Implicit Factoring with Shared Most Significant and Middle Bits

Jean-Charles Faugère, Raphaël Marinier, Guénaël Renault

Implicit Factoring with Shared Most Significant and Middle Bits

Jean-Charles Faugère, Raphaël Marinier, and Guénaël Renault

UPMC, Université Paris 06, LIP6
INRIA, Centre Paris-Rocquencourt, SALSA Project-team
CNRS, UMR 7606, LIP6
4, place Jussieu
75252 Paris, Cedex 5, France
jean-charles.faugere@inria.fr, raphael.marinier@polytechnique.edu,
guenael.renault@lip6.fr

Abstract. We study the problem of integer factoring given *implicit* information of a special kind. The problem is as follows: let $N_1 = p_1 q_1$ and $N_2 = p_2 q_2$ be two RSA moduli of same bit-size, where q_1, q_2 are α-bit primes. We are given the *implicit* information that p_1 and p_2 share t most significant bits. We present a novel and rigorous lattice-based method that leads to the factorization of N_1 and N_2 in polynomial time as soon as $t \geq 2\alpha + 3$. Subsequently, we heuristically generalize the method to k RSA moduli $N_i = p_i q_i$ where the p_i's all share t most significant bits (MSBs) and obtain an improved bound on t that converges to $t \geq \alpha + 3.55\ldots$ as k tends to infinity. We study also the case where the k factors p_i's share t contiguous bits in the middle and find a bound that converges to $2\alpha + 3$ when k tends to infinity. This paper extends the work of May and Ritzenhofen in [9], where similar results were obtained when the p_i's share least significant bits (LSBs). In [15], Sarkar and Maitra describe an alternative but heuristic method for only two RSA moduli, when the p_i's share LSBs and/or MSBs, or bits in the middle. In the case of shared MSBs or bits in the middle and two RSA moduli, they get better experimental results in some cases, but we use much lower (at least 23 times lower) lattice dimensions and so we obtain a great speedup (at least 10^3 faster). Our results rely on the following surprisingly simple algebraic relation in which the shared MSBs of p_1 and p_2 cancel out: $q_1 N_2 - q_2 N_1 = q_1 q_2 (p_2 - p_1)$. This relation allows us to build a lattice whose shortest vector yields the factorization of the N_i's.

Keywords: implicit factorization, lattices, RSA

1 Introduction

Efficient factorization of large integers is one of the most fundamental problem of Algorithmic Number Theory, and has fascinated mathematicians for centuries. It has been particularly intensively studied over the past 35 years, all the more that efficient factorization leads immediately to an attack of the RSA Cryptosystem. In the 1970's, the first general-purpose sub-exponential algorithm for factoring was developed by Morrison and Brillhart in [11] (improving a method described for the first time in [7]), using

continued fraction techniques. Several faster general-purpose algorithms have been proposed over the past years, the most recent and efficient being the general number field sieve (GNFS) [8], proposed in 1993. It is not known whether factoring integers can be done in polynomial time on a classical Turing machine. On quantum machines, Shor's algorithm [16] allows polynomial-time factoring of integers. However, it is still an open question whether a capable-enough quantum computer can be built.

At the same time, the problem of factoring integers given additional information about their factors has been studied since 1985. In [14], Rivest and Shamir showed that $N = pq$ of bit-size n and with balanced factors ($\log_2(p) \approx \log_2(q) \approx \frac{n}{2}$) can be factored in polynomial time as soon as we have access to an *oracle* that returns the $\frac{n}{3}$ most significant bits (MSBs) of p. Beyond its theoretical interest, the motivation behind this is mostly of cryptographic nature. In fact, during an attack of an RSA-encrypted exchange, the cryptanalyst may have access to additional information beyond the RSA public parameters (e,N), that may be gained for instance through side-channel attacks revealing some of the bits of the secret factors. Besides, some variations of the RSA Cryptosystem purposely leak some of the secret bits (for instance, [17]). In 1996, Rivest and Shamir's results were improved in [2] by Coppersmith applying lattice-based methods to the problem of finding small integer roots of bivariate integer polynomials (the now so-called *Coppersmith's method*). It requires only half of the most significant bits of p to be known to the cryptanalyst (that is $\frac{n}{4}$).

In PKC 2009, May and Ritzenhofen [9] significantly reduced the power of the oracle. Given an RSA modulus $N_1 = p_1 q_1$, they allow the oracle to output a new and different RSA modulus $N_2 = p_2 q_2$ such that p_1 and p_2 share at least t least significant bits (LSBs). Note that the additional information here is only *implicit*: the attacker does not know the actual value of the t least significant bits of the p_i's, he only knows that p_1 and p_2 share them. In the rest of the paper, we will refer to this problem as the problem of *implicit factoring*. When q_1 and q_2 are α-bit primes, May and Ritzenhofen's lattice-based method rigorously finds in quadratic time the factorization of N_1 and N_2 when $t \geq 2\alpha + 3$. Besides, their technique heuristically generalizes to $k-1$ oracle queries that give access to k different RSA moduli $N_i = p_i q_i$ with all the p_i's sharing t least significant bits. With $k-1$ queries the bound on t improves to: $t \geq \frac{k}{k-1}\alpha$. Note that these results are of interest for unbalanced RSA moduli: for instance, if $N_1 = p_1 q_1, N_2 = p_2 q_2$ are 1000-bit RSA moduli and the q_i's are 200-bit primes, knowing that p_1 and p_2 share at least 403 least significant bits out of 800 is enough to factorize N_1 and N_2 in polynomial time. Note also that the method absolutely requires that the shared bits be the least significant ones. They finally apply their method to factorize k n-bit balanced RSA moduli $N_i = p_i q_i$ under some conditions and with an additional exhaustive search of $2^{\frac{n}{4}}$.

Very recently, in [15], Sarkar and Maitra applied Coppersmith and Gröbner-basis techniques on the problem of implicit factoring, and improved heuristically the bounds in some of the cases. Contrary to [9], their method applies when either (or both) LSBs or MSBs of p_1, p_2 are shared (or when bits in the middle are shared). Namely, in the case of shared LSBs they obtain better theoretical bounds on t than [9] as soon as $\alpha \geq 0.266n$. Besides, their experiments often perform better than their theoretical bounds, and they improve in practice the bound on t of [9] when $\alpha \geq 0.21n$. Note finally that their bounds

are very similar in the two cases of shared MSBs and shared LSBs. Readers interested in getting their precise bounds may refer to their paper [15].

Unfortunately, Sarkar and Maitra's method is heuristic even in the case of two RSA moduli, and does not generalize to $k \geq 3$ RSA moduli. In fact, when the p_i's share MSBs and/or LSBs, their method consists in building a polynomial f_1 in three variables, whose roots are $(q_2+1, q_1, \frac{p_1-p_2}{2^\gamma})$, where γ is the number of shared LSBs between p_1 and p_2. That is, $\frac{p_1-p_2}{2^\gamma}$ represents the part of p_1-p_2 where the shared bits do not cancel out. To find the integer roots of f_1, they use the Coppersmith-like technique of [5] which consists in computing two (or more) new polynomials $f_2, f_3, \ldots$ sharing the same roots as f_1. If the variety defined by $f_1, f_2, f_3, \ldots$ is 0-dimensional, then the roots can be easily recovered computing resultants or Gröbner basis. However, with an input polynomial with more than two variables, the method is heuristic: there is no guarantee for the polynomials $f_1, f_2, f_3, \ldots$ to define a 0-dimensional variety. We reproduced the results of Sarkar and Maitra and we observed that $f_1, f_2, f_3, \ldots$ almost never defined a 0-dimensional variety. They observed however that it was possible to recover the roots of the polynomials directly by looking at the coefficients of the polynomials in the Gröbner basis of the ideal generated by the f_i's, even when the ideal was of positive dimension. The assumption on which their work relies is that it will always be possible. For instance, in the case of shared MSBs between p_1 and p_2, they found in their experiments that the Gröbner basis contained a polynomial multiple of $x - \frac{q_2}{q_1}y - 1$ whose coefficients lead immediately to the factorization of N_1 and N_2. They support their assumption by experimental data: in most cases their experiments perform better than their theoretical bounds. It seems nevertheless that their assumption is not fully understood.

