A Tool Kit for Finding Small Roots of Bivariate Polynomials over the Integers

Johannes Blömer, Alexander May

Faculty of Computer Science, Electrical Engineering and Mathematics
University of Paderborn
33102 Paderborn, Germany
bloemer,alexx@uni-paderborn.de

Abstract. We present a new and flexible formulation of Coppersmith's method for finding small solutions of bivariate polynomials $p(x, y)$ over the integers. Our approach allows to maximize the bound on the solutions of $p(x, y)$ in a purely combinatorial way. We give various construction rules for different shapes of $p(x, y)$'s Newton polygon. Our method has several applications. Most interestingly, we reduce the case of solving univariate polynomials $f(x)$ modulo some composite number N of unknown factorization to the case of solving bivariate polynomials over the integers. Hence, our approach unifies both methods given by Coppersmith at Eurocrypt 1996.

Keywords: Coppersmith's method, univariate vs. bivariate, RSA

1 Introduction

In 1996, Coppersmith [6–9] introduced two rigorous lattice-based methods for finding small roots of polynomials: One for univariate modular and another one for bivariate integer polynomial equations. Additionally, Coppersmith proposed heuristic multivariate extensions for both approaches. The goal in both methods is to maximize the bounds up to which roots of the polynomials can be found in polynomial time. Coppersmith's method for finding small solutions of modular polynomial equations has been applied in many settings, mainly for cryptanalytic purposes [1, 3, 4, 11] but also for proving the security of schemes [2, 15].

In contrast, the method for finding roots of polynomial equations over the integers has not found so many applications, yet. The most well-known result is the so-called factoring with high bits known [7, 8]: Let $N = pq$ be an RSA modulus and suppose we are given half of the high-order bits of p, then N can be factored in polynomial time. Recently, May [13] gave another application for the bivariate method: He showed that if the RSA secret key is known, then N can be factored in deterministic polynomial time. However, both results can also be proven using univariate polynomial equations.

In 1997, Howgrave-Graham [10] gave an easily applicable reformulation of Coppersmith's univariate modular method. This might be one of the reasons that up to now the univariate modular approach has found more applications than the bivariate integer approach. At Eurocrypt '04, Coron [5] succeeded to give a similar reformulation of Coppersmith's method over the integers.

While it is clear how to optimize a lattice basis for a given univariate polynomial of fixed degree, the construction of an optimal lattice basis for a bivariate polynomial $p(x, y)$ depends on the monomials that appear in $p(x, y)$. Coppersmith [8] analyzed the cases where $p(x, y)$ either has degree δ in x and y separately or degree δ in total.

Let us define the Newton polygon of $p(x, y)$ as the convex hull of the point set

$$\{(i, j) \in \mathbb{N}^2 \mid \text{monomial } x^i y^j \text{ appears in } p(x, y) \text{ with non-zero coefficient}\}.$$

For $p(x, y)$ with degree δ in each variable separately, the shape of the Newton polygon is a square. For $p(x, y)$ with total degree δ, the shape is an equilateral lower triangle (having his right angle in the lower left corner). These two shapes were also analyzed by Coron [5]. In addition, Coppersmith [8] mentions the case where the maximal degree of $p(x, y)$ in x is δ_x and the maximal degree in y is δ_y, which corresponds to a rectangle with side lengths δ_x and δ_y.

In this work, we provide a method that can be used to analyze arbitrary shapes of the Newton polygon of $p(x, y)$. One advantage of our main result is that we can formulate it just in terms of the monomials of $p(x, y)$. Although the proof of our main result requires lattice-based techniques, using our theorem the analysis of different shapes of $p(x, y)$ is purely combinatorial and can be done without any lattice theory. Hence, one can view our approach as a tool kit: If we are given a polynomial $p(x, y)$, we can maximize the bounds up to which a solution can be found in polynomial time. More precisely, let X and Y be upper bounds on the desired roots of $p(x, y)$. I.e., we want to find all solutions (x_0, y_0) such that $p(x_0, y_0) = 0$ and $|x_0| \leq X$, $|y_0| \leq Y$. Our goal is to maximize X and Y. The formulation of our main theorem allows to specify this maximization problem as an optimization problem over two sets of monomials. No lattice theory is required and the theorem can be used as a black box for cryptanalysts.

The proof of our main theorem is a variation of Coppersmith's original proof for the bivariate method [8]. We could use Coron's approach [5] for the proof of our result as well, but we prefer Coppersmith's approach since it has a crucial advantage: We usually obtain bounds of the form $XY \leq W^{g(\delta)-\epsilon}$, where $g(\delta)$ is some function in the degree of $p(x, y)$ in x, y and $W = \|p(xX, yY)\|_\infty$ is the max-norm of the coefficient vector of $p(xX, yY)$. The running time of Coppersmith's algorithm is polynomial in $(\log W, \delta, \frac{1}{\epsilon})$, while Coron's approach is polynomial in $(\log W, \delta)$ but exponential in $\frac{1}{\epsilon}$. This difference is due to a clever trick of Coppersmith which significantly reduces the dimension of the lattice involved by considering only a certain sublattice.

As applications of our main result, we provide rules to analyze different shapes of a Newton polygon of $p(x, y)$, thereby deriving some of the most well-known cryptographic results of Coppersmith's method. Hence, one can also see our new method as a unifying method for certain different approaches to find small roots of polynomial equations. In particular, we obtain the following results for different shapes of the Newton polygons:

Rectangle: The rectangle can be seen as a warm-up example. Let us define $W = \|p(xX, yY)\|_\infty$. For polynomials of degree δ in each variable separately, we show the Coppersmith bound [8]

$$XY \leq W^{\frac{2}{3\delta}-\epsilon}.$$

Lower triangle: We analyze $p(x, y)$ with variable degree in x and y. When the total degree of $p(x)$ is δ, we obtain Coppersmith's bound [8]

$$XY \leq W^{\frac{1}{\delta}-\epsilon}.$$

Moreover, let us consider a univariate modular polynomial equation $f(x) = 0 \bmod N$, where f has degree δ. This can also be written as a bivariate polynomial $p(x, y) = f(x) - yN$ over the integers. The shape of $p(x, y)$'s Newton polygon is also a lower triangle, but with side-lengths δ and 1.

Our analysis shows that one can find all roots (x_0, y_0) of $p(x, y)$ over the integers provided that

$$|x_0| \leq N^{\frac{1}{\delta}},$$

which is exactly Coppersmith's result for univariate modular equations [8]. This unifies both approaches of Coppersmith from Eurocrypt '96 [6, 7]: The univariate modular case is already included in the bivariate integer case.

