New Attacks on RSA with Small Secret CRT-Exponents

Daniel Bleichenbacher[1] and Alexander May[2]

[1] daniel_bleichenbacher@yahoo.com
[2] Department of Computer Science,
TU Darmstadt,
64289 Darmstadt, Germany
may@informatik.tu-darmstadt.de

Abstract. It is well-known that there is an efficient method for decrypting/signing with RSA when the secret exponent d is small modulo $p-1$ and $q-1$. We call such an exponent d a small CRT-exponent. It is one of the major open problems in attacking RSA whether there exists a polynomial time attack for small CRT-exponents, i.e. a result that can be considered as an equivalent to the Wiener and Boneh-Durfee bound for small d. At Crypto 2002, May presented a partial solution in the case of an RSA modulus $N = pq$ with unbalanced prime factors p and q. Based on Coppersmith's method, he showed that there is a polynomial time attack provided that $q < N^{0.382}$. We will improve this bound to $q < N^{0.468}$. Thus, our result comes close to the desired normal RSA case with balanced prime factors. We also present a second result for balanced RSA primes in the case that the public exponent e is significantly smaller than N. More precisely, we show that there is a polynomial time attack if $d_p, d_q \leq \min\{(N/e)^{\frac{2}{5}}, N^{\frac{1}{4}}\}$. The method can be used to attack two fast RSA variants recently proposed by Galbraith, Heneghan, McKee, and by Sun, Wu.

Keywords: RSA, small exponents, lattices, Coppersmith's method.

1 Introduction

Let $N = pq$ be an RSA modulus. The public exponent e and the secret exponent d satisfy the equation $ed = 1 \bmod \phi(N)$, where $\phi(N) = (p-1)(q-1)$ is Euler's totient function. The main drawback of RSA is its efficiency. A normal RSA decryption/signature generation requires time $\Theta(\log d \log^2 N)$.

Therefore, one might be tempted to use small secret exponents to speed up the decryption/signing process. Unfortunately, Wiener[14] showed in 1991 that if $d < N^{\frac{1}{4}}$ then the factorization of N can be found in polynomial time using only the public information (N, e). In 1999, Boneh and Durfee[1] improved the bound to $d < N^{0.292}$. One can view these bounds as a benchmark for attacking RSA (see also the comments in the STORK-roadmap [11]). Thus, improving these bounds is a major research issue in public key cryptanalysis.

M. Yung et al. (Eds.): PKC 2006, LNCS 3958, pp. 1–13, 2006.

It remains an important open problem whether there is an analogue of these attacks in the case of small secret CRT-exponents d, i.e. exponents d such that $d_p = d \bmod p-1$ and $d_q = d \bmod q-1$ both are small. For the construction of such small CRT-exponents with a given bit-size, we refer to Boneh, Shacham [2]. Notice that small CRT-exponents enable to efficiently raise to the d^{th} power modulo p and modulo q, respectively. The results are then combined using the Chinese Remainder Theorem (CRT), yielding a solution modulo N. For the normal RSA case with balanced prime factors p, q and full-size e, the best algorithm that is currently known has time and space complexity $\mathcal{O}(\sqrt{\min\{d_p, d_q\}})$.

At Crypto 2002, May[9] presented two polynomial time attacks for the case of imbalanced prime factors p and q. His attacks are based on Coppersmith's method for finding small roots of modular equations. His first attack is rigorous and solves a polynomial equation modulo p. This attack works whenever $q < N^{0.382}$. May's second attack is a heuristic method that is based on a resultant heuristic for Coppersmith's method in the multivariate modular case. This attack works whenever $q < N^{\frac{3}{8}}$.

Let us have a look at the size of d_p that can be attacked by May's approaches as a function of the size of q. In Fig. 1 we present both of these sizes as a fraction of the bits of N.

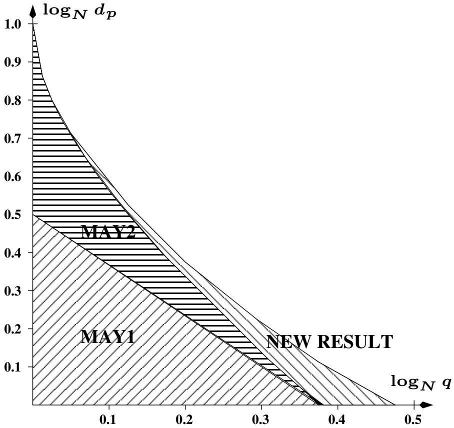

Fig. 1. The attacks of [9] in comparison with the new approach

A close look at the functions presented in Fig. 1 reveals that there is a tiny region where May's first method is better than his second one. Hence, it is a natural question to ask whether there is a unifying method that covers both regions of the key space.

