

Maximizing Small Root Bounds by Linearization and Applications to Small Secret Exponent RSA

Mathias Herrmann and Alexander May*

Horst Görtz Institute for IT-Security
Faculty of Mathematics
Ruhr University Bochum, Germany
mathias.herrmann@rub.de, alex.may@rub.de

Abstract. We present an elementary method to construct optimized lattices that are used for finding small roots of polynomial equations. Former methods first construct some large lattice in a generic way from a polynomial f and then optimize via finding suitable smaller dimensional sublattices. In contrast, our method focuses on optimizing f first which then directly leads to an optimized small dimensional lattice.

Using our method, we construct the first elementary proof of the Boneh-Durfee attack for small RSA secret exponents with $d \leq N^{0.292}$. Moreover, we identify a sublattice structure behind the Jochemsz-May attack for small CRT-RSA exponents $d_p, d_q \leq N^{0.073}$. Unfortunately, in contrast to the Boneh-Durfee attack, for the Jochemsz-May attack the sublattice does not help to improve the bound asymptotically. Instead, we are able to attack much larger values of d_p, d_q in practice by LLL reducing smaller dimensional lattices.

Keywords: linearization, lattices, small roots, small secret exponent, RSA, CRT-RSA.

1 Introduction

The RSA cryptosystem is currently the most widely deployed cryptosystem. To perform a decryption or signature generation, an element $x \in \mathbb{Z}_N$ is raised to the d -th power, where $d \in \mathbb{Z}_{\phi(N)}^*$ is the secret key. In order to speed up this process, one might be tempted to use a small value of d . However, once $d \leq N^{\frac{1}{4}}$, Wiener [Wie90] showed using a continued fraction approach that d can be reconstructed from just the public parameters e and N in polynomial time. This result has been further improved by Boneh and Durfee to $d \leq N^{0.292}$ using a lattice based technique [BD99].

Another possibility to speed up the decryption and signature generation has been proposed by Quisquater and Couvreur [QC82]. They make use of the knowledge of the prime factorization of $N = pq$ to compute x^d modulo p and modulo q

* This research was supported by the German Research Foundation (DFG) as part of the project MA 2536/3-1 and by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II.

and finally combine the result using the Chinese Remainder Theorem. The running time of this process is approx. 4 times faster than a standard decryption. To further lower the number of required operations, one can additionally use *small CRT exponents*, i.e. one can choose d such that $d_p = d \bmod p$ and $d_q = d \bmod q$ are both small.

At Crypto '07, Jochemsz and May [JM07] proposed the first polynomial time attack on CRT exponents that are smaller than $N^{0.073}$. However, the experimental results of Jochemsz and May for small dimensional lattices are much better than theoretically predicted. For example, using a lattice dimension of 56, theoretically the attack should not work at all, while in practice this lattice dimension is sufficient to reconstruct private keys up to a size of $N^{0.01}$. Such a discrepancy between theoretically predicted and practically achieved results is a strong indication that the involved lattice structure is not optimal. This led Jochemsz and May to conjecture that an analysis of sublattice structures could lead to a theoretically superior bound.

In this paper we propose a method that can be applied to attack small CRT-exponents. Our new approach leads to smaller dimensional lattices than in the Jochemsz-May attack and fully explains the gap between the practical results of Jochemsz and May and their theoretical analysis. Unfortunately, our analysis shows that our smaller dimensional lattices asymptotically lead to the same bound $N^{0.073}$ as in [JM07], thereby answering the conjecture of Jochemsz and May that sublattices improve the bound in the negative.

Although we do not achieve an asymptotic improvement, our new approach enables us to attack much larger values of d_p, d_q in practice, compared to [JM07], by using smaller dimensional lattices. We implemented our algorithm and showed that e.g. for a 2000-bit N we can efficiently recover 47-bit d_p, d_q , whereas the technique of [JM07] only allows to recover about 35-bit d_p, d_q in a comparable amount of time.

Our method is lattice-based and uses the technique of *unravelling linearization* introduced by Herrmann and May at Asiacrypt '09 [HM09], which can be seen as a hybrid method between usual linearization and Coppersmith's method [Cop97]. The central idea of unravelled linearization is to perform as a first step a linearization on the initial polynomial and keep the *induced relations* of the linearization in mind. These relations are afterwards used in a second step where we back-substitute in order to eliminate some monomials, thereby partially unravelling the first linearization step. In order to explicitly compute the induced relations, we propose to use a Gröbner basis computation.

We illustrate the technique of unravelled linearization by showing the first elementary proof of the Boneh-Durfee bound $d \leq N^{0.292}$ for small secret RSA exponents. Optimization of bounds is in our framework a simple task. Therefore, we conjecture that the Boneh-Durfee bound cannot be improved unless a different polynomial equation is used.

The rest of the paper is organized as follows: In Section 2 we will review some basic results from lattice theory. Section 3 will describe the method of unravelled linearization for the case of small RSA exponents d with a proof of $d \leq N^{0.292}$.

We will then apply our method to attack small CRT exponents in Section 4, where we achieve the Jochemsz-May bound of $N^{0.073}$ with smaller dimensional lattices. In Section 5, we demonstrate that our improved lattices allow for much better practical results in attacking small CRT-exponents.

2 Basics

Before we explain the details of unravelled linearization and how to use it to improve the analysis of small CRT-exponents, we want to give some necessary background information on lattice theory and the lattice-based method of Coppersmith [Cop97].

A lattice is a discrete additive subgroup of \mathbb{R}^n . That is, for a set of linearly independent basis vectors $b_1, \dots, b_{dim} \in \mathbb{R}^n$, $dim \leq n$, the set

$$L := \left\{ x \in \mathbb{R}^n \mid x = \sum_{i=1}^{dim} a_i b_i \text{ with } a_i \in \mathbb{Z} \right\}$$

is called a lattice. One can describe a lattice by its basis matrix B , where we write the vectors b_i as row vectors.

