Small Private Exponent Partial Key-Exposure Attacks on Multiprime RSA

M. Jason Hinek
School of Computer Science, University of Waterloo,
Waterloo, Ontario, N2L 3G1, Canada.
mjhinek@alumni.uwaterloo.ca

May 25, 2005

Abstract

Given knowledge of one or more of the primes in a multiprime RSA modulus we show that the private exponent can be recovered provided it is sufficiently small. In particular, we present a simple and efficient method that given v of the u primes dividing the modulus N recovers any private exponent d satisfying $d < N^{v/u-\epsilon}$. When only one prime is known, this bound can be increased to approximately $N^{1/u+1/(2u^2)}$ using Boneh & Durfee's techniques for small private exponent attacks on RSA. We also present experimental data which shows that the attack becomes more costly with increasing number of primes in the modulus and increasing modulus sizes.

1 Introduction

In this work we consider attacks on multiprime RSA in which one or more of the primes in the modulus are known. A standard for RSA keys has been proposed in the Public-Key Cryptography Standards (PKCS) #1 v2.1 by RSA Security Inc. In particular, ASN.1 types for RSA public and private keys have been defined [14, Appendix A.1] and are shown in Appendix A. For convenience, we will simplify these keys as follows: for u-prime RSA let the public key be given by the tuple (e, N) and the private key be given by the $(u+1)$-tuple $(d_0, r_1, \ldots, r_u)$. Note that each value in the simplified keys is also found in the formal key types and that remaining information contained in the formal key types can be computed with the information in these simplified keys, so nothing (but efficiency) is lost by using the simplified keys.

The $r_1, \ldots, r_u$ are randomly chosen distinct odd primes called the RSA primes, $N = \prod_{i=1}^{u} r_i$ is called the (multiprime) RSA modulus, e is called the public (or encryption) exponent and d_0 is called the private (or decryption) exponent. The public and private exponents are chosen to satisfy

$$e d_0 \equiv 1 \bmod \lambda(N),$$

where $\lambda(N) = \text{lcm}(r_1 - 1, \ldots, r_u - 1)$. The private exponent d_0 is not the only value that can be used as a decryption exponent though. In fact, any integer equivalent to d_0 modulo $\lambda(N)$ will do. Of the many other decrypting exponents we are interested in the value $d \in \mathbb{Z}^*_{\phi(N)}$ satisfying

$$e d \equiv 1 \bmod \phi(N),$$

where $\phi(N)$ is Euler's totient function. While d may or may not be equal to d_0, we will also refer to d as the private (or decryption) exponent. Converting the above equivalence into an equation we obtain

$$e d = 1 + k\phi(N), \tag{1}$$

where k is some positive integer. We will refer to (1) as the *key equation*. Each of the attacks below use the key equation in some way to obtain d. Once d is obtained, the instance of multiprime RSA with public key (e, N) is broken as any ciphertext can be decrypted with d.

When the RSA primes satisfy

$$\tfrac{1}{2} N^{1/u} < r_1, \ldots, r_u < 2N^{1/u},$$

they are said to be *balanced*. When the primes in the modulus are balanced, we also say that the modulus is balanced. If P is the product of any $1 \leq v \leq u$ distinct primes in N and the primes are balanced, then $P < 2^v N^{v/r}$ and $P - \phi(P) < v2^{v-1} N^{(v-1)/u}$. The second inequality follows from expanding $\phi(N) = \prod_{i=1}^{u}(r_i - 1)$ and collecting terms with equal order of magnitude.

The rest of this work is organized as follows. First we present a failed attack which is the motivation for constructing the next two attacks. Section 3 presents an attack that directly computes sufficiently small private exponents. Section 4 presents an attack that used lattice basis reduction to recover larger private exponents than the direct method when exactly one of the primes in the modulus is known. Section 5 collects the theoretical bounds for both attacks while Section 6 collects all the experimental data. Finally, we conclude with Section 7.

2 Motivation (A Failed Attempt)

In this section, we outline an attack that was the starting point for this work.

