Cryptanalysis of Unbalanced RSA with Small CRT-Exponent

Alexander May

Department of Mathematics and Computer Science
University of Paderborn
33102 Paderborn, Germany
alexx@uni-paderborn.de

Abstract. We present lattice-based attacks on RSA with prime factors p and q of unbalanced size. In our scenario, the factor q is smaller than N^{β} and the decryption exponent d is small modulo $p-1$. We introduce two approaches that both use a modular bivariate polynomial equation with a small root. Extracting this root is in both methods equivalent to the factorization of the modulus $N = pq$. Applying a method of Coppersmith, one can construct from a bivariate modular equation a bivariate polynomial $f(x, y)$ over $\mathbb{Z}$ that has the same small root. In our first method, we prove that one can extract the desired root of $f(x, y)$ in polynomial time. This method works up to $\beta < \frac{3-\sqrt{5}}{2} \approx 0.382$. Our second method uses a heuristic to find the root. This method improves upon the first one by allowing larger values of d modulo $p-1$.

Keywords: RSA, lattice reduction, Coppersmith's method, small secret exponent

1 Introduction

An RSA key is a tuple (N, e) where $N = pq$ is the product of two primes and e is the public key. The corresponding secret key d satisfies the equation $ed = 1 \bmod \frac{(p-1)(q-1)}{2}$ with $\gcd(p-1, \frac{q-1}{2}) = 1$. The Chinese Remainder Theorem (CRT) gives us the equations $ed = 1 \bmod p-1$ and $ed = 1 \bmod \frac{q-1}{2}$.

To speed up the RSA decryption and signature generation process, one is tempted to use small secret decryption exponents d. Unfortunately, Wiener [17] showed that $d < \frac{1}{3}N^{\frac{1}{4}}$ leads to a polynomial time attack on the RSA cryptosystem. This result was generalized by Verheul and Tilborg [16] to the case where one guesses high-order bits of the prime factors. They showed that in order to improve Wiener's bound for r bits one has to guess approximately $2r$ bits.

Recently, Boneh and Durfee [3] showed how to improve the bound of Wiener up to $d < N^{0.292}$. Their attack works in polynomial time and builds upon Coppersmith's method for finding small roots of modular polynomial equations. This method in turn is based on the famous L^3-lattice reduction algorithm of Lenstra, Lenstra and Lovász [9]. Coppersmith's method is rigorous for the univariate case but the proposed generalization in the modular multivariate case is a heuristic.

Since Boneh and Durfee use Coppersmith's method in the bivariate modular case, their attack is a heuristic. In contrast, the approach of Wiener is a provable method. However, the Boneh-Durfee attack works very well in practice. In fact, many other works (e.g. [1, 5, 8]) are based on this useful heuristical multivariate approach.

The results above show that one cannot use a small decryption exponent d. But there is another way to speed up the decryption and signature generation process. One can use a decryption exponent d such that $d_p = d \bmod p - 1$ and $d_q = d \bmod \frac{q-1}{2}$ are small. Such an exponent d is called a small CRT-exponent. In order to sign a message m, one computes $m^{d_p} \bmod p$ and $m^{d_q} \bmod q$. Both terms are combined using the Chinese Remainder Theorem to yield the desired term $m^d \bmod N$. The attacks described before do not work in this case, since d is likely to be large.

It is an open problem if there is a polynomial time algorithm that breaks RSA if d_p and d_q are small. This problem is mentioned several times in the literature, see e.g. [17, 2, 3]. The best algorithm that is known runs in time $O(\min(\sqrt{d_p}, \sqrt{d_q}))$ which is exponentially in the bit-size.

In this work, we give the first polynomial time attack on RSA with small CRT-exponent. Unfortunately, our results are restricted to the case of unbalanced prime numbers p and q. The use of unbalanced primes was first proposed by Shamir [13] to guard the modulus N against different kinds of factorization algorithms and to speed up the computation. There are also other systems that use unbalanced primes [10, 15]. Interestingly, sometimes the use of unbalanced primes decreases the security. For instance, Durfee and Nguyen [5] showed that the Boneh-Durfee attack works for larger exponents d if the prime factors are unbalanced. This breaks the RSA-type scheme of Sun, Yang and Laih [15].

We show in the following work that there is also a decrease in security for unbalanced primes when using small CRT-exponents. The more unbalanced the prime factors are, the larger are the CRT-exponents that can be attacked by our methods.

Let $q < N^\beta$ and $d_p \leq N^\delta$. We show in Section 3 that an RSA public key tuple (N, e) satisfying the condition $3\beta + 2\delta \leq 1 - \log_N(4)$ yields the factorization of N in time $O(\log^2(N))$. Thus, this method does only work provided that $\beta < \frac{1}{3}$.

Like the methods in [1, 3, 5, 8], our approach is based on Coppersmith's technique [4] in the modular multivariate case. More precisely, we use a modular bivariate polynomial equation with a small root. This root gives us the factorization of N. Using a Theorem of Howgrave-Graham [7], we can turn the modular bivariate polynomial into a polynomial $f(x, y)$ over $\mathbb{Z}$ such that the desired small root must be among the roots of $f(x, y)$. Interestingly, for the polynomial $f(x, y)$ we are able to prove that this small root can be extracted easily. This shows that our method provably factors the modulus N. Note, that this is in contrast to other works using the multivariate approach [1, 3, 5, 8] which rely on a heuristic assumption. To our knowledge, this is the first rigorous method using a modular bivariate approach. We think that this method will be useful in other settings as well. As an example, we show that our technique yields an elegant and simple proof of the results of Wiener[17] and Verheul, Tilborg [16].

