A Strategy for Finding Roots of Multivariate Polynomials with New Applications in Attacking RSA Variants

Ellen Jochemsz[1**] and Alexander May[2]

[1] Department of Mathematics and Computer Science,
TU Eindhoven, 5600 MB Eindhoven, the Netherlands
e.jochemsz@tue.nl
[2] Faculty of Computer Science
TU Darmstadt, 64289 Darmstadt, Germany
may@informatik.tu-darmstadt.de

Abstract. We describe a strategy for finding small modular and integer roots of multivariate polynomials using lattice-based Coppersmith techniques. Applying our strategy, we obtain new polynomial-time attacks on two RSA variants. First, we attack the Qiao-Lam scheme that uses a Chinese Remaindering decryption process with a small difference in the private exponents. Second, we attack the so-called Common Prime RSA variant, where the RSA primes are constructed in a way that circumvents the Wiener attack.

Keywords: lattices, small roots, Coppersmith's method, RSA variants, cryptanalysis

1 Introduction

Since Coppersmith introduced new ways of finding small modular and integer roots of polynomials in 1996 [4,5,6], variations of these methods have been widely used in the field of cryptanalysis. Let us give an example that demonstrates the usefulness of computing small roots. In the case of RSA, the public variables (N, e) and the secret variables (d, p, q) satisfy the relation

$$ed - 1 = k(N - (p + q - 1)), \text{ for some (unknown) } k.$$

It is known that one can use Coppersmith techniques to try to find the integer root $(d, k, p + q - 1)$ of the polynomial $f(x, y, z) = ex - yN + yz - 1$, and hence recover the factorization of N. Alternatively, one could look for the modular root $(k, p + q - 1)$ of $f_e(y, z) = y(N - z) + 1$ modulo e.

** The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the author's views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

The success of the application of a Coppersmith technique depends on the size of the root. More precisely, the analysis of the attack results in a bound on the size of roots that can be found with this method in polynomial time. For the case of finding the root $(y^{(0)}, z^{(0)}) = (k, p+q-1)$ of $f_e(y,z) = y(N-z)+1$ modulo e in the example above, Boneh and Durfee [1] used a Coppersmith technique to obtain the bound

$$Y^{2+3\tau} Z^{1+3\tau+3\tau^2} < e^{1+3\tau}, \text{ for } |y^{(0)}| < Y, \text{ and } |z^{(0)}| < Z,$$

where $\tau > 0$ can be optimized once the sizes of Y, Z, and e are known. This led Boneh and Durfee to show that for $d < N^{0.284}$ the secret RSA parameters can be recovered in polynomial time, which they later refined to $d < N^{0.292}$ in the same work [1].

Since the analysis of a polynomial f of which we wish to find a small root heavily depends on the monomials that appear in f, each new polynomial has to be analyzed anew. This is typically a tedious and non-trivial task. In 2005, Blömer and May [3] showed how to find optimal bounds for small integer roots of bivariate polynomials. In this paper we present a heuristic strategy that applies to all multivariate polynomials; having either modular or integer roots.

We apply our strategy to derive new heuristic attacks on two RSA variants, using a polynomial that arises in their cryptanalysis. In the first system, the Chinese Remainder Theorem is used in the decryption phase, with the special property that $d_p \equiv d \mod (p-1)$ and $d_q \equiv d \mod (q-1)$ have a fixed difference $d_p - d_q$. This scheme was proposed in 1998 by Qiao and Lam [17] who suggested to use the small difference $d_p - d_q = 2$. The benefit of the Qiao-Lam scheme is that one has to store only one out of the two keys d_q, d_q and the small difference itself. Up to now, the best attack on the Qiao-Lam scheme was a meet-in-the-middle attack with time and space complexity $\tilde{\mathcal{O}}\{\sqrt{\min\{d_p, d_q\}}\}$ [17].

Qiao and Lam proposed to use a 1024-bit modulus N with 128-bit d_p, d_q. Moreover, they argued that in practice 96-bit private exponents should provide sufficient security. Our results show that private exponents up to $N^{0.099}$ can be recovered in polynomial time. Hence, for 1024-bit RSA moduli one can recover 96-bit d_p, d_q in polynomial time. Furthermore, attacking 128-bit private exponents should also be feasible with our attack by adding some brute force search on the most significant bits. We confirm the validity of our heuristic attack by providing several experiments. Although recovering 96-bit private exponents can theoretically be done in polynomial time, in practice it turns out to be a non-trivial task since it requires an LLL-lattice basis reduction [13] in large dimension.

We would like to point out that our attack works whenever $\max\{d_p, d_q\} \leq N^{0.099-\epsilon}$ for some arbitrarily small constant ϵ, and the difference $d_p - d_q$ is known to the attacker. We do not require that the difference $d_p - d_q$ itself is a small constant like in the Qiao-Lam scheme.

As a second application of our strategy, we give a new attack on an RSA variant called Common Prime RSA. This variant was originally proposed by Wiener [19] as a countermeasure for his attack on small secret exponents $d \leq N^{\frac{1}{4}}$.

The suggestion is to choose p, q such that $p-1$ and $q-1$ share a large gcd. In 1995, Lim and Lee [12] used this Common Prime RSA variant in a server-aided RSA protocol, which was attacked in 1998 by McKee and Pinch [15]. Recently, Hinek [9] revisited the Common Prime RSA variant. He proposed several RSA parameter settings with secret exponents less than $N^{\frac{1}{4}}$. However, our second heuristic attack shows that parts of the proposed key space lead to polynomial time attacks on RSA. We demonstrate the practicality of our second attack by providing several experiments that recover the RSA secret information.

2 Finding Small Roots

In this section we describe some tools that we use to solve the problem of finding small roots, for both the modular and the integer case. Moreover, we present our new strategy.

In [4,5,6], Coppersmith describes rigorous techniques to find small integer roots of polynomials in a single variable modulo N, and polynomials in two variables over the integers. The methods extend to more variables, making them heuristical. Howgrave-Graham reformulated Coppersmith's ideas of finding modular roots in [11], of which we use the following (generalized) lemma.

Lemma 1 (Howgrave-Graham). *Let $h(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$ be an integer polynomial that consists of at most ω monomials. Suppose that*

(1) $h(x_1^{(0)}, \ldots, x_n^{(0)}) \equiv 0 \bmod N$ *for some* $|x_1^{(0)}| < X_1, \ldots, |x_n^{(0)}| < X_n$, *and*
(2) $||h(x_1X_1, \ldots, x_nX_n)|| < \frac{N}{\sqrt{\omega}}$.

Then $h(x_1^{(0)}, \ldots, x_n^{(0)}) = 0$ holds over the integers.

In Lemma 1 the norm of a polynomial $f(x_1, \ldots, x_n) = \sum a_{i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}$ is the Euclidean norm of its coefficient vector: $||f(x_1, \ldots, x_n)||^2 := \sum |a_{i_1 \ldots i_n}|^2$.