In this work, we present a new attack that solves this question. In Fig. 1, we give the improved sizes of d_p that can be attacked by our new approach as a function of q. One can see that the new attack works up to $q < N^{0.468}$ and covers the key spaces of the previously known attacks. Thus, we are able to improve the benchmark for attacking CRT-RSA up to almost balanced prime factors.

Interestingly, we get the improvement by making just a small twist to May's second method. He solved a polynomial equation $f(x, y) = x(N - y) + N$ with a small root (x_0, q) modulo e. In this work, we make additional use of the fact that the desired small solution contains the prime factor q. Namely, we introduce a new variable z for the prime factor p and further use the equation $yz = N$.

Our new approach immediately raises an interesting open problem: The polynomial $f(x, y) = x(N - y) + N$ used here is very similar to the polynomial $g(x, y) = x(N + 1 - y) + 1$ that is used in the Boneh-Durfee approach to show the currently best bound of $d < N^{0.292}$ for attacking small secret exponent RSA. Notice that both polynomials $f(x, y)$ and $g(x, y)$ have the same set of monomials, i.e. the same Newton polytope. In contrast to $f(x, y)$, the polynomial $g(x, y)$ has a small root $(x'_0, p + q)$. It is a natural question to ask whether one can improve the Boneh-Durfee bound by using the fact that this root contains the sum of the prime factors p and q.

We should point out that our new attack works for small d_p and arbitrary sizes of d_q. It is an open problem how to make use of a small parameter d_q in this attack. Maybe a clever use of d_q could already help to push the bound from $q < N^{0.468}$ to the desired normal RSA-case of balanced prime factors.

As a second result, we are able to give a different lattice-based attack on RSA with small CRT-exponents that works in the case of balanced prime factors, but with the restriction that the parameter e is significantly smaller than N. This second attack makes use of small d_p and small d_q. The result is achieved by multiplying the equations $ed_p = 1 \bmod p-1$ and $ed_q = 1 \bmod q-1$ and then using a linearization technique. Our attack works whenever $d_p, d_q < \min\{\frac{1}{4}(N/e)^{\frac{2}{5}}, \frac{1}{3}N^{\frac{1}{4}}\}$, i.e., up to roughly half of the bit-size of p, q for sufficiently small e. The attack requires to find a shortest vector in a 3-dimensional lattice and is extremely fast. As an application of our second result, we show that recently proposed RSA variants by Galbraith, Heneghan and McKee [5] and Sun, Wu [12] are vulnerable to the new attack.

We would like to point out that both new attacks are heuristic methods. We implemented both methods and provide several experiments that show that the heuristics work well in practice.

The organization of the paper is as follows. In Section 2, we state some lattice basis theory and in Section 3 we review May's result. In Section 4, we show how to achieve the improved bound of $q < N^{0.468}$. In Section 5, we present our second attack for $d_p, d_q < \min\{\frac{1}{4}(N/e)^{\frac{2}{5}}, \frac{1}{3}N^{\frac{1}{4}}\}$ and show how this attack breaks recently proposed fast RSA variants. We conclude our work by providing some experimental results for our attacks in Section 6.

2 Lattice Theory and Definitions

Let $b_1, \ldots, b_n \in \mathbb{Z}_n$ be linearly independent. Then these vectors span a lattice of dimension n defined by

$$L := \Big\{ x \in \mathbb{Z}_n \mid x = \sum_{i=1}^{n} a_i b_i, \text{ where } a_i \in \mathbb{Z} \Big\}.$$

We call the set $B = \{b_1, \dots, b_n\}$ a basis of L. There are infinitely many bases. A basis can be transformed into another basis by a unimodular transformation, i.e. a multiplication by a matrix with determinant ±1. Therefore, the absolute value of the determinant of a basis matrix is an invariant of the lattice L. We call this invariant the determinant of L, which is denoted by $\det(L) = |\det(B)|$.

A famous theorem of Minkowski gives an upper bound for the length of a shortest vector v in a lattice in terms of a function of the determinant and the dimension n:

$$\|v\| \leq \sqrt{n}\,\dim(L)^n.$$

In lattices with fixed dimension, a shortest vector can be found in polynomial time. In arbitrary dimension, approximations of a shortest vector can be obtained in polynomial time by applying the well-known L^3 basis reduction algorithm of Lenstra, Lenstra and Lovász [8].

Theorem 1 (Lenstra, Lenstra, Lovász). *Let $B = \{b_1, \dots, b_n\}$ be a basis. On input B, the L^3-algorithm outputs another basis $\{v_1, \dots, v_n\}$ with*

$$\|v_1\| \leq \|v_2\| \leq 2^{\frac{n}{4}} \det(L)^{\frac{1}{n-1}},$$

in time polynomial in n and in the bit-size of the entries in B.

Let $f(x,y) = \sum_{i,j} a_{i,j} x^i y^j \in \mathbb{Z}[x,y]$. We define the norm of f by the Euclidean norm of its coefficient vector: $\|f\|^2 = \sum_{i,j} a_{i,j}^2$.