The attack in Section 3 uses a two-dimensional lattice. In Section 4, we generalize our method to lattices of arbitrary dimension. This improves the condition above to $3\beta - \beta^2 + 2\delta \leq 1 - \epsilon$ for some small error term ϵ. Therefore, this approach works as long as $\beta < \frac{3-\sqrt{5}}{2} = \hat{\phi}^2$, where $\hat{\phi} = \frac{1-\sqrt{5}}{2}$ is the conjugate of the golden ratio. Again, we can show that the desired root can be extracted in polynomial time. This yields a rigorous method for factoring N.

In Section 5, we use a different modular bivariate polynomial. This approach works for larger CRT-exponents than our first attack. Unfortunately, we cannot give a rigorous proof for this method. It relies on Coppersmith's heuristic for modular multivariate polynomials.

Finally, we compare our approaches in Section 6.

2 Preliminaries

Let $\mathbb{Z}_N$ denote the ring of integers modulo N. Let $\mathbb{Z}_N^*$ denote the multiplicative group of invertible integers modulo N. The order of $\mathbb{Z}_N^*$ is given by the Euler phi-function $\phi(N)$. Using RSA, we have $N = pq$ and $\phi(N) = (p-1)(q-1)$. If x is a random element in Z_N^*, we use the notation $x \in_R \mathbb{Z}_N^*$.

Let $f(x,y) = \sum_{i,j} a_{i,j} x^i y^j \in \mathbb{Z}[x,y]$ be a bivariate polynomial with coefficients $a_{i,j}$ in the ring of integers. We will often use the short-hand notation f when the parameters follow from the context. The degree of f is the maximal sum $i+j$ taken over all monomials $a_{i,j}x^i y^j$ with non-zero coefficients. The coefficient vector of f is the vector of the coefficients $a_{i,j}$. The Euclidean norm of f is defined as the norm of the coefficient vector: $\|f\|^2 = \sum_{i,j} a_{i,j}^2$.

In the following, we state a few basic facts about lattices and lattice basis reduction and refer to the textbooks [6, 14] for an introduction into the theory of lattices.

Let $v_1, \ldots, v_n \in \mathbb{R}^m$, $m \geq n$ be linearly independent vectors. A lattice L spanned by $\{v_1, \ldots, v_n\}$ is the set of all integer linear combinations of $v_1, \ldots, v_n$. If $m = n$, the lattice is called a full rank lattice. The set of vectors $B = \{v_1, \ldots, v_n\}$ is called a basis for L.

We denote by $v_1^*, \ldots, v_n^*$ the vectors obtained by applying Gram-Schmidt orthogonalization to the basis vectors. The determinant of L is defined as

$$\det(L) = \prod_{i=1}^{n} \|v_i^*\|,$$

where $\|v\|$ denotes the Euclidean norm of v. Any lattice L has infinitely many bases but all bases have the same determinant. If a lattice is full rank, $\det(L)$ is the absolute value of the determinant of the $(n \times n)$-matrix whose rows are the basis vectors $v_1, \ldots, v_n$. Hence if the basis matrix is triangular, the determinant is very easy to compute.

A well-known result by Minkowski relates the determinant of a lattice L to the length of a shortest vector in L. Minkowski's Theorem shows that every

n-dimensional lattice L contains a non-zero vector v with $\|v\| \leq \sqrt{n}\det(L)^{\frac{1}{n}}$. Unfortunately, the proof of this theorem is non-constructive.

In dimension 2, the Gauss reduction algorithm yields a shortest vector of a lattice. In arbitrary dimension, we can use the famous L^3-reduction algorithm of Lenstra, Lenstra and Lovász [9] to approximate a shortest vector.

Fact 1 (Lenstra, Lenstra and Lovász) *Let L be a lattice spanned by $\{v_1, \ldots, v_n\}$. The L^3-reduction algorithm will output in polynomial time a lattice basis $\{v'_1, \ldots, v'_n\}$ with*

$$\|v'_1\| \leq 2^{\frac{n-1}{4}} \det(L)^{\frac{1}{n}} \ \ \textit{and} \ \ \|v'_2\| \leq 2^{\frac{n}{2}} \det(L)^{\frac{1}{n-1}}.$$

2.1 Key generation using the Chinese Remainder Theorem (CRT)

We briefly describe the key generation process. In our scenario, the RSA modulus N is composed of a large prime factor p and a small prime factor q. The secret decryption exponent d is chosen to be small modulo $p-1$ and of arbitrary size modulo $q-1$.

CRT Key Generation Process

Fix a bit-size n for the public key modulus N. Additionally, fix two positive parameters β, δ with $\beta \leq \frac{1}{2}$ and $\delta \leq 1$.

Modulus: Choose randomly prime numbers p and q with bit-sizes approximately $(1-\beta)n$ and βn. Additionally, $p-1$ and $\frac{q-1}{2}$ must be coprime. Compute the modulus $N = pq$. If the smaller prime factor q does not satisfy $q < N^\beta$, repeat the prime generation.

Secret exponent: Choose a small secret $d_p \in_R Z^*_{p-1}$ such that $d_p \leq N^\delta$. Choose another secret $d_q \in_R Z^*_{\frac{q-1}{2}}$ arbitrarily.