Let (e, N) and $(d, r_1, \ldots, r_u)$ be the public and private keys for an instance of u-prime RSA with balanced primes. Since the primes in N are distinct, we know that $\phi(N)$ can be written as $\phi(N) = (p-1)\phi(Q)$ where p is any one of the primes in N and $Q = N/p$. Assuming we know p we can write the key equation as

$$ed = 1 + k(p-1)\phi(Q), \tag{2}$$

where everything is unknown except e and p. Reducing this equation modulo e, multiplying both sides by $(p-1)^{-1}$ modulo e and then converting back to an equation yields

$$(p-1)^{-1} + k\phi(Q) = \gamma e, \tag{3}$$

for some integer γ. Combining equations (2) and (3), via the common factor $k\phi(Q)$, we see that

$$\gamma e - (p-1)^{-1} = \frac{ed-1}{p-1}. \tag{4}$$

Multiplying both sides of this equation by $p-1$ and rearranging gives

$$ed - (p-1)\gamma e + (p-1)(p-1)^{-1} + 1 = 0, \tag{5}$$

which motivates the bivariate integer polynomial

$$f(x, y) = ex - (p-1)ey + (p-1)(p-1)^{-1} - 1. \tag{6}$$

Notice that $f(d, \gamma) = 0$.

The idea of an attack was to use Coppersmith's method for finding small roots of bivariate integer polynomials (see Coppersmith [5] and Coron [7]) to find $(x_0, y_0) = (d, \gamma)$, provided each was small enough. If the polynomial $f(x, y)$ is irreducible then it can be shown for sufficiently large N that the private exponent can be computed whenever $d < 8eN^{1/u}$, given infinite computing capabilities (time and space). For any $e > N^{(u-1)/u}$ the attack would work for any private exponent $1 < d < \phi(N)$. Initially thinking that this was the case we carried out some experiments to see if any size private exponent could be recovered in practise with modest computing power. No private exponent greater than $N^{1/u}$ could be recovered. Of course, our oversight was that since $(p-1)(p-1)^{-1} - 1 = (1 + \tau e) - 1 = \tau e$ for some

integer τ, the content of $f(x,y)$ is e and so $f(x,y)$ is not irreducible. The attack can still be mounted with the polynomial $f(x,y)/e$, but the range of private exponents that are vulnerable to it is reduced to $d < N^{1/u-\epsilon}$.

The result of this attack is somewhat disappointing as it requires great computational effort (for the lattice basis reduction and resultant computation) to only reach a bound of $d < N^{1/u}$. Essentially, given $\frac{1}{u}\log N$-bits of the modulus, we can recover d if it contains no more than $\frac{1}{u}\log N$-bits of information. In the next two sections we present attacks whose aim was to improve upon this. The first attack achieves the same bound for d but can be mounted with much less computational effort while the second increases the bound.

3 Direct Computation

When the decrypting exponent d is sufficiently small, knowledge of one or more of the primes in N allows us to compute d with a few elementary operations. The main result of this section is the following theorem.

Lemma 1 *Let $N = r_1 r_2 \cdots r_u$ be a balanced u-prime RSA modulus, (e, N) be any valid u-prime RSA public key and $d = N^\delta$ be the inverse of e modulo $\phi(N)$. Given the public key and any $1 \leq v \leq u-2$ of the primes in the factorization of N, if d satisfies*

$$\delta < \frac{v}{u} - (v+1)\log_N 2,$$

then d can be computed in time polynomial in $\log_2 N$.