Let L be a lattice with basis b_1, \dots, b_{dim} , and let b_1^*, \dots, b_{dim}^* be the result of applying Gram-Schmidt orthogonalization to the basis vectors. Then the determinant of L is defined as $\det(L) = \prod_{i=1}^{dim} \|b_i^*\|$. For a lattice of full rank, i.e. $dim = n$, the determinant of a lattice equals the absolute value of the determinant of a lattice basis matrix.

Lattices have proved to be very useful in cryptanalysis mostly because of a powerful and efficient lattice reduction algorithm due to Lenstra, Lenstra and Lovász [LLL82]. This so-called LLL algorithm outputs an approximation of a shortest lattice vector in time polynomial in the bit-length of the entries of the basis matrix and in the dimension of the lattice dim . Using the LLL algorithm as a building block, Coppersmith [Cop96a, Cop96b] designed a rigorous algorithm that allows to efficiently compute *small* roots of bivariate polynomials over the integers or univariate modular polynomials. Additionally, he gave a heuristic extension to multivariate polynomials.

Coppersmith's idea is to construct, on input some polynomial f , a set of coprime polynomials which contain the same roots over the integers. Then one can use standard elimination and root finding techniques to extract these roots. Howgrave-Graham [HG97] gave a simple reformulation of Coppersmith's method that defines the following condition.

Theorem 1 (Howgrave-Graham). *Let $g(x_1, \dots, x_k)$ be a polynomial in k variables with n monomials. Furthermore, let m be a positive integer. Suppose that*

1. $g(r_1, \dots, r_k) = 0 \pmod{b^m}$, where $|r_i| \leq X_i, i = 1, \dots, k$ and
2. $\|g(x_1 X_1, \dots, x_k X_k)\| \leq \frac{b^m}{\sqrt{n}}$,

where the norm of g is defined as the Euclidean norm of its coefficient vector.

Then $g(r_1, \dots, r_k) = 0$ holds over the integers.

3 Unravalled Linearization and the Boneh-Durfee Attack

In this section, we will apply the method of unravalled linearization, introduced by Herrmann and May [HM09], to attack RSA with small secret exponent d . This will lead to an elementary proof of the Boneh-Durfee bound $d \leq N^{0.292}$.

In 1999, Boneh and Durfee [BD99] showed with a lattice-based Coppersmith-type attack, that private RSA keys smaller than $N^{0.284-\epsilon}$ can be recovered in polynomial time. The attack's running time is dominated by LLL-reducing some large dimensional lattice basis B , whose dimension depends on $\frac{1}{\epsilon}$. It turns out that the associated lattice $L(B)$ contains a smaller dimensional sublattice L' that allows to show an improved bound of $N^{0.292-\epsilon}$.

The identification and analysis of this sublattice L' , however, is a complicated task due to the fact that its lattice basis is no longer triangular and, therefore, the computation of the lattice determinant $\det(L')$ is much more involved. Boneh and Durfee developed for the analysis of $\det(L')$ a notion called *geometrically progressive matrices* that allowed for handling these non-triangular lattice bases. Blömer and May [BM01] followed a different approach and showed that asymptotically it does not influence the determinant if some specific columns are removed. This allowed them to rebuild some triangular structure of the basis matrix. Both approaches are, however, quite complex methods for optimizing lattice bases.

As opposed to the methods of [BD99] and [BM01] our new approach will not manipulate a basis matrix but rather it will manipulate the underlying polynomial from which a basis matrix is derived. This will directly lead to a low-dimensional sublattice with a basis of triangular structure that allows for an easy determinant calculation.

The method of our choice for this task is the technique of unravalled linearization [HM09]. However, before we introduce our method we briefly recall the original Boneh-Durfee attack in order to illustrate the similarities and differences.

The polynomial to be analyzed is derived from the RSA key equation $ed = 1 \pmod{\phi(N)}$. Rewrite this as

$$\begin{aligned} ed &= 1 + x\phi(N) \\ \Leftrightarrow ed &= 1 + x \underbrace{(N+1)}_A + \underbrace{(-p-q)}_y \end{aligned}$$

and search for small modular roots of the polynomial

$$f(x, y) := 1 + x(A + y) \pmod{e}.$$

Therefore, we fix an integer m and define the polynomials

$$g_{i,k}(x, y) := x^i f^k e^{m-k} \quad \text{and} \quad h_{j,k}(x, y) := y^j f^k e^{m-k}.$$

A lattice basis is constructed by using the coefficient vectors of the so-called x -shifts $g_{i,k}(xY, yY)$ for $k = 0, \dots, m$ and $i = 0, \dots, m - k$ as basis vectors.

$$\begin{array}{l}
 e^2 \\
 xe^2 \\
 fe \\
 x^2e^2 \\
 xfe \\
 f^2 \\
 ye^2 \\
 yfe \\
 yf^2
 \end{array}
 \left(
 \begin{array}{cccccccccc}
 1 & x & xy & x^2 & x^2y & x^2y^2 & y & xy^2 & x^2y^3 \\
 e^2 & & & & & & & & \\
 e^2X & & & & & & & & \\
 e & eAX & eXY & & & & & & \\
 & & & e^2X^2 & & & & & \\
 eX & & & eAX^2 & eX^2Y & & & & \\
 1 & 2AX & 2XY & A^2X^2 & 2AX^2Y & X^2Y^2 & & & \\
 \hline
 & & & & & & e^2Y & & \\
 & & eAXY & & & & eY & eXY^2 & \\
 \hline
 & & 2AXY & A^2X^2Y & 2AX^2Y^2 & Y & 2XY^2 & X^2Y^3 &
 \end{array}
 \right)$$

Fig. 1. Boneh-Durfee basis matrix for $m = 2, t = 1$

The values X and Y denote upper bounds on the sizes of the solutions. Additionally, we use the so-called y -shifts $h_{j,k}(xX, yY)$ for $k = 0, \dots, m$ and $j = 1, \dots, t$, where t is some parameter that has to be optimized. Figure 1 shows an example for the parameters $m = 2$ and $t = 1$. Note that the coefficient vectors of the shift polynomials $g_{i,k}(xX, yY)$ and $h_{j,k}(xX, yY)$ are written as row vectors.