Howgrave-Graham's lemma is usually combined with LLL reduction of lattice bases, designed by Lenstra, Lenstra, and Lovász [13]. A proof of the following fact can be found in [14].

Fact 1 (LLL). *Let L be a lattice of dimension ω. In polynomial time, the LLL-algorithm outputs reduced basis vectors v_i, $1 \leq i \leq \omega$ that satisfy*

$$||v_1|| \leq ||v_2|| \leq \ldots \leq ||v_i|| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(L)^{\frac{1}{\omega+1-i}}.$$

Thus the condition $2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(L)^{\frac{1}{\omega+1-i}} < \frac{N}{\sqrt{\omega}}$ implies that the polynomials corresponding to the shortest i reduced basis vectors match Howgrave-Graham's bound. This reduces to

$$\det(L) \leq 2^{\frac{-\omega(\omega-1)}{4}} (\frac{1}{\sqrt{\omega}})^{\omega+1-i} N^{\omega+1-i}.$$

In the analysis, we let terms that do not depend on N contribute to an error term ϵ, and simply use the determinant condition $\det(L) \leq N^{\omega+1-i}$.

2.1 Strategy for Finding Small Modular Roots

We will now describe our strategy to find small modular roots of polynomials. Suppose we want to find a small root $(x_1^{(0)}, \ldots, x_n^{(0)})$ of a polynomial f_N modulo a known composite integer N of unknown factorization. We assume that we know an upper bound for the root, namely $|x_j^{(0)}| < X_j$ for some given X_j, for $j = 1, \ldots, n$.

Let l be a leading monomial of f_N, with coefficient a_l. That is, there is no monomial in f_N besides l that is divisible by l. Then $\gcd(N, a_l)$ is 1, or else we have found a factor of N. Therefore, we can use $f'_N = a_l^{-1} f_N \bmod N$.

We start by explaining the basic strategy to find the small modular roots, after which we extend it slightly to obtain the full strategy.

Basic Strategy: Let $\epsilon > 0$ be an arbitrarily small constant. Depending on $\frac{1}{\epsilon}$, we fix an integer m. For $k \in \{0, \ldots, m+1\}$, we define the set M_k of monomials

$$M_k := \{x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \mid \quad x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \text{ is a monomial of } f_N^m$$
$$\text{and } \frac{x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}}{l^k} \text{ is a monomial of } f_N^{m-k}\}.$$

In this definition of M_k and throughout this paper, we assume that the monomials of $f_N, \ldots, f_N^{m-1}$ are all contained in the monomials of f_N^m. If this is not the case, the definition can be slightly changed such that M_k contains all monomials $x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$ of f_N^j for $j \in \{1, \ldots, m\}$ for which $\frac{x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}}{l^k}$ is a monomial of f_N^i for some $i \in \{0, \ldots, m-k\}$. Notice that by definition the set M_0 contains all the monomials in f_N^m, whereas $M_{m+1} = \emptyset$.

Next, we define the following shift polynomials:

$$g_{i_1 \ldots i_n}(x_1, \ldots, x_n) := \frac{x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}}{l^k} f'_N(x_1, \ldots, x_n)^k N^{m-k},$$

for $k = 0, \ldots, m$, and $x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \in M_k \backslash M_{k+1}$.

All polynomials g have the root $(x_1^{(0)}, \ldots, x_n^{(0)})$ modulo N^m. We define a lattice L by taking the coefficient vectors of $g(x_1 X_1, \ldots, x_n X_n)$ as a basis. We can force the matrix describing L to be lower triangular, if we use the following ordering of the columns of the matrix. A column corresponding to the monomial $x_1^{i_1} \ldots x_n^{i_n} \in M_k \backslash M_{k+1}$ has smaller order than a column corresponding to $x_1^{j_1} \ldots x_n^{j_n} \in M_{k'} \backslash M_{k'+1}$ if $k < k'$. If $k' = k$, then a lexicographical ordering of the monomials is used. The columns in the lattice basis appear in increasing order from left to right. The diagonal elements are those corresponding to the monomial l^k in $(f'_N)^k$ for each row. Therefore, the diagonal terms of the matrix are $X_1^{i_1} X_2^{i_2} \ldots X_n^{i_n} N^{m-k}$ for the given combinations of k and i_j.

The intuition behind the choice of the sets M_k can be explained as follows. We aim to have a matrix with a low determinant. To keep the diagonal element corresponding to the monomial $x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$ of f_N^m as small as possible, we use the largest possible powers of f_N in the shifts. The condition that $\frac{x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}}{l^k}$ is a monomial of f_N^{m-k} ensures that no monomials appear that are not in f_N^m.

For a small example, consider the polynomial $f_N(x,y) = 1 + xy^2 + x^2y$. Let us take $l = x^2y$ as our leading term, and $m = 2$. We want to build a lattice whose columns correspond to the monomials $\{1, xy^2, x^2y, x^2y^4, x^3y^3, x^4y^2\}$ of f_N^2. The shifts given by our strategy are:

for	$1 \in M_0 \backslash M_1$: N^2	for	$x^2y \in M_1 \backslash M_2$: $f_N N$
for	$xy^2 \in M_0 \backslash M_1$: xy^2N^2	for	$x^3y^3 \in M_1 \backslash M_2$: $xy^2 f_N N$
for	$x^2y^4 \in M_0 \backslash M_1$: $x^2y^4N^2$	for	$x^4y^2 \in M_2 \backslash M_3$: f_N^2

Note that the monomial x^2y^4 is not in M_1. Although x^2y^4 is divisible by $l = x^2y$ and therefore we could obtain x^2y^4 also by using the shift $y^3 f_N N$, the product $y^3 f_N$ would produce the new monomials y^3 and xy^5, which are not in f_N^2.

In general, we find that our condition $\det(L) < N^{m(\omega+1-n)}$, derived from Lemma 1 and Fact 1, reduces to

$$\prod_{j=1}^{n} X_j^{s_j} < N^{s_N}, \text{ for } \begin{cases} s_j &= \sum_{x_n^{i_1} \ldots x_n^{i_n} \in M_0} i_j, \text{ and} \\ \\ s_N &= \sum_{k=0}^{m} k(|M_k| - |M_{k+1}|) = \sum_{k=1}^{m} |M_k| \end{cases} \tag{1}$$

If we follow this procedure for a given f_N, then (1) will give us an upper bound on the size of the root that we are trying to find. For X_j and N satisfying this bound we obtain n polynomials h_i such that $h_i(x_1^{(0)}, \ldots, x_n^{(0)}) = 0$. If the polynomials h_i are algebraically independent, i.e. they do not share a non-trivial gcd, then resultant computations will reveal the root. Under Assumption 1 this will lead us to finding $(x_1^{(0)}, \ldots, x_n^{(0)})$.

Assumption 1 *The resultant computations for the polynomials h_i yield non-zero polynomials.*

All methods for $n \geq 2$ have a similar assumption concerning the algebraic independence of the polynomials h_i. Therefore one has to keep in mind that (most) attacks using Coppersmith techniques are heuristical, and experiments must be done for specific cases to justify the assumption.