Based on the L^3-algorithm, Coppersmith [4] presented in 1996 a method that finds small solutions to modular polynomial equations. The idea behind Coppersmith's method is to construct a polynomial which has the desired small root over the integers. Howgrave-Graham [7] in turn formulated a useful condition how to find such a polynomial in terms of the norm of a polynomial.

Theorem 2 (Howgrave-Graham). *Let $f(x_1, \dots, x_k)$ be a polynomial in k variables with n monomials. Furthermore, let m be a positive integer. Suppose that*

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

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

3 Revisiting May's Attack on Small CRT-Exponents

Throughout this paper, we assume that $e < \phi(N)$. Furthermore, we assume that $q \leq N^\beta$ for some $\beta \leq \frac{1}{2}$. We start by writing the RSA equation $ed_p = 1 \bmod (p-1)$ in the form

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

for some unknown $k \in \mathbb{N}$. Rewriting terms yields

$$ed_p = (k-1)(p-1) + p. \tag{1}$$

A multiplication with q leaves us with the equation

$$ed_pq = (k-1)(N-q) + N.$$

We assign the variables x and y to the unknown parameters on the right-hand side and obtain a bivariate polynomial

$$f(x,y) = x(N-y) + N, \tag{2}$$

with the root $(x_0, y_0) = (k-1, q)$ modulo e. In order to bound the term $k-1$, we observe that by Eq. (1)

$$k - 1 = \frac{ed_p - p}{p-1} < \frac{e}{p-1}\, d_p < (q-1)X < N^{\beta}X.$$

Let us fix a parameter m. We define the following collection of polynomials that all have the root (x_0, y_0) modulo e^m:

$$\begin{aligned} g_{i,j}(x,y) &= e^{m-i}x^j f^i(x,y) \quad \text{for } i = 0,\ldots,m;\ j = 0,\ldots,m-i \quad \text{and} \\ h_{i,j}(x,y) &= e^{m-i}y^j f^i(x,y) \quad \text{for } i = 0,\ldots,m;\ j = 1,\ldots,t. \end{aligned} \tag{3}$$

The parameter t has to be optimized as a function of m.

Since each polynomial of the collection has the small root (x_0, y_0) modulo e, every linear combination of these polynomials also has the same root modulo e.

A lower triangular lattice basis can be build from the coefficient vectors of $g_{i,j}(xX, yY)$ and $h_{i,j}(xX, yY)$. According to Howgrave-Graham's theorem (Theorem 2), linear combinations of the vectors with sufficiently small norm give raise to bivariate polynomials that have the root (x_0, y_0) not only modulo e but over the integers. Having two polynomials $f_1(x,y)$ and $f_2(x,y)$ with this root over the integers, one can take resultants in order to extract the desired root. However, the last step is a heuristic, since the resultant computation may fail due to a non-trivial gcd of f_1 and f_2.

In [9], it was shown that with the optimal choice of parameters one obtains an attack that works up to $q < N^{\frac{3}{8}}$, see also Fig. 1 in Section 1.

4 An Approach That Works for $q < N^{0.468}$

Our improvement of the algorithm presented in Section 3 is based on the observation that in Eq. (2) the polynomial $f(x,y)$ contains in its small root $(x_0, y_0) = (d_p, q)$ modulo e the prime factor q. We will use the fact that we do not deal with just an arbitrary small root but that q is already determined by N.

Let us introduce a new variable z for p. We multiply the polynomial $f(x,y)$ by a power z^s for some s that has to be optimized. Additionally, we can replace every occurence of the monomial yz by N. Let us look at the following new collection of trivariate polynomials that we obtain by multiplying the former collection from (3) with z^s:

$$\begin{aligned} g'_{i,j}(x,y,z) &= e^{m-i}x^j z^s f^i(x,y) \quad \text{for } i = 0,\ldots,m;\ j = 0,\ldots,m-i \ \text{ and} \\ h'_{i,j}(x,y,z) &= e^{m-i}y^j z^s f^i(x,y) \quad \text{for } i = 0,\ldots,m;\ j = 1,\ldots,t. \end{aligned}$$

What is the impact of a multiplication with z^s, i.e. the changes from the collection g, h to the collection g', h'? Every monomial $x^i y^j$, $j \geq s$ with coefficient $a_{i,j}$ in the former collection is transformed into a monomial $x^i y^{j-s}$ with coefficient $a_{i,j} N^s$ in the new collection. In case of a monomial $x^i y^j$ with $j < s$, we obtain a new monomial $x^i z^{s-j}$ with new coefficient $a_{i,j} N^j$.