Boneh and Durfee's improved analysis showed that one obtains superior values for X and Y , if one takes only a subset of the y -shifts. For our example this means we exclude ye^2 and yfe . Hence, the resulting lattice basis is no longer triangular and, therefore, deriving a closed determinant formula for general m and t is a complex task.

We now use the technique of unravelled linearization to construct a lattice basis which yields the best known asymptotic bound $N^{0.292}$ and yet retains a triangular lattice basis.

The first step in the process is to perform a suitable linearization of the original polynomial. In our case, we glue together the monomials in the following way

$$\underbrace{1 + xy + Ax}_{u} \text{ mod } e.$$

This leaves us with the linear polynomial $\bar{f}(u, x) = u + Ax$ and additionally a relation $xy = u - 1$ derived from the substitution. Although Coppersmith's method is a construction method suited for polynomial equations and does not give improved bounds in the case of linear equations, we now construct a lattice basis using exactly the same x -shifts as in the original Boneh-Durfee attack. I.e., we construct polynomials

$$\bar{g}_{i,k}(u, x) := x^i \bar{f}^k e^{m-k} \quad \text{for } k = 0, \dots, m \text{ and } i = 0, \dots, m - k, \quad (1)$$

and use their coefficient vectors as basis vectors. One can show that this leads to the Wiener bound of $N^{0.25}$.

However, if we also include y -shifts of the form $\bar{h}_{j,k}(u, x, y) := y^j \bar{f}^k e^{m-k}$, then we obtain a benefit. This may sound strange at first glance since the monomial y is not even present in our new polynomial $\bar{f}(u, x)$. The reason for the improved

$$\begin{array}{c}
 e^2 \\
 xe^2 \\
 \bar{f}e \\
 x^2e^2 \\
 x\bar{f}e \\
 \bar{f}^2 \\
 y\bar{f}^2
 \end{array}
 \left(
 \begin{array}{ccccccc}
 1 & x & u & x^2 & ux & u^2 & u^2y \\
 e^2 & & & & & & \\
 & e^2X & & & & & \\
 & eAX & eU & & & & \\
 & & & e^2X^2 & & & \\
 & & & eAX^2 & eUX & & \\
 & & & A^2X^2 & 2AUX & U^2 & \\
 \hline
 & -A^2X & -2AU & & A^2UX & 2AU^2 & U^2Y
 \end{array}
 \right)$$

Fig. 2. Boneh-Durfee lattice for $m = 2, t = 1$ using unravelled linearization

bound becomes clear, when we incorporate the induced relation $xy = u - 1$ and use it to substitute each occurrence of xy by the term $u - 1$.

The advantage can be seen by comparing the shift yf^2 from the original analysis with the new shift $y\bar{f}^2$. As noted previously, the improved analysis uses only the shift yf^2 and neither yfe nor ye^2 . But yf^2 introduces three new monomials y, xy^2 and x^2y^3 in the Boneh-Durfee lattice basis – thereby destroying the triangular structure.

Let us compare this with our new unravelled linearization approach, which we depicted in Figure 2 for the same parameters $m = 2$ and $t = 1$. The shift $y\bar{f}^2$ introduces the monomials x^2y, uxy and u^2y . We replace each occurrence of xy by $u - 1$, i.e., we replace x^2y by $ux - x$ and uxy by $u^2 - u$. But the monomials ux, x, u^2 and u are already present in the lattice bases. Thus, the only new monomial that comes from the shift $y\bar{f}^2$ is u^2y , thereby retaining the triangular structure.

In order to keep the triangular structure in general, we look at an arbitrary shift $y^i\bar{f}^\ell$. Notice that for the ease of notation we will omit the factor $e^{m-\ell}$ as it does not influence the set of monomials. Since $\bar{f} = u + Ax$ we can expand $y^i\bar{f}^\ell$ by the binomial theorem

$$u^\ell y^i + \binom{\ell}{1} Au^{\ell-1}xy^i + \dots + \binom{\ell}{\ell} A^\ell x^\ell y^i.$$

The first term introduces a new monomial $u^\ell y^i$. However, we will now derive a certain restriction under which all other monomials are already present in the lattice basis. Let us therefore look at the monomials of the second term after the substitution of xy

$$u^{\ell-1}xy^i = u^{\ell-1}(u - 1)y^{i-1} = u^\ell y^{i-1} - u^{\ell-1}y^{i-1}.$$

The monomials $u^\ell y^{i-1}$ and $u^{\ell-1}y^{i-1}$ appear in $y^{i-1}\bar{f}^\ell$ and $y^{i-1}\bar{f}^{\ell-1}$, respectively. In general, the $(j + 1)^{th}$ term of the binomial expansion contains monomials that appear in $y^{i-j}\bar{f}^{\ell-k}$ for $k = 0, \dots, j$.

Therefore, the shift $y^i\bar{f}^\ell$ introduces exactly one new monomial $u^\ell y^i$ if all shifts $y^{i-j}\bar{f}^{\ell-k}$ for $j = 1, \dots, i - 1$ and $k = 0, \dots, j$ were used in the construction of the

lattice basis. This is exactly to the restriction that was called *increasing pattern* in [BM01].

Since the y -shifts $h_{j,k}$ in the original Boneh-Durfee attack satisfy this *increasing pattern* restriction as shown in [BM01], we take in our analysis the y -shifts $\bar{h}_{j,k}$ for the same set of indices (j, k) as in [BD99]. I.e., we define the y -shifts

$$\bar{h}_{j,k} = y^j \bar{f}^k e^{m-k} \text{ for } j = 1, \dots, t \text{ and } k = \left\lfloor \frac{m}{t} \right\rfloor j, \dots, m. \quad (2)$$

We show that this set of y -shifts $\bar{h}_{j,k}$ satisfies our requirement, i.e. we show that if $y^i \bar{f}^\ell$ is a y -shift, then all of $y^{i-j} \bar{f}^{\ell-k}$ for $j = 1, \dots, i-1$ and $k = 0, \dots, j$ are also used as shifts. Notice that it is sufficient to show $y^{i-j} \bar{f}^{\ell-j}$ is used as a shift.