Proof: Let P be the product of the v known primes and $Q = N/P$. Since the primes in N are pairwise distinct we can write Euler's totient function of N as $\phi(N) = \phi(Q)\phi(P)$. This allows the key equation, $ed = 1 + k\phi(N)$, to be written as

$$ed = 1 + k\phi(Q)\phi(P), \tag{7}$$

where e and $\phi(P)$ are known quantities. Now, reducing this equation modulo $\phi(P)$ yields $d \equiv e^{-1} \bmod \phi(P)$ and so whenever $d < \phi(P)$ it follows that $d = e^{-1} \bmod \phi(P)$. Since $\phi(P) > \frac{1}{2}P > \frac{1}{2}\frac{1}{2^v}N^{v/u}$, a sufficient condition that $d < \phi(P)$ is given by $d = N^\delta < \frac{1}{2}\frac{1}{2^v}N^{v/u}$, or simply

$$\delta < \frac{v}{u} - (v+1)\log_N 2. \tag{8}$$

Since the cost of computing the inverse of $e < N$ modulo $\phi(P) < N$ is polynomial in $\log_2 N$ the result follows. ❑

Since computing the inverse of e modulo $\phi(P)$ and computing the convergents of $\phi(P)/e$ are intimately related (through the Extended Euclidean Algorithm), it is expected that a similar result can be obtained using Wiener's continued fraction attack on small private exponent RSA (see [16] for details). Indeed, given v of the u primes in a u-prime RSA modulus, it is simple to show that $d/(k\phi(Q))$ will be one of the convergents in the continued fraction expansion of $\phi(P)/e$ whenever

$$\delta < \frac{v}{u} + \alpha - 1 - (u - v + 1)\log_N 2, \tag{9}$$

where as above P is the product of the known v primes and $Q = N/P$. To see this, first recall that if

$$\left|\frac{a}{b} - \frac{c}{d}\right| \leq \frac{1}{2d^2},$$

then c/d is one of the convergents in the continued fraction expansion of a/b. Now, starting with the key equation $ed = 1 + k\phi(P)\phi(Q)$, we can see that

$$\left|\frac{\phi(P)}{e} - \frac{d}{k\phi(Q)}\right| = \left|\frac{1}{ek\phi(Q)}\right| \leq \frac{1}{2(k\phi(Q))^2}$$

whenever $e \geq 2k\phi(Q)$ or equivalentlyg $k \leq e/(2\phi(Q))$. Since $k < d$ and $\phi(Q) \leq 2^{(u-v)}N^{(u-v)/u}$ a sufficient condition for this to hold is given by

$$d < \frac{1}{2^{u-v+1}} eN^{-(u-v)/u}.$$

Letting $d = N^{\delta}$ and $e = N^{\alpha}$ we see that this is equivalent to the bound given in (9).

4 Lattice Basis Reduction

In this section we show that the bound on the private exponent d can be increased using lattice basis reduction. First we recall some necessary facts about lattice basis reduction and roots of polynomials. Next we introduce a new problem (a slight generalization of a known problem) and derive bounds on solutions that can be found using lattice basis reduction. Finally, we present the main result which is an attack for private exponents above the $N^{u/v}$ bound.

The first fact we need deals with finding small vectors in lattices. We will use the LLL algorithm as a black box which finds small vectors in a given lattice. The main result we need is as follows.

Fact 1 (LLL [12]) *Let $\mathcal{L} \in \mathbb{Z}^m$ be a lattice spanned by $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m)$. The LLL algorithm, with input $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m)$, outputs in polynomial time a reduced lattice basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)$ for $\mathcal{L}$ such that*

$$\|\boldsymbol{b}_1\| \leq \|\boldsymbol{b}_2\| \leq 2^{m/4} d(\mathcal{L})^{1/(m-1)},$$

where $d(\mathcal{L})$ is the volume of the lattice $\mathcal{L}$.

The second fact gives a sufficient condition for roots of a polynomial modulo some integer to also be roots of that polynomial over the integers. The univariate form of this result comes from Howgrave-Graham[10].

Fact 2 (Integer Roots) *Let $f(x_1, \ldots, x_r) \in \mathbb{Z}[x_1, \ldots, x_r]$ be a polynomial that is the sum of at most ω monomials. For any positive integer M, if*

1. *$f(\widehat{x}_1, \ldots, \widehat{x}_r) \equiv 0 \bmod M$ where $|\widehat{x}_i| \leq X_i$ for $1 \leq i \leq r$, and*
2. *$\sqrt{\omega} \cdot \|f(x_1 X_1, \ldots, x_r X_r)\| < M$,*

then $f(\hat{x}_1, \ldots, \hat{x}_r) = 0$.

