Extending Wiener's Attack in the Presence of Many Decrypting Exponents

Nicholas Howgrave-Graham[1] and Jean-Pierre Seifert[2]

[1] Mathematical Sciences Department, University of Bath,
Bath, BA2 7AY, U.K.
nahg@math.bath.ac.uk
[2] Department of Mathematics, Johann Wolfgang Goethe-University
Frankfurt am Main, Germany
seifert@mi.informatik.uni-frankfurt.de

Abstract. Wiener has shown that when the RSA protocol is used with a decrypting exponent, d, which is less than $N^{1/4}$ and an encrypting exponent, e, approximately the same size as N, then d can usually be found from the continued fraction approximation of e/N. We extend this attack to the case when there are many e_i for a given N, all with small d_i. For the case of two such e_i, the d_i can (heuristically) be as large as $N^{5/14}$ and still be efficiently recovered. As the number of encrypting exponents increases the bound on the d_i, which enables efficient recovery of the d_i, increases (slowly) to $N^{1-\epsilon}$. However, the complexity of our method is exponential in the number of exponents present, and therefore only practical for a relatively small number of them.

1 Introduction

In the RSA protocol (see [RSA]), Alice publishes a public modulus N and an encrypting exponent e. The modulus N should be the product of two large distinct primes p and q which are kept secret. To make the factoring of N hard, p and q are often chosen with about the same number of digits. With the knowledge of p and q Alice can also calculate d such that $ed = 1 \bmod \lambda(N)$ where $\lambda(N) = \mathrm{lcm}(p-1, q-1)$. Anyone wishing to encrypt a message m for Alice then raises it to the power e modulo N. This can then be decrypted (hopefully only by Alice) by another exponentiation since $(m^e)^d = m \pmod N$. Clearly if one can factor N then one can also decrypt any messages sent to Alice.

Despite twenty years of intensive research on the RSA cryptosystem no devastating attacks on it have been discovered so far. However, under certain circumstances more efficient attacks rather than simply factoring the modulus N are known (see Boneh [B] for a recent survey). One of those is the use of a small private exponent d and another one is the use of a common modulus N for several key pairs e_i, d_i. Let us elaborate these attacks a little bit further.

For efficient RSA signature generation it may be tempting to use a small private exponent d. Unfortunately, Wiener [W] has shown that when the RSA protocol is used with a decrypting exponent, d, less than $N^{1/4}$ and an encrypting

R. Baumgart (Ed.): CQRE'99, LNCS 1740, pp. 153–166, 1999.

exponent, e, approximately the same size as N, then the RSA system can be broken. Very recently Boneh and Durfee [BD] managed to improve Wiener's result by showing how to break the RSA system even when using decrypting exponents of size less than $N^{0.292}$. In order to simplify the RSA key management one may be tempted to use a single modulus for several key pairs e_i, d_i. However, as pointed out by Simmons [Si], whenever a message m is sent to two participants whose public exponents happen to be relatively prime, then the message m can be easily recovered without breaking the system. DeLaurentis [D] described two further attacks in which a participant can break such a common modulus cryptosystem. Particularly, he showed that knowledge of one key pair e_i, d_i gives rise to an efficient probabilistic algorithm for factoring the modulus N. Moreover, he also showed that knowledge of one key pair e_i, d_i gives rise to an efficient deterministic algorithm to generate other key pairs without determining $\lambda(N)$. For a thorough discussion of the common modulus situation when using RSA we refer to Moore [M].

However, we stress that Simmons attack does not break the RSA system at all and that the attack of DeLaurentis assumes that the attacker is also given the secret exponent. Having said all this, it seems to be natural to study the more realistic problem of what an opponent might do, given only several public exponents for a given modulus and the knowledge of the corresponding private exponents being quite small. This is the purpose of this paper. Although, as explained before, this situation is not common in present-day RSA systems, an analysis of this problem sheds some light on the gain of additional public information in attacking RSA and on the security of re-using the modulus N. Moreover, it seems a natural way to better understand and extend Wiener's original idea which might also be useful in other circumstances.

The question of how to combine several public exponents for a given modulus in order to reduce the size constraint on the private exponents for their efficient reconstruction was only very recently initiated by Guo [G]. Still based on the continued fraction approach of Wiener, Guo showed how to break RSA given 3 public exponenents even when their corresponding decrypting exponents are of size less than $N^{1/3}$. Using instead a lattice basis reduction approach we continue this study here, generalising (and improving) the result up to an arbitrary number of exponents. Particularly, we show that with n encrypting exponents e_i, our lattice basis approach allows for the d_i to be as large as N^{α_n} where

$$\alpha_n = \begin{cases} \frac{(2n+1)2^n-(2n+1)\binom{n}{n/2}}{(2n-2)2^n+(4n+2)\binom{n}{n/2}} & \text{if } n \text{ is even,} \\ \\ \frac{(2n+1)2^n-4n\binom{n-1}{(n-1)/2}}{(2n-2)2^n+8n\binom{n-1}{(n-1)/2}} & \text{if } n \text{ is odd.} \end{cases}$$

It is interesting to note that our method already allows for 2 encrypting exponents a decrypting exponent bound of $N^{5/14}$, which is superior to the $N^{1/3}$ bound of Guo for 3 encrypting exponents.

