Further Results on Implicit Factoring in Polynomial Time

Santanu Sarkar and Subhamoy Maitra

Indian Statistical Institute, 203 B T Road, Kolkata 700 108, India
{santanu_r, subho}@isical.ac.in

Abstract. In PKC 2009, May and Ritzenhofen presented some interesting problems related to factoring large integers with some implicit hints and one of the problems is as follows. Consider $N_1 = p_1q_1$ and $N_2 = p_2q_2$, where p_1, p_2, q_1, q_2 are large primes. The primes p_1, p_2 are of same bit-size with the constraint that certain amount of Least Significant Bits (LSBs) of p_1, p_2 are same. Further the primes q_1, q_2 are of same bit-size without any constraint. May and Ritzenhofen proposed a strategy to factorize both N_1, N_2 in poly$(\log N)$ time (N is an integer with same bit-size as N_1, N_2) with the implicit information that p_1, p_2 share certain amount of LSBs. We look at the same problem with a different lattice-based strategy. In a general framework, our method works when implicit information is available related to Least Significant as well as Most Significant Bits (MSBs). Given $q_1, q_2 \approx N^{\alpha}$, we show that one can factor N_1, N_2 simultaneously in poly$(\log N)$ time (under some assumption related to Gröbner Basis) when p_1, p_2 share certain amount of MSBs and/or LSBs. We also study the case when p_1, p_2 share some bits in the middle. Our strategy presents new and encouraging results in this direction. Moreover, some of the observations by May and Ritzenhofen get improved when we apply our ideas for the LSB case.

Keywords: Implicit Information, Prime Factorization.

1 Introduction

Very recently, in [11], a new direction towards factorization with implicit information has been introduced. Consider two integers N_1, N_2 such that $N_1 = p_1q_1$ and $N_2 = p_2q_2$ where p_1, q_1, p_2, q_2 are primes and p_1, p_2 share t least significant bits (LSBs). It has been shown in [11] that when q_1, q_2 are primes of bit-size α, then N_1, N_2 can be factored simultaneously if $t \geq 2(\alpha+1)$. This bound on t has further been improved when $N_1 = p_1q_1, N_2 = p_2q_2, \ldots, N_k = p_kq_k$ and all the p_i's share t many LSBs. The motivation of this problem comes from oracle based complexity of factorization problems. Prior to the work of [11], the main assumption in this direction was that an oracle explicitly outputs certain amount of bits of one prime. The idea of [11] deviates from this paradigm in the direction that none of the bits of the prime will be known, but some implicit information can be available regarding the prime. That is, an oracle, on input to N_1, outputs a different N_2 as described above. A nice motivation towards the importance of this problem is presented in the introduction of [11].

Factoring of large integers is one of the most challenging problems in Mathematics and Computer Science. The quadratic Sieve [13], the elliptic curve method [7] and number field sieve [8] are among the significant works on classical computing model. Till date, there is no known polynomial time factorization algorithm on this model, though in a seminal work Shor [15] has presented a polynomial time algorithm for factorization on quantum computing platforms. Towards the partial results for efficient factorization in classical domain (factoring with explicit information from an oracle according to [11]), Rivest and Shamir [14] showed that N (where $N = pq$) can be factored efficiently when $\frac{3}{5}\log_2 p$ many MSBs of p are known. Later, Coppersmith [2] improved this bound, where $\frac{1}{2}\log_2 p$ many MSBs of p need to be known for efficient factorization.

In this paper we assume the equality of either the MSBs or the LSBs or some portions of LSBs as well as MSBs, i.e., we consider that p_1, p_2 share either t many MSBs or t many LSBs or total t many bits considering LSBs and MSBs together. Further, we consider the case when the primes share certain amount of bits at the middle. Our approach in solving the problem is different from that of [11].

We like to point out that the event of getting two primes with a many LSBs equal is approximately as frequent as getting two primes with a many MSBs equal. This can be noted as follows. Let i be an a bit integer. Consider two sets A and B where

$$A = \{p : p \text{ is a prime of } a + b \text{ bits and } a \text{ many MSBs of } p \text{ is } i\},$$

$$B = \{p : p \text{ is a prime of } a + b \text{ bits and } a \text{ many LSBs of } p \text{ is } i\}.$$

We first calculate cardinality of A. Let $X = 2^b i$. Then from prime number theorem [1, Page 65] $|A| \approx \frac{X+2^b-1}{\log(X+2^b-1)} - \frac{X}{\log X} \approx \frac{2^b}{\log X}$ (assume $b < a$) $\approx \frac{2^b}{\log 2^{a+b}}$, which is $O(2^b)$. Also, we have $B = \{p : p \text{ is a prime of } a + b \text{ bits and } p \equiv i(\text{mod} 2^a)\}$. From Dirichlet's theorem related to prime numbers [1, Page 154], we have $|B| \approx (\frac{2^{a+b}-1}{\log(2^{a+b}-1)} - \frac{2^{a+b-1}-1}{\log(2^{a+b-1}-1)})\frac{1}{\phi(2^a)} \approx \frac{2^{a+b-1}}{\log 2^{a+b}}\frac{1}{2^{a-1}}$, which is again $O(2^b)$. Thus, $|A|$ and $|B|$ are of the same order.

Following this introductory section, in Section 2, we present the technical results considering the LSBs and/or MSBs of p_1, p_2 are same. Section 2.1 considers LSBs and MSBs together and the most general result is presented here. Sections 2.2, 2.3 follow the general idea for specific cases considering only the MSBs and LSBs. Comparisons with the existing work [11] is presented in Section 2.3. Next in Section 3, we consider the case when the primes p_1, p_2 share a contiguous portion of bits at the middle.