The obvious advantage is that the coefficient vectors of $g'(xX, yY, zZ)$ and $h'(xX, yY, zZ)$ contain less powers of Y, which decreases the determinant of the lattice spanned by these vectors. On the other hand, the coefficient vectors contain powers of Z, which in turn increases the determinant. Hence, there is a trade-off and one has to optimize the parameter s subject to a minimization of the lattice determinant.

As in Section 3, the resulting lattice basis built from the coefficient vectors of $g'(xX, yY, zZ)$ and $h'(xX, yY, zZ)$ is lower triangular. Therefore, every polynomial from our new collection contributes with just one coefficient to the diagonal. If the coefficient of this diagonal entry has a factor of N^j, we eliminate this factor by multiplying the polynomial with the inverse of N^j modulo e. I.e., we eliminate powers of N in the diagonal entries in order to keep the lattice determinant as small as possible.

Let B be the lattice basis defined by the coefficient vectors $g'(xX, yY, zZ)$ and $h'(xX, yY, zZ)$, where we eliminated powers of N on the diagonal as explained above. Moreover, let L be the lattice spanned by these vectors with dimension $\dim(L)$ and determinant $\det(L)$.

We have to find two vectors in L that are shorter than the bound $e^m/\sqrt{\dim(L)}$ given in Howgrave Graham's theorem (Theorem 2). These vectors are the coefficient vectors of two trivariate polynomial $f_1(xX, yY, zZ)$ and $f_2(xX, yY, zZ)$. By Howgrave-Graham's theorem, $f_1(x, y, z)$ and $f_2(x, y, z)$ have the root (x_0, q, p) over the integers. We will later show that the desired short vectors can be obtained by applying the L^3-algorithm to our lattice basis B.

Suppose for now that we have computed two such trivariate polynomials f_1 and f_2 with the previous property. Then we can eliminate z from the polynomials by setting $z = N/y$. Since the resulting bivariate polynomials are rational we multiply them by a suitable power of y in order to obtain polynomials $\bar{f}_1, \bar{f}_2$ in $\mathbb{Z}[x, y]$. Afterwards, we take the resultant of these integral polynomials $\bar{f}_1, \bar{f}_2$ with respect to the variable x. We obtain a univariate polynomial $g(y)$ with root q. If $\bar{f}_1$ and $\bar{f}_2$ do not share a non-trivial gcd, $g(y)$ is not the zero-polynomial and we can easily extract q with standard root finding algorithms. This completes the description of the attack. The only heuristic assumption that we make in our approach is that $g(y) \neq 0$.

Assumption 3. *The construction described above yields a non-zero polynomial* $g(y)$.

We are able to confirm Assumption 3 by various experiments in Section 6. This shows that our attack works very well in practice.

It remains to give a condition under which we can efficiently find two sufficiently short vectors in the lattice L spanned by the basis B. The following lemma gives an explicit condition, under which the L^3-algorithm finds two such vectors.

Lemma 4. *Let $\epsilon > 0$, $t = \tau m$ and $s = \sigma m$. Let N and m be sufficiently large and*

$$X^{2+3\tau} Y^{1+3(\tau-\sigma)(1+\tau-\sigma)} Z^{3\sigma^2} \leq e^{1+3\tau-\epsilon}.$$

Then on input B, the L^3-algorithm will output two vectors that are shorter than $\frac{e^m}{\sqrt{\dim(L)}}$.

Proof: Let $n = \dim(L)$. By the L^3 theorem (Theorem 1), the second shortest vector of an L^3-reduced basis satifies

$$\|v_2\| \leq 2^{\frac{n}{4}} \det(L)^{\frac{1}{n-1}}.$$

Suppose that we can upperbound the right-hand side term by $\frac{e^m}{\sqrt{n}}$, then the claim follows. That leaves us with the condition

$$\det(L) < c e^{m(n-1)}, \tag{4}$$

where $c = (2^{-\frac{n}{4}}/\sqrt{n})^{n-1}$. Since c does not depend on N, we let c contribute to the error term ϵ and omit it in the further calculations. Now we have to find an expression for the determinant of L.

It is not hard to see that the contribution of the coefficient vectors in $g'_{i,j}$ to the determinant contains powers of X, Y and Z that correspond to the monomials that appear in $z^s f^m(x,y)$. The coefficient vectors in h' contribute to $\det(L)$ with powers of X, Y and Z from the additional monomials that appear in $z^s y^i f^m(x,y)$, for $i = 1, \ldots, t$. A straight-forward but tedious computation (details are provided in Appendix A) yields that

$$\det(L) = \left((eX)^{2+3\tau} Y^{1+3(\tau-\sigma)(1+\tau-\sigma)} Z^{3\sigma^2} \right)^{\frac{1}{6}m^3(1+o(1))}.$$

Now, we have an expression for the left-hand side of our condition in (4). In order to find an expression for the right-hand side, we observe that $n = \dim(L) = (3+6\tau)\frac{1}{6}m^2(1+o(1))$ (for details of the calculation, see Appendix A). Neglecting low-order terms, we obtain the desired new condition

$$X^{2+3\tau} Y^{1+3(\tau-\sigma)(1+\tau-\sigma)} Z^{3\sigma^2} \leq e^{1+3\tau}. \qquad \square$$

We are now able to state our main theorem for our first attack.

