bits less than the 112 many bits to be known in [10]. More importantly, one needs to know the bits of d by side channel attacks and a reduction of $112 - 80 = 32$ bits makes the chance of this kind of attack more realistic. Further, with higher lattice dimension we get even more interesting results where as less as 53 many MSBs of d are required with the knowledge of 21 many MSBs of p.

One may note that given the constraint $q < p < 2q$, a few bits of p, q can be known in polynomial time (e.g., around 7 bits for 1024 bit N and 9 bits for 2048 bit N following the work of [23]). This will indeed reduce the search effort further for guessing a few MSBs of p.

As we use different notations in this paper compared to [10], let us list the results of [10] with our notations here. Let d be of bitsize $\delta \log_2 N$. Given $(\delta - \gamma) \log_2 N$ many MSBs of d, the product N can be factored in probabilistic polynomial time [10] (we ignore the term ϵ as given in [10]) if

(1) $\gamma \leq \frac{5}{6} - \frac{1}{3}\sqrt{1 + 6\delta}$, or
(2) $\gamma \leq \frac{1}{3}\lambda + \frac{1}{2} - \frac{1}{3}\sqrt{4\lambda^2 + 6\lambda}$, where $\lambda = \max\{\gamma, \delta - \frac{1}{2}\}$.

There are also some results in [10], where cryptanalysis of RSA is studied when some LSBs of d are known.

In this paper we use similar kind of analysis as in [10] and explain different cases relevant to the attacks. The theoretical results are presented in Theorems 2.1, 2.3. The advantages of our work over [10] are as follows.

(1) Given that a few MSBs of p can be guessed, the requirement of MSBs of d in our attack is less than that of [10] (where no guess on p is made).
(2) The total amount of bits, to be known considering the MSBs of both p, d in our case, is less than the number of MSBs to be known for d as reported in [10].
(3) In Theorem 3.1, we have also studied the cryptanalysis of RSA when some MSBs of p along with the LSBs of d are known and our results are better than that of [10].

We also study the RSA variant proposed in [24], where e, d are more than $\sqrt{N}$, but significantly less than N. The data used in [24] considered e, d of the order of N^{τ}, where $\tau \approx 0.6$, say. As, $ed = 1 + k\phi(N)$, the value of k is of the order of $N^{2\tau-1}$, which is significantly smaller than N. Consider that the number of bits in k is k_b; then we find that one needs to know k_b many MSBs of d to cryptanalyze RSA. This result is presented in Theorem 5.1.

The lattice based technique used here is similar to what presented in [10]. However, in [10], a full size public exponent (i.e., e of the order of N) or a full size private exponent (i.e., d of the order of N) has been considered separately. In this paper, we take $e = N^{\alpha}$ and $d \leq N^{\delta}$ to get generalized results.

Our theoretical results are supported by experimental evidences. We have implemented the programs in SAGE 2.10.1 over Linux Ubuntu 7.04 on a computer with Dual CORE Intel(R) Pentium(R) D CPU 2.80 GHz, 1 GB RAM and 2 MB Cache.

While comparing our experimental results with that of [10], we implement the idea of [10] on our own platform. As all the parameters for the experiments in [10] may not be the same with our implementations, the results may vary a little. We point out the exact experimental values presented in [10] as and when required.

The organization of the paper is as follows. Next we present brief preliminaries. In Section 2, we study the key exposure attacks when the MSBs of d are exposed. Similar study continues in Section 3 where it is considered that some LSBs of d is known. Section 4 lists the experimental results corresponding to the theoretical results presented in Sections 2, 3. In Section 5, we discuss the effect of partial key exposure attack on the RSA variant presented in [24]. Section 6 concludes the paper.

1.1. Preliminaries

Let us present some basics on lattice reduction techniques. Consider the linearly independent vectors $u_1, \ldots, u_w \in \mathbb{Z}^n$, when $w \leq n$. A lattice, spanned by $\langle u_1, \ldots, u_w \rangle$, is the set of all linear combinations of $u_1, \ldots, u_w$, i.e., w is the dimension of the lattice. A lattice is called full rank when $w = n$. Let L be a

lattice spanned by linearly independent vectors $u_1, \ldots, u_w$, where $u_1, \ldots, u_w \in \mathbb{Z}^n$. By $u_1^*, \ldots, u_w^*$, we denote the vectors obtained by applying the Gram-Schmidt process to the vectors $u_1, \ldots, u_w$.

The determinant of L is defined as $\det(L) = \prod_{i=1}^{w} ||u_i^*||$, where $||\cdot||$ denotes the Euclidean norm on vectors. Given a polynomial $g(x,y) = \sum a_{i,j}x^i y^j$, we define the Euclidean norm as $\| g(x,y) \| = \sqrt{\sum_{i,j} a_{i,j}^2}$ and infinity norm as $\| g(x,y) \|_\infty = \max_{i,j} |a_{i,j}|$.

It is known that given a basis $u_1, \ldots, u_w$ of a lattice L, LLL algorithm [16] can find a new basis $b_1, \ldots, b_w$ of L with the following properties.

- $\| b_i^* \|^2 \leq 2 \| b_{i+1}^* \|^2$ for $1 \leq i < w$.
- For all i, if $b_i = b_i^* + \sum_{j=1}^{i-1} \mu_{i,j} b_j^*$ then $|\mu_{i,j}| \leq \frac{1}{2}$ for all j.
- $\| b_1 \| \leq 2^{\frac{w}{2}} \det(L)^{\frac{1}{w}}$, $\| b_2 \| \leq 2^{\frac{w}{2}} \det(L)^{\frac{1}{w-1}}$.

By $b_1^*, \ldots, b_w^*$, we mean the vectors obtained by applying the Gram-Schmidt process to the vectors $b_1, \ldots, b_w$.

In [8], techniques have been discussed to find small integer roots of polynomials in a single variable mod n, and of polynomials in two variables over the integers. The idea of [8] extends to more than two variables also, but the method becomes probabilistic. The following theorem is also relevant to the idea of [8].

Theorem 1.1 ([12]). *Let $g(x,y,z)$ be a polynomial which is a sum of ω many monomials. Suppose $g(x_0, y_0, z_0) \equiv 0 \bmod n$, where $|x_0| < X$, $|y_0| < Y$ and $|z_0| < Z$. If $\| g(xX, yY, zZ) \| < \frac{n}{\sqrt{\omega}}$, then $g(x_0, y_0, z_0) = 0$ holds over integers.*

Thus, the condition $2^{\frac{w}{2}} \det(L)^{\frac{1}{w-1}} < \frac{n}{\sqrt{\omega}}$ implies that if polynomials b_1, b_2 (corresponding to the two shortest reduced basis vectors) have roots over 0 mod n, then those roots hold over integers also. The solutions corresponding to each unknown is achieved by calculating the resultant of two polynomials (if they are algebraically independent) and then finding the solution of the resultant.

2. MSBs of d and p known

In this section we consider that certain amount of MSBs of both d, p will be available. We will study to methods following the ideas of [10].

2.1. MSBs of d and p known: Method I

Let us start with the following result.