Since $y^i \bar{f}^\ell$ is in the set of y -shifts, we know that $\ell \in \{\lfloor \frac{m}{t} \rfloor i, \dots, m\}$ and therefore $\ell - j \in \{\lfloor \frac{m}{t} \rfloor (i-j), \dots, m-j\}$. For $y^{i-j} \bar{f}^{\ell-j}$ on the other hand, we have $\ell - j \in \{\lfloor \frac{m}{t} \rfloor (i-j), \dots, m\}$. Our requirement is thus fulfilled if the condition

$$\left\lfloor \frac{m}{t} \right\rfloor (i-j) \leq \left\lfloor \frac{m}{t} \right\rfloor i - j$$

holds. We can rewrite this as $\lfloor \frac{m}{t} \rfloor \geq 1$, which holds if $m \geq t$.

Given the set of shift polynomials, we proceed with the computation of the determinant. For the following asymptotic analysis we let $t = \tau m$. Further, for the optimization we omit roundings as their contribution is negligible for sufficiently large m .

We are able to directly compute the contributions of the shift polynomials from (1) and (2). Here, we denote by s_x the contribution of X to the determinant.

$$\begin{aligned} s_x &= \sum_{k=0}^m \sum_{i=0}^{m-k} i = \frac{1}{6} m^3 + o(m^3) \\ s_y &= \sum_{j=1}^{\tau m} \sum_{k=\frac{1}{\tau} j}^m j = \frac{\tau^2}{6} m^3 + o(m^3) \\ s_u &= \sum_{k=0}^m \sum_{i=0}^{m-k} k + \sum_{j=1}^{\tau m} \sum_{k=\frac{1}{\tau} j}^m k = \left(\frac{1}{6} + \frac{\tau}{3} \right) m^3 + o(m^3) \\ s_e &= \sum_{k=0}^m \sum_{i=0}^{m-k} (m-k) + \sum_{j=1}^{\tau m} \sum_{k=\frac{1}{\tau} j}^m (m-k) = \left(\frac{1}{3} + \frac{\tau}{6} \right) m^3 + o(m^3) \\ \dim(L) &= \sum_{k=0}^m \sum_{i=0}^{m-k} 1 + \sum_{j=1}^{\tau m} \sum_{k=\frac{1}{\tau} j}^m 1 = \left(\frac{1}{2} + \frac{\tau}{2} \right) m^2 + o(m^2) \end{aligned}$$

Using these values together with the upper bounds $X = N^\delta, Y = N^{\frac{1}{2}}, U = N^{\delta + \frac{1}{2}}$ on the variables in the usual enabling condition $\det(L) = X^{s_x} Y^{s_y} U^{s_u} e^{s_e} \leq e^{m \dim(L)}$, we obtain an optimized value of $\tau = (1 - 2\delta)$ and finally derive the desired Boneh-Durfee bound¹

$$\delta \leq \frac{1}{2} \left(2 - \sqrt{2} \right) \approx 0.292.$$

¹ The given bound is for full size e , i.e. we set $e \approx N$.

Notice that our choice of τ fulfills our previous restriction $m \geq t$. To summarize, the method of unravelled linearization provides a simple and elegant way to capture the sublattice structure in the Boneh-Durfee attack. In the following section, we will use the same method to recover the hidden sublattice structure in the Jochemsz-May attack on small CRT-RSA exponents. This sublattice was previously unknown and was conjectured to be the key for improving the CRT-RSA attack bound.

4 CRT Exponents

The task of attacking small CRT exponents was first mentioned as an open problem in Wiener [Wie90]. At PKC '06, Bleichenbacher and May [BM06] gave an attack that worked in the case where e is significantly smaller than N . They started with the CRT-RSA equations $ed_p = 1 + k(p-1)$ and $ed_q = 1 + l(q-1)$, and derived a single polynomial in the unknowns (d_p, d_q, k, l) by setting $q = \frac{N}{p}$ and eliminating p :

$$e^2 d_p d_q - e(d_p + d_q) + e(d_q k + d_p l) - (k + l - 1) - (N - 1)kl = 0. \quad (3)$$

This equation can be linearized to

$$e^2 x_1 + e x_2 - (N - 1)x_3 - x_4 = 0 \quad (4)$$

with unknowns

$$x_1 = d_p d_q, x_2 = d_q k + d_p l - d_p - d_q, x_3 = kl, x_4 = (k + l - 1).$$

For $d_p, d_q \leq N^\delta$ we get $k, l \leq N^{\frac{1}{2} + \delta}$ and Eq. (4) directly leads to a lattice attack provided that $\delta \leq \min\{\frac{1}{4}, \frac{2}{5} - \frac{2}{5}\alpha\}$, where $\alpha = \log_N e$. However, for a full size e , i.e. $\alpha = 1$, this attack does not work.

In 2007, Jochemsz and May [JM07] improved the analysis by exploiting the full algebraic structure of Eq. (3) with a Coppersmith-type attack. For the case $\alpha = 1$, they showed that it is possible to find small solutions if $\delta \leq 0.073$. However, in their experiments they noticed a big gap between the theoretically predicted bound and the experimentally observed bound. Namely, the experiments were far better than theoretically expected indicating the possibility of a better bound.

E.g., using their analysis, a lattice dimension of 56 should not suffice for attacking small CRT-exponents, while practically it allows for solving up to $d_p, d_q \leq N^{0.01}$. Jochemsz and May reported that the smallest LLL vectors came from a sublattice and conjectured that identifying the sublattice structure would improve the bound – analogous to the case of the Boneh-Durfee attack where the sublattice lifts the bound from $N^{0.284}$ to $N^{0.292}$.

In this section, we show that this conjecture is false. By using the method of unravelled linearization, we will capture the sublattice structure behind the Jochemsz-May attack. This will completely explain the experimental behavior in [JM07] and therefore close the gap between practice and theoretical analysis.

As a result, we construct lattices of much smaller dimension than in [JM07], whose theoretical analysis exactly matches the experiments that we present in the subsequent section.