Chinese remaindering: Compute the unique $d \bmod \frac{\phi(N)}{2}$ that satisfies $d = d_p \bmod p-1$ and $d = d_q \bmod \frac{q-1}{2}$.

Public exponent: Compute the inverse e of d in $\mathbb{Z}^*_{\frac{\phi(N)}{2}}$.

Public parameters: Publish the tuple (N, e).

In this work, we will study the following question:
Up to which parameter choices for β and δ does the public key tuple (N, e) yield the factorization of N ?

Note, that the decryption and the signature generation process of a message m are very efficient for small β and δ. Since d_p is small, the computation of $m^{d_p} \bmod p-1$ requires only a small amount of multiplications. On the other hand, the computation of $m^{d_q} \bmod \frac{q-1}{2}$ is cheap because q is small. Both terms can easily be combined to yield the desired term $m^d \bmod \frac{\phi(N)}{2}$ using the Chinese Remainder Theorem(CRT).

In the next section, we will show that given the public key (N, e) there is a provable polynomial time algorithm that factors N if the condition $3\beta + 2\delta \leq 1 - \epsilon$

holds, where ϵ is a small error term. This implies that our method works as long as $\beta < \frac{1}{3}$. The smaller β is chosen, the larger δ can be in the attack. For $\beta = 0$, we obtain $\delta < \frac{1}{2}$. Later, we will improve the bound for β up to $\frac{3-\sqrt{5}}{2} \approx 0.382$ and for δ up to 1.

3 An approach modulo p

Given a public key (N, e) that is constructed according to the CRT Key Generation process. We know that

$$ed_p = 1 \bmod p - 1.$$

Thus, there is an integer k such that

$$ed_p + k(p-1) = 1 \quad \text{over } \mathbb{Z}. \tag{1}$$

We can rewrite this equation as

$$ed_p - (k+1) = -kp \tag{2}$$

In the following, we assume that q does not divide k. Otherwise, the right hand side of the equation is a multiple of N and we can obtain much stronger results. This case will be analyzed later.
Equation (2) gives us the polynomial

$$f_p(x, y) = ex - y$$

with a root $(x_0, y_0) = (d_p, k+1)$ modulo p.
By construction, we have $d_p \leq N^{\delta}$. Since $e < \frac{(p-1)(q-1)}{2}$, we obtain

$$|k+1| = \left|\frac{ed_p - p}{p-1}\right| < \frac{ed_p}{p-1} < \frac{q-1}{2} d_p < N^{\beta+\delta}.$$

Let as define two upper bounds $X = N^{\delta}$ and $Y = N^{\beta+\delta}$. Then, we have a modular bivariate polynomial equation f_p with a small root (x_0, y_0) that satisfies $|x_0| \leq X$ and $|y_0| \leq Y$. This modular equation can be turned into an equation over the integers using a theorem of Howgrave-Graham.

Fact 2 (Howgrave-Graham) *Let $f(x, y)$ be a polynomial that is a sum of at most ω monomial. Suppose $f(x_0, y_0) = 0 \bmod p^m$ for some positive integer m, where $|x_0| \leq X$ and $|y_0| \leq Y$. If $\|f(xX, yY)\| < \frac{p^m}{\sqrt{\omega}}$, then $f(x_0, y_0) = 0$ holds over the integers.*

Using our polynomial $f_p(x, y)$, we want to construct a polynomial $f(x, y)$ that satisfies the conditions of Howgrave-Graham's theorem. Since we have to find a small Euclidean norm polynomial $f(xX, yY)$, we use lattice reduction methods. Our first approach uses a lattice of dimension 2. In that dimension, the Gauss reduction algorithm finds a shortest vector.

Let m be the integer defined in Fact 2. We choose $m = 1$. Next, we use the helper polynomial $f_0(x) = Nx$ that also has the root x_0 modulo p, since N is a multiple of p. Therefore, every integer linear combination of f_0 and f_p has the root (x_0, y_0) modulo p. We construct a lattice L_p that is spanned by the coefficient vectors of the polynomials $f_0(xX)$ and $f_p(xX, yY)$. These coefficient vectors are the row vectors of the following (2×2)-lattice basis B_p:

$$B_p = \begin{bmatrix} NX & \\ eX & -Y \end{bmatrix}$$

We will now give a condition under which the lattice L_p has a vector v with norm smaller than $\frac{p}{\sqrt{2}}$. This vector v can then be transformed into a polynomial $f(x, y)$ satisfying Fact 2.

Lemma 3 *Let $X = N^{\delta}$ and $Y = N^{\beta+\delta}$ with*

$$3\beta + 2\delta \leq 1 - \log_N(4).$$

Then L_p has a smallest vector v with $\|v\| < \frac{p}{\sqrt{2}}$.

Proof: By Minkowski's Theorem, L_p must contain a vector v with $\|v\| \leq \sqrt{2\det(L_p)}$. Thus, v has norm smaller than $\frac{p}{\sqrt{2}}$ if the condition

$$\sqrt{2\det(L_p)} < \frac{p}{\sqrt{2}}$$

holds.
We have $\det(L_p) = NXY$. This implies $NXY < \frac{p^2}{4}$.

