Toward a rigorous variation of Coppersmith's algorithm on three variables

Aurélie Bauer[2] and Antoine Joux[1,2]

[1] DGA
[2] Université de Versailles Saint-Quentin-en-Yvelines
Laboratoire PRISM, 45, Avenue des Etats-Unis
78035 Versailles cedex, France
aurelie.bauer@prism.uvsq.fr, antoine.joux@m4x.org

Abstract. In 1996, Coppersmith introduced two lattice reduction based techniques to find small roots in polynomial equations. One technique works for modular univariate polynomials, the other for bivariate polynomials over the integers. Since then, these methods have been used in a huge variety of cryptanalytic applications. Some applications also use extensions of Coppersmith's techniques on more variables. However, these extensions are heuristic methods. In the present paper, we present and analyze a new variation of Coppersmith's algorithm on three variables over the integers. We also study the applicability of our method to short RSA exponents attacks. In addition to lattice reduction techniques, our method also uses Gröbner bases computations. Moreover, at least in principle, it can be generalized to four or more variables.

Key words: Lattice reduction, Coppersmith's algorithms, Gröbner basis

1 Introduction

In 1996, Coppersmith introduced two methods for finding small roots of polynomial equations using lattice reduction, one for the univariate modular case and another one for the bivariate case over the integers [6, 5, 7]. These algorithms are based on the same idea: using lattice reduction (e.g. LLL) in order to create a second polynomial that has the same root as the first one. In both cases, this construction leads to a rigorous method to recover the root. In particular, in the bivariate case, the use of orthogonal lattice guarantees the independence of the two polynomials and ensures that the root can be recovered. In order to simplify and help understand Coppersmith's methods, Howgrave-Graham [13] and Coron [8] revisited his ideas and proposed alternative constructions.

Since 1996, many cryptanalytic applications have been based on these methods, for example the factorization of $N = pq$ knowing a fraction of the most significant bits on each factor. Another well-known example is the cryptanalysis of RSA with small private key [4, 2].

The applications of these algorithms for finding small roots of polynomial equations can roughly be divided into two parts. On the one hand some researchers try to generalize the original Coppersmith's methods. For example, in [3], Blömer and May present new results using Coppersmith's method for polynomials whose shapes are more complicated than those originally considered in Coppersmith's articles. Another example is the paper of Howgrave-Graham [14] in which he explains how to cast the problem of finding roots for particular polynomials in the more general context of approximate GCD computations. On the other hand, there are researchers trying to adapt all these methods for more than two variables. As an example, several new attacks on RSA are proposed in [9], using variants of the original method on three variables.

However, with more than two variables, one encounters a major obstruction. Indeed, one can not guarantee any more that the polynomials outputted by LLL reduction are algebraically independent. Still in some practical applications, the approach continues to work. Despite this, more and more articles mention problematic cases. For example, in [2] the authors analyze in details one of the heuristic multivariate attacks proposed by Boneh and Durfee in [4]. In [11, 12], Hinek analyzes the problem of algebraic independence of the polynomials. He focuses on the fact that in experiments algebraic dependency often leads to difficulties and he says "in light of the observations in this work, it might be the case that this lack of reported instances is simply due to a lack of experimental observations". In this paper, in order to avoid these difficulties we propose a new generalization of Coppersmith's method in three variables, using a new lattice construction to find a third independent polynomial. Our construction uses Gröbner basis in addition to lattice reduction.

This paper is organized as follows. In section 2, we recall a few facts about lattice reduction and known heuristic variations of Coppersmith's method on three variables over the integers. We discuss the issue of polynomials independence. In section 3, we present an overview of our main idea which generalizes the method by using LLL reduction on a different lattice. To construct it, we show that the use of Gröbner bases and their properties are essential. In section 4 we describe a criterion on the input polynomials that when satisfied allows to develop a rigorous version. In section 5, we focus on one of the RSA attacks proposed in [9] and we show some results of experiments made with our method. The two approaches can then be compared. Finally, in section 6 we discuss the possibility of generalizing to four or more variables.

2 Preliminaries

2.1 Lattices

Since lattices are an essential tool for Coppersmith's attack, let us recall a few facts about lattices and reduced basis. A lattice L is a discrete subgroup of $\mathbb{R}^n$. If L is a non-empty subset of $\mathbb{R}^n$, L is a lattice if and only if there exists r linearly independent vectors over $\mathbb{R}$ (with $r \leq n$) such that

$$L = \mathbb{Z}b_1 \oplus \cdots \oplus \mathbb{Z}b_r$$

The set $\mathcal{B} = (b_1, \ldots, b_r)$ is called a basis of L. In this paper, as in many cryptographic applications, we focus on integer lattices $L \subset \mathbb{Z}^n$.

Let L be a lattice generated by the vectors $\mathcal{B}{=}(b_1, \ldots, b_r)$ and $(b_1^\star, \ldots, b_r^\star)$, the vectors from Gram-Schmidt's orthogonalization of $\mathcal{B}$. Let B be the $r \times n$-matrix whose rows are the b_i's. The determinant of L is defined as

$$\det L = \sqrt{\det(B^t B)} = \prod_{i=1}^{r} \|b_i^\star\|^2$$

where $\|\|$ denotes the Euclidean norm. When L is a full-rank lattice (i.e. when $n = r$) the formula can be simplified to $\det L = |\det B|$.

In 1982, Lenstra, Lenstra and Lovasz [15] introduced the LLL reduction algorithm. Using this algorithm, one can obtain a reduced basis of a lattice L. To analyze Coppersmith's algorithm, we need to know that

$$\|b_r^\star\| \geq (\det L)^{1/r}\, 2^{-(r-1)/4} \tag{1}$$

for any LLL reduced basis $(b_1, \ldots, b_r)$.

2.2 Gröbner basis on three variables

Let $\mathbb{Z}[x, y, z]$ be the polynomial ring in three variables x, y, z over $\mathbb{Z}$. A monomial is an elementary polynomial $x^{\alpha_1} y^{\alpha_2} z^{\alpha_3}$ with $(\alpha_1, \alpha_2, \alpha_3) \in \mathbb{N}^3$ and a term is $\lambda x^{\alpha_1} y^{\alpha_2} z^{\alpha_3}$ with $\lambda \in \mathbb{Z}$. In the following, when we refer to a monomial in a set, we use both the notations $(\alpha_1, \alpha_2, \alpha_3)$ and $x^{\alpha_1} y^{\alpha_2} z^{\alpha_3}$. If p is a polynomial defined over $\mathbb{Z}$, the Newton polygon of p refers to the convex hull of all monomials (viewed as points in $\mathbb{N}^3$) that appear with a non zero coefficient in p.