With these facts in hand, we now consider the *small inverse problem* which was introduced by Boneh & Durfee in [3]. Given bounds X and Y and the polynomial

$$f_M(x, y) = x(A + y) - 1, \tag{10}$$

the small inverse problem is to find all integer pairs (x_0, y_0) such that $|x_0| < X$, $|y_0| < Y$ and $f_M(x_0, y_0) \equiv 0 \pmod{M}$. Alternatively, this can be viewed as finding a small inverse (x_0) of a number close to A $(A + y_0)$ in $\mathbb{Z}_M$, where small is defined by X and close is defined by Y.

Boneh & Durfee present some lattice based methods for solving the small inverse problem based on Coppersmith's method [6] as simplified by Howgrave-Graham [10]. In particular, they give sufficient conditions on the bound X for fixed $Y = 3M^{1/2}$ so that their method will likely succeed[1]. For sufficiently large M, $A \approx M$ and $Y = 3M^{1/2}$ they show that $X < M^{0.284}$. Using sublattices they improve this bound to $X < M^{0.292}$. Both of these bounds require infinite computing capabilities.

[1]The method yields two linearly independent polynomials in $\mathbb{Z}[x, y]$ with root (x_0, y_0) over $\mathbb{Z}$, but does not guarantee that they are algebraically independent.

Building on the problem definition and solution of the small inverse problem, we define the *small multiplier problem* as follows: given bounds X and Y and the function

$$f_M(x, y) = x(A + y) - B, \tag{11}$$

the small multiplier problem is to find all integer pairs (x_0, y_0) such that $|x_0| < X$, $|y_0| < Y$ and $f_M(x_0, y_0) \equiv 0 \pmod{M}$. That is, we wish to find a small multiplier (x_0) of a number close to A $(A - y_0)$ whose product is B in $\mathbb{Z}_M$, where again small is defined by X and close is defined by Y.

The main result concerning the small multiplier problem is given in the following theorem.

Theorem 1 (Small Multiplier Problem) *Let $f_M(x, y) = x(A + y) - B$ with $A, B \in \mathbb{Z}$. For any $\epsilon > 0$, there exists a positive M_0 such that for every integer $M > M_0$ we can find two linearly independent polynomials $p_1, p_2 \in \mathbb{Z}[x, y]$ such that all integer pairs (x_0, y_0) satisfying $|x_0| < X = M^\delta$, $|y_0| < Y = M^\lambda$ and $f_M(x_0, y_0) \equiv 0 \bmod M$ are also roots of p_1 and p_2 over $\mathbb{Z}$ whenever*

$$-3\delta^2 + (2\lambda + 6)\delta + \lambda^2 + 2\lambda - 3 + \epsilon < 0. \tag{12}$$

This inequality can also be written as

$$\delta < 1 + \tfrac{1}{3}\lambda - \tfrac{2}{3}\sqrt{\lambda^2 + 3\lambda - \tfrac{3}{4}\epsilon}, \text{ or}$$

$$\lambda < 2\sqrt{1 - \delta + \delta^2 + \tfrac{1}{4}\epsilon} - \delta - 1,$$

which may be more useful.

As we shall see, this result is simply a generalization of Boneh & Durfee's small inverse problem in which the bound Y is not fixed beforehand. The result is in fact independent of the value of B.

Proof: The proof follows Boneh & Durfee's analysis of the small inverse problem using a full rank lattice [3, Section 4] closely. We begin with the polynomial $f_M \in \mathbb{Z}[x, y]$ given by

$$f_M(x, y) = x(A + y) - B,$$

where M is fixed positive integer but not yet determined.