All the theoretical results are supported by experiments. We have written the programs in SAGE 3.1.1 over Linux Ubuntu 8.04 on a computer with Dual CORE Intel(R) Pentium(R) D CPU 1.83 GHz, 2 GB RAM and 2 MB Cache.

Our strategy is based on lattice reduction [9] followed by Gröbner Basis technique [4, Page 77]. For detailed notion on the technique we use, the readers are referred to [12, 5, 6]. The main idea follows the generalized strategy for finding roots of multivariate polynomials as explained in [5]. In this regard, we like to point out that the polynomials, that we use in Theorems 1, 2, have not been studied earlier following the technique of [5] and one may note that these polynomials are not covered in [6, Table 3.2, Section 3.4].

Before proceeding further, let us clarify an assumption that is required for our theoretical results. Suppose we have a set of polynomials $\{f_1, f_2, \ldots, f_i\}$ on n variables having the roots of the form $(x_{1,0}, x_{2,0}, \ldots, x_{n,0})$. Then it is known that the Gröbner Basis $\{g_1, g_2, \ldots, g_j\}$, of $J = < f_1, f_2, \ldots, f_i >$ (the ideal generated by $\{f_1, f_2, \ldots, f_i\}$), preserves the set of common roots of $\{f_1, f_2, \ldots, f_i\}$. For our problems, we assume that the roots can be collected efficiently from $\{g_1, g_2, \ldots, g_j\}$. Though this is true in practice as noted from the experiments we perform, we formally state the following assumption that we will consider for our theoretical results.

Assumption 1. Consider a set of polynomials $\{f_1, f_2, \ldots, f_i\}$ on n variables having the roots of the form $(x_{1,0}, x_{2,0}, \ldots, x_{n,0})$. Let J be the ideal generated by $\{f_1, f_2, \ldots, f_i\}$. Then we will be able to collect the roots efficiently from the Gröbner Basis of J.

2 Implicit Factoring of Two Large Integers

Here we present the exact conditions on p_1, q_1, p_2, q_2 under which N_1, N_2 can be factored efficiently.

Throughout this paper, we will consider p_1, p_2 are primes of same bit size and q_1, q_2 are primes of same bit size. Thus $N_1 = p_1q_1$ and $N_2 = p_2q_2$ are also of same bit size. We use N to represent an integer of same bit size as of N_1, N_2.

2.1 The General Result

We first consider the case where some amount of LSBs as well as some amount of MSBs of p_1, p_2 are same. Based on this, we present the following generalized theorem.

Theorem 1. *Let $N_1 = p_1q_1$ and $N_2 = p_2q_2$, where p_1, q_1, p_2, q_2 are primes. Let $q_1, q_2 \approx N^{\alpha}$. Consider that $\gamma_1 \log_2 N$ many MSBs and $\gamma_2 \log_2 N$ many LSBs of p_1, p_2 are same. Let $\beta = 1 - \alpha - \gamma_1 - \gamma_2$. Under Assumption 1, one can factor N_1, N_2 in polynomial time if $-4\alpha^2 - 2\alpha\beta - \frac{1}{4}\beta^2 + 4\alpha + \frac{5}{3}\beta - 1 < 0$ provided $1 - \frac{3}{2}\beta - 2\alpha \geq 0$.*

Proof. It is given that $\gamma_1 \log_2 N$ many MSBs and $\gamma_2 \log_2 N$ many LSBs of p_1, p_2 are same. Thus, we can write $p_1 = N^{1-\alpha-\gamma_1}P_0 + N^{\gamma_2}P_1 + P_2$ and $p_2 = N^{1-\alpha-\gamma_1}P_0 + N^{\gamma_2}P_1' + P_2$. Thus, $p_1 - p_2 = N^{\gamma_2}(P_1 - P_1')$. Since $N_1 = p_1q_1$ and $N_2 = p_2q_2$, putting $p_1 = \frac{N_1}{q_1}$ and $p_2 = \frac{N_2}{q_2}$, we get $N^{\gamma_2}(P_1 - P_1')q_1q_2 - N_1q_2 + N_2q_1 = 0$. Thus we need to solve $f'(x, y, z) = N^{\gamma_2}xyz - N_1x + N_2y = 0$ whose roots corresponding to x, y, z are $q_2, q_1, P_1 - P_1'$. Since there is no constant term in f', we define a new polynomial $f(x, y, z) = f'(x - 1, y, z) = N^{\gamma_2}xyz - N^{\gamma_2}yz - N_1x + N_1 + N_2y$. The root (x_0, y_0, z_0) of f is $(q_2 + 1, q_1, P_1 - P_1')$. The idea of modifying the polynomial with a constant term was introduced in [3, Appendix A] and later used in [5] which we follow here.

Let X, Y, Z be the upper bounds of $q_2 + 1, q_1, P_1 - P_1'$ respectively. As given in the statement of this theorem, $X = N^{\alpha}, Y = N^{\alpha}, Z = N^{\beta}$. Following the "Extended Strategy" of [5, Page 274],

$$S = \bigcup_{0 \leq j \leq t} \{x^i y^j z^{k+j} : x^i y^j z^k \text{ is a monomial of } f^m\},$$

$$M = \{ \text{ monomials of } x^i y^j z^k f : x^i y^j z^k \in S\}.$$

We exploit t many extra shifts of z where t is a non-negative integer. Our aim is to find two more polynomials f_0, f_1 that share the root $(q_2 + 1, q_1, P_1 - P_1')$ over the integers.

From [5], we know that these polynomials can be found by lattice reduction if

$$X^{s_1} Y^{s_2} Z^{s_3} < W^{s}, \tag{1}$$

where $s_j = \sum_{x^{i_1} y^{i_2} z^{i_3} \in M \setminus S} i_j$ for $j = 1, \ldots, 3$, and $W = ||f(xX, yY, zZ)||_\infty \geq N_2 X$.