A monomial ordering $<$ on $\mathbb{Z}[x, y, z]$ is a total ordering on the set of monomials which is compatible with multiplication. Among all existing orderings, a frequently encountered one is called *deglex* and is defined as:

$$x^{\alpha_1} y^{\alpha_2} z^{\alpha_3} < x^{\beta_1} y^{\beta_2} z^{\beta_3} \Leftrightarrow \begin{cases} \alpha < \beta \\ \text{ or} \\ \alpha = \beta \text{ and } \ \exists i \in \{1,2,3\}, \alpha_i < \beta_i \\ \qquad\qquad\quad \forall j < i, \alpha_j = \beta_j \end{cases}$$

where $\alpha = (\alpha_1 + \alpha_2 + \alpha_3)$ and $\beta = (\beta_1 + \beta_2 + \beta_3)$. If a monomial ordering is chosen, the initial term of a polynomial p, denoted by $in(p)$, refers to its greatest term. Let I be an ideal of $\mathbb{Z}[x, y, z]$, $in(I)$ is the set of all initial terms of the polynomials which belong to I. If the set $\{q_1, \ldots, q_l\}$ is composed by polynomials of I such that $(in(q_1), \ldots, in(q_l)) = in(I)$, we call it a Gröbner basis of I. In practice, a Gröbner basis can be computed using F4 algorithm [10] implemented in Magma. For a system of generators having d as its maximal degree, the theoretical complexity is polynomial in d when the number of variables is fixed.[3]

[3] According to M. Bardet [1], the complexity in this case is upper bounded by d^{72}. In practice, the computation is very fast under Magma.

2.3 Primary decomposition

Let I be an ideal of $\mathbb{Z}[x, y, z]$. It is said to be prime if the condition $fg \in I$ implies that either f or g belongs to I. The radical of I, denoted by $\sqrt{I}$, refers to the set $\{f \in I, \exists n \in \mathbb{N}, f^n \in I\}$. A primary ideal J satisfies the following condition: if fg belongs to J with $f \notin J$, then g belongs to $\sqrt{J}$. If I is a primary ideal, then $\sqrt{I}$ is a prime one. In a noetherian ring, each ideal can be written as an intersection of primary ideals. In practice, with the help of Magma, a few seconds are needed to compute the primary decomposition of an ideal I or to obtain its radical.

The set defined as $\{(x_1, y_1, z_1) \in \mathbb{Z}^3, \forall p \in I, p(x_1, y_1, z_1) = 0\}$ is denoted as $V(I)$. In the following, we say that (x_1, y_1, z_1) is a root of I if it belongs to $V(I)$. If I has $I_1 \cap \cdots \cap I_r$ as a primary decomposition, then $V(I) = V(I_1) \cup \cdots \cup V(I_r)$. The following property holds: $V(I) = V(\sqrt{I})$.

2.4 Coppersmith's method, a basic variation on 3 variables

Let $p_1(x, y, z)$ be an irreducible polynomial of $\mathbb{Z}[x, y, z]$ having (x_0, y_0, z_0) as root over the integers satisfying $|x_0| < X, |y_0| < Y$ and $|z_0| < Z$. As usual when working with Coppersmith's method, we denote by W_1 the quantity $\|\tilde{p_1}\|_\infty$ where $\|p(x, y, z)\|_\infty$ is the maximum of the absolute values of the coefficients of p and $\tilde{p}_1(x, y, z)$ represents $p_1(xX, yY, zZ)$. Our goal is to recover the root (x_0, y_0, z_0) in polynomial time.

As in [3], we use the notion of admissible sets. Let M be a non-empty set of three variables monomials. A polynomial $p(x, y, z)$ is said to be defined over M if p can be written as linear combination of monomials in M. Let S be another non-empty set and f, g be two polynomials such that $g = fp_1$. The ordered pair (S, M) is said to be admissible for p_1 if the property "g defined over M" is equivalent with "f defined over S". The cardinality of M and S are denoted by m and s.

Coppersmith's algorithm works by finding a second polynomial p_2 algebraically independent from p_1, which has the same root over the integers. When working with two variables, the resultant of p_1 and p_2 is non zero and the root can easily be recovered. However, in our case, since we work with three variables, two polynomials are not enough to recover the root. Still, it is a first important step. Thus, we now describe how Coppersmith's algorithm can be adapted in three variables to find p_2. We start by introducing the notation $\boldsymbol{({x_0}^f {y_0}^g {z_0}^h)_M}$ that refers to the vector $(1, x_0, y_0, z_0, \ldots, {x_0}^f {y_0}^g {z_0}^h, \ldots)$ with $(f, g, h) \in M$, where the order of the coordinates depends on the monomial ordering. Then, let us take the vector $r_0 = \boldsymbol{(x_0^f y_0^g z_0^h)_M}$ and the lattice L_1 generated by the rows of the matrix M_1 (see figure 1).

The right hand part of M_1 is denoted by P_1 and the left hand one by D_M. As $s < m$, there exists a sublattice $L_1' \subset L_1$ of dimension $(m - s)$ such that its vectors have their s last coordinates equal to zero. As (x_0, y_0, z_0) is a root of p_1, the product $s_0 = r_0 M_1$ gives a short vector of L_1' defined by

$$
\begin{array}{cc}
M_1 = \left(\begin{array}{cc|c}
\ddots & & \overbrace{\downarrow\downarrow\downarrow}^{x^iy^jz^kp_1} \\
\underbrace{X^{-f}Y^{-g}Z^{-h}}_{(f,g,h)\in M} & & \\
 & \ddots &
\end{array}\right) & m \updownarrow \\
\quad\quad \overleftrightarrow{\quad m \quad} \qquad \overleftrightarrow{\ s\ } &
\end{array}
$$

Fig. 1.

$s_0 = ((\frac{\boldsymbol{x_0}}{\boldsymbol{X}})^{\boldsymbol{f}}(\frac{\boldsymbol{y_0}}{\boldsymbol{Y}})^{\boldsymbol{g}}(\frac{\boldsymbol{z_0}}{\boldsymbol{Z}})^{\boldsymbol{h}})_{\boldsymbol{M}}|(0,\ldots,0)$ where the symbol $|$ refers to the concatenation of the two vectors. Assume that $(b_1,\ldots,b_r)$ is an LLL reduced basis of L'_1 (with $r = m - s$), then when $\|s_0\| < \|b_r^\star\|$ we know that the inner product $< s_0|b_r^\star >$ is equal to zero. That leads to a new polynomial p_2 that has the same root as p_1.

As in [5], we can show that p_2, as all polynomials obtained from L'_1, is by construction independent from p_1. The crucial point of this proof is based on the fact that, in this case, algebraic independence relies on linear independence.

In order to prove in advance that the inequality $\|s_0\| < \|b_r^\star\|$ holds, we need to compute $|\det L'_1|$. This can be done by adapting the method of [5], see appendix A. From the determinant computation, we derive the conditions on the bounds X, Y, Z:

$$X^{s_x}Y^{s_y}Z^{s_z} < W_1^s 2^{-(6+c)s(d_x^2+d_y^2+d_z^2)} \tag{2}$$

with c a well-chosen constant. In this formula d_x, d_y and d_z denote the maximum degree of p_1 in x, y, z and s_x refers to $\sum_{(f,g,h)\in M\setminus S} f$. The corresponding sums on y and z are denoted by s_y and s_z.