Our contribution consists of a novel and rigorous lattice-based method that address the implicit factoring problem when p_1 and p_2 share *most* significant bits. That is, we obtained an analog of May and Ritzenhofen's results for shared MSBs, and our method is rigorous contrary to the work of Sarkar and Maitra in [15]. Namely, let $N_1 = p_1 q_1$ and $N_2 = p_2 q_2$ be two RSA moduli of same bit-size n. If q_1, q_2 are α-bit primes and p_1, p_2 share t most significant bits, our method provably factorizes N_1 and N_2 as soon as $t \geq 2\alpha + 3$ (which is the same as the bound on t for least significant bits in [9]). This is the first rigorous bound on t when p_1 and p_2 share most significant bits. From this method, we deduce a new heuristic lattice-based for the case when p_1 and p_2 share t bits in the middle. Moreover, contrary to [15], these methods heuristically generalize to an arbitrary number k of RSA moduli and do not depend on the position of the shared bits in the middle, allowing us to factorize k RSA moduli as soon as $t \geq \frac{k}{k-1}\alpha + 6$ (resp. $t \geq \frac{2k}{k-1}\alpha + 7$) most significant bits (resp. bits in the middle) are shared between the p_i's (more precise bounds are stated later in this paper). A summary of the comparison of our method with the methods in [9] and [15] can be found in table 1.

Let's give the main idea of our method with 2 RSA moduli in the case of shared MSB's. Consider the lattice L spanned by the row vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ of the following matrix:

$$\begin{pmatrix} K & 0 & N_2 \\ 0 & K & -N_1 \end{pmatrix} \quad \text{where } K = \lfloor 2^{n-t+\frac{1}{2}} \rfloor$$

Table 1: Comparison of our results against the results of [9] and [15] with k RSA moduli

	May, Ritzenhofen's Results [9]	Sarkar, Maitra's Results [15]	Our results
$k=2$	When p_1, p_2 share t LSBs: rigorous bound of $t \geq 2\alpha+3$ using 2-dimensional lattices of $\mathbb{Z}^2$.	When p_1, p_2 share either t LSBs or MSBs: heuristic bound better than $t \geq 2\alpha+3$ when $\alpha \geq 0.266n$, and experimentally better when $\alpha \geq 0.21n$. In the case of t shared bits in the middle, better bound than $t \geq 4\alpha+7$ but depending on the position of the shared bits. Using 46-dimensional lattices of $\mathbb{Z}^{46}$	When p_1, p_2 share t MSBs: rigorous bound of $t \geq 2\alpha+3$ using 2-dimensional lattices of $\mathbb{Z}^3$. In the case of t bits shared in the middle: heuristic bound of $t \geq 4\alpha+7$ using 3-dimensional lattices of $\mathbb{Z}^3$.
$k \geq 3$	When the p_i's all share t LSBs: heuristic bound of $t \geq \frac{k}{k-1}\alpha$ using k-dimensional lattices of $\mathbb{Z}^k$.	Cannot be directly applied.	When the p_i's all share t MSBs (resp. bits in the middle): heuristic bound of $t \geq \frac{k}{k-1}\alpha+\delta_k$ (resp. $t \geq \frac{2k}{k-1}\alpha+\delta_k$), with $\delta_k \leq 6$ (resp. ≤ 7) and using k-dimensional ($\frac{k(k+1)}{2}$-dimensional) lattices of $\mathbb{Z}^{\frac{k(k+1)}{2}}$.

Consider also the following vector in L:

$$\mathbf{v_0} = q_1\mathbf{v_1} + q_2\mathbf{v_2} = (q_1K, q_2K, q_1q_2(p_2-p_1))$$

The key observation is that the t shared significant bits of p_1 and p_2 cancel out in the algebraic relation $q_1N_2 - q_2N_1 = q_1q_2(p_2-p_1)$. Furthermore, we choose K in order to force the coefficients of a shortest vector of L on the basis $(\mathbf{v_1}, \mathbf{v_2})$ to be of the order of $2^\alpha \approx q_1 \approx q_2$. We prove in the next section that $\mathbf{v_0}$ is indeed a shortest vector of L (thus N_1 and N_2 can be factored in polynomial time) as soon as $t \geq 2\alpha+3$. Besides, we generalized this construction to an arbitrary number of k RSA moduli such that a small vector of the lattice harnesses the same algebraic relation, and to shared middle bits. However, the generalized constructions in both cases become heuristic: we use the Gaussian heuristic to find a condition on t for this vector to be a shortest of the lattice.

Applications of implicit factoring have not yet been extensively studied, and we believe that they will develop. The introduction of [9] gives some ideas for possible applications. They include destructive applications with malicious manipulation of public key generators, as well as possibly constructive ones. Indeed, our work shows that when $t \geq 2\alpha+3$, it is as hard to factorize $N_1 = p_1q_1$, as generating $N_2 = p_2q_2$ with p_2 sharing t most significant bits with p_1. This problem could form the basis of a cryptographic primitive.

Throughout this paper, we heavily use common results on euclidean lattice. A summary of these results can be found in appendix A. The paper is organized as follows. In

section 2, we present our rigorous method in the case of shared MSB's and two RSA moduli, we generalize it to k RSA moduli in section 3. In section 4, we present our method in the case of shared bits in the middle. Finally, in section 5 we present our experiments that strongly support the assumption we made in the case of k RSA moduli and of shared middle bits.

2 Implicit Factoring of Two RSA Moduli with Shared MSBs

In this section, we study the problem of factoring two n-bit RSA moduli: $N_1 = p_1 q_1$ and $N_2 = p_2 q_2$, where q_1 and q_2 are α-bit primes, given only the implicit hint that p_1 and p_2 share t most significant bits (MSBs) that are *unknown* to us. We will show that N_1 and N_2 can be factored in quadratic time as soon as $t \geq 2\alpha + 3$. By saying that the primes p_1, p_2 of maximal bit-size $n - \alpha + 1$ share t MSBs, we really mean that $|p_1 - p_2| \leq 2^{n-\alpha-t+1}$.

Let's consider the lattice L spanned by the row vectors (denoted by $\mathbf{v_1}$ and $\mathbf{v_2}$) of the following matrix:

$$M = \begin{pmatrix} K & 0 & N_2 \\ 0 & K & -N_1 \end{pmatrix} \quad \text{where } K = \lfloor 2^{n-t+\frac{1}{2}} \rfloor$$

We have the following immediate lemma that makes our method work:

Lemma 1. *Let $\mathbf{v_0}$ be the vector of L defined by $\mathbf{v_0} = q_1\mathbf{v_1} + q_2\mathbf{v_2}$. Then $\mathbf{v_0}$ can be rewritten as $\mathbf{v_0} = (q_1 K, q_2 K, q_1 q_2 (p_2 - p_1))$.*

Note that the shared MSBs of p_1 and p_2 cancel each other out in the difference $p_2 - p_1$. Each of the coefficients of $\mathbf{v_0}$ are thus integers of roughly $(n + \alpha - t)$ bits. Provided that t is sufficiently large, $\pm\mathbf{v_0}$ may be a shortest vector of L that can be found using Lagrange reduction on L. Moreover, note that as soon as we retrieve $\mathbf{v_0}$ from L, factoring N_1 and N_2 is easily done by dividing the first two coordinates of $\mathbf{v_0}$ by K (which can be done in quadratic time in n). Proving that $\mathbf{v_0}$ is a shortest vector of L under some conditions on t is therefore sufficient to factorize N_1 and N_2.

We first give an intuition on the bound on t that we can expect, and we give after that a proof that $\pm\mathbf{v_0}$ is indeed the shortest vector of L under a similar condition.