Let $X = M^\delta$ and $Y = M^\lambda$ be bounds on x and y, respectively. Since we are looking for roots of $f_M(x, y)$ in $\mathbb{Z}_M$ we impose the restriction $0 \leq \delta, \lambda \leq 1$.

Also, let $m \geq 1$ and $t \geq 0$ be fixed integers, also not yet determined. Define the x- and y-shift polynomials of $f_N(x,y)$ as

$$g_{i,k}(x,y) = M^{h-k}x^i(f_M(x,y))^k \text{ and } h_{j,k}(x,y) = M^{h-k}y^j(f_M(x,y))^k,$$

respectively. Notice that each (x_0, y_0) satisfying $f_M(x_0, y_0) \equiv 0 \pmod{M}$ also satisfies $g_{i,k}(x_0, y_0) \equiv 0 \pmod{M^m}$ and $h_{j,k}(x_0, y_0) \equiv 0 \pmod{M^m}$ for all $i, j \geq 0$ and $0 \leq k \leq m$. We construct the lattice $\mathcal{L}$ whose basis matrix $\mathcal{M}$ is made up of the coefficient vectors of the $w = (m+1)(m+2)/2 + t(m+1)$, x- and y-shift polynomials

$$\begin{array}{ll} g_{i,k}(xX, yY) & \text{for all } \ 0 \leq i \leq m-k \ , \ 0 \leq k \leq m, \text{ and} \\ h_{j,k}(xX, yY) & \text{for all } \ 1 \leq j \leq t \ , \ 0 \leq k \leq m. \end{array}$$

Notice that each vector in $\mathcal{L}$ is the coefficient vector of a polynomial $p \in \mathbb{Z}[x,y]$ which is an integer linear combination of the $g_{i,k}(x,y)$ and $h_{j,k}(x,y)$. Thus each (x_0, y_0) satisfying $f_M(x_0, y_0) \equiv 0 \pmod{M}$ must also satisfy $p(x_0, y_0) \equiv 0 \pmod{M^w}$.

With a proper ordering of the polynomials we see that $\mathcal{M}$ is triangular with all diagonal elements nonzero. Thus, the lattice is full rank with dimension w. A simple calculation shows that the volume of the lattice is given by

$$d(\mathcal{L}) = d(\mathcal{L}_x)d(\mathcal{L}_y),$$

where

$$d(\mathcal{L}_x) = M^{m(m+1)(m+2)/3}X^{m(m+1)(m+2)/3}Y^{m(m+1)(m+2)/6},$$

is the contribution to the volume by the x-shift polynomials and

$$d(\mathcal{L}_y) = M^{tm(m+1)/2}X^{tm(m+1)/2}Y^{t(m+1)(m+t+1)/2}$$

is the contribution to the volume by the y-shift polynomials.

Letting $X = M^\delta$, $Y = M^\lambda$ and $t = \tau m$ for some real number $\tau \geq 0$, we see that

$$\begin{aligned} \log_M d(\mathcal{L}) \ &= \ \frac{1}{6}\left(3\,\lambda\,\tau^2 + (3\,\delta + 3 + 3\,\lambda)\,\tau + 2 + 2\,\delta + \lambda\right) m^3 \\ &\quad + \frac{1}{6}\left(3\,\lambda\,\tau^2 + (3\,\delta + 3 + 6\,\lambda)\,\tau + 6\,\delta + 6 + 3\,\lambda\right) m^2 \\ &\quad + \frac{1}{6}\left(3\,\lambda\,\tau + 4 + 4\,\delta + 2\,\lambda\right) m, \end{aligned}$$

and

$$m(w-1) = \frac{1}{2}\left(1 + 2\tau\right) m^3 + \frac{1}{2}\left(3 + 2\tau\right) m^2.$$