2.5 Recovering the root

With this method, we have two polynomials p_1 and p_2 that have (x_0, y_0, z_0) as common root over the integers, and are algebraically independent. Two approaches have been proposed to recover the root. The first idea is to compute the (provably non-zero) resultant of p_1 and p_2 in one of the variables. This leads to a polynomial in two variables, on which Coppersmith's algorithm can be reused. However, this polynomial usually has a very high degree. As a consequence, the conditions on the bounds are too restrictive to make the method useful. Another idea is to reuse Coppersmith's method in three variables trying to find another polynomial p_3. The difficulty here is to ensure that p_3 is algebraically independent from p_1 and p_2. Many authors use this approach together with the heuristic hypothesis that p_3 happens to be independent from $\{p_1, p_2\}$.

2.6 The notion of independence

As this works focuses on the problem of algebraic independence, this notion has to be rigorously defined. Three polynomials p_1, p_2, p_3 are algebraically independent if and only if $P(p_1, p_2, p_3) = 0$ implies $P = 0$ for a polynomial P defined over $\mathbb{Q}[x, y, z]$. In general, showing this property is rather difficult. In our case, knowing that p_1 is irreducible and that p_2 does not belong to (p_1), it can be reduced to a simpler problem. When the ideal $I = (p_1, p_2)$ is prime, whenever p_3 does not belong to I, then p_1, p_2 and p_3 are algebraically independent. The proof of this result can be found in appendix B. It uses the fact that (x_0, y_0, z_0) is a common root of these three polynomials.

In the sequel, when we refer to I, it implicitly means a prime ideal. As a consequence, showing that p_3 does not belong to I is a sufficient condition to obtain the independence. Let us now discuss on what happens if I is not a prime ideal. In this case, the analysis is more complicated. Two behaviors are possible depending on the fact that I is a primary ideal or not. If I is primary, it is sufficient to replace it by its radical, which is prime. In the other case, I can be written as an intersection of primary ideals $I_1 \cap \cdots \cap I_r$, such that at least one of the I_j has (x_0, y_0, z_0) as root. One has just to replace I by the well-chosen ideal and to take its radical if it is primary.

3 A new lattice to find a third independent polynomial

Having recovered p_1 and p_2, we now want a method to create a third polynomial p_3 that has again the same root as p_1 and p_2 and moreover does not belong to the ideal $I = (p_1, p_2)$. The main idea is to construct a new lattice very similar to Coppersmith's one that can produce this third independent polynomial.

3.1 Overview of the main idea

Let start with analyzing the first step of Coppersmith's algorithm. The proof concerning the independence of p_2 from p_1 uses the fact that, in this case, algebraic independence relies on linear independence. In three variables, a third polynomial has to be found. As explained before, the main difficulty is less its construction than the proof of its independence from I. Our goal is to adapt the previous construction and to keep information both from p_1 and p_2. If I_M is the set of all polynomials belonging to I that are defined over M, one possible idea would be to create a new lattice by using generators of I_M as a $\mathbb{Z}$-module. Thus, any polynomial belonging to I_M is generated by the columns of this new matrix.

Finding these generators is quite complicated as it is strongly linked to the shape of the set M. For this reason, in the rest of this section, we only focus on a pair $(M, <)$ such that there exists an equivalence between belonging to M and being smaller than a given monomial ($<$ is compatible with the shape of M). As an example, one can consider the set M defined as all $(f, g, h) \in \mathbb{N}^3$ satisfying $(f + g + h) \leq n$ (with n an integer) and the *deglex* ordering. In order to construct these generators, we need an additional tool. In the sequel, we show that the use of truncated Gröbner basis gives us the right tool.

3.2 Truncated Gröbner basis

As explained before, our goal here is to find linear generators of I (up to some degree). Moreover, we want these polynomials to be defined over M in order to preserve the dimension of the lattice. Let us denote by $\mathcal{F}$, the set we are looking for. To sum up the property we require for the construction, one can say that if p is a polynomial defined over M and belonging to I, we want it to be written as linear combination of the polynomials r_i where the set $\{r_1, \dots, r_t\}$ refers to $\mathcal{F}$. To construct such a set, we need the use of "truncated Gröbner basis" whose definition is given as follows:

Definition 1. *Let $G = \{q_1, \dots, q_r\}$ be a minimal Gröbner basis of I. A truncated Gröbner basis of I related to M is the set of polynomials $\{q_{i_1}, \dots, q_{i_l}\}$ of G such that for each $j \in \{i_1, \dots, i_l\}$, q_{i_j} is defined over M. The corresponding set is denoted by G_M.*

The idea is just to keep among all polynomials that generate the ideal I, those which are defined over M. Then, to obtain the set $\mathcal{F}$, it is sufficient to multiply the q_{i_j} by monomials and keep products which remain defined over M. Thus, we have a system of generators of the vector space I_M. The creation of the set G_M from G has a complexity equal to $O(rm)$ whereas those of $\mathcal{F}$ has one of $O(rm^2)$. With this construction, the set $\mathcal{F}$ is not necessarily minimal. As a consequence, one could improve this construction by deleting the extra polynomials. However, keeping them does not increase the dimension of the lattice, and does not change the proofs.

Note: When the set M cannot be described directly by a monomial ordering, finding a set of generators $\mathcal{F}$ is more difficult. However, the examples of section 5 show that it can still be done in practice. The difficulty here is to give a theoretic construction that works for all cases.

3.3 A second Coppersmith's iteration

Knowing how to find the set $\mathcal{F}$, we are able to construct a new lattice to recover a polynomial p_3 having (x_0, y_0, z_0) as root over the integers. The positive point of this construction is that we can now prove that p_3 does not belong to the ideal I. Let us explain in more details, how the lattice is constructed.

Let start by considering the $m \times t$ matrix P_2 whose columns represent the polynomials $\{r_1, \dots, r_t\}$. From P_2, one can construct the lattice L_I generated by the rows of the following matrix:

$$M_I = \begin{array}{cc} \left(\begin{array}{ccc|c} \ddots & & & \overbrace{\downarrow\downarrow\downarrow}^{r_1,\dots,r_t} \\ & \underbrace{X^{-f}Y^{-g}Z^{-h}}_{(f,g,h)\in M} & & \\ & & \ddots & \end{array}\right) & m \updownarrow \\ \xleftrightarrow{\quad m \quad} \qquad \xleftrightarrow{\ t\ } & \end{array}$$

If we assume that $t < m$, there exists a sublattice $L'_I \subset L_I$ whose dimension is $(m-t)$ such that its vectors have their t last coordinates equal to zero. Let see again the vector $r_0 = (\boldsymbol{x_0^f y_0^g z_0^h})_{\boldsymbol{M}}$. As (x_0, y_0, z_0) is a root of all polynomials in I, the product $t_0 = r_0 M_I$ satisfies $t_0 = ((\frac{\boldsymbol{x_0}}{\boldsymbol{X}})^{\boldsymbol{f}} (\frac{\boldsymbol{y_0}}{\boldsymbol{Y}})^{\boldsymbol{g}} (\frac{\boldsymbol{z_0}}{\boldsymbol{Z}})^{\boldsymbol{h}})_{\boldsymbol{M}} |(0, \ldots, 0)$. This is a short vector of L'_I. Assume that $(c_1, \ldots, c_r)$ is an LLL reduced basis of L'_I (with $r = m - t$). When $\|t_0\| < \|c_r^\star\|$, the inner product $< t_0 | c_r^\star >$ is equal to zero, that leads to a new polynomial p_3 that has the same common root as p_1 and p_2.