Theorem 2.1. *Let $d \leq N^\delta, e = N^\alpha$ and consider that d_0, p_0 are exposed such that $|d - d_0| < N^\gamma$ and $|p - p_0| < N^\beta$. Then one can factor N (in probabilistic polynomial time) when*

$$\gamma \leq \frac{(3-\beta) - \sqrt{4\beta^2 + 12\beta\delta + 12\beta\alpha - 12\beta}}{3}.$$

Proof. Let $q_0 = \frac{N}{p_0}$. We have $ed - 1 = k\phi(N) = k(N - (p + q - 1))$. Now writing $d = d_0 + d_1$, the above equation can be written as $e(d_0 + d_1) - 1 = k(N - p_0 - q_0 - (p + q - p_0 - q_0 - 1))$. This can be rewritten as $ed_1 - (N - p_0 - q_0)k + k(p + q - p_0 - q_0 - 1) + ed_0 - 1 = 0$. Let us consider the corresponding polynomial $f_{MSB1} = ex - (N - p_0 - q_0)y + yz + R$, where $R = ed_0 - 1$, and d_1 is renamed as x, k is renamed as y and $p + q - p_0 - q_0 - 1$ is renamed as z. Hence, we have to find the solution $(x_0, y_0, z_0) = (d_1, k, p + q - p_0 - q_0 - 1)$ of the polynomial $f_{MSB1} = ex - (N - p_0 - q_0)y + yz + R$.

Let $X = N^{\gamma}, Y = N^{\alpha+\delta-1}, Z = N^{\beta}$, and one can check that they are the upper bounds of x_0, y_0, z_0. Note that k (renamed as y) is $\frac{ed-1}{\phi(N)}$. As $e = N^{\alpha}$, $d \leq N^{\delta}$ and $\phi(N)$ is order of N, ignoring the constant terms, we get the value of Y, which is the upper bound of y_0. Also, the bound Z should be $|p+q-p_0-q_0|$, which is actually less than $2N^{\beta}$ (however, we ignore the constant term in the proof as in [10]).

Now let us fix the lattice parameters m, t. Define

$$W = ||f_{MSB1}(xX, yY, zZ)||_{\infty} \quad \text{and} \quad n = (XY)^m Z^{m+t} W.$$

In order to work with a polynomial having the constant term 1, we define

$$f \equiv R^{-1} f_{MSB1}(x, y, z) \bmod n \equiv 1 + ax + by + cyz.$$

(During the experiments, as long as $\gcd(R, n) \neq 1$, we keep on increasing n by 1.) Then we use the shifts

$$\begin{aligned}
g_{ijk} &= x^i y^j z^k f(x, y, z) X^{m-i} Y^{m-j} Z^{m+t-k}, \\
&\quad \text{for } \ i = 0, \ldots, m;\ j = 0, \ldots, m - i;\ k = 0, \ldots, j; \\
h_{ijk} &= x^i y^j z^k f(x, y, z) X^{m-i} Y^{m-j} Z^{m+t-k}, \\
&\quad \text{for } \ i = 0, \ldots, m;\ j = 0, \ldots, m - i;\ k = j + 1, \ldots, j + t; \\
g'_{ijk} &= n x^i y^j z^k, \\
&\quad \text{for } \ i = 0, \ldots, m + 1;\ j = m + 1 - i;\ k = 0, \ldots, j; \\
h'_{ijk} &= n x^i y^j z^k, \\
&\quad \text{for } \ i = 0, \ldots, m + 1;\ j = m + 1 - i;\ k = j + 1, \ldots, j + t.
\end{aligned}$$

Now we build a lattice L with the basis elements coming from the coefficient vectors of $g_{ijk}(xX, yY, zZ)$, $h_{ijk}(xX, yY, zZ)$, $g'_{ijk}(xX, yY, zZ)$ and $h'_{ijk}(xX, yY, zZ)$ following the idea of [10]. The vectors are ordered in such a manner that the matrix corresponding to the lattice L becomes triangular, and the diagonal entries of g and h are equal to $(XY)^m Z^{m+t}$. Then we follow the similar computation as in [10, Appendix A], taking $t = \tau m$. If

$$X^{1+3\tau} Y^{2+3\tau} Z^{1+3\tau+3\tau^2} \leq W^{1+3\tau}, \tag{1}$$

we get polynomials f_1 and f_2 (the first two elements after lattice reduction using LLL algorithm) that satisfy the Howgrave-Graham bound as described in Theorem 1.1.

Now we construct two resultants G_1, G_2 taking two different pairs from f_{MSB1}, f_1, f_2 (in our experiments, mostly, G_1 is constructed using f_{MSB1}, f_1 and G_2 is constructed using f_{MSB1}, f_2). Then we construct the resultant of G_1, G_2 to get G. The integer root(s) of G provide z_0, which in turn gives the primes. We assume that the resultant computations for multivariate polynomials constructed in our approach yield non-zero polynomials. This is successful in most of the cases in our experiment. However, as this step involves some probability of success, we consider the algorithm as probabilistic. As each of lattice reduction, resultant computation and root finding is polynomial time algorithm in $\log_2 N$, the product N can be factored in probabilistic polynomial time given the constraints in this theorem.

Here $X = N^{\gamma}, Y = N^{\delta+\alpha-1}, Z = N^{\beta}$ and $W = \max\{eX, (N - p_0 - q_0)Y, YZ, R\} \geq (N - p_0 - q_0)Y \approx NY = N^{\alpha+\delta}$. So the Inequality (1) holds if,

$$\begin{aligned} X^{1+3\tau}Y^{2+3\tau}Z^{1+3\tau+3\tau^2} &\leq (NY)^{1+3\tau} \quad \Leftrightarrow \\ N^{\gamma(1+3\tau)}N^{(\delta+\alpha-1)(2+3\tau)}N^{\beta(1+3\tau+3\tau^2)} &\leq N^{(\alpha+\delta)(1+3\tau)} \quad \Leftrightarrow \\ 3\beta\tau^2 + (3\beta + 3\gamma - 3)\tau + (\alpha + \gamma + \delta + \beta - 2) &\leq 0. \end{aligned} \tag{2}$$

Putting the optimal value of τ, which is $\tau = \frac{1-\beta-\gamma}{2\beta}$ in Inequality (2) we get the required condition

$$\gamma \leq \frac{(3-\beta) - \sqrt{4\beta^2 + 12\beta\delta + 12\beta\alpha - 12\beta}}{3}.$$ □

When e is $O(N)$, we have $e = cN$ for some constant $0 < c < \frac{\phi(N)}{N}$ as $e < \phi(N)$. Thus, putting $\alpha = 1$ and ignoring the constant term, we get the following corollary.

Corollary 2.2. *Let $d \leq N^{\delta}$ and consider that d_0, p_0 are exposed such that $|d - d_0| < N^{\gamma}$ and $|p - p_0| < N^{\beta}$. Then one can factor N (in probabilistic polynomial time) when*

$$\gamma \leq 1 - \frac{\beta + 2\sqrt{\beta(\beta + 3\delta)}}{3}.$$

One may note that putting $\beta = \frac{1}{2}$ in Corollary 2.2, we get the same bound $\gamma \leq \frac{5}{6} - \frac{1}{3}\sqrt{1 + 6\delta}$ as in [10, Theorem 1]. As we have the knowledge of a few MSBs of p, the value of β decreases below $\frac{1}{2}$ in our case, increasing the value of γ. As $\delta - \gamma$ proportion of bits of d needs to be known for the attack, we require less number of MSBs of d to be exposed than [10]. We present some numerical values first. Consider 1024 bits N, e, and 359 bits d when $\delta = 0.35$. Thus, the upper bound of γ using the formula $\gamma \leq \frac{5}{6} - \frac{1}{3}\sqrt{1 + 6\delta}$ of [10] comes to be 0.24644. Then the requirement of MSBs of d to be known is $(0.35 - 0.24644) \times 1024 = 106$ bits. If we consider that 0.039 proportion of MSBs of p (i.e., 0.0195 proportion of $\log_2 N$) is known, then $\beta = 0.5 - 0.0195 = 0.4805$. In this case 20 many MSBs of p is required to be guessed. Using our

Theorem 2.1, the value of γ becomes 0.26813. Thus the requirement of MSBs of d to be known is $(0.35 - 0.26813) \cdot 1024 = 84$ bits.