Theorem 5. *Let $\epsilon > 0$. Under Assumption 3, the following holds for sufficiently large N: Let $N = pq$ be an RSA-modulus with $q \leq N^\beta$ and $p \leq 2N^{1-\beta}$. Moreover, let $e = N^\alpha$ be an RSA-public exponent satisfying $ed_p = 1 \bmod p-1$ for some $d_p = N^\delta$ with*

$$\delta \leq \frac{1}{3}\left(3 - 2\beta - \beta^2 - \sqrt{12\alpha\beta - 12\alpha\beta^2 + 4\beta^2 - 5\beta^3 + \beta^4}\right) - \epsilon.$$

Then N can be factored in polynomial time.

Proof: We can define the upper bounds $Y = N^{\beta}$ and $Z = 2N^{1-\beta}$ for q and p, respectively. Notice that the parameter β must not necessarily be known in advance. If β is unknown, we can brute-force search in polynomial time over the bit-size of q and obtain a suitable parameter β that satisfies our preconditions.

From Section 3, we know that the polynomial $f(x,y) = x(N-y)+N$ has the small root $(x_0, y_0) = (k-1, q)$ modulo e. Using Eq. (1), we obtain

$$x_0 = k - 1 \leq \frac{ed_p}{p-1} \leq \frac{N^{\alpha+\delta}}{N^{1-\beta}-1} \leq 2N^{\alpha+\beta+\delta-1}.$$

Let us define $X = 2N^{\alpha+\beta+\delta-1}$. Now we take the condition from Lemma 4 and plug in our bounds X, Y and Z. Neglecting low-order terms and the error term ϵ, we obtain the new condition

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

Our goal is to minimize the expression on the left-hand side. Therefore, we differentiate the term with respect to τ and σ. After some calculations, we observe that the expression is minimized for the parameter choices

$$\tau = \frac{(1-\beta)^2-\delta}{2\beta(1-\beta)} \quad \text{and} \quad \sigma = \frac{1-\beta-\delta}{2(1-\beta)}.$$

Plugging in these values, we obtain the desired condition

$$\delta \leq \frac{1}{3}\left(3 - 2\beta - \beta^2 - \sqrt{12\alpha\beta - 12\alpha\beta^2 + 4\beta^2 - 5\beta^3 + \beta^4}\right). \qquad \square$$

In Fig. 1 (see Section 1), we presented the function from Theorem 5 for the special case $\alpha = 1$, i.e. for the important case where the magnitude of e is of the order of the size of N. In this case, our attack works up to $\beta = \frac{1}{6}(\sqrt{61}-5) \approx 0.468$.

In [5] and [12], the authors suggested to combine medium size e with small CRT-exponents. In the balanced RSA-case, i.e. for $\beta = \frac{1}{2}$, our bound from Theorem 5 yields a polynomial time attack whenever $\alpha \leq \frac{7}{8}$. However, in the subsequent section we present a polynomial time attack on RSA with balanced prime factors whenever $\alpha < 1$.

5 An Attack for $d_p, d_q < \min\left\{\frac{1}{4}\left(\frac{N}{e}\right)^{\frac{2}{5}}, \frac{1}{3}N^{\frac{1}{4}}\right\}$

In this section we assume both that $d_p < \min\{\frac{1}{4}(N/e)^{2/5}, \frac{1}{3}N^{1/4}\}$ and $d_q < \min\{\frac{1}{4}(N/e)^{2/5}, \frac{1}{3}N^{1/4}\}$. We want to point out that we did not optimize the constant terms $\frac{1}{4}$, $\frac{1}{3}$ in the bounds for d_p, d_q in order to keep the calculations simple. We further assume that $e < \phi(N)$ and $1/2 < p/q < 2$, i.e. that p and

q have about the same size. We show *heuristically* that the modulus N can be factored under these assumptions.