Very disappointingly from a cryptanalytic point of view, the size of the CRT-exponents d_p, d_q that we are able to attack in polynomial time converges for growing lattice dimension to the same bound $N^{0.073}$ as in [JM07]. Thus, asymptotically we are unable to improve on the bound although we fully exploit the sublattice structure. Nevertheless, we think that our method is of independent interest and will prove to be useful for other attacks since it is simple and leads to an easy analysis.

Let us describe the attack in detail. Starting point is the polynomial equation (3). We proceed similar to [BM06] and perform an (almost) identical linearization.

$$e^2 \underbrace{d_p d_q}_u - e \underbrace{(d_p + d_q)}_v + e \underbrace{(d_q k + d_p l)}_w - \underbrace{(k + l - 1)}_x - (N - 1) \underbrace{kl}_y = 0 \quad (5)$$

We now use the method of unravelled linearization with the linear polynomial $f = e^2 u - ev + ew - x - Ay + 1$, where $A = N - 1$. The next step is to build up a lattice following the extended strategy from [JM06]. This means we use the monomials of f^{m-1} as shifts and furthermore include extrashifts in the variables u and v up to some parameter t which has to be optimized later.

The benefit in unravelled linearization comes from the fact that the variables u, v, w, x, y are related. Namely, we have

$$\begin{aligned} vwx &= (d_p + d_q)(d_p l + d_q k)(k + l) \\ &= d_p^2 kl + d_p^2 l^2 + d_p d_q (k + l)^2 + d_q^2 k^2 + d_q^2 kl \\ &= (d_p^2 + d_q^2)kl + (d_p^2 l^2 + d_q^2 k^2) + d_p d_q (k + l)^2 \\ &= ((d_p + d_q)^2 - 2d_p d_q)kl + ((d_p l + d_q k)^2 - 2d_p d_q kl) + d_p d_q (k + l)^2 \\ &= (v^2 - 2u)y + w^2 - 2uy + ux^2. \end{aligned} \quad (6)$$

This non-obvious relation can be computed easily using a Gröbner basis computation. Recall the equations given by the linearization. These are 5 linearization equations in 9 unknowns, so we can eliminate via Gröbner basis computation the four variables d_p, d_q, k, l and obtain Eq. (6) in the unknowns u, v, w, x, y only. This equation now serves in the back-substitution step of unravelled linearization, where we replace each occurrence of vwx by the monomials $v^2 y, uy, w^2$ and ux^2 .

To exemplify our method, we use the parameters $m = 2$ and $t = 1$. This is the smallest choice where Jochemsz and May [JM07] found positive experimental results. In the framework of unravelled linearization, it is obvious why we do not obtain a positive result for smaller parameters. In order to improve upon the bound from Bleichenbacher, May [BM06], we have to use relation (6). However, the lattice parameters $m = 2$ and $t = 1$ are the smallest ones for which the monomial vwx appears.

A lattice basis B for $(m, t) = (2, 1)$ is given in Figure 3. We use here the notation from the original Coppersmith method over the integers – as opposed to the modular approach taken in Section 3. That is, we construct a lattice basis with the coefficient vectors of the shift polynomials as column vectors (refer to [Cop97] for details). For simplicity we omit the left hand side of the basis matrix, which contains just the inverses of the corresponding upper bounds of the monomials on its diagonal. The entries that come from the substitution are printed in bold letters.

For the lattice attack to work, we require the enabling condition $\det(L) > 1$ (see [Cop97]). In our example, computation of the determinant of the basis matrix yields

	f	uf	vf	xf	wf	yf	u^2f	uvf	uwf	uxf	yuf	v^2f	vwf	vxf	yvf
u	e^2	1													
v	e		1												
w	e			1											
x	1				1										
1	1														
u^2	e^2					1									
uv	e	e^2					1								
uw	e				e^2			1							
ux	1		e^2						1						
v^2		e									1				
vw		e			e							1			
vx		1		e									1		
wx			1	e	1									1	
x^2				1											e
w^2					e								1		
u^3						e^2									
u^2v						e	e^2								
u^2w						e		e^2							
u^2x						1			e^2						
uw^2											e^2				
uvw							e	e					e^2		
uvx							1	e	e					e^2	
uw^2								e							
uwx								1	e						
ux^2									1					1	e
v^3											e				
v^2w											e		e		
v^2x											1		e		e
vw^2												e			
wx^2														1	
y	A					1									
yu		A				e^2				1			-4	$-4e$	
yv			A			e									1
yx				A		1									
yw					A	e									
y^2						A									
yu^2							A				e^2				
yuv								A			e				e^2
yuw									A		e				
yu^2x										A	1				
y^2u											A				
yv^2												A	1	e	e
yvw													A	e	1
yvx														A	1
y^2v															A

Fig. 3. Matrix of unravelled linearized polynomial for $m = 2, t = 1$

$$\det(B) = U^{-21}V^{-20}W^{-14}X^{-14}A^{15}.$$

We have upper bounds $(U, V, W, X) = (N^{2\delta}, N^\delta, N^{\frac{1}{2}+2\delta}, N^{\frac{1}{2}+\delta})$ for the unknowns $(d_p d_q, d_p + d_q, d_q k + d_p l, k + l)$, respectively. Thus, with $A \approx N$ the enabling condition $\det(L) > 1$ reduces to $\delta < \frac{1}{104} \approx 0.01$. This perfectly matches the experimental results of Jochemsz and May for parameters $(m, t) = (2, 1)$.

We now proceed to the asymptotic analysis and start by analyzing the simpler case without any extrashifts. I.e., we shift in the monomials of f^{m-1} only, but we have to exclude all monomials that are divisible by $vw x$, since these can be written as the linear combination from Eq. (6).