As our approach combines ideas from both Wiener and Guo into a single lattice the next section reviews the approaches of Wiener and Guo and gives

an overview of our extension approach. Our solution to the general problem of n encrypting exponents is given in section 3 starting with some preliminaries and examining first the cases of 2, 3 and 4 exponents before generalising the approach to n exponents. Section 4 then describes experimental results for our lattice basis.

2 Low Private Exponent Attacks on RSA

2.1 Wiener's Approach

It was shown in Wiener [W] that, if one assumes $\lambda(N)$ and e are both approximately as large as N, and if the decrypting exponent d is less than $N^{1/4}$ then the modulus N can be factored by examining the continued fraction approximation of e/N. This follows because e and d satisfy the relationship $ed - k\lambda(N) = 1$. So letting $\lambda(N) = (p-1)(q-1)/g$, and $s = 1 - p - q$ we have that

$$edg - kN = g + ks. \tag{1}$$

Dividing both sides by dgN gives

$$\frac{e}{N} - \frac{k}{dg} = \frac{g + ks}{dgN} = \left(\frac{k}{dg}\right)\left(\frac{s}{N}\right) + \frac{1}{dN}.$$

Now using the assumption that $e \simeq N$, and that $s \simeq N^{1/2}$ means (from examining equation 1) that $k/(dg) \simeq 1$ so that the right-hand side of the above equation is approximately $N^{-1/2}$. It is well known (see for instance [HW]) that if

$$|x - a/b| < 1/(2b^2)$$

then a/b is a continued fraction approximant of x. Thus if $N^{-1/2} < 1/(2(dg)^2)$ then $k/(dg)$ will be a continued fraction approximant of e/N. This is true whenever

$$d < 2^{-1/2}(1/g)N^{1/4}, \tag{2}$$

and g will be small under the assumption that $\lambda(N) \simeq N$ (though clearly $g \geq 2$ since both p and q are odd). Given dg one may calculate

$$r = (p-1)(q-1) = \frac{edg}{k} - \frac{g}{k} = \lfloor edg/k \rceil \qquad \text{(since } g \text{ is small)},$$

and then we can factor N since the factors p and q satisfy the quadratic relationship $x^2 - (N + 1 - r)x + N = 0$.

2.2 Guo's Approach

The approach taken in Guo [G] assumes that one has more than one e_i for a given N, and that each of these e_i has a relatively small d_i. Guo only considers the problem for 2 and 3 encryption exponents. For 2 exponents we have the following relations:

$$e_1 d_1 g - k_1(p-1)(q-1) = g$$
$$e_2 d_2 g - k_2(p-1)(q-1) = g,$$

so multiplying the first by k_2, the second by k_1, and subtracting gives

$$k_2 d_1 e_1 - k_1 d_2 e_2 = k_2 - k_1. \tag{3}$$

Dividing now both sides of equation 3 by $k_2 d_1 e_2$ implies the following

$$\frac{e_1}{e_2} - \frac{k_1 d_2}{k_2 d_1} = \frac{k_2 - k_1}{k_2 d_1 e_2},$$

and assuming that the d_i (and hence k_i if the e_i are large) are at most N^α means that the right-hand side is about $N^{-(1+\alpha)}$.

For the fraction $k_1 d_2/(k_2 d_1)$ to be a continued fraction approximant of e_1/e_2, we must therefore have that

$$2(k_2 d_1)^2 < N^{1+\alpha},$$

and with the assumptions that k_2 and d_1 are at most N^α and that g is small this condition will be true whenever $\alpha = 1/3 - \epsilon$ for some $\epsilon > 0$.

However, unlike the situation with Wiener's attack, the fraction $k_1 d_2/(k_2 d_1)$ does not break the RSA cryptosystem for two reasons:

- Firstly knowing, say, the numerator $k_1 d_2$, does not allow us to find d_2 or k_1 without factoring this number.
- Secondly there may be a factor in common between $d_1 k_2$ and $d_2 k_1$ in which case the continued fraction method would not give a fraction with numerator $k_1 d_2$ and denominator $k_2 d_1$, but rather the fraction with the common factor removed.

Guo assumes that the second problem does not exist, i.e. that we have $\gcd(k_1 d_2, k_2 d_1) = 1$, and it is estimated that this happens with probability $6/\pi^2 \simeq 0.61$.

To get around the first problem, Guo suggests that one could either try to factor $k_1 d_2$ (a number of size about $N^{2/3}$ and not typically of a hard factorisation shape), or alternatively assume that one has another encrypting exponent e_3 with $d_3 < N^{1/3}$. Then (repeating the above procedure with e_3 and e_2) one can also find $k_3 d_2$, and calculating $\gcd(k_1 d_2, k_3 d_2)$ will hopefully (if $\gcd(k_1, k_3) = 1$) give d_2 and thus allow the factoring of N. The probability of this attack working under the given assumptions is $(6/\pi^2)^3 \simeq 0.23$.