Extended Strategy: For many polynomials, it is profitable to use extra shifts of a certain variable. For instance, if we use extra shifts of x_1, then we can extend our basic strategy by using

$$M_k := \bigcup_{0 \leq j \leq t} \{x_1^{i_1+j} x_2^{i_2} \ldots x_n^{i_n} \mid x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \text{ is a monomial of } f_N^m$$
$$\text{and } \frac{x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}}{l^k} \text{ is a monomial of } f_N^{m-k}\}.$$

Moreover, extra shifts of several variables, or combined shifts should be considered to obtain an optimal bound.

Using this new definition of M_k, the rest of the strategy conforms to the basic strategy as described before. In Appendix A, we show how the known results on small modular roots from [1,2,6] are all special cases of our basic or extended strategy.

2.2 Strategy for Finding Small Integer Roots

Coron reformulated Coppersmith's method of finding small integer roots in [7]. Essentially, Coron picks a 'suitable' integer R and transforms the situation into finding a small root modulo R, after which one can apply Howgrave-Graham's lemma. Analogous to Coron, we will now present our heuristic strategy for finding small integer roots of multivariate polynomials. The result is an extension of the result given by Blömer and May [3], that was meant for the provable special case of bivariate polynomials.

We note that one could also use Coppersmith's original technique instead of Coron's reformulation. The advantage to do so is that in the original Coppersmith technique, lattices of smaller dimension are required. The asymptotic bounds obtained by both methods are equivalent, but the difference is in the size of the error term ϵ. For this paper, we have chosen to use Coron's method for the sake of a simpler notation, an easier implementation and for its similarity to the modular approach.

Suppose we want to find the small integer root $(x_1^{(0)}, \ldots, x_n^{(0)})$ of an irreducible polynomial f. We know that the root is small in the sense that $|x_j^{(0)}| < X_j$, for $j = 1, \ldots, n$.

Analogous to Section 2.1, we fix an integer m depending on $\frac{1}{\epsilon}$. We call d_j the maximal degree of x_j in f, and W the maximal coefficient of $f(x_1X_1, \ldots, x_nX_n)$. We will use $W = \|f(x_1X_1, \ldots, x_nX_n)\|_\infty$, with $\|f(x_1, \ldots, x_n)\|_\infty := \max |a_{i_1 \ldots i_n}|$ for $f(x_1, \ldots, x_n) = \sum a_{i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}$ as a notation. Moreover, we define $R = W \prod_{j=1}^{n} X_j^{d_j(m-1)}$. To work with a polynomial with constant term 1, we define $f' = a_0^{-1} f \bmod R$, where a_0 is the constant term of f. This means that we should have $a_0 \neq 0$ and $\gcd(a_0, R) = 1$. The latter is easy to achieve, analogous to [7, Appendix A], since any X_j with $\gcd(a_0, X_j) \neq 1$ can be changed into an X'_j such that $X_j < X'_j < 2X_j$ and $\gcd(a_0, X'_j) = 1$. The same holds for W.

Let us now consider the case $a_0 = 0$. In [7, Appendix A], Coron discussed this case for bivariate polynomials, and showed a simple way to transfer a polynomial f with zero constant term into a polynomial f^* with non-zero constant term.

A general way to do this for multivariate polynomials would be the following. First, we find a non-zero integer vector $(y_1, \ldots, y_n)$ such that $f(y_1, \ldots, y_n) \neq 0$. This can be constructed in polynomial time since there are only polynomially many roots within the given bounds. Then we define $f^*(x_1, \ldots, x_n) := f(x_1 + y_1, \ldots, x_n + y_n)$, and look for roots of f^*. Since $f^*(0, \ldots, 0) = f(y_1, \ldots, y_n)$, f^* has a non-zero constant term.

We would like to point out that the switch to f^* will affect the set of monomials, and new monomials may appear in f^* that were not in f. This may affect the analysis and lead to a different Coppersmith-type bound. This issue already appears with bivariate polynomials, but it did not affect Coron's analysis since in his case the set of monomials stayed the same.

Let us now describe our strategy for finding integer roots. As before, we start with the basic strategy, that we extend later to obtain the full strategy.

Basic Strategy: Let us first fix an arbitrarily small error term ϵ. We define an integer m depending on $\frac{1}{\epsilon}$. Furthermore, we define the sets S and M of monomials that represent the monomials of f^{m-1} and f^m respectively. We denote by l_j the largest exponent of x_j that appears in the monomials of S, i.e. $l_j = d_j(m-1)$.

Next, we define the following shift polynomials

$$\begin{array}{ll} g: x_1^{i_1}x_2^{i_2}\dots x_n^{i_n} f'(x_1,\dots,x_n)\prod_{j=1}^n X_j^{l_j-i_j} & \text{, for } x_1^{i_1}x_2^{i_2}\dots x_n^{i_n} \in S, \\ g': x_1^{i_1}x_2^{i_2}\dots x_n^{i_n} R & \text{, for } x_1^{i_1}x_2^{i_2}\dots x_n^{i_n} \in M\backslash S. \end{array}$$

All g and g' have the root $(x_1^{(0)},\dots,x_n^{(0)})$ modulo R. The coefficient vectors of $g(x_1X_1,\dots x_nX_n)$ and $g'(x_1X_1,\dots x_nX_n)$ form a lattice basis of a lattice L.

Using lexicographical ordering of the monomials, we can order the basis matrix such that it is upper triangular. The diagonal elements of the rows of g are those corresponding to the constant term in f'. Therefore, the diagonal entries of the matrix are $\prod_{j=1}^n X_j^{d_j(m-1)}$ for the polynomials g and $W\prod_{j=1}^n X_j^{d_j(m-1)+i_j}$ for the polynomials g'.

From Section 2, we know that the determinant condition $\det(L) < R^{\omega+2-n}$ ensures that the $n-1$ smallest vectors in an LLL reduced basis of L correspond to $n-1$ polynomials $h_i(x_1,\dots x_n)$ with $h_i(x_1^{(0)},\dots,x_n^{(0)}) = 0$.

We find that the condition $\det(L) < R^{\omega+2-n}$ reduces to

$$\prod_{j=1}^n X_j^{s_j} < W^{s_W}, \text{ for } s_j = \sum_{x_1^{i_1}\dots x_n^{i_n}\in M\backslash S} i_j \text{ , and } s_W = |S|. \tag{2}$$

So if (2) holds, we obtain $n-1$ polynomials h_i such that $h_i(x_1^{(0)},\dots,x_n^{(0)}) = 0$.