Surprisingly, the lattice basis underlying this result does not use powers of the polynomial $p(x, y)$, whereas in the univariate modular case it seems necessary to use powers of $p(x)$ in order to achieve the bound $N^{\frac{1}{\delta}}$.

Upper triangle: To our knowledge, the shape of an upper triangle (where the right angle is in the upper right corner) has not been analyzed in the literature before.

We use this shape to analyze the factorization algorithm for RSA-moduli $N = p^r q$, $r \geq 1$ of Boneh, Durfee and Howgrave-Graham [4]. In the original work, this is done using a variant of Coppersmith's univariate approach, namely one works modulo the divisor p^r of N. Interestingly, one can solve equations modulo p^r although one knows only N. Boneh, Durfee and Howgrave-Graham propose to exhaustively search approximations $\tilde{p}$ of p. For each guess $\tilde{p}$, they try to solve the polynomial equation $(\tilde{p} + x)^r = 0 \bmod p^r$, which has the solution $p - \tilde{p}$.

Alternatively, for each guess $\tilde{p}$ we consider the bivariate polynomial $f(x, y) = (\tilde{p} + x)^r y - N$ with the solution $(x_0, y_0) = (p - \tilde{p}, q)$. Notice that the shape of $f(x, y)$'s Newton polygon is an upper triangle. Our analysis yields the same result as the one in the work of Boneh, Durfee and Howgrave-Graham: One can find the factorization of N provided that

$$|x_0| \leq N^{\frac{r}{(r+1)^2}}.$$

Surprisingly, for $r > 1$ the following approach gives a smaller bound: Compute $\tilde{q} = \frac{N}{\tilde{p}}$ and try to solve the polynomial $f'(x,y) = (\tilde{p}+x)^r(\tilde{q}+y) - N$. Let X, Y be upper bounds on the desired solution $(x_0, y_0) = (p - \tilde{p}, q - \tilde{q})$. At first glance, the polynomial $f'(x,y)$ seems to be superior since we can decrease the size of Y. On the other hand, $W = \|p(xX, yY)\|_\infty$ decreases as well and the shape of $f'(x,y)$'s Newton polygon now is a rectangle, which has an inferior analysis. These two facts together outweigh the benefit of decreasing Y and we obtain a smaller bound.

In the case $r = 1$, both approaches give the same bound $|x_0| = |p - \tilde{p}| \le N^{\frac{1}{4}}$. But still, the first approach should be preferred in practice since it uses a smaller lattice basis. So counterintuitively, one should sometimes ignore information about one variable in order to obtain a better shape of the Newton polygon. As the moral of this story, one should keep in mind that optimizing Coppersmith's bivariate method is not only a matter of optimizing the bounds X, Y but also of optimizing the structure of the underlying polynomial $p(x,y)$ itself!

In addition to the results above, we also prove general bounds for univariate polynomials of degree δ modulo some divisor b of N. The bounds are functions of the sizes of δ, b and N.

Rectangle and lower triangle: As a last example, we show how to combine two basic shapes such that all results for rectangles and/or for lower triangles follow as special cases by parameter settings.

We expect that similar to Coppersmith's approach [8] our bivariate method extends to a heuristic method for general multivariate equations, but we have not checked this so far.

The paper is organized as follows: In Section 2, we give our main result that allows to formulate the maximization problem of X and Y as an optimization problem for sets of monomials. In Section 3, we formulate our construction rules for the different shapes of Newton polygons of $p(x,y)$. Applications of these shapes are given in Section 4.

2 The Main Theorem

In this section we state our main theorem. We also describe the general setting in which we are going to apply the theorem in the following sections. First we need a couple of preliminary remarks and definitions.

Let M be a set of monomials in the variables x, y. We say that a polynomial $g(x,y)$ is *defined over M* or is a *polynomial over M* iff $g(x,y)$ can be written as

$$g(x,y) = \sum_{\mu \in M} c_\mu \mu, c_\mu \in \mathbb{Z}.$$

The proof of our main result uses a certain resultant that is required to be non-zero. In order to prove this property, the following definition is going to be useful. Later we will elaborate on this definition.

Definition 1 *Let $p(x,y)$ be a bivariate integer polynomial and S, M be finite non-empty sets of monomials in the variables x, y. The sets S, M are called* admissible *for $p(x,y)$ iff*

1. *For every monomial $\alpha \in S$ the polynomial $\alpha \cdot p(x,y)$ is defined over M.*
2. *For every polynomial g defined over M, if $g(x,y) = f(x,y) \cdot p(x,y)$ for some polynomial f, then f is defined over S.*

We say that an integer polynomial $p(x,y) \in \mathbb{Z}[x,y]$ is *irreducible* if $p(x,y) = f(x,y) \cdot g(x,y)$ with $f(x,y), g(x,y) \in \mathbb{Z}[x,y]$ implies that either $f(x,y) = \pm 1$ or $g(x,y) = \pm 1$. In particular, the gcd of all coefficients of an irreducible polynomial $p(x,y)$ must be 1.

Using these definitions we can already state our main theorem. Its proof can be found in the full version of the paper.

Theorem 2 *Let $p(x,y) \in \mathbb{Z}[x,y]$ be an irreducible integer polynomial in two variables with degree at most $d_x, d_y \geq 1$ in the variables x and y, respectively. Let $X, Y \in \mathbb{N}$ and set $W := \|p(xX, yY)\|_\infty$. Furthermore let $S, M, S \subseteq M$, be admissible for $p(x,y)$. Set*

$$s := |S|, \qquad m := |M|$$

$$s_x := \sum_{x^i y^j \in M \setminus S} i, \qquad s_y := \sum_{x^i y^j \in M \setminus S} j.$$

All pairs $(x_0, y_0) \in \mathbb{Z}^2$ satisfying

$$p(x_0, y_0) = 0 \quad \text{with } |x_0| \leq X, |y_0| \leq Y$$

can be found in time polynomial in m, d_x, d_y and $\log(W)$ provided

$$X^{s_x} Y^{s_y} < W^s \cdot 2^{-(8+c)sd_x d_y}, \tag{1}$$

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

In the following we call elements of the set S *shift monomials*. The set S itself will be called *the set of shift monomials*. Let us describe how we are going to apply Theorem 2. To do so, we will identify sets of monomials with sets in the Euclidean plane $\mathbb{R}^2$. More precisely, for a set A of monomials in two variables x, y we define $\{(i,j) \in \mathbb{N}^2 | x^i y^j \in A\}$ and the convex hull $\mathrm{conv}(\{(i,j) \in \mathbb{N}^2 | x^i y^j \in A\})$ of this set. To simplify the notation we call these sets A as well. It will always be clear from the context whether we talk about a set of monomials or about the corresponding sets in the plane. Next, for a polynomial $g(x,y) = \sum c_{ij} x^i y^j, c_{ij} \in \mathbb{R}$ we define a convex set $N(g)$ in the Euclidean plane, called the *Newton polygon* of g. We set

$$N(g) := \mathrm{conv}\{(i,j) \in \mathbb{N}^2 | c_{ij} \neq 0\}.$$

The Newton polygon of the polynomial $p(x,y) = 2 + y + 3xy$ is depicted in Fig. 1.