2.3 Overview of our Extension Approach

As already said in the introduction, our approach also assumes that we have more than one e_i for a given N, and that each of these e_i has a relatively small d_i.

In the remainder we will use, among others, ideas from both Wiener and Guo to solve the general problem of breaking RSA in the presence of n encrypting exponents e_i, all with relatively small $d_i < N^{\alpha_n}$, $i = 1, \ldots, n$. The main technique used in deriving these results is the creation and subsequent reduction of certain lattices. The approach taken by us, however, can currently only be classed as a heuristic method because, although the vectors we search for can be shown to be relatively short, we cannot prove yet that they are indeed among the shortest vectors (and hence bound to be found by lattice basis reduction algorithms). Nevertheless, in section 4 it is shown that our approach performs well in practice, and that the following theoretically derived bounds are frequently achieved. In particular, in the presence of n encrypting exponents e_i, our approach allows for the d_i to be as large as N^{α_n} where

$$\alpha_n = \begin{cases} \frac{(2n+1)2^n - (2n+1)\binom{n}{n/2}}{(2n-2)2^n + (4n+2)\binom{n}{n/2}} & \text{if } n \text{ is even,} \\ \\ \frac{(2n+1)2^n - 4n\binom{n-1}{(n-1)/2}}{(2n-2)2^n + 8n\binom{n-1}{(n-1)/2}} & \text{if } n \text{ is odd.} \end{cases}$$

The first few (from $n = 1$) start 1/4, 5/14, 2/5, 15/34, 29/62. In section 3.5 it is shown that $\alpha_n \to 1$ as $n \to \infty$.

If the LLL algorithm (see [LLL]) is used in order to reduce the lattices underlying our approach, and the (pessimistic) estimate for its complexity of $O(m^6 \log^3 B)$ is assumed (given a lattice of dimension m with largest norm B), then the complexity of our method is $O(2^{6n} n^3 \log^3 N)$, and so clearly the attack is only practical for small n.

3 An Extension in the Presence of Many Small Decryption Exponents

3.1 Preliminaries

In extending the analysis to n encrypting exponents e_i (with small decrypting exponents d_i), we use both Wiener's and Guo's ideas. We shall refer to relations of the form

$$d_i g e_i - k_i N = g + k_i s$$

as Wiener equations, and we shall denote them W_i (see equation 1 for an example). Similarly we shall refer to relations of the form

$$k_i d_j e_j - k_j d_i e_i = k_i - k_j$$

as Guo equations, and shall denote them $G_{i,j}$ (see equation 3 for an example). We shall also assume, for a given n, that the d_i and k_i are at most N^{α_n}, that g is small, and that s is around $N^{1/2}$. Notice that the right-hand sides of W_i and $G_{i,j}$ are therefore quite small; in fact at most $N^{(1/2)+\alpha_n}$, and N^{α_n} respectively. Finally we often refer to composite relations, e.g. $W_u G_{v,w}$, in which case we mean the relation, whose left-hand (resp. right-hand) side is the product of the left-hand (resp. right-hand) sides of W_u and $G_{v,w}$. For example, $W_u G_{v,w}$ which has a relatively small right-hand side, bounded in size by $N^{(1/2)+2\alpha_n}$.

In the following analysis we examine the cases of 2, 3 and 4 exponents before generalising the approach to n exponents. This is done both to give explicit examples of the approach when in the presence of a small number of exponents, and also because it is not until the presence of 4 exponents that the general phenomenon becomes clear. The relations that we choose for the cases of 2, 3 and 4 exponents may seem "plucked from the air", but the pattern is made clear in section 3.5.

3.2 RSA in the Presence of 2 Small Decryption Exponents

Assuming that we have two small decryption exponents, then the following relations hold: $W_1, G_{1,2}, W_1 W_2$; or more explicitly:

$$\begin{aligned} d_1 g e_1 - k_1 N &= g + k_1 s, \\ k_1 d_2 e_2 - k_2 d_1 e_1 &= k_1 - k_2, \\ d_1 d_2 g^2 e_1 e_2 - d_1 g k_2 e_1 N - d_2 g k_1 e_2 N + k_1 k_2 N^2 &= (g + k_1 s)(g + k_2 s). \end{aligned}$$

Multiplying the first of these by k_2 means that the left-hand sides are all in terms of $d_1 d_2 g^2$, $d_1 g k_2$, $d_2 g k_1$, and $k_1 k_2$, and hence we may write these equations in the matrix form below.