One should note that the total requirement of bits to be known in our case is $84 + 20 = 104$, which is less than the requirement of 106 bits in [10]. The number of MSBs of d to be exposed in [10] is $(\delta - \gamma_1) \log_2 N$ (we denote γ by γ_1 here). In our case, the requirement of MSBs in p is $(0.5 - \beta) \log_2 N$ and that of d is $(\delta - \gamma_2) \log_2 N$ (we denote γ by γ_2 here), and adding them we get the total requirement of MSBs (considering both p, d) is $(0.5 - \beta + \delta - \gamma_2) \log_2 N$. One may check that $(\delta - \gamma_1) \log_2 N$ of [10] is greater than $(0.5 - \beta + \delta - \gamma_2) \log_2 N$ when $\beta < \frac{1}{2}$. This theoretically justifies the advantage of our technique.

As we will work with low lattice dimensions, the actual requirement of MSBs to be known will be higher in experimental results than the numerical values arrived from the theoretical results. This is explained in detail in Table 1.

Based on Corollary 2.2, we get the following probabilistic polynomial time algorithm.

Algorithm 1.

Inputs:

$N, e = N^{\alpha}$, and N^{δ}, the upper bound of d.
MSBs of d, p, i.e., d_0, p_0.
Parameters γ, β

Steps:

1. Construct polynomial $f_{MSB1} = ex - (N - p_0 - q_0)y + yz + R$ where
   $q_0 = \frac{N}{p_0}$, and $R = ed_0 - 1$.
2. Initialize $X = N^{\gamma}, Y = N^{\delta}, Z = N^{\beta}$.
3. Fix the lattice parameters m, t.
4. Calculate $W = ||f_{MSB1}(xX, yY, zZ)||_{\infty}$ and $n = (XY)^m Z^{m+t} W$.
5. Construct $f = R^{-1} f_{MSB1}(x, y, z) \bmod n = 1 + ax + by + cyz$.
6. Construct the lattice L from f, i.e., with the coefficients of the shift polynomials
   $g_{ijk}(xX, yY, zZ)$, $h_{ijk}(xX, yY, zZ)$, $g'_{ijk}(xX, yY, zZ)$ and $h'_{ijk}(xX, yY, zZ)$,
   where g, h, g', h' are constructed from f.
7. Reduce L using LLL algorithm to get the first two elements f_1, f_2.
8. Calculate the resultant G_1 using f_{MSB1}, f_1 and the resultant G_2 using f_{MSB1}, f_2.
9. If both G_1, G_2 are nonzero
   then calculate the resultant G of G_1, G_2;
   else
   exit with failure.
10. If G is nonzero and $\gamma \leq 1 - \frac{\beta + 2\sqrt{\beta(\beta + 3\delta)}}{3}$
    then solve G to get the integer root $z = (p + q - p_0 - q_0 - 1)$;
    else
    exit with failure.

One may also consider guessing MSBs of $p+q$ rather than p as the polynomial f_{MSB1} deals with $p + q$ rather than p and q. Experimental results of [23] show that around 12 many MSBs of $p + q$ can be estimated correctly for the 1024-bit N, whereas the estimation gives around 7 many MSBs for p. Consider that b_1 many MSBs of p are known (p is estimated by p') and we estimate q by $q' = \frac{N}{p'}$. Further, let us assume that the estimation $p' + q'$ has b_2 many MSBs identical

with the exact value $p+q$. Then experimentally we observed that $b_2 > b_1$ in general and for $b_1 = 7$, we get $b_2 = 12$ on an average. This shows that the effect of guessing the MSBs of p or $p+q$ are same.

2.2. MSBs of d and p known: Method II

We start this section with the following theorem.

Theorem 2.3. *Let $d \leq N^{\delta}$ and $e = N^{\alpha}$ and consider that d_0, p_0 are exposed such that $|d - d_0| < N^{\gamma}$ and $|p - p_0| < N^{\beta}$. Then one can factor N (in probabilistic polynomial time) when*

$$\gamma \leq 1 + \frac{1}{3}\lambda - \beta - \frac{2}{3}\sqrt{\lambda}\sqrt{\lambda + 3\beta},$$

where $\lambda = \max\{\gamma + \alpha - 1, \delta + \alpha - \frac{3}{2}\}$.

Proof. Note that the attacker can compute $k_0 = \frac{ed_0 - 1}{N}$. Let $k_1 = k - k_0$, the unknown part of k. It can be shown similar to [2] that $|k_1| < \frac{e}{\phi(N)}(N^{\gamma} + 3N^{\delta - \frac{1}{2}})$. So we get $|k_1| < 4N^{\lambda}$, where $\lambda = \max\{\gamma + \alpha - 1, \delta + \alpha - \frac{3}{2}\}$.

Now, $ed-1 = k(N+1-p-q) \Leftrightarrow e(d_0+d_1)-1 = (k_0+k_1)(N-(p+q-1)) \Leftrightarrow e(d_0+d_1)-1 = (k_0+k_1)(N-p_0-q_0-(p+q-p_0q_0-1)) \Leftrightarrow ed_1-(N-p_0-q_0)k_1+k_1(p+q-p_0-q_0-1)+k_0(p+q-p_0-q_0-1)+ed_0-1-(N-p_0-q_0)k_0 = 0$. Hence we have to find the solution of the polynomial

$$f_{MSB2}(x, y, z) = ex - (N - p_0 - q_0)y + yz + k_0 z + R,$$

where $R = ed_0 - 1 - (N - p_0 - q_0)k_0$. That is, the root of $f_{MSB2}(x, y, z)$ is $(x_0, y_0, z_0) = (d_1, k_1, p + q - p_0 - q_0 - 1)$.

Let $X = N^{\gamma}, Y = N^{\lambda}, Z = N^{\beta}$, and one can check that they are the upper bounds of x_0, y_0, z_0 neglecting the small constant multipliers.