Let focus on the most important point which is the independence of p_3 from the ideal I. By construction, the vector which refers to p_3 is orthogonal to all polynomials of the set $\mathcal{F}$. Knowing that in each vector space E, if there exists a vector x such that for all $y \in E$, $< x|y >= 0$, then $x = 0$, we necessarily have $p_3 \notin I$. Indeed, if p_3 is assumed to belong to I, it would be equal to zero, which is not the case.

Then, from p_1 and p_2, we construct a polynomial p_3 that again has (x_0, y_0, z_0) as a root over the integers and that does not belong to I. The resultant computation of the three polynomials leads to a non-zero result and the root can be recovered easily. When trying to check if $\|t_0\| < \|c_r^\star\|$ is verified, some technical difficulties related to the evaluation of the determinant of M_I, are encountered. As the considered lattice L'_I is much more complicated than the initial one used in the first iteration of Coppersmith's algorithm on three variables, it makes the analysis more difficult. In the general case, as we are not able to evaluate the determinant of M_I precisely, this can not give explicit bounds.

4 A criterion that guarantees rigorous success

Starting with the ideal $I = (p_1, p_2)$, we give in this section a criterion on the input polynomials that guarantees that p_3 can be found with no further restrictions on X, Y, Z. Let us consider the set $\mathcal{F}$ related to the ideal I. In the sequel, we use the following criterion: $\mathcal{F}$ should be equal to $\{\{x^i y^j z^k p_1\}_{(i,j,k)\in S}, p_2\}$. The monomial $x^a y^b z^c$ refers to those which verifies $|\tilde{p}_{2,(a,b,c)}| = \|\tilde{p}_2(xX, yY, zZ)\|_\infty = W_2$. The gcd of the coefficients of p_2 is denoted by d. In the sequel, we show that p_3 can be found with no further restrictions than what was required to obtain p_2. The proof relies on a variation of the method explained in appendix A and is written using the same notations.

4.1 Some preliminary results

Consider $\bar{P}_2$ the $(m \times (s+1))$ matrix whose s first columns are multiples of $\tilde{p_1}$ and the last one represent the polynomial $\tilde{p_2}$. Thus, $\bar{P}_2$ is just composed by the matrix $\bar{P}_1$ and one additional column that is $\tilde{p_2}$. Using all results of the appendix A, there exists a subset $\hat{M} \subset M$ of size s such that if $\hat{P}_1$ is the matrix composed by the rows of P_1 related to $\hat{M}$, we are able to evaluate $|\det \hat{P}_1|$. As p_2 is defined over $M \setminus \hat{M}$, we know that (a, b, c) can not belong to $\hat{M}$. Then, let take the set $\dot{M} = \hat{M} \cup \{(a, b, c)\}$. We have $|\dot{M}| = (s+1)$.

If we select from $\bar{P}_2$, the rows related to monomials in $\dot{M}$, we obtain the following matrix $\hat{P}_2$:

$$\hat{P}_2 = \left(\begin{array}{c|c} \hat{P}_1 & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ \hline \times \ \dots \ \times & \pm W_2 \end{array} \right)$$

That leads to

$$|\det \hat{P}_2| \geq W_2 W_1^s 2^{-6s(d_x^2+d_y^2+d_z^2)}$$

4.2 Construction of the lattice $\mathcal{L}_I$

Let P_2 be the $(m \times (s+1))$ matrix constructed as follows: the s first columns represent $x^i y^j z^k p_1$ for all $(i,j,k) \in S$ and the last one is p_2. $\mathcal{L}_I$ is the lattice generated by the rows of the following matrix:

$$N_I = \left(D_{M \setminus \dot{M}} \mid P_2 \right)$$

where $D_{M \setminus \dot{M}}$ is the resulting matrix coming from deletion in D_M of the columns related to monomials in $\dot{M}$. This definition of $\mathcal{L}_I$ is different from those of L_I, which has been announced in the previous section. However using this construction does not change the explanation, moreover it gives an easier analysis. Multiplying the rows of N_I related to $(f,g,h) \in M$ by $X^f Y^g Z^h$ and the s first columns of P_2 related to $(i,j,k) \in S$ by $X^{-i} Y^{-j} Z^{-k}$ leads to the matrix $\bar{N}_I$ satisfying:

$$|\det \bar{N}_I| = |\det N_I| X^{s_x} Y^{s_y} Z^{s_z}$$

Making some elementary row operations on $\bar{N}_I$ leads to:

$$\begin{pmatrix} Id & A' \\ 0 & \hat{P}_2 \end{pmatrix}$$

whose determinant is equal to $(\det \hat{P}_2)$. Thus, we obtain

$$|\det N_I| \geq X^{-s_x} Y^{-s_y} Z^{-s_z} W_2 W_1^s 2^{-6s(d_x^2+d_y^2+d_z^2)}$$

4.3 Using LLL-reduction to construct p_3

The demonstration follows the same idea as in the previous case. r_0 is the vector defined by $r_0 = \boldsymbol{(x_0^f y_0^g z_0^h)_M}$ and $t_0 = r_0 N_I$. We have

$$t_0 = \left(\left(\frac{x_0}{X}\right)^f \left(\frac{y_0}{Y}\right)^g \left(\frac{z_0}{Z}\right)^h {}_{M \setminus \dot{M}} \right) | \underbrace{(0, \dots, 0)}_{s+1}$$

The vector t_0 has its $(s+1)$ last coordinates equal to zero. Moreover, its norm is less than $\sqrt{m-s-1}$. As the polynomial $p_1(x,y,z)$ is irreducible, and the gcd of the coefficients of p_2 is d, some elementary row operations on N_I leads to the following matrix:

$$N'_I = \begin{pmatrix} A_1 \mid B \\ \hline A_2 \mid 0 \end{pmatrix} \begin{array}{l} \updownarrow (s+1) \\ \updownarrow (m-s-1) \end{array}$$

where B is a diagonal matrix having 1 on its s first coefficients and d for the last one. If we call $\mathcal{L}'_I$ the lattice generated by the $(m-s-1)$ last rows of the previous matrix, we have $|d \cdot \det \mathcal{L}'_I| = |\det N_I|$. Moreover, the vector t_0 belongs to $\mathcal{L}'_I$. Let take $r = m - s - 1$, and assume that $(c_1, \ldots, c_r)$ is an LLL-reduced basis of $\mathcal{L}'_I$. Thus $\|c_r^\star\| \geq 2^{-(r-1)/4} |\det \mathcal{L}'_I|^{1/r}$. As t_0 belongs to $\mathcal{L}'_I$, when $\|t_0\| < \|c_r^\star\|$, the inner product $< t_0 | c_r^\star >$ is equal to zero that leads to a polynomial $p_3(x, y, z)$ having the same common root as p_1 and p_2. This condition has to be satisfied:

$$\sqrt{m-s-1} < 2^{-\frac{m-s-2}{4}} |\det \mathcal{L}'_I|^{\frac{1}{m-s-1}}$$

Then we can construct p_3 if the following one is verified:

$$\sqrt{m-s}^{(m-s)} 2^{(m-s-1)/4} < |\det \mathcal{L}'_I|$$
$$\sqrt{m-s}^{(m-s)} 2^{(m-s-1)/4} X^{s_x} Y^{s_y} Z^{s_z} < \frac{W_2}{d} \left(W_1^s 2^{-6s(d_x^2+d_y^2+d_z^2)} \right)$$