To compute the value of the determinant we begin by counting the number of shift polynomials as each one contributes with a factor of A to the determinant. The number of shift polynomials equals the number of monomials in the set

$$\left\{ u^{e_1} v^{e_2} w^{e_3} x^{e_4} y^{e_5} \mid e_i \in \mathbb{N}_0, \sum_{i=1}^5 e_i \leq m-1, e_2 = 0 \text{ or } e_3 = 0 \text{ or } e_4 = 0 \right\}.$$

Their number can be computed as

$$\left| \left\{ (e_1, \dots, e_5) \in \mathbb{N}_0^5 \mid \sum_{i=1}^5 e_i \leq m-1 \right\} \right| - \left| \left\{ (e_1, \dots, e_5) \in \mathbb{N}_0^5 \mid \sum_{i=1}^5 e_i \leq m-1, e_2, e_3, e_4 \geq 1 \right\} \right|.$$

Let us derive the size of the first set by counting. Write $e_1 + e_2 + e_3 + e_4 + e_5 + h = m-1$ for some slack variable $h \in \{0, \dots, m-1\}$ to transform the inequality into an equality. If we set $e'_i = e_i + 1$ and $h' = h + 1$ then the number of tuples that fulfill the equation

$$e'_1 + e'_2 + e'_3 + e'_4 + e'_5 + h' = (m-1) + 6 \quad \text{with } e'_i, h' \geq 1$$

is exactly the number of ordered partitions of $m+5$ in 6 partitions. Let us write $m+5 = 1 + 1 + \dots + 1$, then one obtains an ordered 6-partition of $m+5$ by choosing 5 out of the $m+4$ signs as breakpoints for the partition. We have $\binom{m+4}{5}$ possibilities for this choice.

The size of the second set is derived in a similar fashion, where we require $e'_1 + e_2 + e_3 + e_4 + e'_5 + h' = m+2$. In this case, the number of tuples is $\binom{m+1}{5}$. Summing up, we obtain for the number of shifts

$$\#\text{shifts} = \binom{m+4}{5} - \binom{m+1}{5} = \frac{1}{8}m^4 + o(m^4).$$

The second part contributing to the determinant comes from the monomials that occur in the lattice basis. This is the product of the diagonal entries in the submatrix on the left that has been omitted in Figure 3. As mentioned before,

the diagonal entries consist of the inverses of the upper bound of the monomial corresponding to that row. The explicit computation is given in Appendix A, while we only state the results here.

$$\begin{aligned}\#u &= \binom{m+1}{2} + 3 \binom{m+2}{4} = \frac{1}{8}m^4 + o(m^4) \\ \#v = \#w = \#x &= \binom{m+2}{3} + 2 \binom{m+2}{4} = \frac{1}{12}m^4 + o(m^4).\end{aligned}$$

Recall that the enabling condition for the lattice attack is $\det(L) > 1$. With the previously derived values and neglecting low order terms as well as setting $A = N$, we are able to write the determinant as

$$\det(L) = U^{-\frac{1}{8}m^4} V^{-\frac{1}{12}m^4} W^{-\frac{1}{12}m^4} X^{-\frac{1}{12}m^4} N^{\frac{1}{8}m^4}.$$

If we use the upper bounds $(U, V, W, X) = (N^{2\delta}, N^\delta, N^{\frac{1}{2}+2\delta}, N^{\frac{1}{2}+\delta})$ on the sizes of the variables, we derive the condition

$$\delta < \frac{1}{14} \approx 0.071.$$

This is the same asymptotic bound that was obtained by Jochemsz and May [JM07] without extrashifts. So, unfortunately, our new lattice does not improve the asymptotic bound of [JM07]. But, as opposed to [JM07], our approach requires smaller lattice dimensions. Asymptotically, [JM07] need to LLL-reduce a lattice of size m^3 , while our approach requires only lattice dimension $\frac{1}{2}m^3$. Figure 4 shows a comparison of the two methods in terms of the size of d_p, d_q that can be attacked.

While our approach clearly allows for attacking larger values of CRT-exponents in practice, we would also like to stress the fact that as opposed

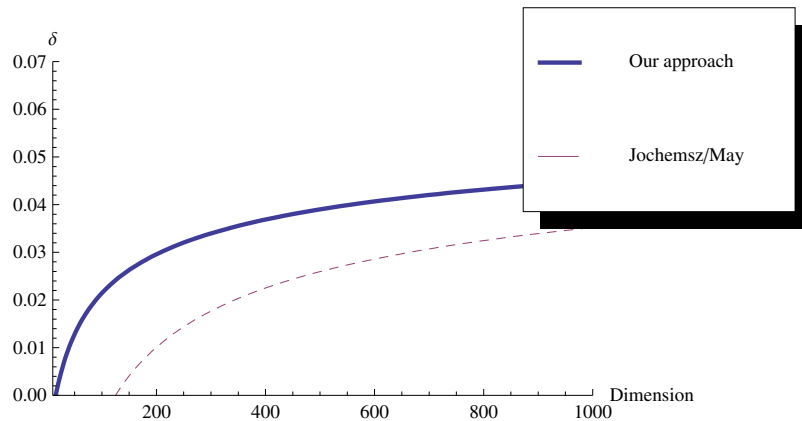


Fig. 4. Comparison of the achievable bound depending on the lattice dimension

to [JM07] the experimental behavior of our attack can be completely explained by our theoretical analysis – thereby also explaining the experimental behavior of [JM07]. We will show this in the subsequent section.

If we also use so-called extrashifts then we end up with a slightly improved bound of $d_p, d_q \leq N^{0.073}$ as in [JM07]. The analysis can be done in a similar fashion to the case without extrashifts. We carry out the calculations in Appendix B.

5 Experiments

The reason for carrying out various experiments for attacking CRT-RSA is twofold. First, we want to show that our analysis from Section 4 is indeed optimal. That is, the experimental behavior can be perfectly predicted by the analysis and there is no hope to improve the bound by this approach. Second, as our lattice-based approach is heuristic, we have to verify that the polynomials that we obtain after the lattice reduction are indeed coprime and thus allow for efficient recovery of their roots.