One can check
$s_1 = \frac{m^3}{2} + \frac{5}{2}m^2 + 4m + 2 + 2t + \frac{3}{2}m^2 t + \frac{7}{2}mt$,
$s_2 = \frac{5}{6}m^3 + 4m^2 + \frac{37}{6}m + 3 + 2t + \frac{3}{2}m^2 t + \frac{7}{2}mt$,
$s_3 = \frac{1}{2}m^3 + \frac{5}{2}m^2 + 4m + 2 + \frac{3}{2}t^2 + \frac{7}{2}t + \frac{3}{2}m^2 t + mt^2 + \frac{9}{2}mt$, and
$s = \frac{1}{3}m^3 + \frac{3}{2}m^2 + \frac{13}{6}m + 1 + t + m^2 t + 2mt$.

Neglecting the lower order terms, form (1), we get the condition as

$$X^{\frac{m^3}{2} + \frac{3}{2}m^2 t} Y^{\frac{5}{6}m^3 + \frac{3}{2}m^2 t} Z^{\frac{m^3}{2} + \frac{3}{2}m^2 t + mt^2} < W^{\frac{m^3}{3} + m^2 t}.$$

Let $t = \tau m$. Then we get the required condition is

$$\tau^2 \beta + (2\alpha + \frac{3}{2}\beta - 1)\tau + (\alpha + \frac{\beta}{2} - \frac{1}{3}) < 0. \tag{2}$$

Now optimal value of τ to minimize the left hand side of (2) is $\frac{1 - \frac{3}{2}\beta - 2\alpha}{2\beta}$. Putting this optimal value, the required condition will be $-64\alpha^2 - 32\alpha\beta - 4\beta^2 + 64\alpha + \frac{80}{3}\beta - 16 < 0$. That is, when this condition holds, according to [5], we get two polynomials f_0, f_1 such that $f_0(x_0, y_0, z_0) = f_1(x_0, y_0, z_0) = 0$. Under Assumption 1, we can extract x_0, y_0, z_0 following the method of [10, Section 6]. □

Remark 1. In the proof of Theorem 1, we have applied extra shifts over z. In fact, we have tried with extra shifts on x, y too. However, we have noted that the best theoretical as well as experimental results are achieved using extra shifts on z.

Looking at Theorem 1, it is clear that the efficiency of this factorization technique depends on the total amount of bits that are equal considering the most and least significant parts together. Next we present an example below.

Example 1. Let us consider 750-bit primes p_1 and p_2
3804472805395186392319221660578496208300951856349524536490291627689678450887
3994603764416042481638726883020251099785398270595309011413652074066298289696
31841459373573878076619162688905451127596423509967449841486470692918256969.
and
3804472805395186392319221660578496208300951856349524536490291627689216610760
8916018165804358808795724349647533346298650637180633006710173703444662098451
70635265772659883844077694434985101401094132819711524954637781487537874249.

Note that p_1, p_2 share 222 many MSBs and 220 many LSBs, i.e., 442 many bits in total. Further, q_1, q_2 are 250-bit primes
1788684495317470472835032661187758515078190921640698934821176591562967327967 and
1706817658439540390758485693495273025642629127144779879402852507986344279931
respectively. Given N_1, N_2, with only the implicit information, we can factorize both of them in poly($\log N$) time. We use lattice of dimension 105 (parameters $m = 3, t = 2$) and the lattice reduction takes 6227.76 seconds. □

2.2 The MSB Case

The study when p_1, p_2 share some MSBs has not been considered in [11], which we present in this section. The following result follows from Theorem 1, noting $\beta = 1 - \alpha - \gamma_1$.

Corollary 1. *Let* $N_1 = p_1 q_1$ *and* $N_2 = p_2 q_2$*, where* p_1, q_1, p_2, q_2 *are primes. Let* $Q_1, Q_2 \approx N^{\alpha}$*, and* $|p_1 - p_2| < N^{\beta}$*. Under Assumption 1, one can factor* N_1, N_2 *in polynomial time if* $-4\alpha^2 - 2\alpha\beta - \frac{1}{4}\beta^2 + 4\alpha + \frac{5}{3}\beta - 1 < 0$ *provided* $1 - \frac{3}{2}\beta - 2\alpha \geq 0$.

It is clear from Corollary 1, that fixing the bit-size of N, if the size of q_1, q_2 (i.e., α) increases, then the equality of the MSBs of p_1, p_2 should increase (i.e., β should decrease) for efficient factorization of N_1, N_2.

The theoretical as well as experimental results are presented in Table 1. The experimental results in each row are based on average of five runs where N_1, N_2 are 1000-bit integers. The experiments in Table 1 are performed with lattice dimension 46 (parameters $m = 2, t = 1$) and each lattice reduction takes around 30 seconds.

To explain the results of Table 1, let us concentrate on the first row. As $\alpha = 0.23$, we have q_1, q_2 are of bit size $0.23 \times 1000 = 230$. Thus, p_1, p_2 are of bit size $1000 - 230 = 770$. Now, the numerical value from Corollary 1 tells that $770 - 0.255 \times 1000 = 515$ many MSBs of p_1, p_2 need to be equal to have efficient factorization of N_1, N_2 simultaneously. However, the average of the experimental results are more encouraging which shows that only $770 - 0.336 \times 1000 = 434$ many MSBs of p_1, p_2 need to be equal to have efficient factorization of N_1, N_2 simultaneously.

α	Numerical upper bound of β following Corollary 1	Results achieved for β from experiments
0.23	0.255	0.336
0.24	0.239	0.313
0.25	0.225	0.296
0.26	0.210	0.269
0.27	0.196	0.250

Table 1. Theoretical and experimental values of α, β for which N_1, N_2 can be factored efficiently.