The volume of L is the square root of the determinant of the Gramian matrix of L given by $MM^t = \begin{pmatrix} K^2 + N_2^2 & -N_1 N_2 \\ -N_1 N_2 & K^2 + N_1^2 \end{pmatrix}$. That is, $vol(L) = K\sqrt{N_1^2 + N_2^2 + K^2}$ which can be approximated by 2^{2n-t} because $K^2 \approx 2^{2(n-t)}$ is small compared to the $N_i^2 \approx 2^{2n}$. The norm of $\mathbf{v_0}$ is approximately $2^{n+\alpha-t}$, because each of its coefficients have roughly $n + \alpha - t$ bits. If $\mathbf{v_0}$ is a shortest vector of L, it must be smaller than the Minkowski bound applied to L: $2^{n+\alpha-t} \approx \|\mathbf{v_0}\| \leq \sqrt{2}\,\mathrm{Vol}(L)^{1/2} \approx 2^{n-t/2}$, which happens when $t \geq 2\alpha$. The following lemma affirms that $\mathbf{v_0}$ is indeed a shortest vector of L under a similar condition on t.

Lemma 2. *Let L be the lattice generated by the row vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ of M and let $\mathbf{v_0} = q_1\mathbf{v_1} + q_2\mathbf{v_2} = (q_1 K, q_2 K, q_1 q_2 (p_2 - p_1))$ as defined in Lemma 1. The vector $\pm\mathbf{v_0}$ is the shortest vector of the lattice L as soon as $t \geq 2\alpha + 3$.*

Proof. Let $(\mathbf{b_1}, \mathbf{b_2})$ be the resulting basis from the Lagrange reduction on L. This reduced basis verifies $\|\mathbf{b_1}\| = \lambda_1(L), \|\mathbf{b_2}\| = \lambda_2(L)$, and, by Hadamard's inequality one have: $\|\mathbf{b_1}\|\|\mathbf{b_2}\| \geq \text{Vol}(L)$. As $\mathbf{v_0}$ is in the lattice, $\|\mathbf{b_1}\| = \lambda_1(L) \leq \|\mathbf{v_0}\|$. Hence we get $\|\mathbf{b_2}\| \geq \frac{\text{Vol}(L)}{\|\mathbf{v_0}\|}$. Moreover, if $\mathbf{v_0}$ is strictly shorter that $\mathbf{b_2}$, $\mathbf{v_0}$ is a multiple of $\mathbf{b_1}$; for otherwise $\mathbf{b_2}$ would not be the second minimum of the lattice. In this case, $\mathbf{v_0} = a\mathbf{b_1} = a(b\mathbf{v_1} + c\mathbf{v_2}), a, b, c \in \mathbb{Z}$, and looking at the first two coefficients of $\mathbf{v_0}$, we get that $ab = q_1$ and $ac = q_2$. Since the q_i's are prime, we conclude that $a = \pm 1$, that is, $\mathbf{v_0} = \pm\mathbf{b_1}$. Using the previous inequality, a condition for $\mathbf{v_0}$ to be strictly shorter than $\mathbf{b_2}$ is:

$$\|\mathbf{v_0}\|^2 < \text{Vol}(L) \tag{1}$$

Let's upper-bound the norm of $\mathbf{v_0}$ and lower-bound $\text{Vol}(L)$. We first provide simple bounds that proves the lemma when $t \geq 2\alpha + 4$ and derive secondly tighter bounds that require only $t \geq 2\alpha + 3$.

The p_i's have at most $n - \alpha + 1$ bits, and they share their t most significant bits so $|p_2 - p_1| \leq 2^{n-\alpha+1-t}$. We thus have the inequality $\|\mathbf{v_0}\|^2 \leq 2^{2(n-t)+1}(q_1^2 + q_2^2) + q_1^2 q_2^2 (p_1 - p_2)^2$ which implies

$$\|\mathbf{v_0}\|^2 \leq 2^{2(n+\alpha-t)+2} + 2^{2(\alpha+n+1-t)} \leq 2^{2(n+\alpha-t)+3} \tag{2}$$

We can lower-bound the volume of L, using that $N_1, N_2 \geq 2^{n-1}$ and $K^2 \geq 2^{2(n-t)}$:

$$\text{Vol}(L)^2 = K^2(N_1^2 + N_2^2 + 2^{2(n-t)}) > 2^{4n-2t-1} \tag{3}$$

Using inequalities (2) and (3), the inequality (1) is true provided that: $2^{2(n+\alpha-t)+3} \leq 2^{2n-t-\frac{1}{2}}$ which is equivalent to (as t and α are an integers):

$$t \geq 2\alpha + 4 \tag{4}$$

We have thus proved the lemma under condition (4). We now refine the bounds on $\|\mathbf{v_0}\|$ and $\text{Vol}(L)$ in order to prove the tight case.

The integers q_1 and q_2 are α-bit primes, therefore $q_i \leq 2^\alpha - 1,\ (i = 1, 2)$. Define ε_1 by $2^\alpha - 1 = 2^{\alpha - \varepsilon_1}$. We get $q_i^2 \leq 2^{2\alpha - 2\varepsilon_1}, (i = 1, 2)$. Moreover, since $K = \lfloor 2^{n-t+\frac{1}{2}} \rfloor$, we have $K^2 \leq 2^{2(n-t)+1}$. From these inequalities, we can upper-bound $K^2 q_i^2$

$$K^2 q_i^2 \leq 2^{2(n-t+\alpha)+1-2\varepsilon_1},\ (i = 1, 2) \tag{5}$$

The p_i's have at most $n - \alpha + 1$ bits and they share t bits, so $(p_2 - p_1)^2 \leq 2^{2(n-\alpha+1-t)}$. Thus, using the upper-bound on the q_i^2, we have

$$q_1^2 q_2^2 (p_2 - p_1)^2 \leq 2^{2(n-t+\alpha+1-2\varepsilon_1)} \tag{6}$$

We can finally bound $\|\mathbf{v_0}\|^2 = K^2(q_1^2 + q_2^2) + q_1^2 q_2^2 (p_2 - p_1)^2$ using (5) and (6):

$$\|\mathbf{v_0}\|^2 \leq 2^{2(n+\alpha-t)+2-2\varepsilon_1} + 2^{2(n-t+\alpha+1-2\varepsilon_1)} \leq 2^{2(n+\alpha-t)+3-\varepsilon_1} \tag{7}$$

Let's now define ε_2 by the equality $2^{n-t+1/2}-1 = 2^{n-t+1/2-\varepsilon_2}$. We have that $K = \lfloor 2^{n-t+\frac{1}{2}} \rfloor \geq 2^{n-t+1/2-\varepsilon_2}$ and $N_i^2 \geq 2^{2n-2}$, we can therefore lower-bound $\mathrm{Vol}(L)^2$:

$$\mathrm{Vol}(L)^2 = K^2(N_1^2+N_2^2+2^{2(n-t)}) > K^2(N_1^2+N_2^2) \geq 2^{4n-2t-2\varepsilon_2} \tag{8}$$

Using the inequalities (7) and (8), the condition (1) is true under the new condition $2^{2(n+\alpha-t)+3-\varepsilon_1} \leq 2^{2n-t-\varepsilon_2}$ which is equivalent to $t \geq 2\alpha+3+\varepsilon_2-\varepsilon_1$.

Since $\varepsilon_1 = \log_2(\frac{1}{1-\frac{1}{2^\alpha}})$, $\varepsilon_2 = \log_2(\frac{1}{1-\frac{1}{2^{n-t+\frac{1}{2}}}})$ and $\alpha \leq n-t$, we have $\varepsilon_2 \leq \varepsilon_1$ and the result follows.

From the preceding Lemmas 1 and 2, one can deduce the following result.

Theorem 1. *Let $N_1 = p_1q_1, N_2 = p_2q_2$ be two n-bit RSA moduli, where the q_i's are α-bit primes and the p_i's are primes that share t most significant bits. If $t \geq 2\alpha+3$, then N_1 and N_2 can be factored in quadratic time in n.*

Proof. Let L be the lattice generated by $\mathbf{v_1}$ and $\mathbf{v_2}$ as above. Since the norms of $\mathbf{v_1}$ and $\mathbf{v_2}$ are bounded by 2^{n+1}, computing the reduced basis $(\mathbf{b_1}, \mathbf{b_2})$ takes a quadratic time in n. By Lemma 2 we know that $\mathbf{b_0} = \pm\mathbf{v_0}$ as soon as $t \geq 2\alpha+3$. The factorization of N_1 of N_2 follows from the description of $\mathbf{v_0}$ given by the lemma 1.