$$(k_1 k_2, d_1 g k_2, d_2 g k_1, d_1 d_2 g^2) \qquad \begin{pmatrix} 1 & -N & 0 & N^2 \\ & e_1 & -e_1 & -e_1 N \\ & & e_2 & -e_2 N \\ & & & e_1 e_2 \end{pmatrix} =$$

$$(k_1 k_2, k_2(g + k_1 s), g(k_1 - k_2), (g + k_1 s)(g + k_2 s).$$

The size of the entries of the vector on the right-hand side are at most $N^{2\alpha_2}$, $N^{(1/2)+2\alpha_2}$, N^{α_2}, and $N^{1+2\alpha_2}$ respectively. These size estimates may be made roughly equivalent by multiplying the first three columns of the matrix by N, $M_1 = N^{1/2}$, and $M_2 = N^{1+\alpha_2}$ respectively, which gives the following matrix:

$$L_2 = \begin{pmatrix} N & -M_1 N & 0 & N^2 \\ & M_1 e_1 & -M_2 e_1 & -e_1 N \\ & & M_2 e_2 & -e_2 N \\ & & & e_1 e_2 \end{pmatrix}$$

In this case the vector $b = (k_1 k_2, d_1 g k_2, d_2 g k_1, d_1 d_2 g^2)$ will be such that

$$\|bL_2\| < 2N^{1+2\alpha_2}.$$

We must now make the assumption that, in the lattice generated by the rows of L_2, the shortest vector has length $\Delta^{1/4-\epsilon}$, where $\Delta := \det(L_2) \simeq N^{(13/2)+\alpha_2}$, and moreover that the next shortest linearly independent vector has a significantly larger norm than the shortest vector in L_2. Indeed, if the lattice L_2 is pretty "random", there are almost surely no lattice points of L_2 significantly shorter than the Minkowski bound $2\Delta^{1/4}$. Under these assumptions, then bL_2 is the shortest vector in the lattice if

$$N^{1+2\alpha_2} < (1/c_2)\left(N^{(13/2)+\alpha_2}\right)^{1/4}$$

for some small c_2, which is true if

$$\alpha_2 < 5/14 - \epsilon'.$$

This implies that the vector $b = (b_1, b_2, b_3, b_4)$ can be found via lattice basis reduction algorithms (e.g. LLL) if $\alpha_2 < 5/14 - \epsilon'$, and then $d_1 g/k_1 = b_2/b_1$ can be calculated, which leads to the factoring of N as shown in section 2.1.

3.3 RSA in the Presence of 3 Small Decryption Exponents

This method extends easily to 3 encrypting exponents. We now have the quantities $1, e_1, e_2, e_1e_2, e_3, e_1e_3$ and $e_1e_2e_3$ from which to form linear relationships, and we already have relationships concerning the first four of these from the 2 exponent case, namely $1, W_1, G_{1,2}$ and W_1W_2. For the remaining relationships we choose $G_{1,3}$, $W_1G_{2,3}$, $W_2G_{1,3}$ and $W_1W_2W_3$. These relations imply looking for the vector

$$b = (k_1k_2k_3, d_1gk_2k_3, k_1d_2gk_3, d_1d_2g^2k_3, \\ k_1k_2d_3g, k_1d_3g, k_2d_3g, d_1d_2d_3g^3),$$

by reducing the rows of the following lattice:

$$L_3 = \begin{pmatrix} 1 & -N & 0 & N^2 & 0 & 0 & 0 & -N^3 \\ & e_1 & -e_1 & -e_1N & -e_1 & 0 & e_1N & e_1N^2 \\ & & e_2 & -e_2N & 0 & e_2N & 0 & e_2N^2 \\ & & & e_1e_2 & 0 & -e_1e_2 & -e_1e_2 & -e_1e_2N \\ & & & & e_3 & -e_3N & -e_3N & e_3N^2 \\ & & & & & e_1e_3 & 0 & -e_1e_3N \\ & & & & & & e_2e_3 & -e_2e_3N \\ & & & & & & & e_1e_2e_3 \end{pmatrix} \times D,$$

where D is the diagonal matrix

$$\mathrm{diag}(N^{3/2}, N, N^{(3/2)+\alpha_3}, N^{1/2}, N^{(3/2)+\alpha_3}, N^{1+\alpha_3}, N^{1+\alpha_3}, 1)$$

used to maximise the determinant of L_3 and still keep

$$\|bL_3\| < \sqrt{8}N^{(3/2)+3\alpha_3}.$$

Again, using the assumptions that the shortest vector in the lattice generated by the rows of L_3 has length $\det(L_3)^{(1/8)-\epsilon}$, and is also significantly shorter than the next shortest linearly independent vector in L_3, means that bL_3 will be the shortest vector in the lattice L_3 if

$$N^{(3/2)+3\alpha_3} < (1/c_3)\left(N^{20+4\alpha_3}\right)^{1/8}$$

for some small c_3 which is true if

$$\alpha_3 < 2/5 - \epsilon'.$$

By using again the first two components of b, as in the 2 exponent case, one may now factor the modulus N as shown in section 2.1.