By the CRT Key Generation Process, we know $p > N^{1-\beta}$. On the other hand, we have $X = N^{\delta}$ and $Y = N^{\beta+\delta}$.
Hence, we obtain

$$N^{1+\beta+2\delta} \leq \frac{1}{4}N^{2-2\beta} < \frac{p^2}{4}.$$

This implies the condition $3\beta + 2\delta \leq 1 - \log_N(4)$ and the claim follows. □

Using Lemma 3, we obtain for every fixed $\epsilon > 0$ the condition $3\beta + 2\delta \leq 1 - \epsilon$ for suitably large moduli N.

Assume we have found a vector v in L_p with norm smaller than $\frac{p}{\sqrt{2}}$ by lattice reduction. Let v be the coefficient vector of the polynomial $f(xX, yY)$. Applying Fact 2, we know that $f(x, y)$ has a root $(x_0, y_0) = (d_p, k + 1)$ over the integers. The next theorem shows that the root (x_0, y_0) can easily be determined.

Lemma 4 *Let $v = (c_0, c_1) \cdot B_p$ be a shortest vector in L_p with $\|v\| < \frac{p}{\sqrt{2}}$. Then $|c_0| = k$ and $|c_1| = qd_p$.*

Proof: We have $v = c_0(NX, 0) + c_1(eX, -Y)$. Define the polynomial $f(xX, yY)$ that has the coefficient vector v. By construction, $\|f(xX, yY)\| < \frac{p}{\sqrt{2}}$ and we can apply Fact 2.
Therefore, the polynomial

$$f(x, y) = c_0 N x + c_1(ex - y)$$

has the root (x_0, y_0) over $\mathbb{Z}$. Plugging (x_0, y_0) into the equation yields

$$c_0 N x_0 = -c_1(ex_0 - y_0).$$

We know that $(x_0, y_0) = (d_p, k + 1)$. That leads to

$$c_0 N d_p = -c_1(ed_p - (k + 1)).$$

Using equation (2) and dividing by p gives us

$$c_0 q d_p = c_1 k.$$

Since we assumed that q does not divide k, we have $\gcd(qd_p, k) = \gcd(d_p, k)$. Now, let us look at equation (1). Every integer that divides both d_p and k must also divide 1. Hence, $\gcd(d_p, k) = 1$.

Thus, we obtain

$$c_0 = ak \quad \text{and} \quad c_1 = aqd_p$$

for some integer a. But v is a shortest vector in L_p. Therefore, we must have $|a| = 1$ and the claim follows. $\square$

Summing up the results gives us the following theorem.

Theorem 5 *Given an RSA public key tuple (N, e) with $N = pq$ and secret exponent d. Let $q < N^{\beta}$, $d_p \leq N^{\delta}$ and*

$$3\beta + 2\delta \leq 1 - \log_N(4).$$

Then N can be factored in time $O(\log^2(N))$.

Proof: Construct the lattice basis B_p and find a shortest vector $v = (c_0, c_1) \cdot B_p$ using Gauss reduction. Compute $\gcd(N, c_1) = q$. The total running time for Gauss reduction and greatest common divisor computation is $O(\log^2(N))$. $\square$

In the previous analysis, we made the assumption that q does not divide k. If we are in the very unlikely case that $k = qr$ for some $r \in \mathbb{Z}$, then we obtain analogous to the reasoning before the following stronger result.

Theorem 6 *Given an RSA public key tuple (N, e) with $N = pq$ and secret exponent d. Let $q < N^{\beta}$, $d_p \leq N^{\delta}$,*

$$k = qr \text{ and } \beta + 2\delta \leq 1 - \log_N(4).$$

Then N can be factored in time $O(\log^2(N))$.

Proof: The polynomial $f_p(x,y) = ex - y$ has the root $(x_0, y_0) = (d_p, k+1)$ not just modulo p but also modulo N. Thus, we can use the modulus N in Fact 2. Analogous to Lemma 3, we conclude that L_p has a shortest vector v with norm smaller than $\frac{N}{\sqrt{2}}$ as long as the condition $\beta + 2\delta \leq 1 - \log_4(N)$ holds. Following the proof of Lemma 4, we see that $v = (c_0, c_1) \cdot B_p$ with $|c_0| = r$ and $|c_1| = d_p$. Since $\frac{1-ed_p}{r} = q(p-1)$ by equation (1), the computation $\gcd(\frac{1-ed_p}{r}, N) = q$ reveals the factorization. $\square$

Interestingly , choosing $\beta = \frac{1}{2}$ in Theorem 6 gives us the bound $\delta \leq \frac{1}{4} - \log_N(4)$. This is similar to Wiener's bound in the attack on low secret exponent RSA [17]. In fact, one can prove the results of Wiener and Verheul, Tilborg [16] in terms of lattice theory in the same manner. We briefly sketch how to obtain their results in a simpler fashion.

Verheul and Tilborg studied the case where they guess high order bits of p. Assume we know $\tilde{p}$ with $|p - \tilde{p}| \leq N^{\frac{1}{2}-\gamma}$ and by calculating $\tilde{q} = \frac{N}{\tilde{p}}$ we know an approximation of q with accuracy $N^{\frac{1}{2}-\gamma}$ as well. The RSA-equation $ed + k(N+1-p-q) - 1 = 0$ gives us a polynomial $f_{N'}(x,y) = ex - y$ with root $(x'_0, y'_0) = (d, k(p - \tilde{p} + q - \tilde{q}) + 1)$ modulo $N + 1 - \tilde{p} - \tilde{q}$. We have $|x'_0| \leq N^{\delta}$ and $|y'_0| \leq N^{\delta + \frac{1}{2} - \gamma}$. Working through the arithmetic, this gives us the condition $\delta \leq \frac{1}{4} + \frac{\gamma}{2} - \epsilon$, where ϵ is a small error term. Wiener's result follows as the special case where $\gamma = 0$.