The choice of R ensures that the h_i are independent of f. This is because all h_i are divisible by $\prod_{j=1}^n X_j^{d_j(m-1)}$. According to a generalization by Hinek/Stinson [10, Corollary 5] of a lemma of Coron [7], a multiple $h(x_1,\dots,x_n)$ of $f(x_1,\dots,x_n)$ that is divisible by $\prod_{j=1}^n X_j^{d_j(m-1)}$ has norm at least

$$2^{-(\rho+1)^n+1}\prod_{j=1}^n X_j^{d_j(m-1)} W = 2^{-(\rho+1)^n+1}R,$$

where ρ is the maximum degree of the polynomials f,h in each variable separately. If we let terms that do not depend on R contribute to ϵ, it follows that if h_i satisfies Howgrave-Graham's bound $||h_i(x_1X_1,\dots,x_nX_n)|| < \frac{R}{\sqrt{\omega}}$, then it also cannot be a multiple of f. Since we assume that f is irreducible, it follows that f and h_i must be algebraically independent. However we cannot prevent that the h_i are pairwise algebraically dependent. So the resultant computations of f and h_i (for $i=1,\dots,n-1$) will only reveal the root under Assumption 1. This makes the techniques heuristical for $n \geq 3$.

Extended Strategy: As in the modular case, our strategy is not finished before exploring the possibilities of extra shifts of a certain variable (or more variables). Suppose we use extra shifts of the variable x_1. Then, instead of $S = \{\text{monomials of } f^{m-1}\}$, and $M = \{\text{monomials of } f^m\}$, we use

$$\begin{aligned} S &= \bigcup_{0\le j\le t}\{x_1^{i_1+j}x_2^{i_2}\dots x_n^{i_n} \mid x_1^{i_1}x_2^{i_2}\dots x_n^{i_n} \text{ is a monomial of } f^{m-1}\},\\ M &= \{\text{monomials of } x_1^{i_1}x_2^{i_2}\dots x_n^{i_n}\cdot f \mid x_1^{i_1}x_2^{i_2}\dots x_n^{i_n} \in S\}. \end{aligned}$$

With the new definitions, the rest of the strategy conforms to the basic strategy as described above, except for the value of R. It is necessary to change $R = W\prod_{j=1}^{n} X_j^{d_j(m-1)}$ into $R = W\prod_{j=1}^{n} X_j^{l_j}$, where l_j is the largest exponent of x_j that appears in the monomials of S. In Appendix B, we show that the known results on small integer roots from [3,6,8] are special cases of our basic or extended strategy. Moreover, a detailed example for a specific polynomial is treated in the next section.

3 A Bound Obtained with the New Strategy

In this section we will give a novel analysis of a trivariate polynomial that will be used in two new attacks on RSA variants in the subsequent sections.

Let $f(x,y,z) = a_0+a_1x+a_2x^2+a_3y+a_4z+a_5xy+a_6xz+a_7yz$ be a polynomial with a small root $(x^{(0)}, y^{(0)}, z^{(0)})$, with $|x^{(0)}| < X$, $|y^{(0)}| < Y$, $|z^{(0)}| < Z$. We show that under Assumption 1 for every fixed ϵ, all sufficiently small roots can be found in time polynomial in $\log W$ provided that

$$X^{7+9\tau+3\tau^2}(YZ)^{5+\frac{9}{2}\tau} < W^{3+3\tau-\epsilon},$$

where we can optimize $\tau > 0$ after the substitution of values for X, Y, Z, and W.

Let us follow the extended strategy described in Section 2.2 to show how this bound can be obtained. Our goal is to construct two polynomials h_1, h_2 with the root $(x^{(0)}, y^{(0)}, z^{(0)})$ that are not multiples of f. To do so, we fix an integer m depending on $\frac{1}{\epsilon}$, and an integer $t = \tau m$ that describes the number of extra x-shifts. We define $R = WX^{2(m-1)+t}(YZ)^{m-1}$ and $f' = a_0^{-1}f \mod R$. The shift polynomials g and g' are given by:

$$\begin{aligned} &g:\ x^{i_1}y^{i_2}z^{i_3}f'(x,y,z)X^{2(m-1)+t-i_1}Y^{m-1-i_2}Z^{m-1-i_3} && \text{for } x^{i_1}y^{i_2}z^{i_3}\in S,\\ &g':\ Rx^{i_1}y^{i_2}z^{i_3} && \text{for } x^{i_1}y^{i_2}z^{i_3}\in M\backslash S, \end{aligned}$$

for

$$\begin{aligned} S &= \bigcup_{0\le j\le t}\{x^{i_1+j}y^{i_2}z^{i_3} \mid x^{i_1}y^{i_2}z^{i_3} \text{ is a monomial of } f^{m-1}\},\\ M &= \{\text{monomials of } x^{i_1}y^{i_2}z^{i_3}\cdot f \mid x^{i_1}y^{i_2}z^{i_3}\in S\}. \end{aligned}$$

It follows that

$$\begin{aligned} x^{i_1}y^{i_2}z^{i_3}\in S &\Leftrightarrow i_2 = 0,\dots,m-1\ ;\ i_3 = 0,\dots,m-1\ ;\\ &\qquad i_1 = 0,\dots,2(m-1)-(i_2+i_3)+t.\\ x^{i_1}y^{i_2}z^{i_3}\in M &\Leftrightarrow i_2 = 0,\dots,m\ ;\ i_3 = 0,\dots,m\ ;\ i_1 = 0,\dots,2m-(i_2+i_3)+t. \end{aligned}$$

All polynomials g and g' have the root $(x^{(0)}, y^{(0)}, z^{(0)})$ modulo R. Let h_1 and h_2 be linear combinations of the polynomials g and g'. As was explained in Section 2.2, if h_1 and h_2 satisfy Howgrave-Graham's bound $||h_i(xX, yY, zZ)|| < \frac{R}{\sqrt{\omega}}$, then we can assume that h_1 and h_2 both have the root $(x^{(0)}, y^{(0)}, z^{(0)})$ over the integers, and also that they are algebraically independent of f.

Using the coefficient vectors of $g(xX, yY, zZ)$ and $g'(xX, yY, zZ)$ as a basis, we build a lattice L. We order the vectors such that the matrix is triangular, with the diagonal entries of g equal to $X^{2(m-1)+t}(YZ)^{m-1}$, and those of g' equal to $RX^{i_1}Y^{i_2}Z^{i_3} = X^{2(m-1)+t+i_1}Y^{m-1+i_2}Z^{m-1+i_3}W$.

Now by (2), provided that $\prod_{j=1}^{n} X_j^{s_j} < W^{|S|}$ with $s_j = \sum_{x_1^{i_1}\ldots x_n^{i_n} \in M \setminus S} i_j$ holds, the polynomials h_1 and h_2 corresponding to the shortest two LLL-reduced basis vectors satisfy Howgrave-Graham's bound. This reduces to

$$X^{(\frac{7}{3}+3\tau+\tau^2)m^3+o(m^2)}(YZ)^{(\frac{5}{3}+\frac{3}{2}\tau)m^3+o(m^2)} \leq W^{(1+\tau)m^3+o(m^2)}.$$

If we let all terms of order $o(m^2)$ contribute to ϵ, the condition simplifies to

$$X^{7+9\tau+3\tau^2}(YZ)^{5+\frac{9}{2}\tau} < W^{3+3\tau-\epsilon}.$$

4 Attack on RSA-CRT with Known Difference

In this section, we explain how a small root of a polynomial $f(x, y, z) = a_0 + a_1x + a_2x^2 + a_3y + a_4z + a_5xy + a_6xz + a_7yz$ results in a new attack on a variant of RSA-CRT proposed by Qiao/Lam [17]. We show the following result.