We start with the RSA equations $ed_p = 1 \bmod p-1$ and $ed_q = 1 \bmod q-1$. We rewrite these equations as

$$\begin{aligned} ed_p &= 1 + k(p-1) \;\text{ and} \\ ed_q &= 1 + \ell(q-1), \end{aligned} \tag{5}$$

where k and ℓ are positive integers. Hence we get

$$\begin{aligned} ed_p + k - 1 &= kp \;\text{ and} \\ ed_q + \ell - 1 &= \ell q. \end{aligned}$$

Multiplying these two equations gives

$$(ed_p + k - 1)(ed_q + \ell - 1) = k\ell N.$$

Next we linearize this equation as

$$ex + y(1-N) + e^2 w = z,$$

with the unknowns

$$\begin{aligned} w &= d_p d_q, \\ x &= d_p(\ell - 1) + d_q(k-1), \\ y &= k\ell, \\ z &= k + \ell - 1. \end{aligned}$$

In the following, we show that the unknowns can be obtained heuristically by lattice reduction techniques. Using our bound $d_p, d_q \le \frac{1}{4} e^{-\frac{2}{5}} N^{\frac{2}{5}}$, we can upper-bound

$$k = \frac{ed_p - 1}{p-1} \le 2ed_p N^{-\frac{1}{2}} \le \frac{1}{2} e^{\frac{3}{5}} N^{-\frac{1}{10}}.$$

The same bound holds for ℓ. This enables us to give the following upper bounds for x, y and z:

$$\begin{aligned} x &\le \frac{1}{4} e^{\frac{1}{5}} N^{\frac{3}{10}}, \\ y &\le \frac{1}{4} e^{\frac{6}{5}} N^{-\frac{1}{5}}, \\ z &\le e^{\frac{3}{5}} N^{-\frac{1}{10}}. \end{aligned}$$

Let us look at the lattice L_1 that is spanned by the row vectors of the following lattice basis

$$B_1 = \begin{pmatrix} 1 & 0 & e \\ 0 & 1 & 1-N \\ 0 & 0 & e^2 \end{pmatrix}.$$

Notice that L_1 contains the target vector $v_1 = (x, y, w) \cdot B_1 = (x, y, z)$. We want to balance the target vector, i.e. to make every entry in v_1 approximately of the same size. Therefore, we multiply the columns of B_1 with suitable factors, such that the size of each entry of the resulting target vector is bounded by $e^{\frac{6}{5}}N^{\frac{3}{10}}$. This gives us the lattice L_2 defined by the span of the row vectors in the basis

$$B_2 = \begin{pmatrix} 4e & 0 & e^{\frac{8}{5}}N^{\frac{2}{5}} \\ 0 & 4N^{\frac{1}{2}} & e^{\frac{3}{5}}N^{\frac{2}{5}}(1-N) \\ 0 & 0 & e^{\frac{13}{5}}N^{\frac{2}{5}} \end{pmatrix}.$$

The new target vector $v_2 = (x, y, w) \cdot B_2$ has norm at most $\|v_2\| \le \sqrt{3}e^{\frac{6}{5}}N^{\frac{3}{10}}$. We want to argue that v_2 is among the shortest vectors in L_2. By Minkowski's theorem, L_2 contains a vector with norm smaller than

$$\sqrt{3}\det(L_2)^{\frac{1}{3}} = \sqrt{3}\left(4^2 e^{\frac{18}{5}}N^{\frac{9}{10}}\right)^{\frac{1}{3}} = 4^{\frac{2}{3}} \cdot \sqrt{3}e^{\frac{6}{5}}N^{\frac{3}{10}}.$$

We use the heuristic assumption that the vector v_2 is the shortest vector in L_2, i.e. v_2 is the only vector with norm below the Minkowski bound. Notice that L_2 also contains the vectors $(x - \lambda e, y, w + \lambda) \cdot B_2 = (x - \lambda e, y, z)$ with $\lambda \in \mathbb{Z}$. Thus $v_2 = (x, y, z)$ clearly is not the shortest vector in L_2 if $x > e/2$. However, this is not a problem because the condition $d_p, d_q < \frac{1}{3}N^{1/4}$ implies

$$x = d_p(\ell - 1) + d_q(k - 1) < \frac{1}{3}N^{\frac{1}{4}}(\ell + k) \le \frac{1}{3}N^{\frac{1}{4}} \cdot \frac{4}{3}eN^{-\frac{1}{4}} < \frac{e}{2}.$$

Under the heuristic assumption that there are no vectors shorter than v_2, we can recover v_2 by a shortest vector computation in L_2. We confirm our heuristic by experiments in Section 6.

Notice that v_2 gives us the unknowns w, x, y and z. From y and z, we can recover the unknowns k and ℓ. This enables us to recover from w and x the unknown parameters d_p and d_q. Finally, we obtain p and q by solving Eq (5). This completes the description of our second attack.

5.1 Applications

As applications of our attack, we present the cryptanalysis of two fast RSA-variants that were recently proposed by Galbraith, Heneghan, McKee [5] and Sun, Wu [12]. In [5], the following parameter choice is suggested: 1024-bit N, 508-bit e and 200-bit d. Similarly in [12], the suggested parameters are: 1024-bit N, 512-bit e and 199-bit d.