4 Improving the bound to $\beta < N^{0.382}$

Using Theorem 5, our approach with the two-dimensional lattice L_p only works provided that $\beta < \frac{1}{3}$. In this section, we use lattices of larger dimension to make our method work for less unbalanced moduli. We are able to improve the bound up to $\beta < \frac{3-\sqrt{5}}{2} \approx 0.382$.

In section 3, we used Fact 2 with the parameter choice $m = 1$. Now, we generalize the method for arbitrary m.
We define the x-shifted polynomials

$$g_{m,i,j}(x,y) = N^{\max(0,m-j)} x^i f_p^j(x,y),$$

where f_p is defined as in section 3. Note, that every integer linear combination of polynomials $g_{m,i,j}$ has the zero $(x_0, y_0) = (d_p, k+1)$ modulo p^m.

We fix a lattice dimension n. Next, we build a lattice $L_p(n)$ of dimension n using as basis vectors the coefficient vectors of $g_{m,i,j}(xX, yY)$ for $j = 0 \ldots n-1$ and $i = n - j - 1$. The parameter m is a function of n and must be optimized.

For example, take $n = 4$ and $m = 2$. The lattice $L_p(n)$ is spanned by the row vectors of the following (4×4)-matrix

$$B_p(4) = \begin{bmatrix} N^2X^3 & & & \\ eNX^3 & -NX^2Y & & \\ e^2X^3 & -2eX^2Y & XY^2 & \\ e^3X^3 & -3e^2X^2Y & 3eXY^2 & -Y^3 \end{bmatrix}.$$

Note, that the lattice L_p of section 3 is equal to $L_p(2)$.

To apply Fact 2, we need a coefficient vector v with norm smaller than $\frac{p^m}{\sqrt{n}}$. The following Lemma gives us a condition for finding such a vector.

Lemma 7 *For every fixed $\epsilon > 0$, there are parameters n and N_0 such that for every $N \geq N_0$ the following holds: Let $X = \frac{n+1}{2}N^{\delta}$ and $Y = \frac{n+1}{2}N^{\beta+\delta}$ with*

$$3\beta - \beta^2 + 2\delta \leq 1 - \epsilon.$$

Then using the L^3-reduction algorithm, we can find a vector v in $L_p(n)$ with norm smaller than $\frac{p^m}{\sqrt{n}}$, where m is a function of n.

Proof: An easy computation shows that

$$\det(L_p(n)) = N^{\frac{m(m+1)}{2}}(XY)^{\frac{n(n-1)}{2}} = \big(\frac{n+1}{2}\big)^{n(n-1)} N^{\frac{m(m+1)}{2}+(2\delta+\beta)\frac{n(n-1)}{2}}$$

for $m < n$. By Fact 1, the L^3-algorithm will find a vector v in $L_p(n)$ with

$$\|v\| \leq 2^{\frac{n-1}{4}} \det(L_p(n))^{\frac{1}{n}}.$$

Using $p > N^{1-\beta}$, we must have

$$2^{\frac{n-1}{4}} \det(L_p(n))^{\frac{1}{n}} \leq \frac{N^{(1-\beta)m}}{\sqrt{n}}.$$

We plug in the value for $\det(L_p(n))$ and obtain the inequality

$$N^{\frac{m(m+1)}{2}+(2\delta+\beta)\frac{n(n-1)}{2}} \leq cN^{(1-\beta)mn},$$

where the factor $c = \left(\left(2^{-\frac{3}{4}}(n+1)\right)^{n-1}\sqrt{n}\right)^{-n}$ does not depend on N. Thus, c contributes to the error term ϵ and will be neglected in the following.

We obtain the condition

$$\frac{m(m+1)}{2} + (2\delta+\beta)\frac{n(n-1)}{2} - (1-\beta)mn \leq 0.$$

Using straightforward arithmetic to minimize the left hand side, one obtains that $m = (1-\beta)n$ is asymptotically optimal for $n \to \infty$. Again doing some calculations, we finally end up with the desired condition $3\beta - \beta^2 + 2\delta \leq 1$. $\square$

Now, we can use the above Lemma 7 in combination with Fact 2 to construct a bivariate polynomial $f(x,y)$ of degree n with at most n monomials and root (x_0, y_0). The problem is how to extract the root (x_0, y_0).

Analogous to Lemma 4, one can show for a vector $v = (c_1, c_2, \ldots, c_n) \cdot B_p(n)$ with norm smaller than $\frac{p^m}{\sqrt{n}}$ that k divides c_1 and d_p divides c_n. But we may not be able to find these factors k and d_p easily.

Therefore, we use another method to obtain the root. This is described in the following Lemma.

Lemma 8 *Let* $X = \frac{n+1}{2} N^{\delta}$ *and* $Y = \frac{n+1}{2} N^{\beta+\delta}$. *Let* $f_p(x,y) = ex - y$ *be a polynomial with root* (x_0, y_0) *modulo* p *that satisfies* $|x_0| \le N^{\delta}$, $|y_0| \le N^{\beta+\delta}$. *Let* v *be a vector in* $L_p(n)$ *with norm smaller than* $\frac{p^m}{\sqrt{n}}$, *where* v *is the coefficient vector of a polynomial* $f(xX, yY)$. *Then, the polynomial* $p(x,y) = y_0 x - x_0 y \in \mathbb{Z}[x,y]$ *must divide* $f(x,y)$. *We can find* p *by factoring* f *over* $\mathbb{Z}[x,y]$.