Table 1. Experimental Results

N	d_p, d_q	δ	lattice parameters	dim JM	LLL-time JM	LLL-time(s)
1000 bit	11 bit	0.0096	$m = 2, t = 1, dim = 30$	56	14	2
1000 bit	18 bit	0.0178	$m = 3, t = 1, dim = 60$	115	6100	258
1000 bit	22 bit	0.0226	$m = 3, t = 2, dim = 93$	–	–	3393
1000 bit	24 bit	0.0244	$m = 4, t = 1, dim = 105$	–	–	7572
1000 bit	29 bit	0.0291	$m = 4, t = 2, dim = 154$	–	–	61298
2000 bit	21 bit	0.0096	$m = 2, t = 1, dim = 30$	56	40	4
2000 bit	35 bit	0.0178	$m = 3, t = 1, dim = 60$	115	20700	613
2000 bit	45 bit	0.0226	$m = 3, t = 2, dim = 93$	–	–	13516
2000 bit	47 bit	0.0244	$m = 4, t = 1, dim = 105$	–	–	34305
5000 bit	48 bit	0.0096	$m = 2, t = 1, dim = 30$	56	379	39
5000 bit	89 bit	0.0178	$m = 3, t = 1, dim = 60$	–	–	5783
5000 bit	113 bit	0.0226	$m = 3, t = 2, dim = 93$	–	–	74417
10000 bit	96 bit	0.0096	$m = 2, t = 1, dim = 30$	56	2500	360
10000 bit	179 bit	0.0178	$m = 3, t = 1, dim = 60$	–	–	31226

We reimplemented the attack of [JM07] and used in the experiments the same modulus sizes and lattice parameters as done in [JM07]. Table 1 clearly shows the speedup for the LLL reduction. For example with parameters $m = 3$ and $t = 1$ our method is 20 to 30 times faster than the one of Jochemsz and May. As previously mentioned, this is due to the reduced lattice dimension². While Jochemsz and May required the reduction of a lattice of dimension 115, our lattice only has dimension 60. Because of this smaller lattice dimension we were

² The lattice we are considering here is the one that serves as input to the LLL reduction routine. That is the sublattice containing zeros in the coordinates corresponding to the shift polynomials.

able to perform experiments on parameter sets that have been out of reach before.

Notice that the experimental results on the achievable sizes of d_p and d_q perfectly match the theoretically predicted bound δ . This is a strong indication that our approach is indeed optimal.

We ran our experiments using sage 4.1.1. and used the L^2 reduction algorithm from Nguyen and Stehlé [NS09]. The calculations were performed on an Quad Core Intel Xeon processor running at 2.66 GHz.

References

- [BD99] Boneh, D., Durfee, G.: Cryptanalysis of RSA with Private Key d Less than $N^{0.292}$. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 1–11. Springer, Heidelberg (1999)
- [BM01] Blömer, J., May, A.: Low Secret Exponent RSA Revisited. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 4–19. Springer, Heidelberg (2001)
- [BM06] Bleichenbacher, D., May, A.: New Attacks on RSA with Small Secret CRT-Exponents. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T.G. (eds.) PKC 2006. LNCS, vol. 3958, pp. 1–13. Springer, Heidelberg (2006)
- [Cop96a] Coppersmith, D.: Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known. In: Maurer [Mau96], pp. 178–189 (1996)
- [Cop96b] Coppersmith, D.: Finding a Small Root of a Univariate Modular Equation. In: Maurer [Mau96], pp. 155–165
- [Cop97] Coppersmith, D.: Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. *J. Cryptology* 10(4), 233–260 (1997)
- [HG97] Howgrave-Graham, N.: Finding Small Roots of Univariate Modular Equations Revisited. In: Darnell, M.J. (ed.) Cryptography and Coding 1997. LNCS, vol. 1355, pp. 131–142. Springer, Heidelberg (1997)
- [HM09] Herrmann, M., May, A.: Attacking Power Generators Using Unravelling Linearization: When Do We Output Too Much? In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 487–504. Springer, Heidelberg (2009)
- [JM06] Jochemsz, E., May, A.: A Strategy for Finding Roots of Multivariate Polynomials with New Applications in Attacking RSA Variants. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 267–282. Springer, Heidelberg (2006)
- [JM07] Jochemsz, E., May, A.: A Polynomial Time Attack on RSA with Private CRT-Exponents Smaller Than $N^{0.073}$. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 395–411. Springer, Heidelberg (2007)
- [LLL82] Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring Polynomials with Rational Coefficients. *Mathematische Annalen* 261(4), 515–534 (1982)
- [Mau96] Maurer, U.M. (ed.): EUROCRYPT 1996. LNCS, vol. 1070. Springer, Heidelberg (1996)
- [NS09] Nguyen, P.Q., Stehlé, D.: An LLL Algorithm with Quadratic Complexity. *SIAM J. Comput.* 39(3), 874–903 (2009)
- [QC82] Quisquater, J.J., Couvreur, C.: Fast Decipherment Algorithm for RSA Public-key Cryptosystem. *Electronics Letters* 18, 905 (1982)
- [Wie90] Wiener, M.J.: Cryptanalysis of Short RSA Secret Exponents. *IEEE Transactions on Information Theory* 36(3), 553–558 (1990)

A Counting $\#u$, $\#v$, $\#w$, $\#x$

The monomials that contribute to the determinant are exactly the monomials of f^m that do not contain the variable y . Denote such a monomial by $u^{e_1}v^{e_2}w^{e_3}x^{e_4}$. In order to count the number of u 's that contribute to the determinant we proceed as follows.

Let $e_1 = 0$. We have $e_2 + e_3 + e_4 \leq m$ with $e_i \in \mathbb{N}_0$, which transform into $e'_2 + e'_3 + e'_4 + h' \leq m + 4$ for a slack variable $h' \in \{1, \dots, m + 1\}$ and $e'_i = e_i + 1$. The number of such tuples is just the number of 4-partitions of $m + 4$, which is $\binom{m+3}{3}$. From these tuples we have to remove the ones with $e_i \geq 1$ for $i = 2, 3, 4$, because of the substitutions of vw . The number of these tuples is $\binom{m}{3}$. For $e_1 = 1$, we proceed similarly and obtain $\binom{m+2}{3} - \binom{m-1}{3}$. We carry this out for all possibilities of e_1 and end up with $e_1 = m$, where we get $\binom{3}{3} - \binom{0}{3}$.