Remark 2. From Table 1 it is clear that we get much better results in experiments than the theoretical bound. This is because, for the parameters we consider here, the shortest vectors belong to some sub-lattice. However, the theoretical calculation in Theorem 1 cannot capture that and further, identifying such optimal sub-lattice seems to be difficult as pointed out in [5, Section 7.1].

We also present evidences to show that higher lattice dimension provides better experimental results. In Examples 2, 3, we find that when $\alpha = 0.25$, the values of β that can be achieved are as high as 0.308, 0.311 respectively for lattice parameters $m = 3, t = 2$. These results are better than the average $\beta = 0.296$ as presented in Table 1 for $m = 2, t = 1$.

Example 2. For a demonstration of the experiment, consider 750-bit primes p_1 and p_2 3967780110926558985695599259225508707353082348138173713914249580078148537872659986732427543412353227686360435307307811045754814960959318503826990494991538951443158292762268189891045388828922478530615979139037853178431738420087 and 3967780110926558985695599259225508707353082348138173713914249580078148537872659986732427543412353227686360435307307811045754814960959767263984990466987511414871763397210786172961055167000499946887837157176166275686743465332147. Note that p_1, p_2 share 442 many MSBs. Further, q_1, q_2 are 250-bit primes 1791405259026492103131865184203435870047916914753003354202248185126637129539 and 1359854273468970113914581544928445498889538930116761650886947228775354080297 respectively. Given N_1, N_2, with only the implicit information, we can factorize both of them in poly$(\log N)$ time. We use lattice of dimension 105 (parameters $m = 3, t = 2$) and the lattice reduction takes 6457.84 seconds. □

Example 3. As another experimental result, consider 750-bit primes p_1 and p_2 3103293851234545621612884177271352199071965229969307590769556901553501696121486894504150753778107049899894702257572943969973109842059427848262110574521661287756193724060104016731225285634163002534645448007119837656454227440177 and 3103293851234545621612884177271352199071965229969307590769556901553501696121486894504150753778107049899894702257572943969973109842063500611566034390188986791515114690594523923567275780555267831035031294553991617471138271288077. Note that p_1, p_2 share 439 many MSBs. Further, q_1, q_2 are 250-bit primes 1761986055485501596400884508719659270275271677762068580864458138443043985389 and 1793915333056311315115475413216227307458109801843263226409813428452265284467 respectively. Given N_1, N_2, with only the implicit information, we can factorize both of them in poly$(\log N)$ time. We use lattice of dimension 105 (parameters $m = 3, t = 2$) and the lattice reduction takes 7150.09 seconds. □

The next example considers the primes p_1, p_2 of 650 bits and q_1, q_2 of 350 bits. This is to demonstrate how our method works experimentally for larger q_1, q_2.

Example 4. As another experimental result, consider 650-bit primes p_1 and p_2 3275958003351638061986916939385797455267819362579720819294801659002592355528289333246983236570140784030169547342941406605698168210875724855956184786453949781113664574387794170322092125817649417089 and 3275958003351638061986916939385797455267819362579720819294801659002592355528289333246983236570140784030169547342941406605698168210875724855956184786453949781116017823491796542769181094911460404833. Note that p_1, p_2 share 529 many MSBs. Further, q_1, q_2 are 350-bit primes 1823227073736496017375980522958217483156482551719830362235263547237757846388546536472532649209077149673483 and 2198082402853042081264929588674625335352875813205705506006454409313585071920396431401126233354206989620787 respectively. Given N_1, N_2, with only the implicit information, we can factorize both of them in poly$(\log N)$ time. We use lattice of dimension 105 (parameters $m = 3, t = 2$) and the lattice reduction takes 10709.84 seconds. □

In Theorem 1, we have considered that given the conditions, we can find f_0, f_1 by lattice reduction. However, in practice, one may get more polynomials. In our experiments, we used four polynomials f_0, f_1, f_2, f_3 that come after lattice reduction. Let J be the ideal generated by $\{f, f_0, f_1, f_2, f_3\}$ and let the corresponding Gröbner Basis be G. We studied the first three elements of G and found that one of them is of the form $y^a(x - \frac{q_2}{q_1}y - 1)$, where a is a small positive integer and we observed $a = 0, 1, 2$ in the experiments. Note that $x_0 = q_2 + 1, y_0 = q_1$ is the root of this polynomial.

Thus the result of Theorem 1 and the experimental evidences show that under certain conditions polynomial time factoring is possible with implicit hints.

2.3 The LSB Case

Let us first explain the ideas presented in [11]. Let $N_1 = p_1q_1$ and $N_2 = p_2q_2$. In [11, Section 3], it has been explained that if $q_1, q_2 \approx N^\alpha$, then for efficient factorization of N_1, N_2, the primes p_1, p_2 need to share at least $2\alpha \log_2 N$ many LSBs.

Our strategy is different from the strategy of [11] and we follow the result of Theorem 1 to get the result.

Corollary 2. *Let $N_1 = p_1q_1$ and $N_2 = p_2q_2$, where p_1, q_1, p_2, q_2 are primes. Let $q_1, q_2 \approx N^\alpha$. Consider that $\gamma \log_2 N$ many LSBs of p_1, p_2 are same, i.e., $p_1 \equiv p_2 \bmod N^\gamma$. Let $\beta = 1 - \alpha - \gamma$. Under Assumption 1, one can factor N_1, N_2 in polynomial time if $-4\alpha^2 - 2\alpha\beta - \frac{1}{4}\beta^2 + 4\alpha + \frac{5}{3}\beta - 1 < 0$ provided $1 - \frac{3}{2}\beta - 2\alpha \geq 0$.*

The numerical values related to the theoretical result of [11] and Corollary 2 as well as experimental results are presented in Table 2. The experimental results in each row are based on one run where N_1, N_2 are 1000-bit integers. The experiments in Table 2 are performed with lattice dimension 46 (parameters $m = 2, t = 1$) and each lattice reduction takes around 30 seconds. Similar to the observation in Section 2.2, we note from Table 2 that better results are obtained in experiments than the theoretical bound. We believe the reason is same as explained in Remark 2 in Section 2.2.