Proof: The point (x_0, y_0) is a root of f_p. For every integer a, the point (ax_0, ay_0) is also a root of f_p. Every root (ax_0, ay_0) with $|a| \le \frac{n+1}{2}$ satisfies the conditions $|ax_0| \le X$ and $|ay_0| \le Y$ of Fact 2. These are at least $n+1$ roots. According to Fact 2, f must contain these roots over $\mathbb{Z}$.

But these roots lie on the line $y = \frac{y_0}{x_0} x$ through the origin. Hence, they are also roots of the polynomial $p(x,y) = y_0 x - x_0 y \in \mathbb{Z}[x,y]$. Note, that p is an irreducible polynomial of degree 1 and f is a polynomial of degree n. Using the Theorem of Bézout (see [12], page 20), either p and f share at most n points or p must divide f. But we know $n+1$ common points of p and f. Thus, the polynomial p must divide f. Since p is irreducible, we can find an integer multiple $p' = (by_0)x - (bx_0)y$ of p by factoring f over $\mathbb{Z}[x,y]$. Note that $\gcd(x_0, y_0) = 1$ since by equation (2) we know that $\gcd(d_p, k+1)$ must divide kp, but $\gcd(d_p, kp) = \gcd(d_p, k) = 1$. Hence, we obtain p by computing $p = \frac{p'}{\gcd(by_0, bx_0)}$. □

Summarizing the results in this section, we obtain the following theorem.

Theorem 9 *Given an RSA public key tuple* (N, e) *with* $N = pq$ *and secret exponent* d. *Let* $q < N^{\beta}$, $\delta \le N^{\delta}$ *and*

$$3\beta - \beta^2 + 2\delta \le 1 - \epsilon,$$

where $\epsilon > 0$ *is arbitrary small for* N *suitably large. Then in deterministic time polynomial in* $\log(N)$, *we can find the factorization of* N.

Proof: Construct the lattice basis $B_p(n)$ according to Lemma 7 and find a short vector v with norm smaller than $\|v\| < \frac{p^m}{\sqrt{n}}$ using the L^3-reduction algorithm. Find the polynomial $p(x,y) = y_0 x - x_0 y$ using Lemma 8 which gives us $(x_0, y_0) = (d_p, k+1)$.

It is known that the factorization of the polynomial $f(x,y) \in \mathbb{Z}[x,y]$ in Lemma 8 can be done in deterministic time polynomial in $\log(N)$. Note that the coefficients of $f(x,y)$ must be of bit-size polynomial in $\log(p)$ since the coefficient vector of $f(xX, yY)$ has norm smaller than $\frac{p^m}{\sqrt{n}}$.

We may assume that we are in the case that k does not divide q in equation (2). Otherwise Theorem 6 proves the claim. Hence $f(x_0, y_0) = -kp$ and $\gcd(f(x_0, y_0), N) = p$ yields the factorization of N. □

In practice, the factorization of polynomials over $\mathbb{Z}[x,y]$ is very fast. Thus, our method is practical even for large n.

5 An approach modulo e

Throughout this section, we assume that e is of the same order of magnitude as N. The results in this section as well as the results in section 3 and 4 can be easily generalized to arbitrary exponents e.

Analogous to the works [3, 17] dealing with small secret exponent RSA, the smaller the exponent e is the better our methods work. On the other hand, one can completely counteract the attacks by adding to e a suitably large multiple of $\phi(N)$. We will give a detailed analysis of this in the full version of the paper.

Let us look again at equation (1) and rewrite it as

$$(k+1)(p-1) - p = -ed_p.$$

Multiplying with q yields

$$(k+1)(N-q) - N = -ed_p q$$

This gives as the polynomial

$$f_e(y,z) = y(N-z) - N$$

with a root $(y_0, z_0) = (k+1, q)$ modulo e.

Let us define the upper bounds $Y = N^{\beta+\delta}$ and $Z = N^{\beta}$. Note, that $|y_0| \leq Y$ and $|z_0| \leq Z$. Analogous to section 3, we can define a three-dimensional lattice L_e that is spanned by the row vectors of the (3×3)-matrix

$$B_e = \begin{bmatrix} e & & \\ & eY & \\ -N & NY & -YZ \end{bmatrix}.$$

Using a similar argumentation as in section 3, one can find a vector $v \in L_e$ with norm smaller than the bound $\frac{e}{\sqrt{3}}$ of Fact 2 provided that $3\beta + 2\delta \leq 1 - \epsilon$. Hence as before, this approach does not work if $\beta \geq \frac{1}{3}$ or $\delta \geq \frac{1}{2}$. In section 4, we used x-shifted polynomials to improve the bound for β. Now, z-shifted polynomials will help us to improve the bound for δ up to $\delta < 1$ and for β up to $\beta < \frac{3}{8}$.

Fix the parameter m. Let us define the y-shifted polynomials

$$g_{i,j}(y,z) = e^{m-i} y^j f_e^i(y,z)$$

and the z-shifted polynomials

$$h_{i,j}(y,z) = e^{m-i} z^j f_e^i(y,z).$$

All these polynomials have the common root (y_0, z_0) modulo e^m. Thus, every integer linear combination of these polynomials also has the root (y_0, z_0).