3.4 RSA in the Presence of 4 Small Decryption Exponents

In the presence of 4 exponents we can now use linear relationships among the quantities $1, e_1, e_2, e_1e_2, e_3, e_1e_3, e_2e_3, e_1e_2e_3, e_4, e_1e_4, e_2e_4, e_3e_4, e_1e_2e_4, e_1e_3e_4, e_2e_3e_4$ and $e_1e_2e_3e_4$. As before we already have linear relationships for the first half of these quantities from the analysis in the presence of 3 equations. For the remaining quantities we use the relations $G_{1,4}$, $W_1G_{2,4}$, $G_{1,2}G_{3,4}$, $G_{1,3}G_{2,4}$, $W_1W_2G_{3,4}$, $W_1W_3G_{2,4}$, $W_2W_3G_{1,4}$ and $W_1W_2W_3W_4$. Putting these relations in matrix form, and multiplying the columns by appropriate factors to make all the relations of size at most $N^{2+4\alpha_4}$, results in a 16×16 matrix, L_4, which has determinant $N^{(109/2)+13\alpha_4}$. The vector b we are now looking for is

$$\begin{aligned} b = (&k_1k_2k_3k_4, d_1gk_2k_3k_4, k_1d_2gk_3k_4, d_1d_2g^2k_3k_4,\\ &k_1k_2d_3gk_4, d_1k_2d_3g^2k_4, k_1d_2d_3g^2k_4, d_1d_2d_3g^3k_4,\\ &k_1k_2k_3d_4g, d_1k_2k_3d_4g^2, k_1d_2k_3d_4g^2, k_1k_2d_3d_4g^2,\\ &d_1d_2k_3d_4g^3, d_1k_2d_3d_4g^3, k_1d_2d_3d_4g^3, d_1d_2d_3d_4g^4). \end{aligned}$$

Therefore, again making the same assumptions as before, implies that the vector bL_4 is the shortest vector in the lattice generated by the rows of L_4 if

$$N^{2+4\alpha_4} < (1/c_4)\left(N^{(109/2)+13\alpha_4}\right)^{1/16}$$

for some small c_4, and this is true if

$$\alpha_4 < 15/34 - \epsilon'.$$

Using again the first two components of b, as in the 2 and 3 exponent case, one may again factor the modulus N as shown in section 2.1.

3.5 The General Approach

Due to space limitations we defer the subtle computation of the general allowable bound on the d_i when we have n encrypting exponents e_i, $i = 1, \ldots, n$, to the appendix and show below simply the graph for

$$\alpha_n = \begin{cases} \dfrac{(2n+1)2^n - (2n+1)\binom{n}{n/2}}{(2n-2)2^n + (4n+2)\binom{n}{n/2}} & \text{if } n \text{ is even,} \\ \dfrac{(2n+1)2^n - 4n\binom{n-1}{(n-1)/2}}{(2n-2)2^n + 8n\binom{n-1}{(n-1)/2}} & \text{if } n \text{ is odd.} \end{cases}$$

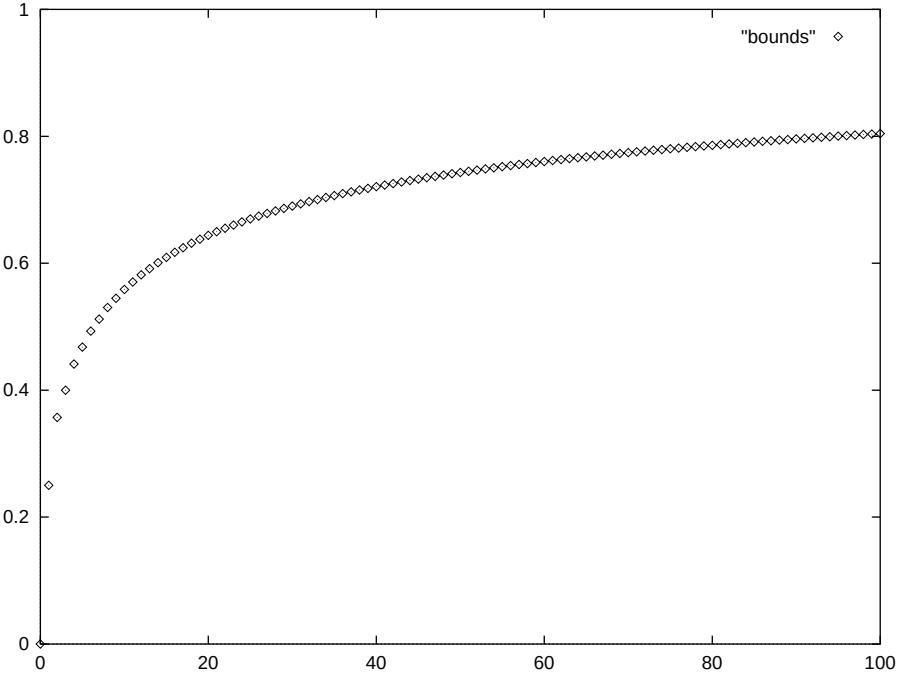

Fig. 1. Graph of bounds α_n for $n \leq 100$.

4 Practical Results

Although our method is at the current time only heuristic, it works well in practice as can be seen from our experimental results below.

Our implementation uses the NTL library [Sh] of Victor Shoup. Timings are given for a 300 MHz AMD K6 running under Linux.