Now suppose we want to use Theorem 2 to determine roots of some polynomial $p(x,y)$. Of course, we want to choose the bounds X, Y as large as possible.

To do so, we need to choose sets S and M carefully under the constraint that S, M are admissible for $p(x, y)$. Once we have chosen S, there is an obvious choice for M in order to guarantee the first property in Definition 1. That is, we choose M as the set of monomials $x^i y^j$ such that (i, j) lies in the so-called *Minkowski sum* $N(p) + S$ of the Newton polygon $N(p)$ and S. Here the Minkowski sum $A+B$ of two sets A, B in R^2 is defined as

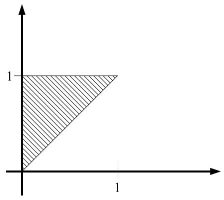

Fig. 1. Newton polygon of $2 + y + 3xy$

$$A+B := \{(a_1, a_2)+(b_1, b_2) \mid (a_1, a_2) \in A, (b_1, b_2) \in B\}.$$

As will be seen in our applications of Theorem 2, setting $M := N(p) + S$ will usually lead to a pair S, M of sets of monomials that also satisfies the second property of Definition 1, i.e. S, M will be admissible for the polynomial $p(x, y)$.

It remains to explain how to choose S in order to achieve large bounds X, Y, that satisfy Equation (1) in Theorem 2. Choosing good sets S requires a trade-off between the size s of S and the quantities s_x, s_y that depend on monomials in $M \setminus S$, where $M = N(p) + S$. We want s to be large, while s_x and s_y should stay relatively small. We have no provable method to find optimal sets S. However, the following general strategy proves to be successful.

We consider a whole class of sets S, that may be parametrized by several parameters. The shape of these sets resembles $N(p)$. Given these parametrized sets we determine the values s, s_x, s_y as functions of the parameters used to describe the sets. Finally, based on Equation (1) we determine the optimal setting for our parameters in order to get sets S, M and large bounds X, Y satisfying the conditions of Theorem 2.

3 The Constructions

Let us explain the construction of parametrized sets S for a few important shapes of Newton polygons $N(p)$ of polynomials $p(x, y)$. Applications of these examples and analysis of the bounds for X and Y that we can derive using these constructions will be given in the following section. First we define some important geometric shapes.

Definition 3 *In the following all parameters are real positive numbers.*

1. *Sets* $R(a, b) := \{x^i y^j \mid 0 \le j \le a, 0 \le i \le b\}$ *are called rectangles.*
2. *Sets* $L(c, a, \lambda) := \{x^{c+i} y^j \mid 0 \le j \le a, 0 \le i \le \lambda(a - j)\}$ *are called lower triangles.*
3. *Sets* $U(c, a, \lambda) := \{x^{c+i} y^j \mid 0 \le j \le a, 0 \le i \le \lambda j\}$ *are called upper triangles.*
4. *Sets* $E(c, a, \lambda) := R(a, c) \cup L(c, a, \lambda)$ *are called extended rectangles.*

Illustrations for these definitions are given in Fig. 2.
With these definitions we can state our main constructions.

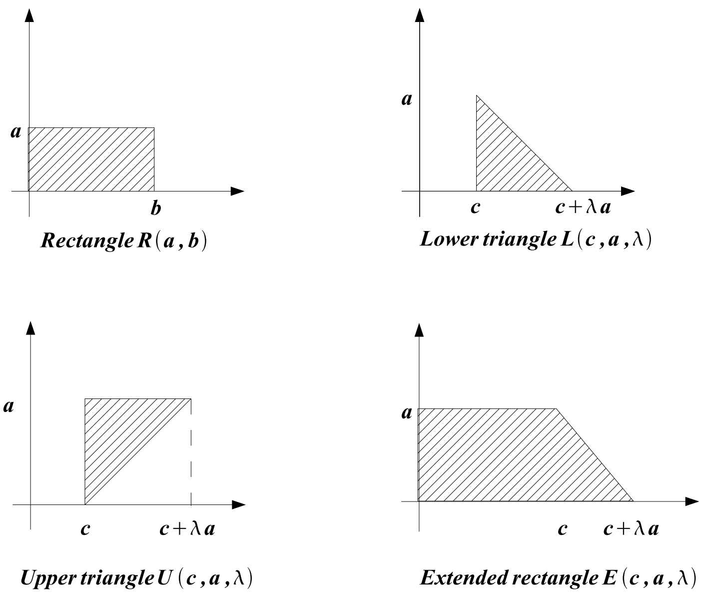

Fig. 2. Illustrations for Definition 3

Construction 4 (Rectangle construction) *Assume the Newton polygon $N(p)$ of polynomial $p(x,y)$ is the rectangle $R(d, \lambda d), \lambda > 0$. Then we use sets S such that*

$$x^i y^j \in S \Leftrightarrow (i,j) \in R(k, \gamma k).$$

Here $k \in \mathbb{N}$ and $\gamma > 0$. Consequently, the sets M of monomials are defined by

$$x^i y^j \in M \Leftrightarrow (i,j) \in R(k+d, \gamma k + \lambda d).$$

Furthermore

$$s = \sum_{j=0}^{k} \sum_{i=0}^{\gamma k} 1, \qquad m = \sum_{j=0}^{k+d} \sum_{i=0}^{\gamma k + \lambda d} 1$$

$$s_x = \sum_{j=0}^{k+d} \sum_{i=0}^{\gamma k + \lambda d} i - \sum_{j=0}^{k} \sum_{i=0}^{\gamma k} i, \qquad s_y = \sum_{j=0}^{k+d} \sum_{i=0}^{\gamma k + \lambda d} j - \sum_{j=0}^{k} \sum_{i=0}^{\gamma k} j.$$

In this construction the parameter γ is used to optimize the bounds X, Y.

In the rectangle construction as well as in the subsequent constructions, the parameter k is not used to optimize X, Y. Mainly it is used to control the size of certain low order error terms.