Remark 1. For our analysis, the value $K = \lfloor 2^{n-t+\frac{1}{2}} \rfloor$ is indeed the best possible value. If we use $K = \lfloor 2^{n-t+\gamma} \rfloor$, we obtain the bound $t \geq 2\alpha + f(\gamma)$ with $f(\gamma) = \frac{3}{2} - \gamma + \log_2(2+2^{2\gamma})$. The minimum of f is 3 and is attained in $\gamma = \frac{1}{2}$.

3 Implicit Factoring of k RSA Moduli with Shared MSBs

The construction of the lattice for 2 RSA moduli naturally generalizes to an arbitrary number k of moduli. Similarly, we show that a short vector $\mathbf{v_0}$ of the lattice allows us to recover the factorization of the N_i's. This vector takes advantage of the relations $q_iN_j - q_jN_i = q_iq_j(p_j-p_i)$ for all $i, j \in \{1,\ldots,k\}$. However, we were unable to prove that $\mathbf{v_0}$ is a shortest vector of the lattice. Therefore, our method relies on the Gaussian heuristic to estimate the conditions under which $\mathbf{v_0}$ should be a shortest vector of the lattice. Experimental data in section 5 confirms that this heuristic is valid in nearly all the cases.

In this section, we are given k RSA moduli of n bits $N_1 = p_1q_1, \ldots, N_k = p_kq_k$ where the q_i's are α-bit primes and the p_i's are primes that all share t most significant bits.

Let us construct a matrix M whose row vectors will form a basis of a lattice L; this matrix will have k rows and $k + \binom{k}{2} = \frac{k(k+1)}{2}$ columns. Denote by $s_1, \ldots, s_m$ with $m = \binom{k}{2}$ all the subsets of cardinality 2 of $\{1,2,\ldots,k\}$. To each of the s_i's, associate a column vector $\mathbf{c_i}$ of size k the following way. Let a,b be the two elements of s_i, with $a < b$. We set the a-th element of $\mathbf{c_i}$ to N_b, the b-th element of $\mathbf{c_i}$ to $-N_a$, and all other elements to zero. Finally, one forms M by concatenating column-wise the matrix $KI_{k\times k}$, where $I_{k\times k}$ is the identity matrix of size k, along with the matrix C_m composed by the m column vectors $\mathbf{c_1}, \ldots, \mathbf{c_m}$. K is chosen to be $\lfloor 2^{n-t+\frac{1}{2}} \rfloor$. We will call $\mathbf{v_1}, \ldots, \mathbf{v_k}$ the row vectors of M.

To make things more concrete, consider the example of $k = 4$. Up to a reordering of the columns (that changes nothing to the upcoming analysis),

$$M = \begin{pmatrix} K & 0 & 0 & 0 & N_2 & N_3 & N_4 & 0 & 0 & 0 \\ 0 & K & 0 & 0 & -N_1 & 0 & 0 & N_3 & N_4 & 0 \\ 0 & 0 & K & 0 & 0 & -N_1 & 0 & -N_2 & 0 & N_4 \\ 0 & 0 & 0 & K & 0 & 0 & -N_1 & 0 & -N_2 & -N_3 \end{pmatrix} \text{ where } K = \lfloor 2^{n-t+\frac{1}{2}} \rfloor \qquad (9)$$

Notice that the columns $k+1$ to $k+m$ correspond to all the 2-subsets of $\{1,2,3,4\}$.

Similarly to the case of 2 RSA moduli (lemma 1), L contains a short vector that allows us to factorize all the N_i's:

Lemma 3. *Let $\mathbf{v_0}$ be the vector of L defined by $\mathbf{v_0} = \sum_{i=1}^{k} q_i \mathbf{v_i}$. Then $\mathbf{v_0}$ can be rewritten as follows:*

$$\mathbf{v_0} = (q_1 K, \ldots, q_k K, \underbrace{\ldots, q_a q_b (p_b - p_a), \ldots}_{\forall \{a,b\} \subset \{1,\ldots,k\}})$$

Proof. For $1 \leq i \leq m$, let a, b be such that $s_i = \{a, b\}$ and $a < b$. By the construction of the $\mathbf{c_i}$'s, we get that the $(k+i)$-th coordinate of $\mathbf{v_0}$ is equal to $q_a N_b - q_b N_a = q_a q_b (p_b - p_a)$. □

Remark that $\mathbf{v_0}$ is short because its m last coordinates harness the cancellation of the t most significant bits between the p_i's. Retrieving $\pm\mathbf{v_0}$ from L leads immediately to the factorization of all the N_i's, dividing its first k coordinates by K.

Assumption 1. *If $\pm\mathbf{v_0}$ is shorter than the Gaussian heuristic $\lambda_1(L) \approx \sqrt{\frac{d}{2\pi e}} \mathrm{Vol}(L)^{\frac{1}{d}}$ applied to the d-dimensional lattice L then it is a shortest vector of L.*

This assumption is supported by experimental data in the section 5. We found it to be almost always true in practice. This condition can be seen as an analog of condition 1 of section 2 in the case of two RSA moduli.

Let's derive a bound on t so that $\mathbf{v_0}$ is smaller than the Gaussian heuristic applied to L. The norm of $\mathbf{v_0}$ can be computed and upper-bounded easily: $\|\mathbf{v_0}\|^2 = K^2 \left(\sum_{i=1}^{k} q_i^2\right) + \sum_{\{i,j\} \subset \{1,\ldots,k\}} q_i^2 q_j^2 (p_i - p_j)^2 \leq k^2 2^{2(n+\alpha-t)+1}$. Computing the volume of L is a bit more involved, we refer to Lemma 5 of appendix B: $\mathrm{Vol}(L) = K \left(K^2 + \sum_{i=1}^{k} N_i^2\right)^{\frac{k-1}{2}}$ and thus $\mathrm{Vol}(L) \geq 2^{n-t} \left(\sqrt{k} 2^{n-1}\right)^{k-1}$

We now seek the condition on t for the norm of $\mathbf{v_0}$ to be smaller than the Gaussian heuristic. Using the two previous inequalities on $\|\mathbf{v_0}\|$ and $\mathrm{Vol}(L)$, we get the stricter condition:

$$k^2 2^{2(n+\alpha-t)+1} \leq \frac{k}{2\pi e} \left(2^{n-t} \left(\sqrt{k} 2^{n-1}\right)^{k-1}\right)^{\frac{2}{k}}$$

Expanding everything and extracting t, we get the following condition:

$$t \geq \frac{k}{k-1}\alpha + 1 + \frac{k}{2(k-1)} \left(2 + \frac{\log_2(k)}{k} + \log_2(\pi e)\right) \qquad (10)$$

When $k \geq 3$, we can derive a simpler and stricter bound on t: $t \geq \frac{k}{k-1}\alpha + 6$

Finally, as $\pm\mathbf{v_0}$ is now the shortest vector of L under Assumption 1, it can be found in time $\mathscr{C}(k, \frac{k(k+1)}{2}, n)$ where $\mathscr{C}(k,s,B)$ is the time to find a shortest vector of a k-dimensional lattice of $\mathbb{Z}^s$ given by B-bit basis vectors. We just proved the following theorem:

Theorem 2. *Let $N_1 = p_1q_1, \ldots, N_k = p_kq_k$ be k n-bit RSA moduli, with the q_i's being α-bit primes, and the p_i's being primes that all share t most significant bits. Under Assumption 1, the N_i's can be factored in time $\mathscr{C}(k, \frac{k(k+1)}{2}, n)$, as soon as t verifies equation* (10)*.*

Remark 2. Note that we can find a shortest vector of the lattice of Theorem 2 using Kannan's algorithm (Theorem 6 in appendix A) in time $\mathscr{O}(\mathscr{P}(n,k)\,k^{\frac{k}{2e}+o(k)})$ where $\mathscr{P}$ is a polynomial. It implies that we can factorize all $N_1, \ldots, N_k$ in time polynomial in n as soon as k is constant or k^k is a polynomial in n. Unfortunately, to the best of our knowledge, this algorithm is not implemented in the computer algebra system Magma [1] on which we implemented the methods. In our experiments, to compute a shortest vector of the lattice, we used instead the Schnorr-Euchner's enumeration algorithm which is well known (see [4,3]) to perform well beyond small dimension (≤ 50) and this step in Magma took less than 1 minute for $k \leq 40$. One may also reduce the lattice using LLL algorithm instead of Schnorr-Euchner's enumeraion. If t is not too close to the bound of Theorem 2, the Gaussian heuristic suggests that the gap (see Definition 1 in the appendix) of the lattice is large, and thus LLL may be able to find a shortest vector of L even in medium dimension (50–200).