Now let us fix the lattice parameters m, t. Define

$$W = ||f_{MSB2}(xX, yY, zZ)||_{\infty} \quad \text{and} \quad n = X^m Y^{m+t} Z^m W.$$

In order to work with a polynomial with constant term 1, we define

$$f \equiv R^{-1} f_{MSB2}(x, y, z) \bmod n \equiv 1 + ax + by + cyz + dz.$$

(During the experiments, as long as $\gcd(R, n) \neq 1$, we keep on increasing n by 1.) Then we use the shifts

$$\begin{aligned}
g_{ijk} &= x^i y^j z^k f(x, y, z) X^{m-i} Y^{m+t-j} Z^{m-k} \\
&\quad \text{for } \ i = 0, \ldots, m;\ j = 0, \ldots, m - i;\ k = 0, \ldots, m - i; \\
h_{ijk} &= x^i y^j z^k f(x, y, z) X^{m-i} Y^{m+t-j} Z^{m-k} \\
&\quad \text{for } \ i = 0, \ldots, m;\ j = m - i + 1, \ldots, m - i + t;\ k = 0, \ldots, m - i; \\
g'_{ijk} &= n x^i y^j z^k \\
&\quad \text{for } \ i = 0, \ldots, m + 1;\ j = 0, \ldots, m + t + 1 - i;\ k = m + 1 - i; \\
h'_{ijk} &= n x^i y^j z^k \\
&\quad \text{for } \ i = 0, \ldots, m;\ j = m + t + 1 - i;\ k = 0, \ldots, m - i.
\end{aligned}$$

Now we build a lattice L with the basis elements coming from the coefficient vectors of $g_{ijk}(xX, yY, zZ)$, $h_{ijk}(xX, yY, zZ)$, $g'_{ijk}(xX, yY, zZ)$ and $h'_{ijk}(xX, yY, zZ)$ following the idea of [10]. The vectors are ordered in such a manner that the matrix corresponding to the lattice L becomes triangular, and the diagonal entries of g and h are equal to $X^m Y^{m+t} Z^m$. We give an example of the lattice below with $m = 1$ and $t = 0$.

1	z	y	yz	x	z^2	yz^2	y^2z^2	xz	xyz	x^2	y^2	y^2z	xy
XYZ	$dXYZ^2$	bXY^2Z	cXY^2Z^2	aX^2YZ	0	0	0	0	0	0	0	0	0
0	XYZ	0	bXY^2Z	0	$dXYZ^2$	cXY^2Z^2	0	aX^2YZ	0	0	0	0	0
0	0	XYZ	$dXYZ^2$	0	0	0	0	0	0	0	bXY^2Z	cXY^2Z^2	aX^2YZ
0	0	0	XYZ	0	0	$dXYZ^2$	cXY^2Z^2	0	aX^2YZ	0	0	bXY^2Z	0
0	0	0	0	XYZ	0	0	0	$dXYZ^2$	cXY^2Z^2	aX^2YZ	0	0	bXY^2Z
0	0	0	0	0	nZ^2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	nYZ^2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	nY^2Z^2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	nXZ	0	0	0	0	0
0	0	0	0	0	0	0	0	0	$nXYZ$	0	0	0	0
0	0	0	0	0	0	0	0	0	0	nX^2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	nY^2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	nY^2Z	0
0	0	0	0	0	0	0	0	0	0	0	0	0	nXY

Now we follow the similar computation as in [10, Appendix B], taking $t = \tau m$. If

$$X^{2+3\tau} Y^{3+6\tau+3\tau^2} Z^{3+3\tau} \leq W^{2+3\tau}, \tag{3}$$

we get polynomials f_1 and f_2 (the first two elements after lattice reduction using LLL algorithm) that satisfy the Howgrave-Graham bound as described in Theorem 1.1.

Similar to the proof of Theorem 2.1, we construct two resultants G_1, G_2 taking two different pairs from f_{MSB2}, f_1, f_2. Then we construct the resultant of G_1, G_2 to get G. The integer root(s) of G provide z_0, which in turn gives the prime. The complete operation works in probabilistic poly($\log_2 N$) time.

Here $X = N^\gamma$, $Y = N^\lambda$, $Z = N^\beta$ and

$$W = \max\{eX, (N - p_0 - q_0)Y, YZ, k_0 Z, R\} \geq (N - p_0 - q_0)Y \approx NY = N^{1+\lambda}.$$

So the Inequality (3) holds if,

$$\begin{aligned} X^{2+3\tau} Y^{3+6\tau+3\tau^2} Z^{3+3\tau} &\leq (NY)^{2+3\tau} \quad \Leftrightarrow \\ N^{\gamma(2+3\tau)} N^{\lambda(3+6\tau+3\tau^2)} N^{\beta(3+3\tau)} &\leq N^{(1+\lambda)(2+3\tau)} \quad \Leftrightarrow \\ 3\lambda\tau^2 + (3\beta + 3\gamma + 3\lambda - 3)\tau + (2\gamma + \lambda + 3\beta - 2) &\leq 0. \end{aligned} \tag{4}$$

Putting the optimal value of τ, which is $\tau = \frac{1-\beta-\gamma-\lambda}{2\lambda}$, in Inequality (4) we get the required condition $\gamma \leq \frac{(6+2\lambda-6\beta)-\sqrt{16\lambda^2+48\lambda\beta}}{6}$. □

When e is $O(N)$, we have $e = cN$ for some constant $0 < c < \frac{\phi(N)}{N}$ as $e < \phi(N)$. Thus, putting $\alpha = 1$ and ignoring the constant term, we get the following corollary.

Corollary 2.4. *Let $d \leq N^{\delta}$ and consider that d_0, p_0 are exposed such that $|d - d_0| < N^{\gamma}$ and $|p - p_0| < N^{\beta}$. Then one can factor N (in probabilistic polynomial time) when*

$$\gamma \leq 1 + \frac{1}{3}\lambda - \beta - \frac{2}{3}\sqrt{\lambda}\sqrt{\lambda + 3\beta},$$

where $\lambda = \max\{\gamma, \delta - \frac{1}{2}\}$.

Putting $\beta = \frac{1}{2}$ in Corollary 2.4, we get the same bound as in [10, Theorem 1]. As we have knowledge of a few MSBs of p, the value of β decreases below $\frac{1}{2}$ in our case, increasing the value of γ. As $\delta - \gamma$ proportion of bits of d needs to be known for the attack, we require less number of MSBs of d to be exposed than [10].

Similar to Algorithm 1 corresponding to Corollary 2.2, one can devise a probabilistic polynomial time algorithm following Corollary 2.4.

2.3. Comparison of Methods I and II

In Theorem 2.1, we have

$$\gamma \leq \frac{(3 - \beta) - \sqrt{4\beta^2 + 12\beta\delta + 12\beta\alpha - 12\beta}}{3},$$

and in Theorem 2.3, we get

$$\gamma \leq 1 + \frac{1}{3}\lambda - \beta - \frac{2}{3}\sqrt{\lambda}\sqrt{\lambda + 3\beta},$$

where $\lambda = \max\{\gamma + \alpha - 1, \delta + \alpha - \frac{3}{2}\}$. Now $\lambda = \gamma + \alpha - 1$ implies that $\gamma \leq \frac{\frac{4}{3}\alpha + \beta^2 - 2\alpha\beta - \frac{1}{3}\alpha^2}{\frac{4}{3}\alpha}$ is valid for $\delta \leq \frac{1}{2} + \frac{\frac{4}{3}\alpha + \beta^2 - 2\alpha\beta - \frac{1}{3}\alpha^2}{\frac{4}{3}\alpha}$. If $\lambda = \delta + \alpha - \frac{3}{2}$, we get that

$$\gamma \leq \frac{1}{3}\alpha + \frac{1}{2} - \beta + \frac{\delta}{3} - \frac{2}{3}\sqrt{\alpha^2 + 2\alpha\delta + \delta^2 + 3\alpha\beta + 3\delta\beta - 3\alpha - 3\delta - \frac{9\beta}{2} + \frac{9}{4}}$$

is valid for $\delta \geq \frac{1}{2} + \frac{\frac{4}{3}\alpha + \beta^2 - 2\alpha\beta - \frac{1}{3}\alpha^2}{\frac{4}{3}\alpha}$. We need $(\delta - \gamma)\log_2 N$ many MSBs of d to factor N and thus when the upper bound of γ is larger, one gets the better result.