Using the LLL-algorithm, we know from Fact 1 that we can find two vectors in $\mathcal{L}$ that are the coefficient vectors of two polynomials $p_1(x, y)$ and $p_2(x, y)$ satisfying

$$\|p_1(xX, yY)\| \leq \|p_2(xX, yY)\| \leq 2^{w/4} d(\mathcal{L})^{1/(w-1)}.$$

Since the coefficient vectors of p_1 and p_2 are in $\mathcal{L}$ we know that each (x_0, y_0) satisfying $f_M(x_0, y_0) \equiv 0 \pmod{M}$ also satisfies $p_1(x_0, y_0) \equiv 0 \pmod{M^w}$ and $p_2(x_0, y_0) \equiv 0 \pmod{M^w}$. Further, if

$$\|p_1(xX, yY)\| \leq \|p_2(xX, yY)\| < w^{-1/2} M^w,$$

then by Fact 2, we know that each such (x_0, y_0) is also a root of p_1 and p_2 over $\mathbb{Z}$. A sufficient condition for this to occur is given by

$$2^{w/4} d(\mathcal{L})^{1/(w-1)} < w^{-1/2} M^m,$$

or

$$d(\mathcal{L}) < c(w) M^{m(w-1)}, \tag{13}$$

where $c(w) = 2^{-w(w-1)/4} w^{-(w-1)/2}$. Defining $M_0 = c(w)^{1/m^2}$ we see that $c(w) = M^{o(m^3)}$ whenever $M > M_0$. Notice that M_0 cannot be fixed until m and t are determined though. Assuming that $M > M_0$, we can rewrite (13) as

$$M^{\frac{1}{6}\left(3\,\lambda\,\tau^2+3(1+\delta+\lambda)\tau+2+2\,\delta+\lambda\right)m^3+o(m^3)} < M^{\frac{1}{2}(1+2\tau)m^3+o(m^3)}.$$

Focusing only on the exponents of M and assuming m is large enough so that the $o(m^3)$ terms are negligible this inequality becomes

$$\frac{1}{6}\left(3\,\lambda\,\tau^2 + 3\,(-1+\delta+\lambda)\,\tau - 1 + 2\,\delta + \lambda + \epsilon\right) m^3 < 0,$$

where $\epsilon > 0$ is introduced to account for the $o(m^3)$ terms. The size of ϵ is of the order of $o(m^3)/m^3$. For any values of δ and $\lambda > 0$, the left-hand side of this inequality is minimized when τ is chosen to be

$$\tau_{\text{opt}} = \frac{1-\delta-\lambda}{2\lambda}.$$

Substituting τ_{opt} back into the inequality yields

$$-3\delta^2 + (2\lambda+6)\delta + \lambda^2 + 2\lambda - 3 + \epsilon < 0,$$

which is the desired inequality. ❑

We can now give the result of the attack on multi-prime RSA.

Theorem 2 *For every $\epsilon > 0$ there exists an N_0 such that for every balanced u-prime RSA modulus $N = r_1 r_2 \cdots r_u > N_0$ with public exponent $e = N^\alpha$ and private exponent $d = N^\delta = e^{-1}$ modulo $\phi(N)$, given the public key and $1 \leq v \leq u-2$ of the primes in the factorization of N, if*

$$\delta < \alpha + \frac{1}{3}(u-v-1)(\frac{1}{u} + \log_N 2) + \frac{1}{3}\log_N(u-v) - \frac{2\alpha}{3}\sqrt{D^2 + 3\alpha D + \epsilon},$$

where D is given by

$$D = \frac{u-v-1}{u} + (u-v-1)\log_N 2 + \log_N(u-v).$$

then two polynomials in $\mathbb{Z}[x,y]$ can be found with a common root that yields the private exponent d.