Both schemes are vulnerable to our new attack, i.e. the factorization of N can be obtained from the public parameters (N, e) in a fraction of a second. However, the construction in [5] allows to arbitrarily tune the RSA parameters within some constraints. Thus, the parameters can easily be adapted in such a way that our attack becomes infeasible. Indeed, after learning from our attack Galbraith, Heneghan and McKee [6] as well as Hinek, Sun and Wu [13] revised their constructions in such a way that the present attack does not work. On the other hand, we want to warn that a lack of an attack for a certain part of the RSA key space is not a guarantee of security!

6 Experiments

We implemented the attack described in Section 4 using Shoup's NTL [10]. We ran our experiments on a 2.4Ghz-Pentium under Linux. In each test, we used an 1000-bit RSA-modulus N with varying bit-size of q. The sizes of d_p and the lattice parameters are given in Fig. 2. We would like to point out that we could not find one example, where Assumption 3 failed. Thus, the resultant heuristic seems to work perfectly in practice.

q	d_p	Lattice parameters	L^3-time
405 bit	10 bit	$m = 3$, $t = s = 2$, $\dim(L) = 18$	5 sec
370 bit	50 bit	$m = 3$, $t = s = 2$, $\dim(L) = 18$	5 sec
330 bit	100 bit	$m = 3$, $t = s = 2$, $\dim(L) = 18$	5 sec
280 bit	160 bit	$m = 3$, $t = s = 2$, $\dim(L) = 18$	5 sec
420 bit	10 bit	$m = 4$, $t = s = 3$, $\dim(L) = 30$	50 sec
385 bit	50 bit	$m = 4$, $t = s = 3$, $\dim(L) = 30$	50 sec
340 bit	100 bit	$m = 4$, $t = s = 3$, $\dim(L) = 30$	50 sec
290 bit	160 bit	$m = 4$, $t = s = 3$, $\dim(L) = 30$	50 sec
430 bit	10 bit	$m = 5$, $t = s = 4$, $\dim(L) = 45$	6 min
395 bit	50 bit	$m = 5$, $t = s = 4$, $\dim(L) = 45$	6 min
345 bit	100 bit	$m = 5$, $t = s = 4$, $\dim(L) = 45$	7 min
300 bit	160 bit	$m = 5$, $t = s = 4$, $\dim(L) = 45$	9 min
440 bit	10 bit	$m = 6$, $t = s = 5$, $\dim(L) = 63$	35 min
405 bit	50 bit	$m = 6$, $t = s = 5$, $\dim(L) = 63$	35 min
355 bit	100 bit	$m = 6$, $t = s = 5$, $\dim(L) = 63$	44 min
305 bit	160 bit	$m = 6$, $t = s = 5$, $\dim(L) = 63$	53 min

Fig. 2. Experimental results for the attack from Section 4

An implementation of the attack in Section 5 using PARI/GP [3] needs approximately 15 ms on an 3Ghz-Pentium to find the factors of an 1024-bit RSA modulus. In a test we generated 1000 RSA moduli with 512-bit e, and $d_p, d_q < 2^{200}$. Our implementation was in all cases successful. The success rate however fell to about 90% when we generated the moduli such that $d_p, d_q < 2^{204}$.

Acknowledgements. We thank the anonymous reviewers of PKC 2006 for their very helpful comments.

References

1. D. Boneh, G. Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$", IEEE Trans. on Information Theory, Vol. 46(4), pp. 1339–1349, 2000
2. D. Boneh, H. Shacham, "Fast Variants of RSA", CryptoBytes Vol. 5, No. 1, pp. 1–9, 2002
3. H. Cohen et al. "PARI/GP", http://www.pari.math.u-bordeaux.fr

4. D. Coppersmith, “Small solutions to polynomial equations and low exponent vulnerabilities”, Journal of Cryptology, Vol. 10(4), pp. 223–260, 1997.
5. S. D. Galbraith, C. Heneghan, and J. F. McKee, “Tunable Balancing of RSA”, Proceedings of ACISP 2005, Lecture Notes in Computer Science Vol. 3574, pp. 280–292, 2005
6. S. D. Galbraith, C. Heneghan, and J. F. McKee, “Tunable Balancing of RSA”, full version of [5], online available at `http://www.isg.rhul.ac.uk/~sdg/full-tunable -rsa.pdf`
7. N. Howgrave-Graham, “Finding small roots of univariate modular equations revisited”, Proceedings of Cryptography and Coding, Lecture Notes in Computer Science Vol. 1355, Springer-Verlag, pp. 131–142, 1997
8. A. K. Lenstra, H. W. Lenstra, and L. Lovász, ”Factoring polynomials with rational coefficients,” Mathematische Annalen, Vol. 261, pp. 513–534, 1982
9. A. May, “Cryptanalysis of Unbalanced RSA with Small CRT-Exponent”, Advances in Cryptology – Crypto 2002, Lecture Notes in Computer Science Vol. 2442, Springer-Verlag, pp. 242–256, 2002
10. V. Shoup, NTL: A Library for doing Number Theory, online available at `http://www.shoup.net/ntl/index.html`
11. STORK, Strategic Roadmap for Crypto, `http://www.stork.eu.org/index.html`
12. H.-M. Sun, M.-E. Wu, “An Approach Towards Rebalanced RSA-CRT with Short Public Exponent”, Cryptology ePrint Archive: Report 2005/053, online available at `http://eprint.iacr.org/2005/053`
13. H.-M. Sun, M. J. Hinek, and M.-E. Wu, “An Approach Towards Rebalanced RSA-CRT with Short Public Exponent”, revised version of [12], online available at `http://www.cacr.math.uwaterloo.ca/techreports/2005/cacr2005-35.pdf`
14. M. Wiener, “Cryptanalysis of short RSA secret exponents”, IEEE Transactions on Information Theory, Vol. 36, pp. 553–558, 1990