Now,

$$\frac{\frac{4}{3}\alpha + \beta^2 - 2\alpha\beta - \frac{1}{3}\alpha^2}{\frac{4}{3}\alpha} \leq \frac{(3 - \beta) - \sqrt{4\beta^2 + 12\beta\delta + 12\beta\alpha - 12\beta}}{3}$$

if and only if

$$\delta \leq \frac{1}{12\beta}(\alpha^4 - 12\alpha^3\beta + \frac{26}{3}\alpha^2\beta^2 - 28\alpha\beta^3 + 9\beta^4 + \frac{64}{3}\alpha^2\beta)\frac{16}{9}\alpha^2.$$

Since

$$\frac{1}{12\beta}(\alpha^4 - 12\alpha^3\beta + \frac{26}{3}\alpha^2\beta^2 - 28\alpha\beta^3 + 9\beta^4 + \frac{64}{3}\alpha^2\beta)\frac{16}{9}\alpha^2$$
$$\leq \frac{1}{2} + \frac{\frac{4}{3}\alpha + \beta^2 - 2\alpha\beta - \frac{1}{3}\alpha^2}{\frac{4}{3}\alpha}$$

for our α, β. Hence, we can conclude that Method I is more effective when

$$\delta \leq \frac{1}{12\beta}(\alpha^4 - 12\alpha^3\beta + \frac{26}{3}\alpha^2\beta^2 - 28\alpha\beta^3 + 9\beta^4 + \frac{64}{3}\alpha^2\beta)\frac{16}{9}\alpha^2,$$

but for higher values of δ, Method II will perform better.

3. LSBs of d and MSBs of p known

In [10, Theorem 3], the cryptanalysis of RSA has been studied when some LSBs of d are exposed. We here extend the idea with the additional idea that a few MSBs of p are also known. This gives the following theorem. We present the proof briefly as the technique is similar to Theorem 2.1.

Theorem 3.1. *Let $d < N^\delta$ and $e = N^\alpha$. Given $(\delta - \gamma)\log_2 N$ many LSBs of d and p_0 when $|p - p_0| < N^\beta$, N can be factored in probabilistic polynomial time when*

$$\gamma \leq \frac{(3-\beta) - \sqrt{4\beta^2 + 12\beta\delta + 12\beta\alpha - 12\beta}}{3}.$$

Proof. Consider that d_0 is the integer corresponding to the exposed LSBs of d. Thus, $d_0 \equiv d \bmod M$ for some M, i.e., $d = d_0 + d_1 M$ for some d_1. Now we have $ed-1 = k(N-(p+q-1))$, which can be written as $e(d_0+d_1M)-1 = k(N-p_0-q_0-(p+q-p_0-q_0-1)) \Leftrightarrow eMd_1-(N-p_0-q_0)k+k(p+q-p_0-q_0)+ed_0-1 = 0$. Hence we have to find the solution of the polynomial

$$f_{LSB}(x, y, z) = eMx - (N - p_0 - q_0)y + yz + R,$$

where $R = ed_0 - 1$. So, the root of $f_{LSB}(x, y, z)$ is $(x_0, y_0, z_0) = (d_1, k, p + q - p_0 - q_0 - 1)$. This polynomial is same as the polynomial f_{MSB1} in the proof of Theorem 2.1. Thus, using similar analysis as in the proof of Theorem 2.1, we get the constraint as

$$X^{1+3\tau}Y^{2+3\tau}Z^{1+3\tau+3\tau^2} \leq W^{1+3\tau}.$$

Putting $X = N^\gamma, Y = N^{\alpha+\delta-1}, Z = N^\beta$ we get $\gamma \leq \frac{(3-\beta)-\sqrt{4\beta^2+12\beta\delta+12\beta\alpha-12\beta}}{3}$. □

When e is $O(N)$, we have $e = cN$ for some constant $0 < c < \frac{\phi(N)}{N}$ as $e < \phi(N)$. Thus, putting $\alpha = 1$ and ignoring the constant term, we get the following corollary.

Corollary 3.2. *Let $d \leq N^{\delta}$ and consider that d_0, p_0 are exposed such that $|d - d_0| < N^{\gamma}$ and $|p - p_0| < N^{\beta}$. Then one can factor N (in probabilistic polynomial time) when*

$$\gamma \leq 1 - \frac{\beta + 2\sqrt{\beta(\beta + 3\delta)}}{3}.$$

Putting $\beta = \frac{1}{2}$ in Corollary 3.2, we get the same bound as in [10, Theorem 3]. As we have the knowledge of a few MSBs of p, the value of β decreases below $\frac{1}{2}$ in our case, increasing the value of γ. As $\delta - \gamma$ proportion of bits of d needs to be known for the attack, we require less number of LSBs of d to be exposed than [10].

Similar to Algorithm 1 corresponding to Corollary 2.2, one can devise a probabilistic polynomial time algorithm following Corollary 3.2.

In the next section we present experimental results with all the relevant data that highlights the improvement achieved by our technique.

4. Experimental results

As we will work with low lattice dimensions, the actual requirement of MSBs to be known will be higher in experimental results than the numerical values arrived from the theoretical results. In all the examples in this section, we consider e is $O(N)$, i.e., $\alpha = 1$.

Let us first present an example corresponding to Corollary 2.2.

Example 1. We consider 1024 bits N, where p, q are as follows:
125076192352751041131507009460095319151891488205387463013857272133794535733443372033786891784694556227753494467523090187993837113578541321880095737053207991107912156937047618049134072984642192716736685911164684230293246833316600383916744711068174787341479864802074044896764353805764428925176190701388649979938 3.

The public encryption exponent e and the private decryption exponent d ($> N^{0.3}$) are as follows:
411141953148270330221315221524982019936529761031745298555857276797330634647691153459856956000333796180934856263680695803317014371713991035411585833035097935179306334968838354246222965614977094438717597912073932796183294924469326214709544940416156185452374908280364653971826687426168385755769098614735090957011, and
911260046070098225464230311775052873569746472764337803805303583934395253129269343722635765941.

First we work with the case $m = t = 1$, i.e., getting a lattice with dimension $w = 16$ which corresponds to a 16×16 matrix (one may refer to [10, Section 4.1.1, Page 378] for the exact matrix). Factoring N requires the knowledge of 112 many MSBs of d using the method of [10], whereas, our technique

requires 80 many MSBs of d and 21 many MSBs of p. Both the techniques require around 1.5 seconds on our platform. Following the idea of [23], around 7 MSBs of p may be known in polynomial time and hence we need 2^{21-7} many guesses for p, which requires less than 7 hours in our experimental set-up. The existing works on partial key exposure attacks will not work with the knowledge of only 80 bits of MSBs that we achieve here.

Considering a higher lattice dimension, $m = t = 2$, i.e., $w = 40$, factoring N requires knowledge of 110 many MSBs of d using the idea of [10]. This requires 53.03 seconds. According to experimental results in [10, Figure 5], this should require around 93 MSBs of d. In our case, we require only 53 MSBs of d and 21 MSBs of p to factor N that requires 46.25 seconds; thus the total requirement is $53 + 21 = 74$ many bits. Considering that 7 many MSBs of p may be known using the idea of [23], the overall attack will take a day in a cluster of 9 machines.