As X, Y, Z already verify the equation (2), we obtain that if $d < W_2$ (this is always the case), then we can construct p_3. In this case, no further restrictions on the bounds are needed to construct p_3. This polynomial is independent from the ideal $I = (p_1, p_2)$, as explained in section 3.3.

For a well-chosen pair $(M, <)$ (see section 3.2), the previous condition on $\mathcal{F}$ can be stated in terms of the truncated Gröbner basis. More precisely, we should have $G_M = \{p_1, p'_2\}$ and no multiples of p'_2 should be defined over M.[4] In practice, when $G_M = \{p_1, p'_2\}$, the other condition is often true.

5 Application to "Partial key exposure attack on RSA"

In this section, in order to better understand the way the algorithm works in practice, we apply it to one of the partial key exposure attacks on RSA which have been proposed in [9]. We start by describing the basis of this attack in section 5.1 and 5.2.

5.1 The RSA equation

Let $N = pq$ be a RSA modulus. The RSA encryption exponent e and decryption exponent d both satisfy the well-known equation $ed \equiv 1 \mod \phi(N)$ which can be rewritten into $ed = 1 + k(N - (p+q-1))$. We focus on the particular case of a small exponent d but without any restrictions on e except that $e < \phi(N)$. In addition, part of the high order bits of d $(\tilde{d})$ are known to an attacker. As a consequence,

[4] The polynomial p'_2 is obtained by replacing in p_2 all multiples of the initial term of p_1 by multiples of $p_1^{(1)}$ where $p_1^{(1)} = p_1 - in(p_1)$.

d can be rewritten as $\tilde{d} + d_0$ such that $|d| \leq N^\beta$ and $|d_0| = |d - \tilde{d}| \leq N^\delta$. The values of the two parameters β and δ will be used later. Putting these entries into the RSA equation leads to the following polynomial:

$$f_{MSB1}(x, y, z) = ex - yN + yz + R \text{ with } R = e\tilde{d} - 1$$

The problem remains to find the root $(x_0, y_0, z_0) = (d_0, k, p + q - 1)$ of the polynomial $p_1 = f_{MSB1}(x, y, z)$ with $|x_0| < X, |y_0| < Y$ and $|z_0| < Z$ knowing that $X = N^\delta, Y = N^\beta$ and $Z = 3\sqrt{N}$.

5.2 A heuristic attack

We only sketch here the general idea of the attack proposed in [9], for further details, the reader can refer to it. Let m and t be two small integers which are taken in $\{0, 1, 2\}$ for the experiments. Let S and M be two sets of $\mathbb{N}^3$ defined as:

$$S = \{(i, j, k) | (i + j) \leq \mathsf{m}, k \leq j + \mathsf{t}\} \quad M = \{(f, g, h) | (f + g) \leq \mathsf{m} + 1, h \leq g + \mathsf{t}\}$$

By multiplying p_1 by monomials in S and n (a well-chosen integer) by monomials in M, a collection of polynomials is obtained whose Newton polygons are included in M and that have (x_0, y_0, z_0) as root modulo n. A lattice is then constructed with the coefficients of all these polynomials and an LLL reduction is performed. By taking the two shortest vectors of the lattice (under some conditions on the bounds, see [9]) two polynomials p_2 and p_3 can be constructed such that they have (x_0, y_0, z_0) as a root over the integers. If the three resulting polynomials p_1, p_2, p_3 are algebraically independent, it leads to the root by resultant computations. Unfortunately, one can not guarantee the independence, which makes this attack be a heuristic one.

5.3 Our attack

Let us now explain our attack, that is the way we manage to recover the root of $p_1 = f_{MSB1}(x, y, z)$ by using the construction exposed in section 3.3. Starting with two independent polynomials, our construction allows us to construct a third one having the same common root and algebraically independent from the two others. Constructing the first two polynomials from a single one is simply done by using the construction of section 5.2. Indeed, while this construction is heuristic for the third polynomial, it rigorously yields the second one. We denote by p_2 the second polynomial thus found.

Let us consider the ideal $I = (p_1, p_2)$ which has to be prime for our construction. If this is not the case, some preliminary computations have to be performed in order to replace I by another prime ideal which still has (x_0, y_0, z_0) as a root. If I is primary, it is sufficient to replace it by its radical. If I is not primary, we can compute its primary decomposition $I = I_1 \cap \cdots \cap I_r$ and replace it by the corresponding ideal I_j (or $\sqrt{I_j}$ if necessary). In practice, testing each I_j to find the correct one is very fast since there is a small number of such ideals in this decomposition. Finally, from I_j we construct a lattice L_{I_j} using an auxiliary set $\mathcal{F}$ as in section 3.2. After reducing L'_{I_j}, we obtain a third independent polynomial p_3.

The polynomial p_2 is derived from p_1 using Coron's and Howgrave-Graham's variations [13, 8] instead of the original Coppersmith method. As a consequence, we cannot apply the criterion of section 4 to ensure that the construction of p_3 is always easier than the construction of p_2. Nevertheless, it works extremely well in practice.

5.4 Experiments

Let us take N as a 256-bit modulus for the experiments. The following tables show the results we obtained with some fixed values of the parameters m, t and β for both the attack proposed in [9] (which we refer to as "Method 1") and ours ("Method 2"). One hundred polynomials p_1 are created for each value of δ. The first column gives the number of times the original attack only provides one polynomial, the second column refers to the number of times it provides two polynomials. In this case, the number of (p_1, p_2, p_3) really independent is given in column 3. This number is counted too with our method (column 4). The value of δ in bold corresponds to the best bound obtained in practise in [9].