α	Numerical upper bound of β following [11]	Numerical upper bound of β following Corollary 2	Results achieved for β from experiments
0.23	0.31	0.255	0.336
0.24	0.28	0.239	0.314
0.25	0.25	0.225	0.296
0.26	0.22	0.210	0.268
0.27	0.19	0.196	0.251

Table 2. Theoretical and experimental values of α, β for which N_1, N_2 can be factored efficiently.

In our notation, the number of MSBs in each of p_1, p_2 that are unshared is $\beta \log_2 N$. Thus $\beta = (1 - \alpha) - 2\alpha = 1 - 3\alpha$, where $\alpha \log_2 N$ is the bit size of q_1, q_2. Table 2 identifies that while our theoretical result is either worse or better than that of [11] based on the values of α, the experimental results that we obtain are always better than [11]. In the introduction of [11], it has been pointed out that for 250-bit q_1, q_2 and 750-bit p_1, p_2, the primes p_1, p_2 need to share 502 many LSBs. We have implemented the strategy of [11] and observed similar results.

On the other hand, our experimental results are better as evident from Table 2, when $\alpha = 0.25$. In fact, we experimented with a higher lattice dimension as explained in Examples 5, 6 and our strategy requires only 440 and 438 many LSBs respectively to be shared in p_1, p_2. These results are much better than [11], where 502 many LSBs have been shared.

Example 5. In this experiment, consider 750-bit primes p_1 and p_2
5232464401790173496889776813731992463007796797197958752484439607191540455235
6608087324378089911735572744300332234102069657955934461989289309962068103250
78810654140616439325724089448684722792481034854929045247229685114499401607 and
4311796718402237315332622037900773800355832324549261614699895316190733254104
0376948850231036794311185546576317750184830286997614825307318419096215142451
35730269665188193197190838441262406453523279005533091728042442492020950919.
Note that p_1, p_2 share 440 many LSBs which will be clear if one writes p_1, p_2 in binary and checks the LSBs. Further, q_1, q_2 are 250-bit primes
1631651738790114027147107602960138604308539138427653628254827153426896347739 and
1776124692833044236475237348456766321872003926797460168161822934670015844393
respectively. Given N_1, N_2, with only the implicit information, we can factorize both of them in poly$(\log N)$ time. We use lattice of dimension 105 (parameters $m = 3, t = 2$) and the lattice reduction takes 7160.63 seconds. □

Example 6. Here we consider 750-bit primes p_1 and p_2
5895254139679228077142387416586490039613283191466241401307494261824605966908
4690420722716275439075281566487074700579275565739610880278518405272767367010
03322173329476277711235116947599147048863366019662261619304575961682668297 and
4392119049423447468690947059559090008016802774014559696547174955333794465234
2861564934625350120675407265601224878945969002652471346685040069850301681742
01428949181076294088915910886847055459554005392066246146594876423472933641.
Note that p_1, p_2 share 438 many LSBs. Further, q_1, q_2 are 250-bit primes
916010977814643010666950783967979656772444969801926690589674791043059104197 and
1587061752065032326280290326014711341044827082150757395718254111544994945759
respectively. Given N_1, N_2, with only the implicit information, we can factorize both of them in poly$(\log N)$ time. We use lattice of dimension 105 (parameters $m = 3, t = 2$) and the lattice reduction takes 7273.52 seconds. □

The next example considers the primes p_1, p_2 of 650 bits and q_1, q_2 of 350 bits. This is to demonstrate how our method works experimentally for larger q_1, q_2.

Example 7. Here we consider 650-bit primes p_1 and p_2
31370558899010969090775314583271711200148784533831527325125302572763631682927
85241218747273712763711037157637711966791419526760377688029885676273831127205

611509045644179511599106554189421550654601 and
24514360109308139038143105060866330207163283877575874117266619411272093212167
40545001634090447037011441230660481097503555238640524767415889480913091786359
014934176726120292021849927924906510931081.
Note that p_1, p_2 share 531 many LSBs. Further, q_1, q_2 are 350-bit primes
18514205888865174789397135953034924041903821127915515977985711433395162336134
45774636517955322189132943773 and
22583503051484782188700251613256676376586234088559388990147583389496665081155
61055599847183651567682695481
respectively. Given N_1, N_2, with only the implicit information, we can factorize both of them in poly$(\log N)$ time. We use lattice of dimension 105 (parameters $m = 3, t = 2$) and the lattice reduction takes 15016.42 seconds. □

We now discuss in more details how our strategy compares with that of [11]. It is indeed clear from Table 2, that our experimental results provide much better performance than the theoretical results presented in our paper as well as in [11]. Moreover, we now explain how the technique of [11] and our strategy compare in terms of theoretical results.

Let us first concentrate on the formula $\beta = 1 - 3\alpha$, that characterizes the bound on the primes for efficient factoring in [11]. When $\alpha = \frac{1}{3}$, then β becomes zero, implying that p_1, p_2 need to have all the bits shared. Thus, the upper bound on the smaller primes q_1, q_2 is $N^{\frac{1}{3}}$, where sharing of LSBs in p_1, p_2 helps in efficient factoring.