In Table 1, we consider different 1024 bits N and present the results of 10 runs of Algorithm 1 for two cases, one when $d > N^{0.3}$ (308-bit d) and the other when $d > N^{0.35}$ (359-bit d). Let MSB_d, MSB_p be the number of MSBs exposed in d, p respectively and b_d, b_N be the number of bits in d, N respectively. For the experiments, we have taken $X = 2^{b_d - MSB_d - \tau} + 3$, $Y = 2^{b_d - \tau} + 3$ and $Z = 2^{\frac{b_N}{2} - 1 - MSB_p - \tau} + 3$, where τ is assigned to either 0 or 1.

TABLE 1. Our results for 1024 bits N with lattice dimension $m = 1, t = 1$, i.e., $w = 16$.

308-bit d and # MSBs of d revealed in our case is 80 bits										
# MSBs of d [10]	112	112	107	111	122	114	115	114	113	113
# MSBs of p (our)	21	22	26	27	33	20	23	27	24	17

359-bit d and # MSBs of d revealed in our case is 150 bits										
# MSBs of d [10]	213	213	224	221	210	213	213	209	214	209
# MSBs of p (our)	55	58	64	63	56	58	58	60	57	64

First, we consider that only 80 MSBs of d will be leaked and studied the requirement of the MSBs of p for the attack. In each case, the algorithm of [10] has also been executed and the requirement of the minimum number of MSBs for d is presented. Next, we consider that 150 MSBs of d will be exposed for our attack. The results of Table 1 clearly identifies the improvement through our approach over the idea of [10].

Now we present an example corresponding to Corollary 2.4.

Example 2. We consider 1024 bits N, where p, q are as follows:
12900954999005375207385920181416356418902368468039150115133837 67
02098744712580162829362111710263879758520746505779736380616669 75
87560825229347694650364315 3 and

100018509329865935646436400634421440180345180969932799051114353462459764015419519476055271010012194150583838878020173194022682316782601191836891187015992 91.

The public encryption exponent e and the private decryption exponent d ($> N^{0.635}$) are as follows:
264642794496370529083200104026432106451833064401427278190117669212757479951849910627005043663570362373485826106594523765744413906848604272574339602928280657237457953663021451655943042945578450102419616363485965292375381930771310725466811883801452448440797553199552275119277450247772914173533837855915317872037161023303467486069671927956706449459095092348532240745792204228848640890584976007853666974474085220376561849594212667546760685105870728672799323285469369900587950978784699041414105582850665589707.

First we work with the case $m = t = 1$, i.e., lattice dimension $w = 20$. Factoring N requires the knowledge of 572 many MSBs of d using the method of [10], whereas, our technique requires 517 many MSBs of d and 31 many MSBs of p. Both the techniques require around 7.5 seconds on our platform. Following the idea of [23], around 7 MSBs of p may be known in polynomial time and hence we need 2^{31-7} many guesses for p, which requires around a day in a cluster of 2^{10} machines. The existing works on partial key exposure attacks will not work with the knowledge of only 517 bits of MSBs that we achieve here. Further the total requirement of unknown bits in our case is $517 + 31 = 548$ which is less than 572.

With higher lattice dimension, $m = t = 2$, i.e., $w = 50$, factoring N requires 527 many MSBs of d using the idea of [10]. This takes 859.64 seconds. In our case, it is enough to know 494 MSBs of d with 31 MSBs of p. The time required is 887.22 seconds.

We now present the experimental details of 10 runs with 10 different 1024 bits N in Table 2. We consider that only 517 many MSBs of d will be leaked and then study the requirement of the MSBs of p for our attack. In each case, the algorithm of [10] has also been executed and the requirement of the minimum number of MSBs for d is presented. The results of Table 2 clearly identifies the improvement through our approach over the idea of [10].

TABLE 2. Our results for 1024 bits N with lattice dimension $m = 1, t = 1$, i.e., $w = 20$.

651-bit d and # MSBs of d revealed in our case is 517 bits										
# MSBs of d [10]	572	573	572	573	573	571	570	569	578	575
# MSBs of p (our)	31	34	35	35	35	32	38	33	33	35

Now we present an example corresponding to the Corollary 3.2.

Example 3. We consider 1024 bits N, where p, q are as follows:
120345555452049651309296431229078115451502115011463732197427366040366045510514324016989233753142232193527761166689925629539776014948123702173905117450646 09 and
117016223242807604327596324209239490299204404169992276518274549116877945870694719394591078917009532387658528255891957655231772210613634373575810563853451 93.

The public encryption exponent e and the private decryption exponent d ($> N^{0.30}$) are as follows:
926284084883281809972592323129091068228437747986105793515923839221529080071271482166645655318455503177949951672784415983929081494300715331067535008047871523708599866902351068839273181735190226333386409790895575209623822107359490619936495064143986099800469310297155386364637607527939582944789365867808994343691 and
500972702758950805167354427743616028216073987403943201936640167969825484681181534595620036481.

First we work with the case $m = t = 1$, i.e., lattice dimension $w = 16$. Factoring N requires the knowledge of 115 many LSBs of d using the method of [10], whereas, our technique requires 80 many LSBs of d and 23 many MSBs of p. Both the techniques requires little less than 1.5 seconds on our platform. Following the idea of [23], around 7 MSBs of p may be known in polynomial time and hence we need 2^{23-7} many guesses for p, which requires a day in our experimental set-up.

When we work with higher lattice dimension $m = t = 2$, i.e., $w = 40$, factoring N requires 112 LSBs of d using the idea of [10]. It takes 46.39 seconds. In our case, we need 48 LSBs of d with 25 MSBs of p (requires 38.21 seconds) or 62 LSBs of d with 23 MSBs of p (requires 39.41 seconds).

We now present the experimental details of 10 runs in Table 3 considering 10 different 1024 bits N. We consider that only 80 many LSBs of d will be leaked and then study the requirement of the MSBs of p for the attack. In each case, the algorithm of [10] has also been executed and the requirement of the minimum number of LSBs for [10] is presented. The results of Table 3 clearly identifies the improvement through our approach over the idea of [10].

TABLE 3. Our results for 1024 bits N with lattice dimension $m = 1, t = 1$, i.e., $w = 16$.

308-bit d and # LSBs of d revealed in our case is 80 bits										
# LSBs of d [10]	115	107	105	108	109	109	114	116	112	108
# MSBs of p (our)	23	24	23	29	24	27	30	27	20	19

5. Key exposure attacks on RSA variant proposed in [24]

In [24], Sun and Yang proposed a variant of RSA where the public encryption exponent e and the private decryption exponent d are such that $\log_2 e + \log_2 d \approx \log_2 N + l_k$, where l_k is a positive integer. The main idea was to keep the bit size d as well as e quite less and the value of l_k is related to the security of this variant of RSA. The examples in [24] used $l_k = 112$. Below we present the result that gives a view of key exposure attack in such a scenario.