As it turns out the optimal γ is given by $\sqrt{\lambda}$, not by λ itself. Using the convex hulls of S and M instead of S, M itself, this construction is shown in Fig. 3.

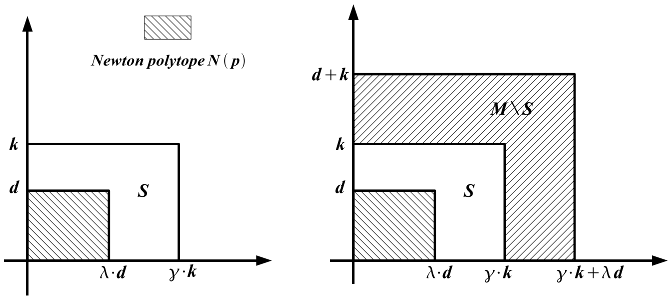

Fig. 3. The rectangle construction

Similarly, we define constructions for the lower and upper triangle, shown in Fig. 4. In the lower triangle construction we need no parameter to optimize the bounds X, Y.

Construction 5 (Lower triangle construction) *Assume the Newton polygon $N(p)$ of polynomial $p(x,y)$ is the lower triangle $L(0,d,\lambda), \lambda > 0$. Then we use sets S such that*

$$x^i y^j \in S \Leftrightarrow (i,j) \in L(0,k,\lambda).$$

Here $k \in \mathbb{N}$. Consequently, the sets M of monomials are defined by

$$x^i y^j \in M \Leftrightarrow (i,j) \in L(0,k+d,\lambda).$$

Using Definition 3, the formulas for s, m, s_x, and s_y can expressed in a similar fashion as in the rectangle construction.

Construction 6 (Upper triangle construction) *Assume the Newton polygon $N(p)$ of polynomial $p(x,y)$ is the upper triangle $U(0,d,\lambda), \lambda > 0$. Then we use sets S such that*

$$x^i y^j \in S \Leftrightarrow (i,j) \in R(k,ck) \cup U(ck,k,\lambda).$$

Here $k \in \mathbb{N}$ and $c \geq 0$. Consequently, the sets M of monomials are defined by

$$x^i y^j \in M \Leftrightarrow (i,j) \in R(k+d,ck) \cup U(ck,k+d,\lambda).$$

Again using Definition 3, the formulas for s, m, s_x, and s_y can expressed in a similar fashion as in the rectangle construction.

Of course, one can combine some or even all of these constructions into a single construction using several parameters to describe the shapes of $N(p)$ and S. For example, combining the rectangle and the lower triangle construction leads to the *extended rectangle construction.* This construction is shown in Fig. 5.

Our applications of Theorem 2 only use the constructions defined above. The following lemma shows that these constructions always yield admissible sets S and M. Hence in the subsequent sections we need not worry about the admissibility of the sets S and M that are used.

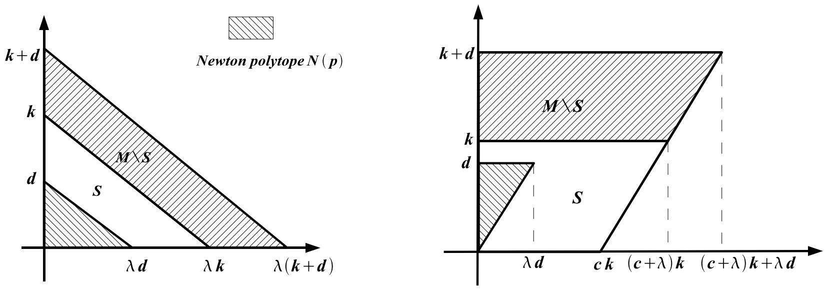

Fig. 4. Lower and upper triangle construction

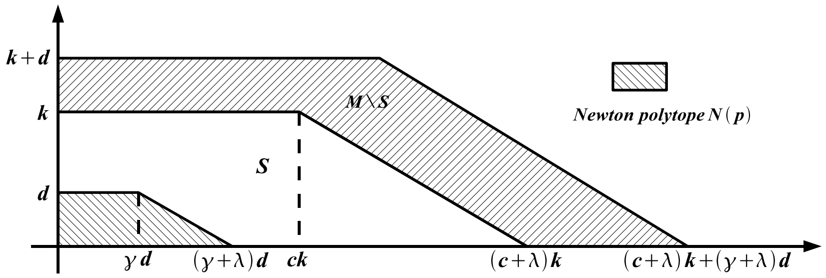

Fig. 5. The extended rectangle construction

Lemma 7 *The rectangle, lower triangle, upper triangle, and extended rectangle constructions as defined above lead to admissible sets S and M for the respective polynomials.*

Proof: We only show the lemma for the rectangle construction. The proofs for the other constructions are similar. As mentioned above, since M is the Minkowski sum of $N(p)$ and S, the sets S, M have the first property of Definition 1. To see that S, M also have the second property, consider a polynomial $f(x, y) = \sum f_{ij} x^i y^j$ that is not defined over S. We need to show that $f(x, y) \cdot p(x, y)$ is not defined over M. By l_x, l_y denote the degree of f in x, y, respectively. Since $f(x, y)$ is not defined over S, we have that $l_x > \gamma k$ or $l_y > k$. Since the two cases are symmetric, we only consider the case that $l_y > k$.

Let g be maximal over all i with $f_{il_y} \neq 0$. Then the coefficient of $x^{i+\lambda d} y^{l_y+d}$ in $f(x, y) \cdot p(x, y)$ will be non-zero. Since $l_y > k$ we get $l_y + d > k + d$ and $x^{i+\lambda d} y^{l_y+d} \notin M$. Hence $f(x, y) \cdot p(x, y)$ is not defined over M. $\sqcup\!\!\!\!\sqcap$

4 Applications of Our Method

The following lemma is due to Coppersmith [8]. It is often used in the subsequent proofs to remove small error terms from the bounds. Namely, whenever we have a bound of B for the size of our solution, we can enlarge this bound to cB by doing some brute-force search. This search increases the time complexity also by a factor of c.

Lemma 8 (Coppersmith) *Let $p(x, y) \in \mathbb{Z}[x, y]$. Assume that we have an algorithm A that finds all pairs $(x_0, y_0) \in \mathbb{Z}^2$ satisfying*

$$p(x_0, y_0) = 0 \quad \text{with } |x_0 \cdot y_0| \leq B$$

in time complexity T. Then one can find all (x_0, y_0) satisfying

$$p(x_0, y_0) = 0 \quad \text{with } |x_0 \cdot y_0| \leq cB$$

in time complexity cT.