We build a lattice $L_e(m)$ that is defined by the span of the coefficient vectors of the y-shifted polynomials $g_{i,j}(yY, zZ)$ and $h_{i,j}(yY, zZ)$ for certain parameters i, j. We take the coefficient vectors of $g_{i,j}$ for all non-negative i, j with $i + j \leq m$

and the coefficient vectors $h_{i,j}$ for $i = 0 \dots m$ and $j = 1 \dots t$ for some t. The parameter t has to be optimized as a function of m.

For example, choose $m = 2$ and $t = 1$. We take the coefficient vectors of $g_{0,0}$, $g_{0,1}$, $g_{1,0}$, $g_{0,2}$, $g_{1,1}$, $g_{2,0}$ and the coefficient vectors of $h_{0,1}$, $h_{1,1}$, $h_{2,1}$ to build the lattice basis $B_e(2)$:

$$
\left[\begin{array}{ccccccccc}
e^2 & & & & & & & & \\
 & e^2Y & & & & & & & \\
-eN & eNY & -eYZ & & & & & & \\
 & & & e^2Y^2 & & & & & \\
 & -eNY & & eN^2Y^2 & -eY^2Z & & & & \\
N^2 & -2N^2Y & 2NYZ & N^2Y^2 & -2NY^2Z & Y^2Z^2 & & & \\
\hline
 & & & & & & e^2Z & & \\
 & & eNYZ & & & & -eNZ & -eYZ^2 & \\
 & & -2N^2YZ & & N^2Y^2Z & -2NY^2Z^2 & N^2Z & 2NYZ^2 & Y^2Z^3
\end{array}\right]
$$

The row vectors of $B_e(2)$ span the lattice $L_e(2)$.

In order to apply Fact 2, we need a vector in $L_e(m)$ with norm smaller than $\frac{e^m}{\sqrt{\dim L_e(m)}}$. The following lemma gives us a condition under which we can find such a vector.

Lemma 10 *For every constant $\epsilon > 0$ there exist m, N_0 such that for every $N \geq N_0$ the following holds: Let $Y = N^{\beta+\delta}$, $Z = N^\beta$ with*

$$\frac{2}{3}(\beta + \sqrt{3\beta + \beta^2}) + \delta \leq 1 - \epsilon,$$

where ϵ is arbitrary small for N suitably large. Then we can find a vector v in $L_e(m)$ with norm smaller than $\frac{e^m}{\sqrt{\dim L_e(m)}}$ using the L^3-algorithm.

Proof: A straightforward computation shows that

$$\det L_e(m) = (eY)^{\frac{1}{6}(2m^3+(6+3t)m^2+(4+3t)m)} Z^{\frac{1}{6}(m^3+(3+3t)m^2+(2+6t+3t^2)m+3t+3t^2)}.$$

Let $t = \tau m$ and $e = N^{1-o(1)}$. Using $Y = N^{\beta+\delta}$ and $Z = N^\beta$, we obtain

$$\det L_e(m) = N^{\frac{1}{6}m^3((\beta+\delta+1)(2+3\tau)+\beta(1+3\tau+3\tau^2)+o(1))}.$$

Analogous to the reasoning in Lemma 7, we obtain the condition

$$\det L_e(m) < cN^{(1-o(1))m \dim L_e(m)},$$

where c does not depend on N and contributes to the error term ϵ. An easy calculation shows that $\dim(L) = \frac{(m+1)(m+2)}{2} + t(m+1)$. We plug in the value for $\det L_e(m)$ and $\dim L_e(m)$. Neglecting all low order terms yields the condition

$$3\beta(\tau^2 + 2\tau + 1) + \delta(3\tau + 2) - 3\tau - 1 < 0$$

for $m \to \infty$. Using elementary calculus to minimize the left hand side, we obtain an optimal choice for the value $\tau = \frac{1-2\beta-\delta}{2\beta}$. Plugging this value in, we finally end up with the condition $\frac{2}{3}(\beta + \sqrt{3\beta + \beta^2}) + \delta \leq 1$. $\square$

Using Lemma 10, we can again apply Fact 2 and obtain a polynomial $f(y, z)$ with root (y_0, z_0) over $\mathbb{Z}$. But in contrast to the previous sections, we are not able to give a rigorous method to extract this root. Instead, we follow a heuristic approach due to Coppersmith [4]. Using the bounds of Fact 1 and a slightly different error term ϵ in Lemma 10, the L^3-algorithm must find a second vector that is short enough. This gives us another polynomial $g(y, z)$ with the same root (y_0, z_0) over $\mathbb{Z}$.

We take the resultant of f and g with respect to y. The resultant is a polynomial in z that can be solved by standard root finding algorithms. This gives as the unknown $z_0 = q$ and with it the factorization of N. Unfortunately, we cannot prove that the resultant is not the zero polynomial. It may happen that f and g share a non-trivial greatest common divisor. In this case, the resultant vanishes.

We carried out several experiments. If both y-shifted and z-shifted polynomials were used, we did not find any example where the resultant vanished. Thus although we cannot state the result as a theorem due to the gap in theory, the method works very well in practice. In fact, there are many results in cryptanalysis that rely on this heuristic, this includes among others [1, 3, 5, 8].