RSA-500 with 2 public exponents			
α_2	bit length of d_i	avg. time in secs.	success rate
0.356	178	0.441	40%
0.354	177	0.421	100%

Fig. 2. Average running time (in seconds) and success rate for 10 random experiments as a function of α_2.

RSA-700 with 2 public exponents			
α_2	bit length of d_i	avg. time in secs.	success rate
0.357143	250	1.075	0%
0.355714	249	1.117	70%
0.354286	248	0.93	80%
0.352857	247	1.33	100%

Fig. 3. Average running time (in seconds) and number of success rate for 10 random experiments as a function of α_2.

RSA-500 with 3 public exponents			
α_3	bit length of d_i	avg. time in secs.	success rate
0.4	200	3.632	0%
0.398	199	3.567	40%
0.396	198	3.599	90%
0.394	197	3.726	90%
0.392	196	3.595	90%
0.39	195	3.529	100%

Fig. 4. Average running time (in seconds) and success rate for 10 random experiments as a function of α_3.

RSA-200 with 4 public exponents			
α_4	bit length of d_i	avg. time in secs.	success rate
0.44	88	14.538	0%
0.435	87	14.496	50%
0.43	86	14.328	80%
0.425	85	14.159	100%

Fig. 5. Average running time (in seconds) and success rate for 10 random experiments as a function of α_4.

RSA-200 with 5 public exponents			
α_5	bit length of d_i	avg. time in secs.	success rate
0.45	90	424.756	0%
0.445	89	427.275	60%
0.44	88	422.74	100%

Fig. 6. Average running time (in seconds) and success rate for 10 random experiments as a function of α_5.

5 Open Problems

The major open problem raised by our work is the following. To work out the manageable bound on α_n for the secret exponents we had to make two heuristic assumptions concerning "random" lattices. As the experimental results strongly support the derived bounds it is natural to ask whether our attack can be turned into a rigorous theorem?

References

B. D. Boneh, "Twenty years of attacks on RSA", *Notices of the AMS* Vol. 46, pp. 203-213, 1999.

BD. D. Boneh, G. Durfee, "New results on the cryptanalysis of low exponent RSA", to appear in *Proc. of EUROCRYPT '99*.

D. J. M. DeLaurentis, "A further weakness in the common modulus protocol for the RSA cryptoalgorithm", *Cryptologia* Vol. 8, pp. 253-259, 1984.

G. C. R. Guo, "An application of diophantine approximation in computer security", to appear in *Mathematics of Computation*.

HW. G. H. Hardy, E. M. Wright, *An introduction to the theory of numbers*, 5^{th} edn., Oxford University Press, 1979.

LLL. A. K. Lenstra, H. W. Lenstra, L. Lovasz, "Factoring polynomials with integer coefficients", *Mathematische Annalen* Vol. 261, pp. 513-534, 1982.

M. J. H. Moore, "Protocol failures in cryptosystems", in G. J. Simmons (ed.), *Contemporary Cryptology*, IEEE Press, 1992.

RSA. R. L. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Commun. ACM* Vol. 21, pp. 120-126, 1978.

Sh. V. Shoup, "Number Theory Library (NTL)", `http://www.cs.wisc.edu/~shoup.ntl`.

Si. G. J. Simmons, "A 'weak' privacy protocol using the RSA cryptalgorithm", *Cryptologia* Vol. 7, pp. 180-182, 1983.

VvT. E. R. Verheul, H. C. A. van Tilborg, "Cryptanalysis of 'Less Short' RSA secret exponents", *Applicable Algebra in Engeneering, Communication and Computing* Vol. 8, pp. 425-435, 1997.

W. M. Wiener, "Cryptanalysis of short RSA exponents", *IEEE Trans. on Information Theory* Vol. 36, pp. 553-558, 1990.

Appendix

We now work out the general bound on the d_i when we have n encrypting exponents. The reader is encouraged to refer back to the previous sections (when $n = 2, 3$ and 4) as examples.

Given that there are n exponents e_i, then there are 2^n different quantities, h_j, involving the e_i's, and the product of all of these (assuming $e \simeq N$) is $N^{(n2^{n-1})}$. This means that one considers a diagonal matrix, L_n, of dimension 2^n, and that the determinant of this matrix, before multiplying the rows to increase the allowable bound, is $N^{(n2^{n-1})}$.

The last relation $W_1 W_2 \dots W_n$ has a right-hand side of at most $N^{(n/2)+n\alpha_n}$, and thus we increase the right-hand side of all the other relations up to this bound, making the desired vector b such that $\|bL_n\|_\infty$ is (still) approximately $N^{(n/2)+n\alpha_n}$. The general form of the desired vector b is that its j^{th} entry is the product of n unknown quantitities a_i for $i = 1 \dots n$, where a_i is either $d_i g$ or k_i depending on whether e_i is present in the j^{th} quantity h_j or not.