	Method 1			Method 2
δ	p_2	(p_2, p_3)	Indep.	OK
0.09	0	100	98	100
0.10	0	100	92	100
0.11	0	100	95	100
0.12	0	100	92	100
0.13	0	100	80	100
0.132	0	100	86	100
0.134	0	100	77	100
0.136	0	100	71	100
0.138	1	99	75	100
0.140	0	100	71	100
0.142	1	99	72	100
0.144	0	100	55	100
0.146	4	95	57	99
0.148	7	89	50	96
0.150	6	91	43	97

Table 1. $\mathsf{m} = 1, \mathsf{t} = 1, \beta = 0.35$

	Method 1			Method 2
δ	p_2	(p_2, p_3)	Indep.	OK (Root Pb.)
0.14	0	100	100	100 (0)
0.15	0	100	97	100 (0)
0.16	0	100	97	100 (0)
0.17	0	100	82	100 (1)
0.18	0	100	60	100 (8)
0.182	0	100	47	100 (13)
0.184	0	100	47	100 (13)
0.186	0	100	33	100 (26)
0.188	0	100	18	100 (36)
0.190	0	100	16	100 (50)
0.192	0	100	6	100 (79)
0.194	7	82	0	89 (89)
0.196	14	49	0	63 (63)
0.198	4	42	0	46 (46)
0.20	4	25	0	29 (29)

Table 2. $\mathsf{m} = 2, \mathsf{t} = 0, \beta = 0.3$

The first table really show that there are no more problems due to independence. Thus, our method can be applied beyond that of [9]. In the second table, a different problem seems to appear during the computation. This behavior can be explained quite simply. Indeed, as we noticed before, in this application, we can not predict in advance the restrictions on the bounds in order to obtain p_3 such that $p_3(x_0, y_0, z_0) = 0$. Surprisingly, this is not a problem to recover the root (see the next section).

5.5 Special cases of interest

We saw in the previous section that even if there are sometimes root problems, it does not prevent us to recover the root. The reason why is that, in all cases, the Gröbner basis of the ideal $I = (p_1, p_2)$ is so simple that it allows to recover the root, without needing a third polynomial. Let us show some examples where the "root problem" appears for I prime, primary and non-primary. These toys examples use the tiny parameter $N \simeq 2^{50}$ with $\beta = 0.3$, $\mathsf{m} = 2$, $\mathsf{t} = 0$, and $\delta = 0.190$.

***I* is prime** The initial parameters are:

$$p_1 = 9450886190201x + ((z - 155155341747587)y + 72582805940743679)$$
$$(x_0 = 233, y_0 = 482, z_0 = 25517171) \rightarrow (X = 496, Y = 18080, Z = 37368409)$$

After using the polynomial p_2, which is too large to print, coming from the attack of [9], we have the following Gröbner basis:

$$\begin{cases} q_1 = xz - 39521501447/12x + 46079/6z + 6785552382017/12 \\ q_2 = y - 12/197x - 92158/197 \end{cases}$$

In particular, the polynomial q_2 has (x_0, y_0, z_0) as a root. By multiplying all its coefficients by 197 and taking the equation modulo 197, we find $x_0 \equiv 36 \mod 197$. By testing then $36, 233$, we find the root.

***I* is primary** The initial parameters are:

$$p_1 = -32390526593433x + ((z - 96130883093383)y - 215591345005890049)$$
$$(x_0 = 87, y_0 = -2272, z_0 = 20056623) \rightarrow (X = 453, Y = 15661, Z = 29413906)$$

The polynomial p_2 is taken to construct $I = (p_1, p_2)$. As this ideal is not prime, it is replaced by its radical, what gives:

$$\begin{cases} r_1 = xz - 128929299037/31x + 206327/31z + 7024533450267/31 \\ r_2 = y + 31/92x + 206327/92 \end{cases}$$

The polynomial r_2 has (x_0, y_0, z_0) as a root. By multiplying it by 92 and taking the equation modulo 92, we obtain that $x_0 \equiv 87 \mod 92$. We find the root $x_0 = 87$.

***I* is non-primary** The initial parameters are:

$$p_1 = 1581190442669x + ((z - 3199926510559)y + 7690910313142015)$$
$$(x_0 = 165, y_0 = 2485, z_0 = 4282719) \rightarrow (X = 237, Y = 5642, Z = 5366501)$$

By taking p_2, we consider $I = (p_1, p_2)$ which is not a primary ideal. Its primary decomposition gives the two following prime ideals:

$$\begin{cases} q_1 = xz + 4274183387/42x + 29185/6z - 268309596605/7 \\ q_2 = y - 42/85x - 40859/17 \end{cases}$$

$$\begin{cases} q_1' = xz + 4274183387/42x + 34049/7z + 1590068930929/42 \\ q_2' = y - 42/85x - 204294/85 \end{cases}$$

By checking which of the two previous ideals has (x_0, y_0, z_0) as root, we find that it is the first one. In particular, the polynomial q_2 has the right root. By taking the equation modulo 85, we obtain that $x_0 \equiv 80 \mod 85$. This gives the right root $x_0 = 165$.

Some comments First of all, it seems for the previous examples to work very well because of the size of the parameters, but in fact we have the same behaviors with $N \simeq 2^{256}$. As the previous equations only give the modular value of x_0 and not its integer value, some tests have to be performed to recover the root. In almost cases, it has to be tested less than five times (we even often recover the root directly). There are nevertheless some cases where the value of x_0 is more difficult to find. Another important point to notice is that in almost cases, the ideals have the shape of those studied previously. It means that we can recover the root with only two polynomials instead of three, except for very rare cases.

6 Possible generalizations in more variables

We expose here a method to replace the heuristic that appears in all articles concerning small roots of polynomial equations in three variables by weaker conditions. We show that the method can, in principle, be generalized to more variables. Starting with an irreducible polynomial p_1 having $(x_{0,1}, \ldots, x_{0,n})$ as root, the classical Coppersmith's method provides a second polynomial p_2 that has the same root and that is independent from p_1. For each $j \in \{3, \ldots, n\}$, considering the ideal $I_{j-1} = (p_1, \ldots, p_{j-1})$, the polynomial p_j can be constructed such that $p_j \notin I_{j-1}$. If the ideal I_{j-1} is prime, the polynomials $p_1, \ldots, p_j$ are algebraically independent. As a consequence, using a successive sequence of prime ideals, we can obtain n polynomials algebraically independent and which have the same common root, that leads to $(x_{0,1}, \ldots, x_{0,n})$.

7 Conclusion

The main result of this paper is a new variation of Coppersmith's algorithm on three variables, that uses both lattice reduction and Gröbner bases computations. In general, the success of this method is controlled by the shape of the Gröbner basis of the ideal $I = (p_1, p_2)$ produced by a straight adaption of Coppersmith's algorithm to the trivariate case. This is a first important step toward rigorous applications of Coppersmith's method with more than two variables. We also show how variations on our technique can improve applications of cryptographic interest.

Open problems are to generalize the method to more applications and to determine general criteria yielding rigorous variants of Coppersmith's algorithm with a wide range of applicability.