Theorem 1 (RSA-CRT with Fixed Known Difference $d_p - d_q$). *Under Assumption 1, for every $\epsilon > 0$, there exists n_0 such that for every $n > n_0$, the following holds: Let $N = pq$ be an n-bit RSA modulus, and p, q primes of bitsize $\frac{n}{2}$. Let $ed \equiv 1 \mod \phi(N)$, and d_p and d_q be such that $d_p \equiv d \mod (p-1)$ and $d_q \equiv d \mod (q-1)$. Assume that d_p and d_q are chosen such that $d_p = d_q + \bar{c}$ for some known $\bar{c}$ and bitsize(d_p), bitsize(d_q) $\leq \delta n$ for some $0 < \delta < \frac{1}{2}$. Then N can be factored in time polynomial in $\log N$ provided that*

$$\delta < \frac{1}{4}(4 - \sqrt{13}) - \epsilon.$$

Notice that $\frac{1}{4}(4 - \sqrt{13}) \approx 0.099$. Hence, our attack applies whenever d_p or d_q is smaller than $N^{0.099-\epsilon}$ and the difference $\bar{c} = d_p - d_q$ is known to an attacker.

4.1 RSA-CRT with Known Difference $d_p - d_q$

In 1990, Wiener [19] showed that choosing $d < N^{\frac{1}{4}}$ makes RSA insecure. As an alternative, Wiener suggested to use the Chinese Remainder Theorem (CRT) for the decryption phase of RSA: Instead of computing $m \equiv c^d \mod N$ for some ciphertext c, compute $m_1 \equiv c^{d_p} \mod p$ and $m_2 \equiv c^{d_q} \mod q$ and then combine these results using CRT to obtain m. Wiener pointed out that both exponents

$d_p \equiv d \mod (p-1)$ and $d_q \equiv d \mod (q-1)$ could be chosen small to obtain a fast decryption. Then usually e is of the same size as the modulus N.

Qiao and Lam [17] proposed to use d_p and d_q such that $d_p - d_q = 2$ in their method for fast signature generation on a low-cost smartcard. For the size of d_p and d_q, they suggest 128 bits to counteract the birthday attack that they describe in [17]. Moreover, they state that 96 bits should be enough to counteract this attack in practice. In current proposals, a minimum of 160 bits is advised for the private exponents to counteract the birthday attack.

4.2 Description of the New Attack

When $d_p - d_q = \bar{c}$, the public and private variables of RSA-CRT satisfy the following relations.

$$\begin{cases} ed_p = 1 + k(p-1), \\ e(d_p - \bar{c}) = 1 + l(q-1), \end{cases} \quad \text{or equivalently} \quad \begin{cases} ed_p - 1 + k = kp, \\ ed_p - \bar{c}e - 1 + l = lq. \end{cases}$$

Multiplying the two equations results in

$$(1 + \bar{c}e) - (2e + \bar{c}e^2)d_p + e^2 d_p^2 - (\bar{c}e + 1)k - l + ed_p k + ed_p l + (1 - N)kl = 0,$$

in which the unknowns are d_p, k, and l. We can extract from this equation that

$$f(x,y,z) = (1 + \bar{c}e) - (2e + \bar{c}e^2)x + e^2x^2 - (\bar{c}e + 1)y - z + exy + exz + (1 - N)yz$$

has a small root (d,k,l). From (d,k,l), the factorization of N can easily be found. Suppose $\max\{d_p, d_q\}$ is of size N^δ for some $\delta \in (0, \frac{1}{2})$. Then k and l are both bounded by $N^{\delta+\frac{1}{2}}$ (here we omit constants and let these contribute to the error term ϵ). Therefore, we put $X = N^\delta, Y = Z = N^{\delta+\frac{1}{2}}$, and $W = N^{2+2\delta}$.

In Section 3 we showed that for this polynomial, the asymptotic bound is

$$X^{7+9\tau+3\tau^2}(YZ)^{5+\frac{9}{2}\tau} < W^{3+3\tau},$$

where $\tau > 0$ can be optimized. Substituting the values for X, Y, Z, and W, we obtain

$$(7 + 9\tau + 3\tau^2)\delta + (5 + \tfrac{9}{2}\tau)(2\delta + 1) - (3 + 3\tau)(2\delta + 2) < 0, \text{ or}$$

$$3\delta\tau^2 + 3(4\delta - \tfrac{1}{2})\tau + (11\delta - 1) < 0.$$

For the optimal value $\tau = \frac{\frac{1}{2} - 4\delta}{2\delta}$, this reduces to $\delta < \frac{1}{4}(4 - \sqrt{13}) \approx 0.099$.

Therefore, for a 1024 bit modulus N, the system should be considered unsafe when d_p is at most $0.099 \cdot 1024 \approx 101$ bits. This breaks the system of Qiao and Lam for the proposed 96 bit exponents in time polynomial in the bit-size of N.

We can add an exhaustive search on the most significant bits of d_p and try the attack for each candidate for $\tilde{d_p}$. Here, $d_p = \tilde{d_p} + d_0$, where the unknown part of d is d_0. The corresponding polynomial f will change, but it will still have the same monomials. Therefore, the analysis will follow easily. The proposal of Qiao and Lam to use 128 bit private exponents can also be considered unsafe when applying such an extra exhaustive search, although performing such an attack may be costly in practice.

We performed several experiments to test the validity of Assumption 1 and to show which results can be achieved with relatively small lattices. We implemented the new attacks on a 2.4GHz Pentium running Linux. The LLL lattice reduction was done using Shoup's NTL [18]. For the attack on RSA-CRT with known difference described in Section 4, the parameters d_p, d_q were chosen with difference $d_p - d_q = 2$ as suggested in the Qiao-Lam scheme. For $m = 2$ the choice $t = 8$ maximizes the size of the attackable d_p.

N	d_p	lattice parameters	LLL-time
1000 bit	10 bit	$m = 2, t = 3, \dim = 54$	32 min
2000 bit	22 bit	$m = 2, t = 3, \dim = 54$	175 min
3000 bit	42 bit	$m = 2, t = 3, \dim = 54$	487 min
4000 bit	60 bit	$m = 2, t = 3, \dim = 54$	1015 min
5000 bit	85 bit	$m = 2, t = 3, \dim = 54$	1803 min
500 bit	9 bit	$m = 2, t = 8, \dim = 99$	105 min
1000 bit	18 bit	$m = 2, t = 8, \dim = 99$	495 min
500 bit	13 bit	$m = 3, t = 3, \dim = 112$	397 min

In each experiment we obtained two polynomials $h_1(x, y, z)$, $h_2(x, y, z)$ with the desired root $(x^{(0)}, y^{(0)}, z^{(0)})$. Solving $g(z) = \mathrm{Res}_y(\mathrm{Res}_x(h_1, f), \mathrm{Res}_x(h_2, f)) = 0$ yielded the unknown $z^{(0)}$. The parameters $y^{(0)}$ and $x^{(0)}$ could be obtained by back substitution. The resultant heuristic of Assumption 1 worked perfectly in practice. For every instance, we could recover the secrets and hence factor N.