Theorem 5.1. *Let $d = N^\delta$ and $e = N^\alpha$. Consider that d_0, p_0 are exposed such that $|d - d_0| < N^\gamma$ and $|p - p_0| < N^\beta$. Further, $e > p + q$. Let $\gamma + \alpha - 1 \approx 0$ and $\delta + \alpha - \frac{3}{2} \approx 0$ and $\lambda = \max\{\gamma + \alpha - 1, \delta + \alpha - \frac{3}{2}\}$. If N^λ is bounded by some value V, then one can factor N in $O(V)$ time.*

Proof. First compute $k_0 = \frac{ed_0 - 1}{N}$. Let $k_1 = k - k_0$, the unknown part of k. It can be shown similar to [2] that $|k_1| < \frac{e}{\phi(N)}(N^\gamma + 3N^{\delta - \frac{1}{2}})$. So we get $|k_1| < 4N^\lambda$, where $\lambda = \max\{\gamma + \alpha - 1, \delta + \alpha - \frac{3}{2}\}$. Given the conditions in the statement of this theorem, $\lambda \approx 0$ and N^λ is bounded by V. Thus k_1 becomes small and one can get k by attempting $O(V)$ many guesses around k_0 as $k \approx k_0$.

Since, $ed = 1 + k(N + 1 - p - q)$, we get $p + q \equiv N + 1 + k^{-1} (\mathrm{mod} e)$. Since $e > p + q$, we get the exact value of $p + q$ from the above relation, when k is known. From $p + q$ one can factor N easily. □

For the examples presented in [24], we note that $\delta + \alpha - \frac{3}{2} < 0$. In such a case, we get the following corollary from the above theorem.

Corollary 5.2. *Let $\delta + \alpha - \frac{3}{2} \leq 0$ and k_b be the number of bits in k. If k_b many MSBs of d are exposed, then one can factor N in $O(V)$ time.*

Proof. Here k_b is the number of bits in k. Also assume $\delta + \alpha - \frac{3}{2} \leq 0$. From $ed = 1 + k(N + 1 - p - q)$, putting $e = N^\alpha$ and $d = N^\delta$, we get $k \approx N^{\alpha + \delta - 1}$. So we can write $k_b \approx (\alpha + \delta - 1) \log_2 N$. Consider that k_b many MSBs of d are exposed. Then one can find an integer d_0 such that k_b many MSBs of d and d_0 are same; rest of the bits of d_0 are set to 0. Then $|d - d_0| \approx N^{\delta - (\alpha + \delta - 1)} = N^{1 - \alpha}$. Thus, in this case, $\gamma \approx 1 - \alpha$. Given that

- $\delta + \alpha - \frac{3}{2} < 0$ and
- $\gamma + \alpha - 1 \approx 0$,

the condition that $\lambda \approx 0$ is satisfied and k may be obtained correctly in $O(V)$ time if N^λ is bounded by V. Thus k can be found from exposure of k_b many MSBs of d. □

In the examples of [24], we have $\delta + \alpha - \frac{3}{2} < 0$ and the number of bits of k is around 112. If 112 many MSBs of d are known in the examples of [24] then $\gamma + \alpha - 1 \approx 0$ and for these examples, N^λ is bounded by a small value. Hence, by Theorem 5.1, N can be factored with a few attempts, when 112 MSBs of d are exposed.

Let us take one example from [24] and show how the idea of Theorem 5.1 can be exploited in partial key exposure attack.

Example 4. Consider 1023 bit N, where p, q are as follows:
694629802315215123411992148068083243649754990479910717574507319329536676037202975544946028000824106343814965452795507729312204569387283256891949436717968 3 and
929801037865377498615289802383557463563813895298809697252646730106780222869581005796206684243065666502657988401133624674582335169496301411652140636711219 1.

The 624 bit public encryption exponent e and the 512 bits private decryption exponent d are as follows:
572106130179471018239068381502984244146036272490787956096327698669756108624382389748196147956425590174084491462173511252659291252631042065144975092250576973180336864547315548211840430871417602283866781159820947913228610686097130386000077350755390256771895186851063443804523752325090168789772551330531099955626119981664756390141139678043720501.

One can check that k is as follows:
6734064074495379225554574165269558.

To the attacker, only e, N and 112 many MSBs of d are known. In this case, d_0 is as follows:
760228386678115982094791322861068471023517944549543613708756406712792088547689088216849188663046156484101181774133083112979691089683519188685400058521190 4.

We get $k_0 = \frac{(ed_0-1)}{N}$ as follows:
6734064074495379225554574165269556.

For each integer k', relatively prime to e around k_0, we calculate k'^{-1} mod (e). Then we test whether $p+q \equiv N+1+k'^{-1} (\mathrm{mod}\, e)$ gives the factorization of N or not. In this manner we get the correct k. Then the value of k^{-1} mod e is as follows:
8024132632238591697576274438536382308684483967207580783523882571596593407610400954547833395862172890057368335078985151244628444583674593058099194589485024742085444514803006551931927028037, which in turn gives $p+q$.

Example 4 demonstrates that 112 many MSBs of d are required to mount the key exposure attack. Now we study the performance of the idea presented in Section 2. For experimentation we use low lattice dimensions, and hence the numerical values arrived from the theoretical results may not be reached.

First we concentrate on Method I (based on the idea of Theorem 2.1).

- If we consider that 20 many MSBs of p are known then $\beta = 0.48093$. Here we have $\gamma = 0.42299$ and we need to know $(\delta - \gamma) \log_2 N$ many

MSBs of d which is 79 in this case. Thus the total requirement of bits to be known is $20 + 79 = 99$.
- We perform an experiment corresponding to Method I, with parameters $m = 2, t = 2$, i.e., lattice dimension $\omega = 40$. In this case we need 95 many MSBs of d and 20 many MSBs of p and the time required is 22 seconds. Thus the total requirement of bits to be known is $20 + 95 = 115$.

Our results in Section 2 points out that lower values of e makes this RSA variant [24] more vulnerable. With the same primes presented in Example 4, we take the value of d two more than what presented in Example 4 and the corresponding e becomes a 1023 bit number, which is $O(N)$. With this new setup, the results using Method I are as follows.

- If we consider that 20 many MSBs of p are known then $\beta = 0.48093$. Here we have $\gamma = 0.18953$ and we need to know $(\delta - \gamma)\log_2 N$ many MSBs of d which is 318 in this case. Thus the total requirement of bits to be known is $20 + 318 = 338$.
- We perform an experiment corresponding to Method I, with parameters $m = 2, t = 2$, i.e., lattice dimension $\omega = 40$. In this case we need 388 many MSBs of d and 20 many MSBs of p and the time required is 37.5 seconds. Thus the total requirement of bits to be known is $20 + 388 = 408$.

Now we present examples with Method II (based on the idea of Theorem 2.3). First we consider the p, q, e, d as in Example 4.

- If we consider that 20 many MSBs of p are known then $\beta = 0.48093$. Here we have $\gamma = 0.41050$ and we need to know $(\delta - \gamma)\log_2 N$ many MSBs of d which is 92 in this case. Thus the total requirement of bits to be known is $20 + 92 = 112$.
- We perform an experiment corresponding to Method II, with parameters $m = 2, t = 2$, i.e., lattice dimension $\omega = 50$. In this case we need 99 many MSBs of d and 20 many MSBs of p and the time required is 553.8 seconds. Thus the total requirement of bits to be known is $20 + 99 = 119$.
- It is clear that Method I performs better than Method II when we consider the RSA variant of [24].

With the same primes presented in Example 4, we take the value of d two more than what presented in Example 4. With this new setup, the results using Method II are as follows.