However, in our case, the bound on the primes is characterized by $-4\alpha^2 - 2\alpha\beta - \frac{1}{4}\beta^2 + 4\alpha + \frac{5}{3}\beta - 1 < 0$ provided $1 - \frac{3}{2}\beta - 2\alpha \geq 0$. We find that β becomes zero when $\alpha = \frac{1}{2}$. Thus in our case, the upper bound on smaller primes q_1, q_2 is $N^{\frac{1}{2}}$, where sharing of LSBs in p_1, p_2 helps in efficient factoring.

Theoretically, our method starts performing better (i.e., β in our case is greater than that of [11]) than [11] when $\alpha \geq 0.266$. Thus for $q_1, q_2 \geq N^{0.266}$, our method will require less number of LSBs of p_1, p_2 to be equal than that of [11]. This is also presented in Figure 1. Referring Figure 1, we like to reiterate that our experimental results outperforms the theoretical results presented by us as well as in [11].

Though our result does not generalize for the case where $N_1, N_2, \ldots, N_k$ immediately, we like to compare the result of Example 7 with [11, Table 1, Section 6.2] when $\alpha = 0.35$ and N is of 1000 bits. This is presented in Table 3. One may note that the idea of [11] requires 10 many N_i's as the input where $N_i = p_i q_i$, $1 \leq i \leq 10$. In such a case, 391 many LSBs need to be same for $p_1, \ldots, p_{10}$. On the other hand, we require higher number of LSBs, i.e., 531 to be same, but we require only N_1, N_2.

Reference	Bit sizes of p_i, q_i	Number of N_i's required	Number of shared bits
[11, Table 1, Section 6.2]	650, 350	10	391
Our Example 7	650, 350	2	531

Table 3. Comparison of experimental results when $\alpha = 0.35$.

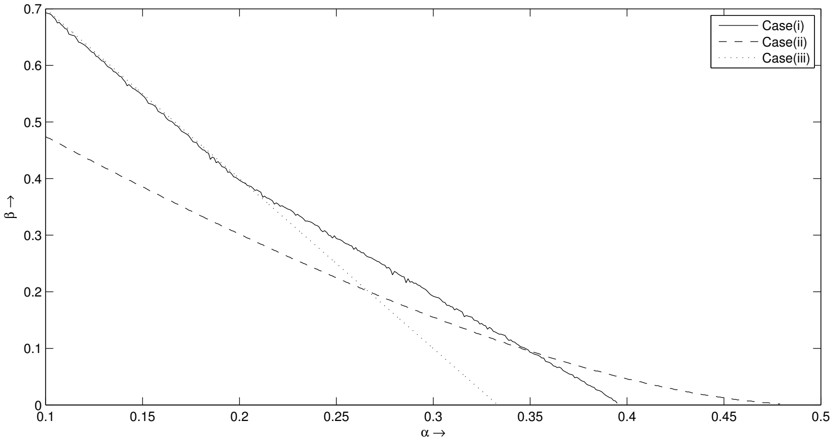

Fig. 1. Comparison of our experimental (case (i)) and theoretical results (case (ii)) with that of [11] (case (iii)). The numerical values of the theoretical results are generated using the formulae $\beta = 1 - 3\alpha$ for [11] and Corollary 2 for our case. The experimental results are generated by one run in each case with lattice dimension 46 (parameters $m = 2, t = 1$) for 1000 bits N_1, N_2. The values of α are considered in $[0.1, 0.5]$, in a step of 0.01.

The analysis of our results related to LSBs, presented in this section, will apply similarly for our analysis related to MSBs or LSBs and MSBs taken together as explained earlier. The other case remaining in this direction is to study what happens when p_1, p_2 share some bits at the middle. This is studied in the next section.

3 Primes p_1, p_2 Share a Contiguous Portion of Bits at the Middle

Now we consider the case when p_1, p_2 share a contiguous portion of bits at the middle.

Theorem 2. *Let $N_1 = p_1q_1$ and $N_2 = p_2q_2$, where p_1, q_1, p_2, q_2 are primes. Let $q_1, q_2 \approx N^{\alpha}$. Consider that a contiguous portion of bits of p_1, p_2 are same at the middle leaving the $\gamma_1 \log_2 N$ many MSBs and $\gamma_2 \log_2 N$ many LSBs. Then under Assumption 1, we can factor both N_1, N_2 if there exist $\tau_1, \tau_2 \geq 0$ for which $h(\tau_1, \tau_2, \alpha, \gamma_1, \gamma_2) < 0$ where $h(\tau_1, \tau_2, \alpha, \gamma_1, \gamma_2) = (3\tau_1\tau_2 + \frac{7}{3}\tau_1 + \frac{7}{3}\tau_2 + \frac{17}{24})\alpha + (\tau_1^2\tau_2 + \frac{3}{2}\tau_1\tau_2 + \frac{3}{4}\tau_1^2 + \frac{2}{3}\tau_1 + \frac{2}{3}\tau_2 + \frac{1}{6})\gamma_1 + (\tau_1\tau_2^2 + \frac{3}{2}\tau_1\tau_2 + \frac{3}{4}\tau_2^2 + \frac{2}{3}\tau_1 + \frac{2}{3}\tau_2 + \frac{1}{6})\gamma_2 - (\tau_1\tau_2 + \frac{\tau_1}{2} + \frac{\tau_2}{2} + \frac{1}{8})(1 + \gamma) < 0$ and $\gamma = \max\{\gamma_1, \gamma_2\}$.*

Proof. We can write $p_1 = N^{1-\alpha-\gamma_1}p_{10} + N^{\gamma_2}p_{11} + p_{12}$ and $p_2 = N^{1-\alpha-\gamma_1}p_{20} + N^{\gamma_2}p_{11} + p_{22}$. So $p_1 - p_2 = N^{1-\alpha-\gamma_1}(p_{10} - p_{20}) + (p_{12} - p_{22})$. Since $p_1 = \frac{N_1}{q_1}$ and $p_2 = \frac{N_2}{q_2}$ we have $N_1q_2 - N_2q_1 - N^{1-\alpha-\gamma_1}(p_{10} - p_{20})q_1q_2 - (p_{12} - p_{22})q_1q_2 = 0$. Thus we need to solve $f'(x, y, z, v) = N_1x - N_2y - N^{1-\alpha-\gamma_1}xyz - xyv = 0$ whose roots corresponding to x, y, z, v are $q_2, q_1, p_{10} - p_{20}, p_{12} - p_{22}$.