One should note that our experiments are quite far from solving the proposed 96-bit d_p, d_q instances of the Qiao-Lam scheme. Theoretically, the smallest m for which one obtains the 96-bit bound is $m = 61$ together with $t = 36$, resulting in a lattice dimension of 376712. Reducing lattice bases in this dimension is clearly out of reach.

However, we would like to point out that we did not optimize the performance of our attack. For optimization of the running-time, one should combine brute-force guessing of most significant bits of d_p with the described lattice attack. Moreover, one should apply faster lattice reduction methods like the recently proposed L^2-method of Nguyen, Stehlé [16]. Additionally, a significant practical improvement should be obtained by implementing Coppersmith's original method instead of Coron's method, since in Coppersmith's method one has to reduce a lattice basis of smaller dimension.

5 New Attack on Common Prime RSA

In this section, we explain how a small root of a polynomial $f(x, y, z) = a_0 + a_1x + a_2x^2 + a_3y + a_4z + a_5xy + a_6xz + a_7yz$ results in a new attack on a variant of RSA called Common Prime RSA. We show the following result.

Theorem 2 (Common Prime RSA).
Under Assumption 1, for every $\epsilon > 0$, there exists n_0 such that for every $n > n_0$, the following holds: Let $N = pq$ be an n-bit RSA modulus, and p, q primes of bitsize $\frac{n}{2}$ such that $p - 1 = 2ga$ and $q - 1 = 2gb$, for some prime g of bitsize γn, with $0 < \gamma < \frac{1}{2}$. Let $ed \equiv 1 \mod 2gab$, with bitsize$(e) = (1-\gamma)n$ and bitsize$(d) = \delta n$, with $0 < \delta < (1-\gamma)n$. Then d can be found in time polynomial in $\log N$ provided that

$$\delta < \frac{1}{4}(4 + 4\gamma - \sqrt{13 + 20\gamma + 4\gamma^2}) - \epsilon.$$

5.1 Common Prime RSA

In Section 4, we mentioned that a small d is unsafe in Wiener's attack [19]. Therefore, RSA-CRT is often used when efficient decryption is needed. However, there is also a possibility to choose $d < N^{\frac{1}{4}}$ in RSA while avoiding Wiener's attack. There is a variant of RSA where Wiener's attack works less well, as was already shown by Wiener, namely when $\gcd(p-1, q-1)$ has a large prime factor. Lim and Lee used this fact in a proposal [12], which was attacked a few years later by McKee and Pinch [15]. Recently Hinek [9] revisited this variant, calling it Common Prime RSA, and investigated its potential and its weaknesses.

In Common Prime RSA, we have $N = pq$ for primes p and q such that $p = 2ga + 1$ and $q = 2gb + 1$, for g a large prime, and a, b coprime integers. The exponents e and d are mutually inverse modulo $\mathrm{lcm}(p-1, q-1) = 2gab$:

$$ed = 1 + k \cdot 2gab, \text{ with } 0 < e, d < 2gab.$$

The goal is to safely choose an exponent $d < N^{\frac{1}{4}}$, which enables a fast RSA decryption process. We set $g = N^{\gamma}$ and $d = N^{\delta}$ for some $0 \leq \gamma < \frac{1}{2}$, $0 < \delta < 1 - \gamma$. Then, e is of size $N^{1-\gamma}$, k is of size N^{δ}, and a and b are both of size $N^{\frac{1}{2}-\gamma}$.

A large number of security issues were addressed in [9]. After excluding all parameter choices of Common Prime RSA that should be considered unsafe by the known attacks, Hinek concludes that there are still plenty of safe choices for $d = N^{\delta}$ with $\delta < \frac{1}{4}$ (see Fig. 1).

5.2 Description of the New Attack

An improved attack can be obtained by treating the equation in Hinek's second lattice attack in a different way. In his attack, Hinek starts by multiplying the following two equations:

$$ed = 1 + k(p-1)b, \quad ed = 1 + k(q-1)a.$$

This can be written as $e^2d^2 + ed(ka + kb - 2) - (N-1)k^2ab - (ka + kb - 1) = 0$. Next, he uses the fact that the polynomial $f(x, y, z, u) = e^2x + ey - (N-1)z - u$

has a small root $(d^2, d(k(a+b-2)), k^2ab, (ka+kb-1))$. This leads to the bound $\delta < \frac{2}{5}\gamma$, for which the secret information can be revealed.

Now let us take another look at the equation

$$e^2d^2 + ed(ka + kb - 2) - (ka + kb - 1) - (N - 1)k^2ab = 0,$$

in which the unknowns are d, k, a and b. We can extract from this equation that the polynomial $f(x, y, z) = e^2x^2 + ex(y + z - 2) - (y + z - 1) - (N - 1)yz$ has a small root (d, ka, kb) with $X = N^{\delta}, Y = N^{\delta+\frac{1}{2}-\gamma}, Z = N^{\delta+\frac{1}{2}-\gamma}$. Moreover, $W = N^{2+2\delta-2\gamma}$.

Substituting these in the asymptotical bound $X^{7+9\tau+3\tau^2}(YZ)^{5+\frac{9}{2}\tau} < W^{3+3\tau}$ from Section 3 yields

$$3\delta\tau^2 + 3(4\delta - \frac{1}{2} - \gamma)\tau + (11\delta - 1 - 4\gamma) < 0.$$

For the optimal $\tau = \frac{\frac{1}{2}+\gamma-4\delta}{2\delta}$, this reduces to $\delta < \frac{1}{4}(4 + 4\gamma - \sqrt{13 + 20\gamma + 4\gamma^2})$.

Fig. 1 shows the new attack region as well as the known attacks, for any size of modulus N. Combinations of d and g that should be considered unsafe by the new attack are in the dark shaded area, whereas the lighter shaded area was already unsafe by the known attacks. It can be seen that the number of 'safe' combinations $\{d, g\}$ with $d < N^{\frac{1}{4}}$ has significantly decreased.

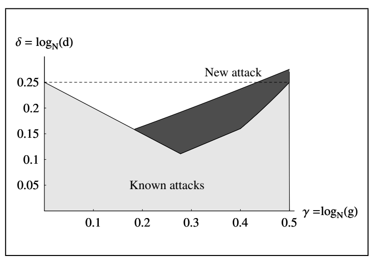

Fig. 1. New attack region

We note that for 'small' N (such as the regular 1024 bits), other attacks such as factoring attacks may apply, see [9]. Also, depending on the size of N, the attacks in the figure could be extended by an additive exhaustive search.