Similarly to the case of 2 RSA moduli, $K = \lfloor 2^{n-t+\frac{1}{2}} \rfloor$ is optimal for our analysis. Indeed, if we redo the analysis with $K = \lfloor 2^{n-t+\gamma} \rfloor$, we find that the optimal value for γ is the one that minimizes the function $f_k = \gamma \mapsto \frac{1}{2}k\log_2(k-1+2^{2\gamma-1}) - \gamma$, which is $\gamma = \frac{1}{2}$ regardless of k.

Finally, note that a slightly tighter bound (differing to equation 10 by a small additive constant) may be attained by bounding $\|\mathbf{v_0}\|$ and $\mathrm{Vol}(L)$ more precisely.

4 Implicit Factoring with Shared Bits in the Middle

In this section, we are given k RSA moduli of n bits $N_1 = p_1q_1, \ldots, N_k = p_kq_k$ where the q_i's are α-bit primes and the p_i's are primes that all share t bits from position t_1 to $t_2 = t_1 + t$. More precisely, these RSA moduli all verify:

$$N_i = p_iq_i = (p_{i_2}2^{t_2} + p2^{t_1} + p_{i_0})q_i$$

where p is the integer part shared by all the moduli. Contrary to the LSB case presented in [9] and the MSB one developed in the previous sections, the method we present here is heuristic even when $k = 2$. We sketch now our method when $k = 2$ and present the details on the general result later. When $k = 2$, we have a system of two equations in four variables p_1, q_1, p_2, q_2: $N_1 = p_1q_1 = (p_{1_2}2^{t_2} + p2^{t_1} + p_{1_0})q_1$ and $N_2 = p_2q_2 =$

$(p_{2_2}2^{t_2}+p2^{t_1}+p_{2_0})q_2$. Similarly to the LSB's case (see [9]), this system can be reduced modulo 2^{t_2}. One obtains a system of two equations with 5 variables p, p_{1_0}, p_{2_0}, q_1, q_2:

$$\begin{cases} (p2^{t_1}+p_{1_0})q_1 = N_1 \mod 2^{t_2} \\ (p2^{t_1}+p_{2_0})q_2 = N_2 \mod 2^{t_2} \end{cases} \tag{11}$$

The problem can now be seen as a modular implicit factorization of N_1 and N_2 with shared MSBs. Thus, we adapt the method we proposed in section 2 to the modular case. More precisely, we consider the lattice L defined by the rows of the matrix

$$M = \begin{pmatrix} K & 0 & N_2 \\ 0 & K & -N_1 \\ 0 & 0 & 2^{t_2} \end{pmatrix} \tag{12}$$

Let $\mathbf{v_0}$ be the vector (q_1K, q_2K, r) with r being the unique remainder of $q_1N_2 - q_2N_1$ modulo 2^{t_2} in $]-2^{t_2-1}, 2^{t_2-1}]$. Clearly, $\mathbf{v_0}$ is in L. As in the section 3, we search for a condition on the integer t under which $\pm\mathbf{v_0}$ is the shortest vector in L under Assumption 1 (here, the dimension of the lattice L is 3). The integer K will be set at the end of the analysis.

We have $\|\mathbf{v_0}\|^2 = K^2(q_1^2+q_2^2)+r^2$ and $]-2^{t_2-1}, 2^{t_2-1}] \ni r = q_1N_2 - q_2N_1 \mod 2^{t_2}$ $= q_1q_2(p_{2_0}-p_{1_0}) \mod 2^{t_1+t}$ with $|p_{2_0}-p_{1_0}| \leq 2^{t_1}$ and $q_i \leq 2^{\alpha}$. Thanks to the upper-triangular shape of M, the volume of L is easily computed: $\mathrm{Vol}\, L = K^2 2^{t_2}$. Thus, we can respectively upper-bound and lower-bound $\|\mathbf{v_0}\|^2$ and $\mathrm{Vol}\, L$ by $2^{2\alpha+1}K^2 + 2^{2t_1+4\alpha}$ and $K^2 2^{t_2}$; a condition on t so that $\mathbf{v_0}$ is smaller than the Gaussian heuristic follows: $2^{2\alpha+1}K^2 + 2^{2t_1+4\alpha} \leq \frac{3}{2\pi e}(K^2 2^{t_2})^{\frac{2}{3}}$. This condition is equivalent to

$$t \geq \frac{3}{2}\left[\log_2(2^{2\alpha+1-\frac{2}{3}t_1}K^{\frac{2}{3}} + 2^{\frac{4}{3}t_1+4\alpha}K^{-\frac{4}{3}}) + \log_2(\frac{2\pi e}{3})\right]$$

and the integer value of K which minimizes the right-hand of this inequality is $K = 2^{\alpha+t_1}$. Hence, under Assumption 1, one can factorize N_1, N_2 in polynomial-time as soon as

$$t \geq 4\alpha + \frac{3}{2}(1+\log_2(\pi e)) \tag{13}$$

A stricter and simpler condition on t is: $t \geq 4\alpha + 7$.

We now inspect when Assumption 1 is not verified, that is we study the possible existence of exceptional short vectors in L that are smaller than $\mathbf{v_0}$. These vectors may appear when there exists small coefficients c_1, c_2 ($< 2^{\alpha}$) such that $c_1N_1 - c_2N_2 \mod 2^{t_2}$ is small (say $\approx 2^{t_2-\gamma}$). In particular, to make easier the analysis, we examine the case when the simple vector $\mathbf{v_1}$ defined with $c_1 = c_2 = 1$ is smaller than $\mathbf{v_0}$. The inequality $\|\mathbf{v_1}\|^2 < \|\mathbf{v_0}\|^2$ is equivalent to $t - \gamma < 2\alpha$. So this inequality is possible only for small t and large γ which can be considered as an exception. In our experiments, these exceptional shorts vectors (and, in particular, simple vectors $\mathbf{v_1}$) almost never appear in the $k = 2$ case with t verifying the bound 13.

The method for $k \geq 3$ is a straightforward generalization of the $k = 2$ case by using the results of section 3. Let's consider the lattice L defined by the rows of the matrix M given by

$$M = \left(\begin{array}{c|c} K\,\mathrm{I}_{k\times k} & C_m \\ \hline 0 & 2^{t_2}\,\mathrm{I}_{m\times m} \end{array} \right)$$

where C_m is the matrix defined in section 3 and formed by the concatenation of $m = \binom{k}{2}$ column vectors of k rows and $I_{k\times k}$ (resp. $I_{m\times m}$) is the identity matrix of size $k\times k$ (resp. $m\times m$). Thus, M is a square upper triangular matrix of size $(m+k)\times(m+k)$ and the volume of the $m+k$-dimensional lattice L is easily computed: $\mathrm{Vol}\,L = K^k 2^{mt_2}$.