References

1. M. Bardet. *Etude de sytèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie*. PhD thesis, University of Paris 6, 2004.
2. J. Blömer and A. May. Low Secret Exponent RSA Revisited. In *CaLC '01: Revised Papers from the International Conference on Cryptography and Lattices*, pages 4–19, London, UK, 2001. Springer-Verlag.
3. J. Blömer and A. May. A Tool Kit for Finding Small Roots of Bivariate Polynomials over the Integers. *Proceedings of Eurocrypt 2005, Lecture Notes in Computer Science*, 3494:251–257, 2005.
4. D. Boneh and G. Durfee. Cryptanalysis of RSA with Private Key Less Than $N^{0.292}$. *IEEE Transactions on Information Theory*, 46:1339–1349, July 2000.
5. D. Coppersmith. Finding a Small Root of a Bivariate Integer Equation; Factoring with high bits known. In *Advances in Cryptology-Eurocrypt '96, Lecture Notes in Computer Science*, volume 1070, pages 178–189. Springer-Verlag, 1996.
6. D. Coppersmith. Finding a Small Root of a Univariate Modular Equation. In *Advances in Cryptology-Eurocrypt '96, Lecture Notes in Computer Science*, volume 1070, pages 155–165. Springer Verlag, 1996.
7. D. Coppersmith. Finding Small Solutions to Small Degree Polynomials. In *Cryptography and Lattice Conference, Lecture Notes in Computer Science*, volume 2146. Springer-Verlag, 2001.
8. J.-S. Coron. Finding Small Roots of Bivariate Integer Polynomial Equations Revisited. In *Advances in Cryptology-Eurocrypt '04, Lecture Notes in Computer Science*, pages 492–505. Springer-Verlag, 2004.
9. M. Ernst, E. Jochemsz, A. May, and B.de Weger. Partial Key Exposure Attacks on RSA up to Full Size Exponents. *In Advances in Cryptology (Eurocrypt 2005), Lecture Notes in Computer Science Volume 3494, pages 371-386, Springer-Verlag*, 2005.
10. J.-C. Faugère. A New Efficient Algorithm for Computing Gröbner Bases (F4). *Journal of Pure and Applied Algebra*, (139):61–88, 1999.
11. M. J. Hinek. New partial key exposure attacks on RSA revisited. Technical report, CACR, Centre for Applied Cryptographic Research, University of Waterloo, 2004.
12. M. J. Hinek. Small Private Exponent Partial Key-Exposure Attacks on Multiprime RSA. Technical report, CACR, Centre for Applied Cryptographic Research, University of Waterloo, 2005.
13. N. Howgrave-Graham. Finding Small Roots of Univariate Modular Equations Revisited. In *Proceedings of the 6th IMA International Conference on Cryptography and Coding*, pages 131–142, London, UK, 1997. Springer-Verlag.
14. N. Howgrave-Graham. Approximate Integer Common Divisor. In *CaLC '01: Lecture Notes in Computer Science*, volume 2146, pages 51–66. Springer-Verlag, 2001.
15. A.K. Lenstra, Jr. H.W. Lenstra, and L. Lovasz. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:513–534, 1982.

A First iteration using a basic variation of Coppersmith's method on three variables

Let $p_1(x, y, z)$ be an irreducible polynomial of $\mathbb{Z}[x, y, z]$ having (x_0, y_0, z_0) as root over the integers such that $|x_0| < X$, $|y_0| < Y$ and $|z_0| < Z$. Let S and M be sets of monomials over $\mathbb{N}^3$.

Theorem 1. *If S and M are admissible sets for p_1, we can find in polynomial time $p_2(x, y, z)$ which has (x_0, y_0, z_0) as a root over the integers and is algebraically independent from p_1, provided that*

$$X^{s_x} Y^{s_y} Z^{s_z} < W_1^s 2^{-(6+c)s(d_x^2+d_y^2+d_z^2)} \tag{3}$$

where we assume that $(m - s)^2 \leq cs(d_x^2 + d_y^2 + d_z^2)$ for some constant c.

A.1 Preliminaries

We denote by $\bar{P}_1$ the $(m \times s)$ matrix defined in section 2.4 whose columns refer to the coefficients of $x^i y^j z^k \tilde{p_1}$ for all $(i, j, k) \in S$. The following result holds:

Lemma 1. *There exists a subset $\hat{M} \subset M$ of size s such that if $\hat{P}_1$ is the matrix composed by the rows of $\bar{P}_1$ corresponding to monomials in $\hat{M}$, we have*

$$|\det \hat{P}_1| \geq W_1^s 2^{-6s(d_x^2+d_y^2+d_z^2)}$$

We omit this proof because it follows the same idea as in [5], the point is that we work on three variables instead of two.

A.2 Construction of the lattice $\mathcal{L}_1$

The $(m \times s)$ matrix whose columns represent the polynomials $x^i y^j z^k p_1$ for all $(i, j, k) \in S$ is denoted by P_1. Moreover, we call D_M the $(m \times m)$ diagonal matrix whose entries are $X^{-f} Y^{-g} Z^{-h}$ with $(f, g, h) \in M$. $\mathcal{L}_1$ is the lattice generated by the rows of the following matrix:

$$N_1 = \begin{pmatrix} D_{M \setminus \hat{M}} \mid P_1 \end{pmatrix}$$

where $D_{M \setminus \hat{M}}$ is the resulting matrix coming from deletion in D_M of the columns related to monomials in $\hat{M}$. One can observe that the definition of $\mathcal{L}_1$ is different from those of L_1, which has been introduced in section 2.4. In fact, the same explanation holds with this definition, however, in this case, the conditions on the bounds are easier to determine. By multiplying the rows of N_1 related to $(f, g, h) \in M$ by $X^f Y^g Z^h$ and the columns of P_1 related to $(i, j, k) \in S$ by $X^{-i} Y^{-j} Z^{-k}$, a matrix $\bar{N}_1$ is constructed and satisfies:

$$|\det \bar{N}_1| = |\det N_1| X^{s_x} Y^{s_y} Z^{s_z}$$

Making some elementary row operations on $\bar{N}_1$ leads to:

$$\begin{pmatrix} Id & A \\ 0 & \hat{P}_1 \end{pmatrix}$$

whose determinant is equal to $(\det \hat{P}_1)$. Thus, we obtain that

$$|\det N_1| \geq X^{-s_x} Y^{-s_y} Z^{-s_z} W_1^s 2^{-6s(d_x^2+d_y^2+d_z^2)}$$

A.3 Using LLL-reduction to construct p_2

Let consider the vector $r_0 = (\boldsymbol{x_0^f y_0^g z_0^h})_{\boldsymbol{M}}$ and $s_0 = r_0 N_1$. We have

$$s_0 = ((\frac{x_0}{X})^f (\frac{y_0}{Y})^g (\frac{z_0}{Z}))^h{}_{M \setminus \hat{M}} | (\underbrace{0, \ldots, 0}_{s})$$

This vector satisfies the two following conditions :

- $\|s_0\|_2 \leq \sqrt{m-s}$
- Its s last coordinates are equal to 0.