We performed experiments to check the validity of Assumption 1 and to demonstrate the practicality of our attack. We have implemented the new attack for the parameter setting $m = 2$, $t = 0$ (without the possible additional exhaustive search), to give an impression on what a realistic bound is for the smallest lattice possible. Of course, extending to $m = 3$, $m = 4$, etc. and using x-shifts will give results closer to the theoretical attack bound $\delta < \frac{1}{4}(4 + 4\gamma - \sqrt{13 + 20\gamma + 4\gamma^2})$, but will also result in a longer time needed for the lattice basis reduction. For $m = 2$, $t = 0$ the reduction time (the longest part of the attack) is about one minute.

The following table summarizes the experimental results performed for $m = 2$, $t = 0$, and $\log_2(N) = 1024$. As one can see, the results are already outside the *asymptotical* range of the two other lattice attacks described in [9].

γ	maximal δ (asymptotic) new attack	obtained δ $(m = 2, t = 0)$ new attack	maximal δ (asymptotic) known attacks
0.10	0.130	0.07	0.20
0.20	0.164	0.10	0.15
0.30	0.200	0.13 $(*)$	0.12
0.40	0.237	0.17 $(*)$	0.16
0.50	0.275	0.2	0.25

The resultant heuristic of Assumption 1 worked perfectly in most cases. However, in the rare situation that both δ and γ were very small (e.g. $\gamma = 0.1$ and $\delta = 0.05$), we encountered cases where some of the polynomials h_i were algebraically dependent. In these cases, we could still recover the secret information in two different ways. One way was to use combinations of h_1 and the somewhat 'larger' h_i for $i > 2$, instead of only h_1 and h_2. The other way was by examining the cause of the zero resultant. In essence, $\mathrm{Res}_y(\mathrm{Res}_x(h_1, f), \mathrm{Res}_x(h_2, f)) = 0$ because $\mathrm{Res}_x(h_1, f)$ and $\mathrm{Res}_x(h_2, f)$ have a common polynomial factor, whose coefficients immediately reveal the secrets.

Acknowledgements: We thank Benne de Weger, Arjen Lenstra, Jason Hinek, and the anonymous reviewers for their helpful comments.

References

1. D. Boneh, G. Durfee: Cryptanalysis of RSA with Private Key d Less Than $N^{0.292}$, IEEE Transactions on Information Theory **46** [2000], 1339–1349.
2. J. Blömer, A. May: New Partial Key Exposure Attacks on RSA, Proceedings of CRYPTO 2003, LNCS **2729** [2003], 27–43.
3. J. Blömer, A. May: A Tool Kit for Finding Small Roots of Bivariate Polynomials over the Integers, Proceedings of EUROCRYPT 2005, LNCS **3494** [2005], 251–267.
4. D. Coppersmith: Finding a Small Root of a Univariate Modular Equation, Proceedings of EUROCRYPT 1996, LNCS **1070** [1996], 155–165.
5. D. Coppersmith: Finding a Small Root of a Bivariate Integer Equation, Proceedings of EUROCRYPT 1996, LNCS **1070** [1996], 178–189.
6. D. Coppersmith: Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities, Journal of Cryptology **10** [1997], 233–260.
7. J.-S. Coron: Finding Small Roots of Bivariate Integer Equations Revisited, Proceedings of EUROCRYPT 2004, LNCS **3027** [2004], 492–505.
8. M. Ernst, E. Jochemsz, A. May, B. de Weger: Partial Key Exposure Attacks on RSA up to Full Size Exponents, Proceedings of EUROCRYPT 2005, LNCS **3494** [2005], 371–386.
9. M.J. Hinek: Another Look at Small RSA Exponents, Proceedings of CT-RSA 2006, LNCS **3860** [2006], 82–98.
10. M.J. Hinek, D.R. Stinson: An Inequality About Factors of Multivariate Polynomials" [2006], http://www.cacr.math.uwaterloo.ca/techreports/2006/cacr2006-15.pdf.

11. N. Howgrave-Graham: Finding Small Roots of Univariate Modular Equations Revisited, Cryptography and Coding, LNCS **1355** [1997], 131–142.
12. C.H. Lim, P.J. Lee: Security and performance of server-aided RSA computation protocols, Proceedings of CRYPTO 1995, LNCS **963** [1995], 70-83.
13. A. Lenstra, H. Lenstra, Jr., L. Lovász: Factoring Polynomials with Rational Coefficients, Mathematische Ann. **261** [1982], 513–534.
14. A. May: New RSA Vulnerabilities Using Lattice Reduction Methods, PhD Thesis, University of Paderborn [2003].
15. J. McKee, R. Pinch: Further attacks on server-aided RSA cryptosystems [1998], http://citeseer.ist.psu.edu/388295.html.
16. P. Nguyen, D. Stehlé: Floating-Point LLL Revisited, Proceedings of EUROCRYPT 2005, LNCS **3494** [2005], 215–233.
17. G. Qiao, K.-Y. Lam: RSA Signature Algorithm for Microcontroller Implementation, Proceedings of CARDIS 1998, LNCS **1820** [2000], 353–356.
18. V. Shoup: NTL: A Library for doing Number Theory, online available at http://www.shoup.net/ntl/index.html.
19. M. Wiener: Cryptanalysis of Short RSA Secret Exponents, IEEE Transactions on Information Theory **36** [1990], 553–558.

A Small Modular Roots, Known Results

In this appendix, we give the known results for finding small modular roots [1,2,6] that can also be obtained by following the new strategy. Due to limited space, we only give the definitions of M_k that reproduce the known bounds. In all cases where the extended strategy is used, we use the notation $t = \tau m$ for some $\tau > 0$ that can be optimized later.

Boneh/Durfee [1]: $f_N(x_1, x_2) = a_0 + a_1x_1 + a_2x_1x_2$
The bound $X_1^{2+3\tau}X_2^{1+3\tau+3\tau^2} < N^{1+3\tau}$ can be found with the extended strategy using $x_1^{i_1}x_2^{i_2} \in M_k \Leftrightarrow i_1 = k, \ldots, m; i_2 = k, \ldots, i_1 + t$

Blömer/May [2]: $f_N(x_1, x_2, x_3) = a_0 + a_1x_1 + a_2x_2 + a_3x_2x_3$
The bound $X_1^{1+4\tau}X_2^{2+4\tau}X_3^{1+4\tau+6\tau^2} < N^{1+4\tau}$ can be found with the extended strategy, with $x_1^{i_1}x_2^{i_2}x_3^{i_3} \in M_k \Leftrightarrow i_1 = k, \ldots, m; i_2 = 0, \ldots, m-i_1; i_3 = 0, \ldots, i_2+t$.