The vector

$$\mathbf{v_0} = (q_1 K, \ldots, q_k K, \underbrace{\ldots, r_{(a,b)}, \ldots}_{\forall\{a,b\}\subset\{1,\ldots,k\}})$$

with $r_{(a,b)}$ defined as the unique remainder of $q_a q_b(p_b - p_a) = q_a N_a - q_b N_b$ modulo 2^{t_2} in $]-2^{t_2-1}, 2^{t_2-1}]$, is clearly a vector of L. As we do above, we search for a condition on the integer t under which $\pm\mathbf{v_0}$ is the shortest vector in L under Assumption 1. The integer K will be set at the end of the analysis to be optimal.

We have $\|\mathbf{v_0}\|^2 = K^2(q_1^2 + \cdots + q_k^2) + \sum_{\{a,b\}\subset\{1,\ldots,k\}} r_{(a,b)}^2$, that we can bound by $\|\mathbf{v_0}\|^2 \leq k2^{2\alpha}K^2 + m2^{2t_1+4\alpha}$. A condition on t, under Assumption 1, follows:

$$k2^{2\alpha}K^2 + m2^{4\alpha+2t_1} \leq \frac{m+k}{2\pi e}(K^k 2^{mt_2})^{\frac{2}{m+k}}.$$

This condition is equivalent to

$$t \geq \frac{m+k}{2m}\left[\log_2\left(k2^{2\alpha-\frac{2m}{m+k}t_1}K^{\frac{2m}{m+k}} + m2^{4\alpha+\frac{2k}{m+k}t_1}K^{-\frac{2k}{m+k}}\right) + \log_2\left(\frac{2\pi e}{m+k}\right)\right] \qquad (14)$$

The value of K which minimizes the right-hand of this inequality is given by the zero of the derivative of the function $K \mapsto k2^{2\alpha-\frac{2m}{m+k}t_1}K^{\frac{2m}{m+k}} + m2^{4\alpha+\frac{2k}{m+k}t_1}K^{-\frac{2k}{m+k}}$. Actually, K is given by the solution of the equation

$$\frac{2mk}{m+k}2^{2\alpha-\frac{2m}{m+k}t_1}K^{\frac{m-k}{m+k}} = \frac{2km}{m+k}2^{4\alpha+\frac{2k}{m+k}t_1}K^{-\frac{m+3k}{m+k}}$$

and thus, after simplification, $K = 2^{\alpha+t_1}$ which is an integer value. A general condition on t becomes

$$t \geq \frac{m+k}{2m}\left[\log_2\left((m+k)2^{2\alpha\frac{2m+k}{m+k}}\right) + \log_2\left(\frac{2\pi e}{m+k}\right)\right]$$

and the general result immediately follows.

Theorem 3. *Let $N_1 = p_1 q_1, \ldots, N_k = p_k q_k$ be k n-bit RSA moduli, where the q_i's are α-bit primes and the p_i's are primes that all share t bits from the position t_1 to $t_2 = t_1 + t$. Under Assumption 1, the N_i's can be factored in time $\mathscr{C}(\frac{k(k+1)}{2}, \frac{k(k+1)}{2}, n)$, as soon as*

$$t \geq 2\alpha + \frac{2}{k-1}\alpha + \frac{k+1}{2(k-1)}\log_2(2\pi e)$$

As in the case of $k = 2$, we inspect the general case $k \geq 3$ for the existence of exceptional vectors $\mathbf{v_1} = (c_1K, \ldots, c_kK, \ldots, c_iN_i - c_jN_j \mod 2^{t_2}, \ldots)$ which will disprove Assumption 1, that is, with c_i's ($< 2^\alpha$) and $c_iN_i - cjN_j \mod 2^{t_2}$ small (say $\approx 2^{t_2-\gamma}$). The condition under which the simple vector $\mathbf{v_1}$ with $c_1 = c_2 = \cdots = c_k = 1$ verify $\|\mathbf{v_1}\|^2 < \|\mathbf{v_0}\|^2$ is given by

$$t - \gamma < \alpha + \frac{1}{2} + \frac{1}{2}\log(\frac{(k+1)2^{2\alpha-1} - 1}{(k-1)}) \approx 2\alpha$$

Thus, as in the case of $k = 2$, for t and α small and γ large enough, this type of simple vectors may appear. Moreover, the degree of liberty for choosing the c_i increases with k, thus, exceptional vectors may appear more frequently when k grows. This fact was observed during our experiments.

Remark 3. During our first experiments, in few cases, our method fails to factor the N_i's. After analysis of the random generation functions used in our code, it turns out that the q_i where randomly generated in the interval $\left]2^{\alpha-1}, 2^\alpha\right]$. Thus, the probability that a lot of q_i's have exactly size α is high. If, moreover, α is small enough compared to t_2 ($\alpha < t_2 = t + t_1$), the corresponding $N_i - N_j \mod 2^{t_2}$ may be very small. This could be explained by the following fact: some of the most significant bits (and at least the highest bit) of $N_i \mod 2^{t_2}$ and $N_j \mod 2^{t_2}$ will be a part of the shared bits between the p_i's and thus they cancel themselves in $(N_i - N_j) \mod 2^{t_2}$. Hence, in this case, we have an exceptional short vector in L and our method fails; on the other hand, if one use these moduli then an attacker may use this extra information to easily factor them with another method.

5 Experimental results

Table 2: Results for $k = 2$ and 1024-bit RSA moduli with shared MSBs

α (bit-size of the q_i's)	Bound of Theorem 1 $t \geq 2\alpha + 3$	Best experimental t
150	303	302
200	403	402
250	503	502
300	603	602

In order to check the validity of Assumption 1 and the quality of our bounds on t, we implemented the methods on Magma 2.15 [1].

5.1 Shared MSBs

We generated many random 1024-bit RSA moduli, for various values of α and t. We observed that the results were similar for other values of n. In the case where $k = 2$,

Table 3: Results for $k = 3, 10, 40$ and 1024-bit RSA moduli with shared MSBs

α (bit-size of the q_i's)	Theoretical bound t	Best experimental t using LLL algo.	Best experimental t using Schnorr-Euchner's algo.	Failure rate of Assumption 1
Results for $k = 3$ (Theoretical bound of Theorem 2: $t \geq \frac{3}{2}\alpha + 5.2\ldots$)				
150	231	228	228	0% ($t = 227$)
200	306	303	303	0% ($t = 302$)
250	381	378	378	0% ($t = 377$)
300	456	453	453	0% ($t = 452$)
350	531	528	528	0% ($t = 527$)
400	606	603	603	0% ($t = 602$)
Results for $k = 10$ (Theoretical bound of Theorem 2: $t \geq \frac{10}{9}\alpha + 4.01\ldots$)				
150	171	169	169	0% ($t = 168$)
200	227	225	225	3% ($t = 224$)
250	282	280	280	3% ($t = 279$)
300	338	336	336	1% ($t = 335$)
350	393	391	391	2% ($t = 390$)
400	449	447	447	0% ($t = 446$)
Results for $k = 40$ (Theoretical bound of Theorem 2: $t \geq \frac{40}{39}\alpha + 3.68\ldots$)				
150	158	156	155	2% ($t = 154$)
200	209	208	207	3% ($t = 206$)
250	261	259	258	1% ($t = 257$)
300	312	310	309	1% ($t = 308$)
350	363	362	361	0% ($t = 360$)
400	414	413	412	2% ($t = 411$)

we used the Lagrange reduction to find with certainty a shortest vector of the lattice, and for $3 \leq k \leq 40$ we compared Schnorr-Euchner's algorithm (that provably outputs a shortest vector of the lattice) with LLL (that gives an exponential approximation of a shortest vector). We used only LLL for $k = 80$.