Proof: We split our interval $[-cB, cB]$ into c subintervals of the size $2B$ centered at some x_i. For each of the subintervals with center x_i, we apply algorithm A to the polynomial $p(x - x_i, y)$ and output the roots in this subinterval. $\square$

By Lemma 8, whenever we derive a bound of $B2^{-\mathcal{O}(\delta)}$ in the following theorems, we can also derive a bound of B by increasing the time complexity by a factor polynomial in 2^δ.

4.1 Rectangular Shape

We start by analyzing the case, where $p(x, y)$ has degree δ in x and y seperately.

Theorem 9 (Coppersmith) *Let $p(x, y) \in \mathbb{Z}[x, y]$ be an irreducible polynomial of degree δ in each variable separately. Let $X, Y \in \mathbb{N}$ and define $W = \|p(xX, yY)\|_\infty$. Then we can find all pairs $(x_0, y_0) \in \mathbb{Z}^2$ satisfying*

$$p(x_0, y_0) = 0 \quad \text{with } |x_0| \leq X, |y_0| \leq Y$$

in time polynomial in $\log W$ and δ provided that

$$XY \leq W^{\frac{2}{3\delta}} 2^{-\mathcal{O}(\delta)}.$$

Proof: Since the Newton polygon of our polynomial $p(x, y)$ is a rectangle, we apply Construction 4. We use the parameter setting

$$k = \max\{\log W, \delta\}, \ \gamma = 1 \text{ and } \lambda = 1.$$

According to Construction 4, we shift our polynomial $p(x, y)$ with all the monomials in $S = R(k, k)$. Let $M = R(k + \delta, k + \delta)$. By Lemma 7, the sets S and M are admissible for $p(x, y)$ and Theorem 2 is applicable.

Plugging our values of $\gamma = \lambda = 1$ in the formulas for s_x, s_y, s and m gives us

$$s_x = s_y = \frac{3\delta}{2}k^2\Big(1 + \mathcal{O}\Big(\frac{\delta}{k}\Big)\Big),\ s \geq k^2 \text{ and } s, m = \mathcal{O}(k^2).$$

Furthermore, we have $d_x = d_y = \delta$. One easily checks the condition $(m-s)^2 = \mathcal{O}(sd_xd_y)$ of Theorem 2. An application of Theorem 2 with the values of s_x, s_y, s, d_x and d_y leaves us with the condition

$$(XY)^{\frac{3\delta}{2}k^2(1+\mathcal{O}(\frac{\delta}{k}))} \leq W^{k^2}2^{-\mathcal{O}(k^2\delta^2)}$$

This implies the bound

$$XY \leq W^{\frac{k^2}{\frac{3\delta}{2}k^2(1+\mathcal{O}(\frac{\delta}{k}))}}2^{-\mathcal{O}(\delta)}.$$

Now we observe that for any x, we have $\frac{1}{1+x} \leq 1 - x$. Therefore, we can bound the exponent of W by $\frac{2}{3\delta}(1 - \mathcal{O}(\frac{\delta}{k}))$. This leads to the new condition

$$XY \leq W^{\frac{2}{3\delta}}W^{-\mathcal{O}(\frac{1}{k})}2^{-\mathcal{O}(\delta)}.$$

Since we chose $k \geq \log W$, our term $W^{\mathcal{O}(\frac{1}{k})}$ is of constant size. An application of Lemma 8 shows that we can omit this term by increasing the running time only by a constant factor. This concludes the proof of the theorem. $\square$

4.2 Lower Triangular Shape

First, we state the case where $p(x, y)$ has total degree δ. The proof of the following theorem can be found in the full version of the paper.

Theorem 10 (Coppersmith) *Let $p(x, y) \in \mathbb{Z}[x, y]$ be an irreducible polynomial of total degree δ. Let $X, Y \in \mathbb{N}$ and define $W = \|p(xX, yY)\|_\infty$. Then we can find all pairs $(x_0, y_0) \in \mathbb{Z}^2$ satisfying*

$$p(x_0, y_0) = 0 \quad \text{with } |x_0| \leq X, |y_0| \leq Y$$

in time polynomial in $\log W$ and δ provided that

$$XY \leq W^{\frac{1}{\delta}}2^{-\mathcal{O}(\delta)}.$$

Next, let us analyze the case $p(x, y) = f(x) - yN$, where $f(x)$ is a univariate polynomial of degree δ. This is exactly the univariate modular case and the following result reduces Coppersmith's univariate modular method [6] to the bivariate integer method [7].

In order to state Theorem 11, we use the following notation: Let $a_1, a_2, \ldots, a_n \in \mathbb{Z}$. We denote by $\gcd(a_1, a_2, \ldots, a_n)$ the greatest integer that divides all $a_i, i = 1 \ldots n$.

Theorem 11 (Coppersmith) *Let N be a composite integer of unknown factorization. Let $f(x) = \sum f_i x^i \in \mathbb{Z}[x]$ be a polynomial of degree δ with* $\gcd(f_1, f_2, \ldots, f_\delta, N) = 1$. *Furthermore, let $X \in \mathbb{N}$. Then we can find all point $x_0 \in \mathbb{Z}$ satisfying*

$$f(x_0) = 0 \bmod N \quad \textit{with } |x_0| \leq X$$

in time polynomial in $\log N$ and δ provided that

$$X \leq N^{\frac{1}{\delta}}.$$

Proof: We define the following bivariate polynomial

$$p(x, y) = f_N(x) - yN,$$

where $f_N(x) = f(x) \bmod N$. I.e., we reduce the coefficients of $f(x)$ modulo N. Notice that x_0 is a root of $f(x_0)$ modulo N iff $p(x, y)$ has the root (x_0, y_0) for some y_0 over the integers. Furthermore, $p(x, y)$ is irreducible. Since we reduced $f(x)$ by N, we can upper bound the size of y_0 by

$$|y_0| \leq \frac{|f_N(x_0)|}{N} \leq X^\delta + X^{\delta-1} + \cdots + X^0 \leq (\delta + 1)X^\delta.$$

Let us define $Y = (\delta + 1)X^\delta$. Then we obtain $W = \|f(xX, yY)\|_\infty = YN$.

The shape of the Newton polygon of $p(x, y)$ is a lower triangle. Therefore, we apply Construction 5. Here, we use the parameter setting

$$k = \max\{\log W, \delta\},\ d = 1 \text{ and } \lambda = \delta.$$

That means, we apply the shifts with the monomials in $S = L(0, k, \delta)$ to the polynomial $f(x, y)$. Let $M = L(0, k + \delta, \delta)$. By Lemma 7 the sets S and M are admissible for $p(x, y)$, and Theorem 2 is applicable.