We now consider the interesting problem of which relations to consider for n equations. Observe that a general relation of the form

$$R_{u,v} = W_{i_1} \dots W_{i_u} G_{j_1,l_1} \dots G_{j_v,l_v},$$

(where the $i_1, \dots, i_u, j_1, \dots, j_v, l_1, \dots, l_v$ are unique), has a left-hand side composed of products of $(u + 2v)$ of the e_i's with coefficients that are products of $(u + v)$ of the unknown quantities a_i (where a_i is again either $d_i g$ or k_i). Also notice that the right-hand side of $R_{u,v}$ has size at most $N^{(u/2)+(u+v)\alpha_n}$.

Our method requires all the coefficients to be roughly the same size (a product of n of the quantities a_i). This means that relations which have coefficients less than this must be multiplied (on both sides) by some missing k_i. For example, in the the 2 exponent case we multiplied the first equation by k_2 to make all the coefficients of size $N^{2\alpha_2}$. This has the effect of increasing the right-hand side of relation $R_{u,v}$ to a size bounded by $N^{(u/2)+(n-v)\alpha_n}$.

Given this new relation $R_{u,v}$ we now need to make it's right-hand side as large as the right-hand side of $W_1 W_2 \dots W_n$, which means multiplying (both sides) by $N^{(n-u)/2+v\alpha_n}$. For example, these multiplication factors are the (diagonal) entries of the diagonal matrix D in the example when $n = 3$.

Say that the product of these multiplication factors (i.e. the determinant of D in the $n = 3$ example) is N^{β_n}, where $\beta_n = x + y\alpha_n$, and let L_n denoted the lattice of (modified) relations as before. This means that (under the usual assumptions) the vector bL_n is the shortest vector of the lattice if

$$N^{n/2+n\alpha_n} < (1/c_n) \left(N^{n2^{n-1}+x+y\alpha_n} \right)^{1/2^n}$$

for some small c_n, i.e. when

$$\alpha_n < \frac{x}{n2^n - y} - \epsilon'. \tag{4}$$

In order to maximise α_n we wish both x and y to be large. This means that the relations should be chosen to maximise v (and minimise u). For instance when $n = 2$ we choose the relations $W_1, G_{1,2}$ and $W_1 W_2$ rather than W_1, W_2 and $W_1 W_2$ because $\beta_2 = 2$ in the latter case rather than $5/2 + \alpha_2$ in the former.

With this general principle in mind we still need to explain exactly which relations we use. In order to mantain the triangularity of L_n we only consider relations which introduce one new quantity h_j. The choices for $n \leq 5$ can be seen in the below figure.

h_j	relation	size of coeffs	size of h_j	size of rhs	contribution to β_n
1	–	0	0	0	$(n/2)$
e_1	W_1	1	1	$(1/2)+\alpha_n$	$(n-1)/2$
e_2	$G_{1,2}$	2	1	α_n	$(n/2)+\alpha_n$
e_1e_2	W_1W_2	2	2	$1+2\alpha_n$	$(n-2)/2$
e_3	$G_{1,3}$	2	1	α_n	$(n/2)+\alpha_n$
e_1e_3	$W_1G_{2,3}$	3	2	$(1/2)+2\alpha_n$	$(n-1)/2+\alpha_n$
e_2e_3	$W_2G_{1,3}$	3	2	$(1/2)+2\alpha_n$	$(n-1)/2+\alpha_n$
$e_1e_2e_3$	$W_1W_2W_3$	3	3	$(3/2)+3\alpha_n$	$(n-3)/2$
e_4	$G_{1,4}$	2	1	α_n	$(n/2)+\alpha_n$
e_1e_4	$W_1G_{2,4}$	3	2	$(1/2)+2\alpha_n$	$(n-1)/2+\alpha_n$
e_2e_4	$G_{1,2}G_{3,4}$	4	2	$2\alpha_n$	$(n/2)+2\alpha_n$
e_3e_4	$G_{1,3}G_{2,4}$	4	2	$2\alpha_n$	$(n/2)+2\alpha_n$
$e_1e_2e_4$	$W_1W_2G_{3,4}$	4	3	$1+3\alpha_n$	$(n-2)/2+\alpha_n$
$e_1e_3e_4$	$W_1W_3G_{2,4}$	4	3	$1+3\alpha_n$	$(n-2)/2+\alpha_n$
$e_2e_3e_4$	$W_2W_3G_{1,4}$	4	3	$1+3\alpha_n$	$(n-2)/2+\alpha_n$
$e_1e_2e_3e_4$	$W_1W_2W_3W_4$	4	4	$2+4\alpha_n$	$(n-4)/2$
e_5	$G_{1,5}$	2	1	α_n	$(n/2)+\alpha_n$
e_1e_5	$W_1G_{2,5}$	3	2	$(1/2)+2\alpha_n$	$(n-1)/2+\alpha_n$
e_2e_5	$G_{1,2}G_{3,5}$	4	2	$2\alpha_n$	$(n/2)+2\alpha_n$
e_3e_5	$G_{1,3}G_{4,5}$	4	2	$2\alpha_n$	$(n/2)+2\alpha_n$
e_4e_5	$G_{1,4}G_{2,5}$	4	2	$2\alpha_n$	$(n-2)/2+\alpha_n$
$e_1e_2e_5$	$W_1W_2G_{4,5}$	4	3	$1+3\alpha_n$	$(n-1)/2+2\alpha_n$
$e_1e_3e_5$	$W_1G_{2,3}G_{4,5}$	5	3	$(1/2)+3\alpha_n$	$(n-1)/2+2\alpha_n$
$e_1e_4e_5$	$W_1G_{2,4}G_{3,5}$	5	3	$(1/2)+3\alpha_n$	$(n-1)/2+2\alpha_n$
$e_2e_3e_5$	$W_2G_{1,3}G_{4,5}$	5	3	$(1/2)+3\alpha_n$	$(n-1)/2+2\alpha_n$
$e_2e_4e_5$	$W_2G_{1,4}G_{3,5}$	5	3	$(1/2)+3\alpha_n$	$(n-1)/2+2\alpha_n$
$e_3e_4e_5$	$W_3G_{2,4}G_{1,5}$	5	3	$(1/2)+3\alpha_n$	$(n-1)/2+2\alpha_n$
$e_1e_2e_3e_5$	$W_1W_2W_3G_{4,5}$	5	4	$(3/2)+4\alpha_n$	$(n-3)/2+\alpha_n$
$e_1e_2e_4e_5$	$W_1W_2W_4G_{3,5}$	5	4	$(3/2)+4\alpha_n$	$(n-3)/2+\alpha_n$
$e_1e_3e_4e_5$	$W_1W_3W_4G_{2,5}$	5	4	$(3/2)+4\alpha_n$	$(n-3)/2+\alpha_n$
$e_2e_3e_4e_5$	$W_2W_3W_4G_{1,5}$	5	4	$(3/2)+4\alpha_n$	$(n-3)/2+\alpha_n$
$e_1e_2e_3e_4e_5$	$W_1W_2W_3W_4W_5$	5	5	$(5/2)+5\alpha_n$	$(n-5)/2$