We conducted experiments for $k = 2, 3, 10, 40$ and 80, and for several values for α. For specific values of k, α and t, we said that a test was successful when the first vector of the reduced basis of the lattice was of the form $\pm\mathbf{v_0}$ (that is, it satisfies Assumption 1 in the heuristic case $k \geq 3$). For each k and each α, we generated 100 tests and found experimentally the best (lowest) value of t that had 100% success rate. We compared this experimental value to the bounds we obtained in Theorems 2 and 1. For the first value of t that does not have 100% success rate and for $k \geq 3$, we analyzed the rate of failures due to Assumption 1 not being valid. Note that failures can be of two different kinds: the first possibility is that $\|\mathbf{v_0}\|$ is greater than the Gaussian heuristic, and the second one is that $\|\mathbf{v_0}\|$ is smaller than the Gaussian heuristic yet $\mathbf{v_0}$ is not a shortest vector of the lattice (that is, Assumption 1 does not hold). We wrote down the percentage of the cases where Assumption 1 was not valid among all the cases where $\|\mathbf{v_0}\|$ was smaller than the Gaussian heuristic. These results are shown in tables 2 and 3. Let's take an ex-

Table 4: Results for $k = 5$ and 1024-bit RSA moduli with shared bits in the middle ($\alpha \in \{99, 100\}$, $t_1 = 20$, theoretical bound $t \geq 254$)

Experimental t	Failure rate of $\|\mathbf{v_0}\| <$ Gaussian heuristic	Failure rate with Schnorr-Euchner's algo.	Failure rate with LLL's algo.
261	0%	0%	0%
260	0%	1%	1%
259	0%	1%	1%
258	0%	1%	0%
257	0%	3%	2%
256	0%	6%	5%
255	0%	17%	10%
254	0%	33%	19%
253	0%	58%	28%
252	2%	90%	58%
251	96%	100%	89%

ample. For $k = 10$ and $\alpha = 200$ (second line of the part corresponding to $k = 10$ in table 3), Theorem 2 predicts that $\mathbf{v_0}$ is a shortest vector of the lattice as soon as $t \geq 227$. It turned out that it was always the case as soon as $t \geq 225$, which is better than expected. For $t = 224$, Assumption 1 was not valid in 3% of the cases.

Let's analyze the results now. In the rigorous case $k = 2$, we observe that the attack consistently goes one bit further with 100% success rate than our bound in Theorem 1.

In all our experiments concerning the heuristic cases $k \geq 3$, we observed that we had 100% success rate (thus, Assumption 1 was always true) when t was within the bound (10) of Theorem 2. That means that Theorem 2 was always true in our experiments. Moreover, we were often able to go a few bits (up to 3) beyond the theoretical bound on t. When the success rate was not 100% (that is, beyond our experimental bounds on t), we found that Assumption 1 was not true in a very limited number of the cases (less than 3%). Finally, up to dimension 80, LLL was always sufficient to find $\mathbf{v_0}$ when t was within the bound of Theorem 2, and Schnorr-Euchner's algorithm allowed us to go one bit further than LLL in dimension 40.

5.2 Shared bits in the middle

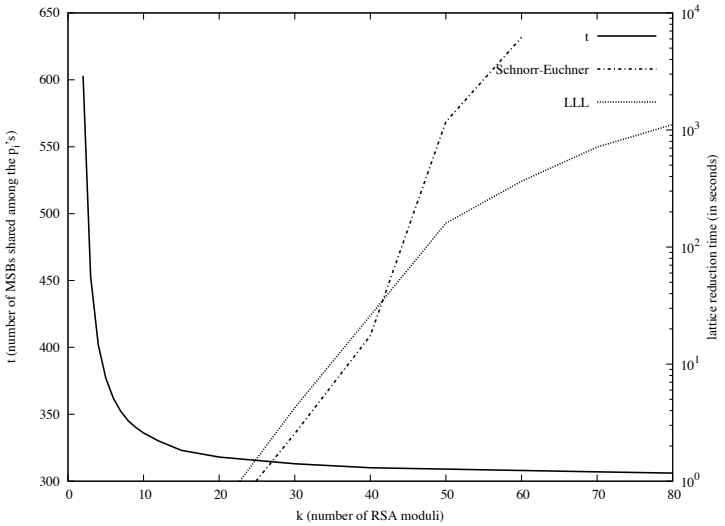
Contrary to the case of shared MSBs, Assumption 1 may fail when we apply our method with shared bits in the middle (see section 4). When $k = 2$ the phenomenon of exceptional short vectors rarely appeared when t was within the bound of Theorem 3 (less than 1% of failure and did not depend on the position t_1, moreover, we were generally allowed to go 2 or 3 bits further with 90% of success). When $k \geq 3$ it was not still the case. When Schnorr-Euchner's algorithm did not return $\mathbf{v_0}$, we tried to find it in a reduced basis computed by LLL. If neither of these algorithms was able to find $\mathbf{v_0}$ then our method failed. The table 4 shows the result of our experiments for $k = 5$ RSA moduli of size $n = 1024$ and q_i's of size $\alpha \in \{100, 99\}$ (see Remark 3). As one can

see, our method can be successfully applied in this case. During these experiments, the failure rate of our method was equal to the failure rate of finding $\mathbf{v_0}$ in a reduced basis computed by LLL. More generally, our experiments showed that for the same size of problems the rate of success is approximately 80% when t was within the bound of Theorem 3 and allowed us to go one or two bits further with success rate $\approx$50%.

5.3 Efficiency comparisons

Additionally, we show in table 5 the lowest value of t with 100% success rate and the running-time of LLL and Schnorr-Euchner's algorithm for several values of k (k RSA moduli with p_i's factors sharing t MSBs). For each k, we show the worst running-time we encountered when running 10 tests on an Intel Xeon E5420 at 2.5Ghz. We see that all individual tests completed in less than 1 second for $2 \leq k \leq 20$. We used Schnorr-Euchner's algorithm up to $k = 60$ where it took at most 6200 seconds. LLL completes under one minute for $20 \leq k \leq 40$ and in less than 30 minutes for $40 \leq k \leq 80$.

Table 5: Running time of LLL and Schnorr-Euchner's algorithm, and bound on t as k grows. (Shared MSBs with $\alpha = 300$ and $n = 1024$)



6 Conclusion

In this article we have studied the problem of integers factorization with implicit hints. We have presented new lattice based methods in order to factorize $k \geq 2$ RSA moduli $N_i = p_i q_i$ with polynomial complexity in $\log(N_i)$ when p_i's share unknown MSBs or contiguous bits in the middle. In the case $k = 2$ and shared MSBs, our method is the first one to be completely rigorous. These new results can be seen as an extension of the ones presented in [9] and [15] where, respectively, May and Ritzenhofen gave same type of results in the case where the p_i's share LSBs and Sarkar and Maitra presented heuristic methods based on the Coppersmith's algorithm for finding small roots of polynomials for $k = 2$ moduli with shared MSBs (and/or LSBs) or bits in the middle . Our method gives comparable theoretical results as the one of May and Ritzenhofen and it is more efficient than the Sarkar and Maitra's method.

Whether the method can be applied for $k \geq 3$ N_i's RSA moduli with p_i's sharing MSBs and LSBs remains an open issue. In this case, the problem has much more variables and our method can not be directly applied. One possible way to follow for attacking this problem is to use algebraic techniques, in particular elimination theory, jointly with lattice based methods. This would be an interesting focus for future research.

Acknowledgments

We would like to thank the referees for their valuable comments. We thank Alexander May and Maike Ritzenhofen for their very helpful comments on a draft version of this article and, more particularly, for those which initiated the results of section 4.

The work described in this paper has been supported in part by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II. The authors were also supported in part by the french ANR under the Computer Algebra and Cryptography (CAC) project ANR-09-JCJCJ-0064-01.

References

1. Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system I: The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).
2. Don Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In Ueli M. Maurer, editor, *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 178–189. Springer, 1996.
3. Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51. Springer, 2008.
4. Guillaume Hanrot and Damien Stehlé. Improved analysis of Kannan's shortest lattice vector algorithm. In Alfred Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 170–186. Springer, 2007.
5. Ellen Jochemsz and Alexander May. A strategy for finding roots of multivariate polynomials with new applications in attacking rsa variants. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 267–282. Springer, 2006.

6. Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *STOC*, pages 193–206. ACM, 1983.
7. D.H. Lehmer and R.E. Powers. On factoring large numbers. *Bulletin of the AMS*, 37:770–776, 1931.
8. Arjen K. Lenstra and Hendrik W. Jr. Lenstra, editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1993.
9. Alexander May and Maike Ritzenhofen. Implicit factoring: On polynomial time factoring given only an implicit hint. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2009.
10. Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.
11. Michael A. Morrison and John Brillhart. A method of factoring and the factorization of F_7. *Mathematics of Computation*, 29(129):183–205, 1975.
12. Phong Q. Nguyen and Damien Stehlé. Floating-point lll revisited. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 215–233. Springer, 2005.
13. Xavier Pujol and Damien Stehlé. Rigorous and efficient short lattice vectors enumeration. In Josef Pieprzyk, editor, *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 390–405. Springer, 2008.
14. Ronald L. Rivest and Adi Shamir. Efficient factoring based on partial information. In Franz Pichler, editor, *EUROCRYPT*, volume 219 of *Lecture Notes in Computer Science*, pages 31–34. Springer, 1985.
15. Santanu Sarkar and Subhamoy Maitra. Further Results on Implicit Factoring in Polynomial Time. *Advances in Mathematics of Communications*, 3(2):205–217, 2009.
16. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *FOCS*, pages 124–134. IEEE, 1994.
17. Scott A. Vanstone and Robert J. Zuccherato. Short rsa keys and their generation. *J. Cryptology*, 8(2):101–114, 1995.

A Common results on lattice

An integer lattice L is an additive subgroup of $\mathbb{Z}^n$. Equivalently, it can be defined as the set of all integer linear combinations of d independent vectors $\mathbf{b_1}, \ldots, \mathbf{b_d}$ of $\mathbb{Z}^n$. The integer d is called the *dimension* of L, and $B = (\mathbf{b_1}, \ldots, \mathbf{b_d})$ is one of its *bases*. All the bases of L are related by a unimodular transformation. The *volume* (or *determinant*) of L is the d-dimensional volume of the parallelepiped spanned by the vectors of a basis of L and is equal to the square root of the determinant of the Gramian matrix of B. It does not depend upon the choice of B. We denote it by $\mathrm{Vol}(L)$.

We state (without proofs) common results on lattices that will be used throughout this paper. Readers interested in getting more details and proofs can refer to [10].

Definition 1. *For $1 \leq r \leq d$, let $\lambda_r(L)$ be the least real number such that there exist at least r linearly independent vectors of L of euclidean norm smaller than or equal to $\lambda_r(L)$. We call $\lambda_1(L), \ldots, \lambda_d(L)$ the d* minima *of L, and we call $g(L) = \frac{\lambda_2(L)}{\lambda_1(L)} \geq 1$ the* gap *of L.*

Lemma 4 (Hadamard). *Let $B = (\mathbf{b_1}, \ldots, \mathbf{b_d})$ be a basis of a d-dimensional integer lattice of $\mathbb{Z}^n$. Then the inequality $\prod_{i=1}^{d} \|\mathbf{b_i}\| \geq Vol(L)$ holds.*

Theorem 4 (Minkowski). *Let L be a d-dimensional lattice of $\mathbb{Z}^n$. Then there exists a non zero vector $\mathbf{v}$ in L which verifies $\|\mathbf{v}\| \leq \sqrt{d}\,\mathrm{Vol}(L)^{\frac{1}{d}}$. An immediate consequence is that $\lambda_1(L) \leq \sqrt{d}\,\mathrm{Vol}(L)^{\frac{1}{d}}$*

Theorem 5 (Lagrange reduction). *Let L be a 2-dimensional lattice of $\mathbb{Z}^n$, given by a basis $B = (\mathbf{b_1}, \mathbf{b_2})$. Then one can compute a Lagrange-reduced basis $B' = (\mathbf{v_1}, \mathbf{v_2})$ of L in time $\mathcal{O}(n\log^2(\max(\|\mathbf{b_1}\|, \|\mathbf{b_2}\|)))$. Besides, it verifies $\|\mathbf{v_1}\| = \lambda_1(L)$ and $\|\mathbf{v_2}\| = \lambda_2(L)$. More information about the running time of the Lagrange reduction may be found in [10].*

Theorem 6 (Kannan's algorithm, see [6,13,4]). *Let L be a d-dimensional lattice of $\mathbb{Z}^n$ given by a basis $(\mathbf{b_1}, \ldots, \mathbf{b_d})$. One can compute a shortest vector of L (with norm equal to $\lambda_1(L)$) in time $\mathcal{O}(\mathscr{P}(\log B, n)\, d^{\frac{d}{2e}+o(d)})$ where $\mathscr{P}$ is a polynomial and $B = \max_i(\|\mathbf{b_i}\|)$. This is done by computing a HKZ-reduced basis of L.*

Theorem 7 (LLL). *Let L be a d-dimensional lattice of $\mathbb{Z}^n$ given by a basis $(\mathbf{b_1}, \ldots, \mathbf{b_d})$. Then LLL algorithm computes a reduced basis $(\mathbf{v_1}, \ldots, \mathbf{v_d})$ that approximates a shortest vector of L within an exponential factor $\|\mathbf{v_1}\| \leq 2^{\frac{d-1}{4}}\,\mathrm{Vol}(L)^{\frac{1}{d}}$. The running time of Nguyen and Stehlé's version is $\mathcal{O}(d^5(d+\log B)\log B)$ where $B = \max_i(\|\mathbf{b_i}\|)$, see [12].*

In practice, LLL algorithm is known to perform much better than expected. It has been experimentally established in [3] that we can expect the bound $\|\mathbf{v_1}\| \leq 1.0219^d\,\mathrm{Vol}(L)^{\frac{1}{d}}$ on $\|\mathbf{v_1}\|$ on random lattices and that finding a shortest vector of a lattice with gap greater than 1.0219^d should be easy using LLL.

B Exact computation of the Volume of lattice L of section 3

In this section, we compute exactly the volume of the lattice L defined at the beginning of section 3. As a visual example of the construction of this lattice, the reader may take a look at the matrix defined in equation (9) in the case of $k = 4$. We use the notations of section 3.

Lemma 5. *Let L be the lattice whose construction is described at the beginning of section 3. Then its volume is equal to $\mathrm{Vol}(L) = K\left(K^2 + \sum_{i=1}^{k} N_i^2\right)^{\frac{k-1}{2}}$.*

Proof. Let G be the Gramian matrix (of size $k \times k$) of L. Its diagonal terms are $\langle \mathbf{v_i}, \mathbf{v_i} \rangle = K^2 + \sum_{\substack{u=1 \\ u \neq i}}^{k} N_u^2$ and its other terms are: $\langle \mathbf{v_i}, \mathbf{v_j} \rangle = -N_i N_j$. Observe that we can rewrite G as follows $G = \left(K^2 + \sum_{i=1}^{k} N_i^2\right) I_{k\times k} + J$ where $I_{k\times k}$ is the identity matrix of size k and J is the $k \times k$ matrix with terms $-N_i N_j$. If we let χ_J be the characteristic polynomial of J and $\lambda_0 = K^2 + \sum_{i=1}^{k} N_i^2$, we observe that $\det(G) = \chi_J(-\lambda_0)$.

All the columns of J are multiples of $(N_1, N_2, \ldots, N_k)^t$. The rank of J is thus 1. The matrix J has therefore the eigenvalue 0 with multiplicity $k-1$. The last eigenvalue is computed using its trace: $\mathrm{Tr}(J) = -\sum_{i=1}^{k} N_i^2$. Therefore, up to a sign $\chi_J(X) = X^{k-1}\left(X + \sum_{i=1}^{k} N_i^2\right)$. We conclude that $\det(G) = \chi_J\left(-K^2 - \sum_{i=1}^{k} N_i^2\right)$, hence $\det(G) = K^2\left(K^2 + \sum_{i=1}^{k} N_i^2\right)^{k-1}$ and $\mathrm{Vol}(L) = \sqrt{\det(G)} = K\left(K^2 + \sum_{i=1}^{k} N_i^2\right)^{\frac{k-1}{2}}$ □