Generalized Rectangle (generalization of a bound of Coppersmith[6]):
$f_N(x_1, \ldots, x_n)$ is a polynomial such that the degree of x_i is $\lambda_i D$.
The bound $X_1^{\lambda_1} \cdot \ldots \cdot X_n^{\lambda_n} < N^{\frac{2}{(n+1)D}}$ can be obtained with the basic strategy using $x_1^{i_1} \cdot \ldots \cdot x_n^{i_n} \in M_k \Leftrightarrow i_j = \lambda_j Dk, \ldots, \lambda_j Dm$ (for $j = 1, \ldots, n$)

Generalized Lower Triangle (generalization of a bound of Coppersmith[6]):
$f_N(x_1, \ldots, x_n)$ is a polynomial with monomials $x_1^{i_1} \ldots x_n^{i_n}$ for $i_1 = 0, \ldots, \lambda_1 D$, $i_2 = 0, \ldots, \lambda_2 D - \frac{\lambda_2}{\lambda_1} i_1$, …, $i_n = 0, \ldots, \leq \lambda_n D - \sum_{r=1}^{n-1} \frac{\lambda_n D}{\lambda_r} i_r$.
The bound $X_1^{\lambda_1} \cdot \ldots \cdot X_n^{\lambda_n} < N^{\frac{1}{D}}$ can be obtained with the basic strategy, with $x_1^{i_1} \cdot \ldots \cdot x_n^{i_n} \in M_k \Leftrightarrow i_1 = \lambda_1 Dk, \ldots, \lambda_1 Dm; i_j = 0, \ldots, \lambda_j Dm - \sum_{r=1}^{j-1} \frac{\lambda_j}{\lambda_r} i_r$ (for $j = 2, \ldots, n$).

B Small Integer Roots, Known Results

In this appendix, we give the known results for finding small integer roots [3,8,6] that can also be obtained with the basic or extended strategy. Due to limited space, we only give the definitions of S and M that reproduce the known bounds. In all cases where the extended strategy is used, we use the notation $t = \tau m$ for some $\tau > 0$ that can be optimized later.

Blömer/May, Upper Triangle [3]:
$f(x_1, x_2)$ is a polynomial with monomials $x_1^{i_1} x_2^{i_2}$ for $i_1 = 0 \dots D$, $i_2 = 0 \dots \lambda i_2$.
The bound $X_1^{(\lambda+\tau)^2} X_2^{2(\lambda+\tau)} < W^{\frac{1}{D}(\lambda+2\tau)}$ can be obtained with the extended strategy, with $x_1^{i_1} \cdot \ldots \cdot x_n^{i_n} \in S \Leftrightarrow i_2 = 0, \dots, D(m-1); i_1 = 0, \dots, \lambda i_2 + t$, and $x_1^{i_1} \cdot \ldots \cdot x_n^{i_n} \in M \Leftrightarrow i_2 = 0, \dots, Dm; i_1 = 0, \dots, \lambda i_2 + t$.

Blömer/May, Extended Rectangle [3]:
$f(x_1, x_2)$, with monomials $x_1^{i_1} x_2^{i_2}$ for $i_2 = 0, \dots, D$, $i_1 = 0, \dots, \gamma D + \lambda(D - i_2)$, e.g. $f(x_1, x_2) = a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 + a_4 x_2 + a_5 x_1 x_2$ (where $D = 1, \gamma = 1, \lambda = 2$).
The bound $X_1^{\lambda^2+3\gamma\lambda+2\tau\lambda+4\tau\gamma+\tau^2+3\gamma^2} X_2^{\lambda+3\gamma+2\tau} < W^{\frac{1}{D}(\lambda+2\gamma+2\tau)}$ can be obtained with the extended strategy, using $x_1^{i_1} x_2^{i_2} \in S \Leftrightarrow i_2 = 0, \dots, D(m-1)$; $i_1 = 0, \dots, \gamma D(m-1) + \lambda(D(m-1) - i_2) + t$, and $x_1^{i_1} x_2^{i_2} \in M \Leftrightarrow i_2 = 0, \dots, Dm; i_1 = 0, \dots, \gamma Dm + \lambda(Dm - i_2) + t$.

Ernst et al. 1 [8]: $f(x_1, x_2, x_3) = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_2 x_3$.
The bound $X_1^{1+3\tau} X_2^{2+3\tau} X_3^{1+3\tau+3\tau^2} < W^{1+3\tau}$ can be found with the extended strategy, with $x_1^{i_1} x_2^{i_2} x_3^{i_3} \in S \Leftrightarrow i_1 = 0, \dots, m-1; i_2 = 0, \dots, m-1-i_1$; $i_3 = 0, \dots, i_2 + t$, and $x_1^{i_1} x_2^{i_2} x_3^{i_3} \in M \Leftrightarrow i_1 = 0, \dots, m; i_2 = 0, \dots, m - i_1; i_3 = 0, \dots, i_2 + t$.

Ernst et al. 2 [8]: $f(x_1, x_2, x_3) = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 x_2 x_3$.
The bound $X_1^{2+3\tau} X_2^{3+3\tau} X_3^{3+6\tau+3\tau^2} < W^{2+3\tau}$ can be found with the extended strategy, using $x_1^{i_1} x_2^{i_2} x_3^{i_3} \in S \Leftrightarrow i_1 = 0, \dots, m-1; i_2 = 0, \dots, m-1-i_1+t; i_3 = 0, \dots, m-1-i_1$, and $x_1^{i_1} x_2^{i_2} x_3^{i_3} \in M \Leftrightarrow i_1 = 0, \dots, m; i_2 = 0, \dots, m - i_1 + t; i_3 = 0, \dots, m - i_1$.

Generalized Rectangle (generalization of a bound of Coppersmith [6]):
$f(x_1, \dots, x_n)$ is a polynomial where the degree of x_i is $\lambda_i D$.
The bound $X_1^{\lambda_1} \cdot \ldots \cdot X_n^{\lambda_n} < W^{\frac{2}{(n+1)D}}$ can be found with the basic strategy, with $x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \in S \Leftrightarrow i_j = 0, \dots, \lambda_j D(m-1)$, and $x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \in M \Leftrightarrow i_j = 0, \dots, \lambda_j Dm$ (for $j = 1, \dots, n$).

Generalized Lower Triangle (generalization of a bound of Coppersmith [6]):
$f(x_1, \dots, x_n)$ is a polynomial monomial are $x_1^{i_1} \dots x_n^{i_n}$ for $0 \le i_1 \le \lambda_1 D$, $0 \le i_2 \le \lambda_2 D - \frac{\lambda_2}{\lambda_1} i_1$, $\dots$, $0 \le i_n \le \lambda_n D - \sum_{r=1}^{n-1} \frac{\lambda_n}{\lambda_r} i_r$.
The bound $X_1^{\lambda_1} \cdot \ldots \cdot X_n^{\lambda_n} < W^{\frac{1}{D}}$ can be found with the basic strategy, with $x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \in S \Leftrightarrow i_j = 0, \dots, \lambda_j D(m-1) - \sum_{r=1}^{j-1} \frac{\lambda_j}{\lambda_r} i_r$, and $x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \in M \Leftrightarrow i_j = 0, \dots, \lambda_j Dm - \sum_{r=1}^{j-1} \frac{\lambda_j}{\lambda_r} i_r$ (for $j = 1, \dots, n$).