Setting the values $d = 1$ and $\lambda = \delta$ in our formulas for s_x, s_y, s and m provides us with the bounds

$$s_x = \frac{\delta^2}{2}k^2\Big(1 + \mathcal{O}\Big(\frac{1}{k}\Big)\Big),\ s_y = \frac{\delta}{2}k^2\Big(1 + \mathcal{O}\Big(\frac{1}{k}\Big)\Big),\ s \geq \frac{\delta}{2}k^2 \text{ and } s, m = \mathcal{O}(\delta k^2).$$

Furthermore, we observe that $d_x = \delta$ and $d_y = 1$. One easily checks that our parameters satisfy the condition $(m - s)^2 = \mathcal{O}(s d_x d_y)$ of Theorem 2.
Using these values in combination with Theorem 2 leads to the condition

$$X^{\frac{\delta^2}{2}k^2(1+\mathcal{O}(\frac{1}{k}))} Y^{\frac{\delta}{2}k^2(1+\mathcal{O}(\frac{1}{k}))} \leq W^{\frac{\delta}{2}k^2} 2^{-\mathcal{O}(\delta^2 k^2)}$$

Since $W = YN$, we obtain

$$X^{\frac{\delta^2}{2}k^2(1+\mathcal{O}(\frac{1}{k}))} \leq N^{\frac{\delta}{2}k^2} Y^{-\mathcal{O}(\delta k)} 2^{-\mathcal{O}(\delta^2 k^2)}$$

Analogous to the reasoning in the proof of Theorem 9, this implies the bound

$$X \leq N^{\frac{1}{\delta}} N^{-\mathcal{O}(\frac{1}{\delta k})} Y^{-\mathcal{O}(\frac{1}{\delta k})} 2^{-\mathcal{O}(1)}. \tag{2}$$

By our setting, we have $k \geq \log W$ which bounds the term $(NY)^{-\mathcal{O}(\frac{1}{\delta k})} = W^{-\mathcal{O}(\frac{1}{\delta k})}$ by a constant. An application of Lemma 8 shows that we can increase the bound in (2) to the desired bound $X \leq N^{\frac{1}{\delta}}$ by increasing the running time by a constant factor.

By Theorem 2, we know that the running time of our algorithm is polynomial in $\log W$ and δ. It remains to show that $\log W$ is also a polynomial in $\log N$ and δ. Since our condition in inequality (2) implies that $X \leq N^{\frac{1}{\delta}}$, we have $W = YN = (\delta+1)X^{\delta}N \leq (\delta+1)N^2$ or equivalently $\log W \leq \log(\delta+1) + 2\log N$. This concludes the proof of the theorem. $\square$

4.3 Upper Triangular Shape

In this subsection, we analyze a variant of Coppersmith's univariate modular approach, where one solves polynomial equations modulo a divisor of N. We start by reproducing the Boneh, Durfee and Howgrave-Graham [4] lattice-based factoring for RSA-moduli $N = p^r q$, $r \geq 1$, which is a generalization of "factoring with high bits known" of Coppersmith [8].

Theorem 12 (BDH) *Let $N = p^r q$ be an RSA modulus, where p and q are primes of the same bit-size and $r \geq 1$ is an integer. Suppose we are given an approximation $\tilde{p}$ of p with*

$$|\, p - \tilde{p} \,| \leq N^{\frac{r}{(r+1)^2}}.$$

Then we can find the factorization of N in time polynomial in $\log N$ and r.

Proof: We define the polynomial

$$f(x, y) = (\tilde{p} + x)^r y - N.$$

with the root $(x_0, y_0) = (p - \tilde{p}, q)$. Let $X = N^{\frac{r}{(r+1)^2}}$, then by our assumption $|x_0| \leq X$. Now, let us also find an upper bound Y for the size of $y_0 = q$. Since p and q are of the same bit-size, we know that $p > \frac{q}{2}$. Therefore, we obtain $q = \frac{N}{p^r} < \frac{2^r N}{q^r}$ which gives us $q^{r+1} < 2^r N$. This yields the upper bound $q < 2N^{\frac{1}{r+1}}$. Thus, we set $Y = 2N^{\frac{1}{r+1}}$. Obviously, we have $W = \|f(xX, yY)\|_{\infty} \geq N$.

Since the structure of the Newton polygon of our polynomial $f(x, y)$ is an upper triangle, we apply Construction 6. Here we use the parameter setting

$$k = \max\{\log N, r\}, d = 1, \lambda = r \text{ and } c = 1.$$

Thus, we use the shifts of the polynomial $f(x, y)$ with all the monomials in $S = R(k, k) \cup U(k, k, r)$. Let $M = R(k+1, k) \cup U(k, k+1, r)$. By Lemma 7, the sets S and M are admissible. Therefore, Theorem 2 is applicable.

Plugging the values $d = 1, \lambda = r$ and $c = 1$ into our formulas for s_x, s_y, s and m yields

$$s_x = \tfrac{(r+1)^2}{2} k^2 \Big(1 + \mathcal{O}\Big(\tfrac{1}{k}\Big)\Big), s_y = \big(r+1\big) k^2 \Big(1 + \mathcal{O}\Big(\tfrac{1}{k}\Big)\Big)$$
$$s \geq \Big(\tfrac{r}{2} + 1\Big) k^2 \text{ and } s, m = \mathcal{O}(rk^2)$$

Furthermore, we have $d_x = r$ and $d_y = 1$. One can check that these parameters meet the condition $(m-s)^2 = \mathcal{O}(sd_xd_y)$ of Theorem 2.
Now we apply Theorem 2 with the above parameters, which gives us

$$X^{\frac{(r+1)^2}{2}k^2(1+\mathcal{O}(\frac{1}{k}))} Y^{(r+1)k^2(1+\mathcal{O}(\frac{1}{k}))} \leq W^{(\frac{r}{2}+1)k^2} 2^{-\mathcal{O}(r^2k^2)}.$$

Using $Y = 2N^{\frac{1}{r+1}}$ and $W \geq N$ leads to the new condition

$$X^{\frac{(r+1)^2}{2}k^2(1+\mathcal{O}(\frac{1}{k}))} \leq N^{\frac{r}{2}k^2 - \mathcal{O}(k)} 2^{-\mathcal{O}(r^2k^2)}.$$

This in turn gives us

$$X \leq N^{\frac{\frac{r}{2}k^2}{\frac{(r+1)^2}{2}k^2(1+\mathcal{O}(\frac{1}{k}))}} N^{-\mathcal{O}(\frac{1}{r^2k})} 2^{-\mathcal{O}(1)},$$

which can be transformed into

$$X \leq N^{\frac{r}{(r+1)^2}} N^{-\mathcal{O}(\frac{1}{rk})} 2^{-\mathcal{O}(1)}.$$

Since $k \geq \log N$, an application of Lemma 8 gives us the desired bound $X \leq N^{\frac{r}{(r+1)^2}}$ by an increase of the running time by a constant factor. $\square$

For the special case $r = 1$, we use the polynomial $p(x, y) = (\tilde{p} + x)y - N$ in the analysis of the proof of Theorem 12. In contrast, Coppersmith [8] proposed to use the polynomial $p'(x, y) = (\tilde{p} + x)(\tilde{q} + y) - N$, where $\tilde{q} = \frac{N}{\tilde{p}}$.