- If we consider that 20 many MSBs of p are known then $\beta = 0.48093$. Here we have $\gamma = 0.20207$ and we need to know $(\delta - \gamma)\log_2 N$ many MSBs of d which is 305 in this case. Thus the total requirement of bits to be known is $20 + 305 = 325$.

- We perform an experiment corresponding to Method II, with parameters $m = 2, t = 2$, i.e., lattice dimension $\omega = 50$. In this case we need 363 many MSBs of d and 20 many MSBs of p and the time required is 1217.9 seconds. Thus the total requirement of bits to be known is $20 + 363 = 383$.
- One may note that Method II performs better than Method I when we consider RSA with e of the order of N.

Our study shows that much higher key exposure is required when e is $O(N)$ than in case of the RSA variant [24], when e is smaller.

6. Conclusion

In this paper we have studied cryptanalysis of RSA when either certain amount of MSBs or certain amount of LSBs of d are exposed. Our additional idea is to guess a few MSBs of the secret prime p. With this additional information, we find that our technique is more efficient than that of [10] (where no guess on the bits of p is attempted) in terms of the amount of bits of d to be exposed. Our technique is also better if one considers total number of bits to be known from d, p together than that of d only in [10]. Our theoretical results are implemented and we present experimental evidences of 1024 bits N, that can be factored with the exposure of considerably less amount of bits in d than [10] with a guess of a few MSBs in p that can be searched exhaustively (say around 20 to 30 bits). We also study an RSA variant proposed in [24] and analyze the effect of partial key exposure attack on this scheme.

Acknowledgments. The authors like to thank the anonymous reviewer for detailed comments that 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.

References

[1] J. Blömer and A. May, *Low secret exponent RSA revisited*, Cryptography and lattices (Providence, RI, 2001), 4–19, Lecture Notes in Comput. Sci., 2146, Springer, Berlin, 2001.

[2] ———, *New partial key exposure attacks on RSA*, Advances in cryptology—CRYPTO 2003, 27–43, Lecture Notes in Comput. Sci., 2729, Springer, Berlin, 2003.

[3] ———, *A generalized Wiener attack on RSA*, Public key cryptography—PKC 2004, 1–13, Lecture Notes in Comput. Sci., 2947, Springer, Berlin, 2004.

[4] D. Boneh, *Twenty years of attacks on the RSA cryptosystem*, Notices Amer. Math. Soc. **46** (1999), no. 2, 203–213.

[5] D. Boneh and G. Durfee, *Cryptanalysis of RSA with private key d less than* $N^{0.292}$, IEEE Trans. Inform. Theory **46** (2000), no. 4, 1339–1349.

[6] D. Boneh, G. Durfee, and Y. Frankel, *Exposing an RSA private key given a small fraction of its bits*, AsiaCrypt'98, LNCS 1514, pp. 25–34, Springer-Verlag, 1998.

[7] D. Boneh, R. DeMillo, R. Lipton, *On the importance of checking cryptographic protocols for faults* (*extended abstract*), Advances in cryptology—EUROCRYPT '97 (Konstanz), 37–51, Lecture Notes in Comput. Sci., 1233, Springer, Berlin, 1997.

[8] D. Coppersmith, *Small solutions to polynomial equations, and low exponent RSA vulnerabilities*, J. Cryptology **10** (1997), no. 4, 233–260.
[9] A. Duejella, *Continued fractions and RSA with small secret exponent*, Tatra Mt. Math. Publ. **29** (2004), 101–112.
[10] M. Ernst, E. Jochemsz, A. May, and B. de Weger, *Partial key exposure attacks on RSA up to full size exponents*, Advances in cryptology—EUROCRYPT 2005, 371–386, Lecture Notes in Comput. Sci., 3494, Springer, Berlin, 2005.
[11] J. Hastad, *On using RSA with low exponent in a public key network*, Advances in cryptology—CRYPTO '85 (Santa Barbara, Calif., 1985), 403–408, Lecture Notes in Comput. Sci., 218, Springer, Berlin, 1986.
[12] N. Howgrave-Graham, *Finding small roots of univariate modular equations revisited*, Cryptography and coding (Cirencester, 1997), 131–142, Lecture Notes in Comput. Sci., 1355, Springer, Berlin, 1997.
[13] E. Jochemsz, *Cryptanalysis of RSA variants using small roots of polynomials*, Ph. D. thesis, Technische Universiteit Eindhoven, 2007.
[14] P. Kocher, *Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems*, Proc. Crypto'96, 104–113, Lecture Notes in Comput. Sci., 1109, Springer-Verlag, 1996.
[15] P. Kocher, J. Jaffe, and B. Jun, *Differential power analysis*, CRYPTO '99, 388–397, Lecture Notes in Comput. Sci., 1666, Springer, 1999.
[16] A. K. Lenstra, H. W. Lenstra, and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann. **261** (1982), no. 4, 515–534.
[17] A. May, *Using LLL-Reduction for Solving RSA and Factorization Problems: A Survey*, LLL+25 Conference in honour of the 25th birthday of the LLL algorithm, 2007. Available at http://www.informatik.tu-darmstadt.de/KP/alex.html [last accessed 23 December, 2008].
[18] A. Nitaj, *Another Generalization of Wiener's Attack on RSA*, Progress in Cryptology—AFRICACRYPT 2008, 174–190, Lecture Notes in Comput. Sci., 5023, Springer-Verlag, 2008.
[19] J. M. Pollard, *Theorems on factorization and primality testing*, Proc. Cambridge Philos. Soc. **76** (1974), 521–528.
[20] R. L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM **21** (1978), no. 2, 120–126.
[21] R. Steinfeld, S. Contini, H. Wang, and J. Pieprzyk, *Converse results to the Wiener attack on RSA*, Public key cryptography—PKC 2005, 184–198, Lecture Notes in Comput. Sci., 3386, Springer, Berlin, 2005.
[22] D. R. Stinson, *Cryptography – Theory and Practice*, 2nd Edition, Chapman & Hall/CRC, 2002.
[23] H.-M. Sun, M.-E. Wu, and Y.-H. Chen, *Estimating the prime-factors of an RSA modulus and an extension of the Wiener attack*, Applied Cryptography and Network Security, 116–128, Lecture Notes in Comput. Sci., 4521, Spriger, 2007.
[24] H.-M. Sun and C.-T. Yang, *RSA with balanced short exponents and its application to entity authentication*, Public key cryptography—PKC 2005, 199–215, Lecture Notes in Comput. Sci., 3386, Springer, Berlin, 2005.
[25] E. R. Verheul and H. C. A. van Tilborg, *Cryptanalysis of 'less short' RSA secret exponents*, Appl. Algebra Engrg. Comm. Comput. **8** (1997), no. 5, 425–435.
[26] B. de Weger, *Cryptanalysis of RSA with small prime difference*, Appl. Algebra Engrg. Comm. Comput. **13** (2002), no. 1, 17–28.
[27] M. Wiener, *Cryptanalysis of short RSA secret exponents*, IEEE Trans. Inform. Theory **36** (1990), no. 3, 553–558.
[28] H. C. Williams, *A $p+1$ method of factoring*, Math. Comp. **39** (1982), no. 159, 225–234.

Santanu Sarkar
Indian Statistical Institute
203 B T Road, Kolkata 700 108, India
E-mail address: santanu_r@isical.ac.in

Subhamoy Maitra
Indian Statistical Institute
203 B T Road, Kolkata 700 108, India
E-mail address: subho@isical.ac.in