Since there is no constant term in f', we define a new polynomial $f(x, y, z, v) = f'(x - 1, y, z, v) = N_1x - N_2y - N_1 - N^{1-\alpha-\gamma_1}xyz + N^{1-\alpha-\gamma_1}yz - xyv + yv$. The root (x_0, y_0, z_0, v_0) of f is $(q_2 + 1, q_1, p_{10} - p_{20}, p_{12} - p_{22})$.

Let $X = N^{\alpha}, Y = N^{\alpha}, Z = N^{\gamma_1}, V = N^{\gamma_2}$. Then we can take X, Y, Z, V as the upper bound of x_0, y_0, z_0, v_0 respectively.

Following the "Extended Strategy" of [5, Page 274], we have the following definitions of S, M, where m, t_1, t_2 are non-negative integers.

$$S = \bigcup_{0 \le j_1 \le t_1, 0 \le j_2 \le t_2} \{x^{i_1} y^{i_2} z^{i_3+j_1} w^{i_4+j_2} : x^{i_1} y^{i_2} z^{i_3} w^{i_4} \text{ is a monomial of } f^m\},$$

$$M = \{\text{monomials of } x^{i_1} y^{i_2} z^{i_3} w^{i_4} f : x^{i_1} y^{i_2} z^{i_3} w^{i_4} \in S\}.$$

From [5], we know that these polynomials can be found by lattice reduction if

$$X^{s_1} Y^{s_2} Z^{s_3} V^{s_4} < W^s, \tag{3}$$

where $s_j = \sum_{x^{i_1} y^{i_2} z^{i_3} v^{i_4} \in M \setminus S} i_j$ for $j = 1, \ldots, 4$, and
$W = ||f(xX, yY, zZ)||_\infty \ge \max\{N_1 Y, n_2 X\} = N^{1+\gamma}$. One can check
$s_1 = \frac{3}{2} t_1 t_2 m^2 + t_1 m^3 + t_2 m^3 + \frac{1}{4} m^4 + o(m^4)$,
$s_2 = \frac{3}{2} t_1 t_2 + \frac{4}{3} t_1 m^3 + \frac{4}{3} t_2 m^3 + \frac{11}{24} m^4 + o(m^4)$,
$s_3 = t_1^2 t_2 m + \frac{3}{2} t_1 t_2 m^2 + \frac{3}{4} t_1^2 m^2 + \frac{2}{3} t_1 m^3 + \frac{2}{3} t_2 m^3 + \frac{1}{6} m^4 + o(m^4)$,
$s_4 = t_1 t_2^2 m + \frac{3}{2} t_1 t_2 m^2 + \frac{3}{4} t_2^2 m^2 + \frac{2}{3} t_1 m^3 + \frac{2}{3} t_2 m^3 + \frac{1}{6} m^4 + o(m^4)$ and
$s = t_1 t_2 m^2 + \frac{1}{2} t_1 m^3 + \frac{1}{2} t_2 m^3 + \frac{1}{8} m^4 + o(m^4)$.

For a given integer m, let $t_1 = \tau_1 m$ and $t_2 = \tau_2 m$. Then substituting the values of X, Y, Z, V and lower bound of W in 3 and neglecting the lower order terms of s_j we get the required condition. □

When $\alpha, \gamma_1, \gamma_2$ are available, we need to take the partial derivative of h with respect to τ_1, τ_2 and equate each of them to 0 to get non-negative solutions of τ_1, τ_2. Given any pair of such non-negative solutions, if h is less than zero, then N_1, N_2 can be factored in polynomial time. When $\gamma_1 = \gamma_2 = \gamma$, then one can consider t_1 to be equal to t_2. In that case we get the following result.

Corollary 3. *Let $N_1 = p_1 q_1$ and $N_2 = p_2 q_2$, where p_1, q_1, p_2, q_2 are primes. Let $q_1, q_2 \approx N^\alpha$. Consider that a contiguous portion of bits of p_1, p_2 are same at the middle leaving $\gamma \log_2 N$ many MSBs and as well as LSBs. Then under Assumption 1, we can factor both N_1, N_2 if there exists $\tau \ge 0$ for which $2\gamma\tau^3 + (3\alpha + \frac{7}{2}\gamma - 1)\tau^2 + (\frac{14}{3}\alpha + \frac{5}{3}\gamma - 1)\tau + (\frac{17}{24}\alpha + \frac{5}{24}\gamma - \frac{1}{8}) < 0$.*

In Table 4, we present some numerical values of $\alpha, \gamma_1, \gamma_2$ following Theorem 2 for which N_1, N_2 can be factored in polynomial time. It is clear from Table 4 that the requirement, of bits at the middle of p_1, p_2 to be same, is quite high compared to the case presented in Section 2, where we have considered that MSBs and/or LSBs are same. Thus, the kind of lattice based technique we consider in this paper works more efficiently when the bits of p_1, p_2 are same at MSBs and/or LSBs compared to the case when some contiguous bits at the middle are same. Below we present an experimental result in this regard.