A table showing the chosen relations for $n \leq 5$.

After the initial "base relation" (which requires that the first component of b should be small), we seek a linear relation between e_1 and 1 (or a multiple of this e.g. N), and our only choice for this is W_1. With the introduction of the next exponent e_2 we now look for a relation between $1, e_1$ and e_2. For this we can either choose W_2 or $G_{1,2}$, and as explained above $G_{1,2}$ is the right choice.

A more interesting situation arises when the fourth exponent e_4 has been introduced, and one looks for a relation regarding e_1e_4 and the previous ones. The best choice in this case turns out to be $W_1G_{2,4}$. However, when considering the next relation regarding e_2e_4 and the previous ones we may now use $G_{1,2}G_{3,4}$ because the left-hand side of this relation contains e_1e_3, e_1e_4, e_2e_3 and e_2e_4 all of which are now present.

In general when looking for a relation regarding $e_{i_1}e_{i_2}\dots e_{i_s}$ and the previous ones, one can use any relation $R_{u,v}$ where $u+v=s$, subject to the required h_j being present earlier. It can be shown that the number of relations R_{u+v} with $v=t$ should be $\binom{n}{t}-\binom{n}{t-1}$ regardless of the size $s=u+v$ of the relation (though of course this is subject to $t\le s$ and $s+2t\le n$). The contribution to β_n for such a relation is $(n-s+t)/2+t\alpha_n$, and thus (summing over the possible n) the total contribution to β_n is shown below.

$$\beta_n=\sum_{s=0}^{n}\sum_{t=0}^{\min(s,n-s)}\left(\binom{n}{t}-\binom{n}{t-1}\right)\left(\frac{n-s+t}{2}+t\alpha_n\right)$$

Assuming n is even this sum can be simplified to

$$\beta_n=\frac{(2n+1)2^n-(2n+1)\binom{n}{n/2}}{4}+\frac{(n+1)2^n-(2n+1)\binom{n}{n/2}}{2}\,\alpha_n,$$

or if n is odd then the sum becomes

$$\beta_n=\frac{(2n+1)2^n-4n\binom{n-1}{(n-1)/2}}{4}+\frac{(n+1)2^n-4n\binom{n-1}{(n-1)/2}}{2}\,\alpha_n.$$

Using equation 4 this means that if n is even, then

$$\alpha_n=\frac{(2n+1)2^n-(2n+1)\binom{n}{n/2}}{(2n-2)2^n+(4n+2)\binom{n}{n/2}},\tag{5}$$

whilst if n is odd, then

$$\alpha_n=\frac{(2n+1)2^n-4n\binom{n-1}{(n-1)/2}}{(2n-2)2^n+8n\binom{n-1}{(n-1)/2}}.\tag{6}$$

Either way, using Stirling's formula $n!\simeq\sqrt{2\pi n}n^ne^{-n}$ we get that

$$\binom{2k}{k}\;=\;\frac{(2k)!}{(k!)^2}\;\simeq\;\frac{1}{\sqrt{\pi k}}2^{2k}\;\ll\;2^{2k}$$

as $k\to\infty$, and then we have that $\alpha_n\to 1$ as $n\to\infty$.