As the polynomial $p_1(x, y, z)$ is irreducible, some elementary row operations on N_1 leads to the following matrix:

$$N_1' = \begin{pmatrix} A_1 \mid Id \\ \hline A_2 \mid \ 0 \end{pmatrix} \begin{array}{l} \updownarrow s \\ \updownarrow m - s \end{array}$$

If we call $\mathcal{L}_1'$ the lattice generated by the $(m-s)$ last rows of the previous matrix, we obtain $|\det \mathcal{L}_1'| = |\det N_1|$. Moreover, s_0 belongs to $\mathcal{L}_1'$. Let take $r = m - s$, and assume that $(b_1, \ldots, b_r)$ is an LLL-reduced basis of $\mathcal{L}_1'$. We know that $\|b_r^\star\| \geq 2^{-(r-1)/4} |\det \mathcal{L}_1'|^{1/r}$. As s_0 is a vector belonging to the lattice $\mathcal{L}_1'$, when $\|s_0\| < \|b_r^\star\|$, the inner product $< s_0 | b_r^\star >$ is equal to zero, that leads to a polynomial $p_2(x, y, z)$ which has the same root as $p_1(x, y, z)$. The following condition has to be satisfied:

$$\sqrt{m-s} < 2^{-\frac{m-s-1}{4}} |\det \mathcal{L}_1'|^{\frac{1}{m-s}}$$

to allow the construction of p_2. Let see the more restrictive condition:

$$\sqrt{m-s} < 2^{-\frac{m-s-1}{4}} (2^{-6s(d_x^2+d_y^2+d_z^2)} W_1^s X^{-s_x} Y^{-s_y} Z^{-s_z})^{\frac{1}{m-s}}$$

Finally, if $X^{s_x} Y^{s_y} Z^{s_z} < W_1^s 2^{-(6+c)s(d_x^2+d_y^2+d_z^2)}$ is verified (for c a constant such that $(m-s)^2 \leq cs(d_x^2 + d_y^2 + d_z^2)$), the polynomial p_2 can be constructed in polynomial time. With this construction, the monomials of p_2 belong to $M \setminus \hat{M}$. It remains to prove that the polynomial p_2 is independent from p_1. In fact, if this is not the case, the vector related to p_2 is a linear combination of the columns of P_1. Knowing that p_2 is orthogonal to all multiples of p_1, this can not be possible.

B Algebraic independence between p_1, p_2 and p_3

Here is the proof of the result given in section (2.6). If the ideal $I = (p_1, p_2)$ is prime and $p_3 \notin I$, we show that p_1, p_2 and p_3 are algebraically independent. Assume there exists a polynomial P defined over $\mathbb{Q}[x, y, z]$ such that $P(p_1, p_2, p_3) = 0$, our goal is to prove that $P = 0$. In the following, we denote by $\Delta(P)$ the set of all points $(a, b, c) \in \mathbb{N}^3$ such that $x^a y^b z^c$ appears in P with a non-zero coefficient.

Starting with $\sum_{(a,b,c)\in\Delta(p)} \lambda_{(a,b,c)} p_1^a p_2^b p_3^c = 0$, the polynomial $Q(x)$ can then be defined as $\sum_{c=0}^{n} \mu_c x^c$ with $\mu_c = \sum_{(a,b)|(a,b,c)\in\Delta(P)} \lambda_{(a,b,c)} p_1^a p_2^b$. This polynomial also has p_3 as a root and we can assume that $\mu_0 \neq 0$, otherwise the polynomial $Q(x)$ can be replaced by $Q(x)/x$. As p_1 and p_2 are already algebraically independent, proving that $Q = 0$ implies that $P = 0$.

The first step of the proof is to show that $\mu_0, \dots, \mu_n \in I$. Indeed, let us take $Q(p_3)$ evaluated in (x_0, y_0, z_0). Knowing that p_3 has (x_0, y_0, z_0) as a root, it implies that $\mu_0(x_0, y_0, z_0) = 0$. As a consequence, its constant coefficient $\lambda_{(0,0,0)}$ is equal to zero and then $\mu_0 \in I$. Using the equation $Q(p_3) = 0$, we obtain $p_3(\mu_1 + \cdots + \mu_n p_3^{n-1}) \in I$. As the ideal I is prime and as $p_3 \notin I$, we have $\mu_1 + \cdots + \mu_n p_3^{n-1} \in I$. Evaluating again this quantity in (x_0, y_0, z_0) leads to $\mu_1(x_0, y_0, z_0) = 0$, that implies $\mu_1 \in I$. We can then go on the proof by doing the same for $\mu_2, \dots, \mu_n$.

The previous results allow us to rewrite each μ_c as $\mu_c = p_1\ F_c(p_1, p_2) + p_2\ G_c(p_2)$ with $F_c \in \mathbb{Q}[x, y]$ such that $deg_x(F_c) < deg_x(\mu_c)$ and G_c a polynomial of $\mathbb{Q}[x]$ defined by $G_c(x) = \sum_{i=0}^{g_c} l_{c,i} x^i$. The equation $Q(p_3) = 0$ becomes then:

$$p_1\left(F_0(p_1, p_2) + \cdots + F_n(p_1, p_2) p_3^n\right) = \underbrace{-p_2(G_0(p_2) + \cdots + G_n(p_2) p_3^n)}_{\in (p_1)} \qquad (4)$$

As the ideal (p_1) is prime and as p_2 does not belong to (p_1), it implies that $G_0(p_2) + \cdots + G_n(p_2) p_3^n \in (p_1) \subset (p_1, p_2)$. As before, by evaluating this expression in (x_0, y_0, z_0), we can show that $l_{0,0}, \dots, l_{n,0} = 0$. It implies that the polynomials $G_c(x)$ can be expressed as $G_c(x) = p_2 G_{c,2}(x)$ with $G_{c,2}(x)$ defined as $\sum_{i=0}^{g_c-1} l_{c,i+1} x^i$. As a consequence, the right hand part of the equation (4) which belongs to (p_1), can be rewritten into $-p_2^2(G_{0,2}(p_2) + \cdots + G_{n,2}(p_2) p_3^n)$. By the same explanation, we finally show that $G_c(x) = 0$ for all $c \in \{0, \dots, n\}$.

Using the previous result, we can then rewrite μ_c as $p_1 F_c(p_1, p_2)$. The equation $Q(p_3) = 0$ becomes:

$$p_1\left(\sum_{c=0}^{n} F_c(p_1, p_2) p_3^c\right) = 0 \Rightarrow R(p_3) = \sum_{c=0}^{n} \nu_c p_3^c = 0$$

with $\nu_c = F_c(p_1, p_2)$. The polynomial $R(x)$ satisfies $Q(x) = p_1 R(x)$ and the coefficients ν_c are such that $deg_x(\nu_c) < deg_x(\mu_c)$. We then separate again ν_c as $p_1 H_c(p_1, p_2) + p_2 I_c(p_2)$ and we show that $I_c = 0$ for all $c \in \{0, \dots, n\}$. By recurrence, we finally obtain that $Q(x) = p_1^k V(x)$ with $V(x)$ a polynomial defined over $\mathbb{Q}[x]$ which has p_3 as a root. It implies that $V = 0$, and thus $P = 0$. This concludes the proof.