One can improve the shape of the curve for the approach modulo e slightly by using only a certain subset of the z-shifted polynomials. This approach leads to non-triangular lattice bases. We will analyze this in the full version of the paper.

We do not know if our lattice based approach yields the optimal bound. But there is a heuristic argument that gives us an upper bound for our method when using the polynomial $f_e(y, z)$.

Assume that the function $h(y, z) = y(N - z) \bmod e$ takes on random values in $\mathbb{Z}_e$ for $|y| \leq Y$ and $|z| \leq Z$. Every tuple (y, z) with $h(y, z) = N \bmod e$ is a root of f_e. The expected number of those tuples is $\Omega(\frac{YZ}{e}) = \Omega(N^{2\beta+\delta-1})$. As soon as $2\beta + \delta - 1$ is larger than some positive fixed constant, the number of small roots satisfying f_e is exponentially in $\log(N)$. All of these roots fulfill the criterion in Fact 2. But we require that $f(y, z)$ has a unique root over the integers in order to extract this root by resultant computation.
Thus heuristically, we cannot expect to obtain a bound better than $2\beta + \delta \leq 1$ using the polynomial f_e.

It is an open problem if one can really reach this bound.

6 Comparison of the methods

We compare the methods introduced in section 4 and section 5. In the figure below, we plotted the maximal δ as a function of β for which our two approaches succeed. The method modulo p is represented by the dotted line $\delta = \frac{1}{2} - \frac{3}{2}\beta + \frac{1}{2}\beta^2$ resulting from Theorem 9. The approach modulo e gives as the curve $\delta = 1 - \frac{2}{3}(\beta + \sqrt{3\beta + \beta^2})$ by Lemma 10. The points below the curves are the feasible region of parameter choices for our attacks. We see that our method modulo e yields much better results for small β.

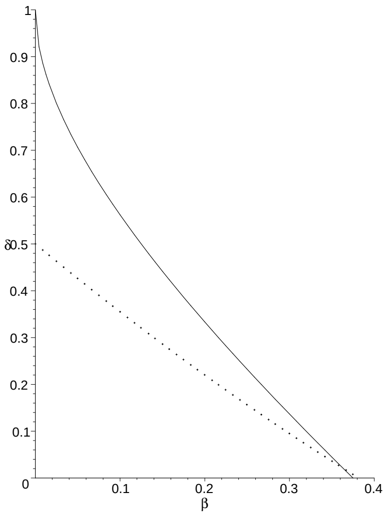

Fig. 1. Comparison of the two methods

One might we tempted to combine the two approaches and use the polynomials $ez \cdot f_p(x,y)$ and $N \cdot f_e(y,z)$ in a single lattice basis (i.e. working modulo eN). However, such a lattice will always contain an extremely short coefficient vector corresponding to the polynomial $f(x,y,z) = exz + y(N-z) - z$ over $\mathbb{Z}$. But this polynomial can be obtained by multiplying equation (1) with q and does not help us any further. It is an open problem if there is a successful way how to combine the methods.

7 Acknowledgement

I want to thank Johannes Blömer for many helpful discussions. I am also grateful to Damien Stehle who pointed out a fault in the determinant calculation of Lemma 10 in an earlier version of this paper.

References

1. D. Bleichenbacher, “On the Security of the KMOV public key cryptosystem”, Proc. of Crypto ’97

2. D. Boneh, “Twenty years of attacks on the RSA cryptosystem”, Notices of the AMS, 1999
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), 2000
4. D. Coppersmith, “Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities”, Journal of Cryptology 10(4), 1997
5. G. Durfee, P. Nguyen, “Cryptanalysis of the RSA Schemes with Short Secret Exponent from Asiacrypt ’99”, Proc. of Asiacrypt ’2000
6. M. Gruber, C.G. Lekkerkerker, “Geometry of Numbers”, North-Holland, 1987
7. N. Howgrave-Graham, “Finding small roots of univariate modular equations revisited”, Proc. of Cryptography and Coding, LNCS 1355, Springer-Verlag, 1997
8. C. Jutla, “On finding small solutions of modular multivariate polynomial equations”, Proc. of Eurocrypt ’98
9. A. Lenstra, H. Lenstra and L. Lovasz, “Factoring polynomials with rational coefficients”, Mathematische Annalen, 1982
10. N. Modadugu, D. Boneh, M. Kim, “Generating RSA Keys on a Handheld Using an Untrusted Server”, INDOCRYPT 2000, pp. 271-282, 2000
11. R. Rivest, A. Shamir and L. Adleman, “A method for obtaining digital signatures and public key cryptosystems”, Communications of the ACM, volume 21, 1978
12. I.R. Shafarevich, “Basic Algebraic Geometry”, Springer-Verlag, 1994
13. A. Shamir, “RSA for paranoids”, CryptoBytes vol. 1, no. 3, pp. 1-4, 1995
14. C.L. Siegel, “Lectures on the Geometry of Numbers”, Springer Verlag, 1989
15. H.-M. Sun, W.-C. Yang and C.-S. Laih, “On the design of RSA with short secret exponent”, Proc. of Asiacrypt ’99, LNCS vol. 1716, pp. 150-164, 1999
16. E. Verheul, H. van Tilborg, “Cryptanalysis of less short RSA secret exponents”, Applicable Algebra in Engineering, Communication and Computing, Springer Verlag, vol. 8, 1997
17. M. Wiener, “Cryptanalysis of short RSA secret exponents”, IEEE Transactions on Information Theory, vol. 36, 1990