Now we know the number of occurrences for each power u^i , $i = 0, \dots, m$. In order to count the total number of u we compute the weighted sum as follows.

$$\begin{aligned} \#u &= \sum_{i=3}^{m+3} (m+3-i) \binom{i}{3} - \sum_{i=0}^m (m-i) \binom{i}{3} \\ &= \sum_{i=m+1}^{m+3} (m+3-i) \binom{i}{3} + \sum_{i=3}^m ((m+3-i) - (m-i)) \binom{i}{3} \\ &= 2 \binom{m+1}{3} + \binom{m+2}{3} + 3 \sum_{i=3}^m \binom{i}{3} = \binom{m+2}{3} - \binom{m+1}{3} + 3 \sum_{i=3}^{m+1} \binom{i}{3} \end{aligned}$$

Using the identities $\binom{n}{k} - \binom{n-1}{k} = \binom{n-1}{k-1}$ and $\sum_{i=0}^n \binom{i}{k} = \binom{n+1}{k+1}$ we eventually obtain

$$\#u = \binom{m+1}{2} + 3 \binom{m+2}{4}.$$

Thus, $\#u = \frac{1}{8}m^4 + o(m^4)$.

Counting the number of occurrences of v , w and x can be done in a similar way and we obtain

$$\begin{aligned} \#v = \#w = \#x &= \sum_{i=3}^{m+3} (m+3-i) \binom{i}{3} - \sum_{i=1}^{m+1} (m+1-i) \binom{i}{3} \\ &= 2 \sum_{i=0}^{m+1} \binom{i}{3} + \binom{m+2}{3} = 2 \binom{m+2}{4} + \binom{m+2}{3} \\ &= \frac{1}{12}m^4 + o(m^4). \end{aligned}$$

B Improving the Bound Using Extrashifts

In the following we will show that it is possible to improve the bound $\delta < \frac{1}{14} \approx 0.0714$ to $\delta \approx 0.0734$ by using so-called extrashifts. In this case, we use the set of shifts

$$S = \bigcup_{t_1=0}^t \bigcup_{t_2=0}^{t-t_1} \{u^{e_1+t_1} v^{e_2+t_2} w^{e_3} x^{e_4} y^{e_5} \mid u^{e_1} v^{e_2} w^{e_3} x^{e_4} y^{e_5} \text{ is monomial of } f^{m-1}\}.$$

To estimate the number of shifts, one may use a combinatorial proof as in Section 4 and count the number of all monomials minus the monomials having $e_2, e_3, e_4 \geq 1$. However, we choose to use a computational approach here and simply evaluate a series of sums.

The shift monomials can be characterized by the set $S_1 \setminus S_2$, where S_1 is the set of all shifts and S_2 are the shifts that have to be removed due to the substitution of vwx .

$$u^{e_1} v^{e_2} w^{e_3} x^{e_4} y^{e_5} \in S_1 \Leftrightarrow \begin{cases} e_5 = 0, \dots, m-1 \\ e_4 = 0, \dots, m-1 - e_5 \\ e_3 = 0, \dots, m-1 - e_5 - e_4 \\ e_2 = 0, \dots, m-1 - e_5 - e_4 - e_3 + t \\ e_1 = 0, \dots, m-1 - e_5 - e_4 - e_3 - e_2 + t \end{cases}$$

$$u^{e_1} v^{e_2} w^{e_3} x^{e_4} y^{e_5} \in S_2 \Leftrightarrow \begin{cases} e_5 = 0, \dots, m-1 \\ e_4 = 1, \dots, m-1 - e_5 \\ e_3 = 1, \dots, m-1 - e_5 - e_4 \\ e_2 = 1, \dots, m-1 - e_5 - e_4 - e_3 + t \\ e_1 = 0, \dots, m-1 - e_5 - e_4 - e_3 - e_2 + t \end{cases}$$

Setting $t = \tau m$, the resulting number of shifts is

$$|S_1 \setminus S_2| = \left(\frac{1}{8} + \frac{\tau}{2} + \frac{\tau^2}{2} \right) m^4 + o(m^4).$$

In a similar fashion we derive the exponents of the variables u, v, w and x contributing to the determinant. For example, to calculate the number of occurrences of u , we compute

$$\begin{aligned} s_u &= \sum_{e_4=0}^m \sum_{e_3=0}^{m-e_4} \sum_{e_2=0}^{m-e_4-e_3+t} \sum_{e_1=0}^{m-e_4-e_3-e_2+t} e_1 \\ &\quad - \sum_{e_4=1}^m \sum_{e_3=1}^{m-e_4} \sum_{e_2=1}^{m-e_4-e_3+t} \sum_{e_1=0}^{m-e_4-e_3-e_2+t} e_1 \\ &= \left(\frac{1}{8} + \frac{\tau}{2} + \frac{3\tau^2}{4} + \frac{\tau^3}{3} \right) m^4 + o(m^4). \end{aligned}$$

For the other values we obtain

$$\begin{aligned} s_v &= \left(\frac{1}{12} + \frac{\tau}{3} + \frac{\tau^2}{2} + \frac{\tau^3}{3} \right) m^4 + o(m^4) \\ s_w &= \left(\frac{1}{12} + \frac{\tau}{3} + \frac{\tau^2}{4} \right) m^4 + o(m^4) \\ s_x &= \left(\frac{1}{12} + \frac{\tau}{3} + \frac{\tau^2}{4} \right) m^4 + o(m^4). \end{aligned}$$

We use these values together with the upper bounds $(U, V, W, X) = (N^{2\delta}, N^\delta, N^{\frac{1}{2}+2\delta}, N^{\frac{1}{2}+\delta})$ to compute the determinant of the lattice. After that, we are able to solve the enabling condition $\det(L) > 1$ for δ and optimize the value of τ to maximize δ . We obtain $\tau \approx 0.381788$, which finally leads to the bound

$$\delta \leq 0.0734142.$$