Example 8. In this experiment, we consider 850-bit primes p_1 and p_2
6010063291745673411630355586520987527501854123507752031634410338922261325200
7908441224870562785926663459589770176998171796837046320523689364119368753632

α	γ_1	γ_2
0.25	0.019	0.019
0.25	0.010	0.030
0.20	0.061	0.061
0.20	0.052	0.078
0.15	0.146	0.146
0.15	0.137	0.175
0.10	0.277	0.277
0.10	0.268	0.314

Table 4. Values of $\alpha, \gamma_1, \gamma_2$ for which N_1, N_2 can be factored in polynomial time.

4043967528101378987325022248122037708305617873964005637745986435542948539825
7581885621415132927137653573 and
5984825641870931585823382220962926344220670532554403933352105675571066672703603259730323516347307682331134375921535484093155453663109803357469969320260504
32396316389068925325234493940473852769406714934240353469418641140783489390073
2303380146620012528421339.
Note that, p_1, p_2 share middle 504 many bits (leaving 177 bits from the least significant side). Further, q_1, q_2 are 150-bit primes
1038476608131498405684472704928794724111541861 and
1281887704228770097092001008195142506836912053 respectively. Given N_1, N_2, with only the implicit information, we can factorize both of them in poly$(\log N)$ time. We use lattice of dimension 70 (parameters $m = 1, t_1 = 1, t_2 = 1$) and the lattice reduction takes 175.83 seconds. □

4 Conclusion

In this paper we have studied poly$(\log N)$ time factorization strategy when two integers N_1, N_2 (of same size) are given where $N_1 = p_1 q_1$ and $N_2 = p_2 q_2$ and p_1, p_2 share certain amount of LSBs and/or MSBs taken together. We also study the case when p_1, p_2 share some bits at the middle. Our results extend the idea presented in [11]. Further, for the LSB case, we obtain better results than [11] under certain conditions.

However, the technique presented here can not immediately be extended to the following generalized problems studied in [11].

1. Our strategy is hard to extend with more than two integers, i.e., when one considers that $N_1, N_2, \ldots, N_k$ are available such that $N_1 = p_1 q_1, N_2 = p_2 q_2, \ldots, N_k = p_k q_k$ and all the p_i's share t many LSBs and/or MSBs. This is because, the idea presented in Theorem 1 exploits the term $p_1 - p_2$. It is not clear how to extend the idea when $p_i - p_j$ is considered in general.
2. Our idea does not work well when one considers that p_i, q_i are of same bit-size. The bound presented in Theorem 1 does not provide encouraging results as q_i increases. Even if we consider that some information regarding q_i's are available, that also does not help much.

This is because, under such information the structure of the polynomial f in Theorem 1 changes and more number of monomials arrive, that prevents to achieve a good bound.

3. Referring to Theorems 1, 2 together, one may be tempted to consider the case that a few contiguous intervals of bits are same in p_1, p_2. However, in such a scenario, the polynomials contain increased number of variables as well as monomials. Thus, encouraging results cannot be obtained in this method.

Settling these issues are left open for future research.

Still, we like to point out that the problem of factorization with two integers N_1, N_2 in this domain is harder than the case of factorization with more than two integers $N_1, N_2, \ldots, N_k$. For the case of two integers, we present results that could not be achieved earlier.

The strategy presented in [11] used lattice dimension 2 only for the case with two integers N_1, N_2 and it is also not immediate whether similar technique can be extended with higher lattice dimensions. However, our strategy allows to exploit larger lattice dimensions and thus during experiments we get better results as lattice dimension increases.

Acknowledgments: The authors like to thank Prof. A. May for important comments on an earlier draft of this paper 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. T. Apostol. An Introduction to Analytic Number Theory. Narosa Publishing House, 1979.
2. D. Coppersmith. Finding a Small Root of a Bivariate Integer Equations. Eurocrypt 1996, LNCS 1070, pp. 178–189, 1996.
3. J. -S. Coron. Finding Small Roots of Bivariate Integer Equations Revisited. Eurocrypt 2004, LNCS 3027, pp. 492–505, 2004.
4. D. Cox, J. Little, D. O'Shea. Ideals, Varieties, and Algorithms. Second Edition, Springer-Verlag, 1998.
5. E. Jochemsz and A. May. A Strategy for Finding Roots of Multivariate Polynomials with new Applications in Attacking RSA Variants. Asiacrypt 2006, LNCS 4284, pp. 267–282, 2006.
6. E. Jochemsz. Cryptanalysis of RSA Variants Using Small Roots of Polynomials. Ph. D. thesis, Technische Universiteit Eindhoven, 2007.
7. H. W. Jr. Lenstra. Factoring Integers with Elliptic Curves. Annals of Mathematics, 126, pp. 649–673, 1987.
8. A. K. Lenstra and H. W. Jr. Lenstra. The Development of the Number Field Sieve. Springer-Verlag, 1993.
9. A. K. Lenstra, H. W. Lenstra and L. Lovász. Factoring Polynomials with Rational Coefficients. Mathematische Annalen, 261:513–534, 1982.
10. E. Jochemsz and A. May. A Polynomial Time Attack on RSA with Private CRT-Exponents Smaller Than $N^{0.073}$. Crypto 2007, LNCS 4622, pp. 395–411, 2007.
11. A. May and M. Ritzenhofen. Implicit Factoring: On Polynomial Time Factoring Given Only an Implicit Hint. PKC 2009. Available at http://www.cits.rub.de/personen/may.html [last accessed 20 March, 2009].
12. 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 20 March, 2009].
13. C. Pomerance. The Quadratic Sieve Factoring Algorithm. Eurocrypt 1984, LNCS 209, pp. 169–182, 1985.
14. R. Rivest and A. Shamir. Efficient Factoring Based on Partial Information. Eurocrypt 1985, LNCS 219, pp. 31–34, 1986.
15. P. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. 35th Annual Symposium on Foundations of Computer Science, IEEE Computer Science Press, pp. 124–134, 1994.