A Details of the Calculations in Lemma 4

It remains to give the dimension and determinant calculation from the proof of Lemma 4. Therefore, we recall our collection of polynomials from Section 4:

$$\begin{aligned} g'_{i,j}(x,y,z) &= e^{m-i}x^j z^s f^i(x,y) \quad \text{for } i=0,\ldots,m;\ j=0,\ldots,m-i \text{ and} \\ h'_{i,j}(x,y,z) &= e^{m-i}y^j z^s f^i(x,y) \quad \text{for } i=0,\ldots,m;\ j=1,\ldots,t. \end{aligned} \tag{6}$$

The dimension of L is the number of polynomials in this collection:

$$\dim(L) = \sum_{i=0}^{m}\sum_{j=0}^{m-i} 1 = (3+6\tau)\cdot\frac{1}{6}m^3(1+o(1)).$$

We order the monomials in our collection such that the coefficient of the monomial which appears on the lattice basis diagonal corresponds to the monomial $x^i y^i$ in $f^i(x,y)$. I.e., the coefficient of the monomial from $g'_{i,j}(xX,yY,zZ)$ which contributes to the lattice determinant is the coefficient of $x^j z^s (xy)^i$, where we cancel out all terms yz using the relation $yz = N$. As explained in Section 4, we also eliminate all powers of N from the coefficient. Analogously, we proceed with the coefficient vectors of $h'_{i,j}(xX,yY,zZ)$.

Let us first calcute the contribution of the coefficient vectors of $g_{i,j}(xX, yY, zZ)$ to the determinant. We denote by e_g, X_g, Y_g and Z_g the contribution of all of the coefficient vectors of $g_{i,j}(xX, yY, zZ)$ to the exponents of e, X, Y, Z in the determinant of L, respectively.

From the description of our collection in (6), we derive

$$\begin{aligned}
e_g &= \sum_{i=0}^{m}\sum_{j=0}^{m-i} m - i = 2 \cdot \frac{1}{6}m^3(1 + o(1)), \\
X_g &= \sum_{i=0}^{m}\sum_{j=0}^{m-i} i + j = 2 \cdot \frac{1}{6}m^3(1 + o(1)), \\
Y_g &= \sum_{i=s}^{m}\sum_{j=0}^{m-i} i - s = (1 - \sigma)^3 \cdot \frac{1}{6}m^3(1 + o(1)), \\
Z_g &= \sum_{i=0}^{s}\sum_{j=0}^{m-i} s - i = (3\sigma^2 - \sigma^3) \cdot \frac{1}{6}m^3(1 + o(1)).
\end{aligned}$$

Similarly, we derive the contribution of the coefficient vectors of $h_{i,j}(xX, yY, zZ)$ to the determinant of L:

$$\begin{aligned}
e_h &= \sum_{i=0}^{m}\sum_{j=1}^{t} m - i = 3\tau \cdot \frac{1}{6}m^3(1 + o(1)), \\
X_h &= \sum_{i=0}^{m}\sum_{j=1}^{t} i = 3\tau \cdot \frac{1}{6}m^3(1 + o(1)), \\
Y_h &= \sum_{i=0}^{m}\sum_{j=\max\{1,s-i\}}^{t} j + i - s = (\sigma^3 + 3(\tau + \tau^2) - 6\sigma\tau) \cdot \frac{1}{6}m^3(1 + o(1)), \\
Z_h &= \sum_{i=0}^{s}\sum_{j=1}^{s-i} s - i - j = \sigma^3 \cdot \frac{1}{6}m^3(1 + o(1)).
\end{aligned}$$

Summarizing we obtain the determinant

$$\begin{aligned}
\det(L) &= e^{e_g+e_h} X^{X_g+X_h} Y^{Y_g+Y_h} Z^{Z_g+Z_h} \\
&= \left((eX)^{2+3\tau} Y^{1+3(\tau-\sigma)(1+\tau-\sigma)} Z^{3\sigma^2}\right)^{\frac{1}{6}m^3(1+o(1))}.
\end{aligned}$$