For $r = 1$, both polynomials give the same bound (but $p(x, y)$ yields smaller lattice bases, so it should lead to a faster algorithm in practice). Interestingly, for $r > 1$ the polynomial $(\tilde{p} + x)^r y - N$ yields a better bound than its counter-part with $\tilde{q}$, although we have to increase the bound on y_0. But this disadvantage is outweighed by the fact that the shape of $p(x, y)$ is upper triangular rather than rectangular, and that we can increase W to N.

In the following theorem, we analyze the more general case where we want to solve a univariate polynomial $f(x)$ with $f(x_0) = \bar{c}b$ for some small root x_0 and some (unknown) divisor b of N. Here, we assume that $\bar{c}$ is a known constant. By the result of the theorem, a large $\bar{c}$ helps to improve the bound. Unfortunately, we are not aware of an application with $\bar{c} > 1$.

Theorem 13 *Let N be a composite integer of unknown factorization with divisor $b \geq N^{\beta}$. Let $f(x) = \sum f_i x^i \in \mathbb{Z}[x]$ be a polynomial of degree δ with* $\gcd(f_1, f_2, \ldots, f_\delta, \bar{c}N) = 1$. *Then we can find all points $x_0 \in \mathbb{Z}$ satisfying $f(x_0) = \bar{c}b$ for some known constant $\bar{c} = N^{\gamma}, \gamma \geq 0$ in time polynomial in $\log N, \delta$ and γ provided that*

$$|x_0| \leq N^{\frac{(\beta+\gamma)^2}{\delta(1+\gamma)}}.$$

Proof: We define the following bivariate polynomial

$$p(x,y) = f(x)y - \bar{c}N.$$

Notice that $p(x, \frac{N}{b})$ has the same roots as $f(x) - \bar{c}b$ over the integers. Furthermore, $p(x,y)$ is irreducible. Define $y_0 = \frac{N}{b}$. Since $b \geq N^{\beta}$, we know that $y_0 \leq N^{1-\beta}$. Let $Y = N^{1-\beta}$ denote this upper bound for y_0.

Next, we will determine all integer roots (x_0, y_0) of $p(x,y)$ with the property that $|x_0| \leq X$ and $|y_0| \leq Y$. Among these roots must be all roots of $f(x) - \bar{c}b$. (It may happen that we additionally find roots of $f(x) - \bar{c}b'$ for some other divisor b' of N.)
We observe that $W = \|f(xX, yY)\|_\infty \geq \bar{c}N$.

Notice that the structure of the Newton polygon of $p(x,y)$ is an upper triangle. Therefore, we apply Construction 6. In this case, we use the parameter setting

$$k = \max\{\log N, \delta, \gamma\},\ d = 1,\ \lambda = \delta \text{ and } c = \frac{(1-\beta)\delta}{\beta+\gamma}$$

That means that we shift the polynomial $p(x,y)$ with all the monomials in the set $S = R(k, ck) \cup U(ck, k, \delta)$. Let $M = R(k+1, ck) \cup U(ck, k+1, \delta)$. By Lemma 7 the sets S and M are admissible for $p(x,y)$. Therefore, Theorem 2 is applicable.

If we plug in the values of d, λ and c in our formulas for s_x, s_y, s and m, we obtain

$$s_x = \tfrac{\delta^2(1+\gamma)^2}{2(\beta+\gamma)^2} k^2 \big(1 + \mathcal{O}(\tfrac{1}{k})\big),\ s_y = \tfrac{\delta(1+\gamma)}{\beta+\gamma} k^2 \big(1 + \mathcal{O}(\tfrac{1}{k})\big),$$
$$s \geq \tfrac{\delta(2-\beta+\gamma)}{2(\beta+\gamma)} k^2 \text{ and } s, m = \mathcal{O}(\delta k^2)$$

Notice that $d_x = \delta$ and $d_y = 1$. We easily check that the condition $(m-s)^2 = \mathcal{O}(s d_x d_y)$ of Theorem 2 is satisfied.
Using $Y = N^{1-\beta}$ and $W \geq \bar{c}N = N^{1+\gamma}$, an application of Theorem 2 yields

$$X^{\frac{\delta^2(1+\gamma)^2}{2(\beta+\gamma)^2} k^2 (1+\mathcal{O}(\frac{1}{k}))} N^{\frac{\delta(1+\gamma)(2-2\beta)}{2(\beta+\gamma)} k^2 (1+\mathcal{O}(\frac{1}{k}))} \leq N^{\frac{\delta(1+\gamma)(2-\beta+\gamma)}{2(\beta+\gamma)} k^2} 2^{-\mathcal{O}(\delta^2 k^2)}.$$

This can be rewritten as

$$X^{\frac{\delta^2(1+\gamma)^2}{2(\beta+\gamma)^2} k^2 (1+\mathcal{O}(\frac{1}{k}))} \leq N^{\left(\frac{\delta(1+\gamma)(2-\beta+\gamma)}{2(\beta+\gamma)} - \frac{\delta(1+\gamma)(2-2\beta)}{2(\beta+\gamma)}\right) k^2} N^{-\mathcal{O}(\delta k)} 2^{-\mathcal{O}(\delta^2 k^2)},$$

which simplifies to

$$X^{\frac{\delta^2(1+\gamma)^2}{2(\beta+\gamma)^2} k^2 (1+\mathcal{O}(\frac{1}{k}))} \leq N^{\frac{\delta(1+\gamma)}{2} k^2} N^{-\mathcal{O}(\delta k)} 2^{-\mathcal{O}(\delta^2 k^2)}$$

This in turn gives us the new condition

$$X \leq N^{\frac{(\beta+\gamma)^2}{\delta(1+\gamma)}} N^{-\mathcal{O}(\frac{1}{\delta k})} 2^{-\mathcal{O}(1)}$$

Since $k \geq \log N$, an application of Theorem 8 yields the desired bound. $\square$

As the special case $\bar{c} = 1$ of Theorem 13, we obtain the following corollary.

Corollary 14 *Let N be a composite integer of unknown factorization with divisor $b \geq N^\beta$. Let $f(x) = \sum f_i x^i \in \mathbb{Z}[x]$ be a polynomial of degree δ with $\gcd(f_1, f_2, \ldots, f_\delta, N) = 1$. Then we can find all points $x_0 \in \mathbb{Z}$ satisfying $f(x_0) = b$ in time polynomial in $\log N$ and δ provided that*

$$|x_0| \leq N^{\frac{\beta^2}{\delta}}.$$

An application of Corollary 14 is again "factoring with high bits known" [8]: Let $N = pq$ with $p > q$. Define $f(x) = \tilde{p} + x$. We want to find $x_0 = p - \tilde{p}$ with $f(x_0) = p$. We have $p \geq N^{\frac{1}{2}}$, which implies $\beta = \frac{1}{2}$. Hence, we obtain the well-known bound $|x_0| \leq N^{\frac{1}{4}}$.

Another application is the deterministic reduction of May [13]: Let $N = pq$ be an RSA modulus and let (e, d) satisfy $ed = 1 \bmod \phi(N)$. Suppose, we are given (N, e, d). Define $f(x) = N - x$. We want to find $x_0 = p + q - 1 \approx N^{\frac{1}{2}}$ with $f(x_0) = \phi(N)$. Notice that we know the multiple $ed - 1$ of $\phi(N)$. Let $ed - 1 = N^\alpha$ with $\alpha \leq 2$. Then we can set $\beta = \frac{1}{\alpha}$. Therefore, we can recover x_0 as long as $|x_0| \leq N^{\frac{1}{\alpha}}$. Since $\alpha \leq 2$, our bound is at least of the desired size $N^{\frac{1}{2}}$.

Similar to the case of "factoring with high bits known", the reduction yields another polynomial than originally proposed by May. Here, we obtain the polynomial $p(x, y) = (N - x)y + 1 - ed$, whereas May suggested to use $p'(x, y) = (N - x)(\tilde{k} + y) + 1 - ed$ with $\tilde{k} = \frac{ed-1}{N}$. Again, we can ignore the knowledge provided by $\tilde{k}$ in the analysis without affecting the bound. As before, $p(x, y)$ should be preferred in practice since it yields smaller lattice bases.

We want to point out that a result similar to the bound given in Corollary 14 has been given by Howgrave-Graham [11]. He showed a bound of N^{β^2} for solving $f(x) = 0 \bmod b$, where $f(x)$ has degree 1. This was later generalized by May [14] to $N^{\frac{\beta^2}{\delta}}$ for $f(x)$ of degree δ. Notice that these approaches allow to solve $f(x) = c'b$ for some *unknown* c' as opposed to $f(x) = \bar{c}b$ for some *known* $\bar{c}$ as in Theorem 13.

We pose the open problem to reduce this case of unknown c' to the bivariate integer case or a provable trivariate integer case. To our knowledge, this is the only rigorous variant of Coppersmith's method which is not covered by our new approach.

References

1. J. Blömer, A. May, “New Partial Key Exposure Attacks on RSA”, Advances in Cryptology – Crypto 2003, Lecture Notes in Computer Science Vol. 2729, pp. 27–43, Springer-Verlag, 2003
2. D. Boneh, “Simplified OAEP for the RSA and Rabin Functions”, Advances in Cryptology – Crypto 2001, Lecture Notes in Computer Science Vol. 2139, pp. 275–291, Springer-Verlag, 2001
3. D. Boneh, G. Durfee, “Cryptanalysis of RSA with private key d less than $N^{0.292}$”, IEEE Trans. on Information Theory, Vol. 46(4), pp. 1339–1349, 2000
4. D. Boneh, G. Durfee, and N. Howgrave-Graham, “Factoring $N = p^r q$ for large r”, Advances in Cryptology – Crypto ’99, Lecture Notes in Computer Science Vol. 1666, Springer-Verlag, pp. 326–337, 1999
5. J.-S. Coron, “Finding Small Roots of Bivariate Integer Polynomial Equations Revisited”, Advances in Cryptology – Eurocrypt ’04, Lecture Notes in Computer Science Vol. 3027, Springer-Verlag, pp. 492–505,2004
6. D. Coppersmith, “Finding a Small Root of a Univariate Modular Equation”, Advances in Cryptology – Eurocrypt ’96, Lecture Notes in Computer Science Vol. 1070, Springer-Verlag, pp. 155–165, 1996
7. D. Coppersmith, “Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known”, Advances in Cryptology – Eurocrypt ’96, Lecture Notes in Computer Science Vol. 1070, Springer-Verlag, pp. 178–189, 1996
8. D. Coppersmith, “Small solutions to polynomial equations and low exponent vulnerabilities”, Journal of Cryptology, Vol. 10(4), pp. 223–260, 1997.
9. D. Coppersmith, “Finding Small Solutions to Small Degree Polynomials”, Cryptography and Lattice Conference (CaLC 2001), Lecture Notes in Computer Science Volume 2146, Springer-Verlag, pp. 20–31, 2001.
10. N. Howgrave-Graham, “Finding small roots of univariate modular equations revisited”, Proceedings of Cryptography and Coding, Lecture Notes in Computer Science Vol. 1355, Springer-Verlag, pp. 131–142, 1997
11. N. Howgrave-Graham, “Approximate Integer Common Divisors”, Cryptography and Lattice Conference (CaLC 2001), Lecture Notes in Computer Science Vol. 2146, Springer-Verlag, pp. 51–66, 2001
12. A. K. Lenstra, H. W. Lenstra, and L. Lovász, ”Factoring polynomials with rational coefficients,” Mathematische Annalen, Vol. 261, pp. 513–534, 1982
13. A. May, “Computing the RSA Secret Key is Deterministic Polynomial Time Equivalent to Factoring”, Advances in Cryptology – Crypto ’04, Lecture Notes in Computer Science Vol. 3152, Springer Verlag, pp. 213–219, 2004
14. A. May, “Secret Exponent Attacks on RSA-type Schemes with Moduli $N = p^r q$”, Practice and Theory in Public Key Cryptography – PKC 2004, Lecture Notes in Computer Science Vol. 2947, Springer-Verlag, pp. 218–230, 2004
15. V. Shoup, “OAEP Reconsidered”, Advances in Cryptology – Crypto 2001, Lecture Notes in Computer Science Vol. 2139, Springer-Verlag, pp. 239–259, 1998