Proof: As in the proof of Lemma 1, let P be the product of the v known primes, $Q = N/P$ and consider equation (7)

$$ed = 1 + k\phi(P)\phi(Q).$$

Reducing this equation modulo e, multiplying by $\phi(P)^{-1}$ modulo e and replacing $\phi(Q)$ with $\phi(Q) = Q - \Lambda_Q$ we obtain

$$k(Q - \Lambda_Q) + \phi(P)^{-1} \equiv 0 \pmod{e}. \tag{14}$$

Again, as $ed \equiv 1 \bmod \phi(N)$ and $\phi(N) = \phi(P)\phi(Q)$, we know that the inverse of $\phi(P)$ modulo e exists. Replacing k by x and $-\Lambda_Q$ by y in (14) leads to the following instance of the small multiplier problem: find all small roots (x_0, y_0) of

$$f_e(x,y) = x(Q+y) - (-\phi(P)^{-1}), \tag{15}$$

modulo e. In particular, we are trying to find the solution $(k, -\Lambda_Q)$. Since $k < d = N^\delta$ and $\Lambda_Q < (u-v)2^{u-v-1}N^{(u-v-1)/u}$ we define the bounds

$$X = N^\delta \quad \text{and} \quad Y = (u-v)2^{u-v-1}N^{(u-v-1)/u}.$$

Applying the result of Lemma 1 we obtain two polynomials, f_1 and f_2 say, with a common root $(k, -\Lambda_Q)$. Since $\phi(Q) = Q - \Lambda_Q$, the private exponent is simply given by $d = e^{-1} \bmod \phi(P)\phi(Q)$. ❑

Of course, for this result to be of any use we must be able to find the root $(k, -\Lambda_Q)$ from f_1 and f_2. If f_1 and f_2 are algebraically independent (which in practise is usually the case), a univariate polynomial $g \in \mathbb{Z}[y]$ with root $y_0 = -\Lambda_Q$ over the integers can be found by computing the resultant of f_1 and f_2 to remove the variable x. Standard root finding techniques can then be applied to g to recover $y_0 = -\Lambda_Q$.

5 Theoretical Bounds

In Table 1, we give some bounds on the private exponent $d = N^{\delta}$ so that the attacks of Lemma 1 and Theorem 2 will succeed. Bounds are given for various sizes of N, values of u (number of primes in the modulus), v (number of primes known) and public exponent sizes. It should be pointed out that these bounds are sufficient conditions for the attacks to succeed.

The maximum number of primes for each modulus size is based on projections for the year 2010 by Lenstra [11], except for $v = 5$ which should not appear until the modulus size is 8192 bits. Since we also want to estimate how close in practise one can come to the theoretical bounds (see Section 6 for experimental results) we only consider moduli with bitsize 1024, 2048 and 4096 in order to keep the runtime of the lattice basis reduction reasonable. Because of this, we include $v = 5$ for 4096-bit moduli to give a clearer indication of how the bounds change with increasing number of primes.

From the data we see that the bounds for the lattice based attack are greater than the bounds for the direct computation attack only when exactly one prime is known. If we consider this case (one prime known) we can estimate the bounds on the private exponent for very large modulus given by Lemma 1 and Theorem 2. Letting δ_{direct} be the bound given in Lemma 1, $\delta_{lattice}$ be the bound given in Theorem 2 observe that in the limiting case of $v = 1$ and $N \to \infty$ we have

$$\delta^*_{direct} = \lim_{N\to\infty} \delta_{direct}\big|_{v=1} = \frac{1}{u},$$

and

$$\delta^*_{lattice} = \lim_{N\to\infty} \delta_{lattice}\big|_{\substack{v=1\\ \alpha=1}} = \frac{4}{3} - \frac{2}{3u} - \frac{2\sqrt{2}}{3}\sqrt{\frac{2}{u^2} - \frac{5}{u} + 2} \approx \frac{1}{u} + \frac{1}{2u^2}.$$

The approximation $\delta^*_{lattice} \approx \frac{1}{u} + \frac{1}{2u^2}$ underestimates $\delta_{lattice}$ for $u = 3, 4, 5$ by at most 0.0195 at $u = 3$ (decreasing to 0.0035 for $u = 4$ and 0.0002 for $u = 5$). For all $u \geq 6$ the approximation overestimates $\delta_{lattice}$ by at most 0.0009. Thus, when exactly one prime in the modulus is known, we see that for small values of u the bound on the private exponent can be increased from about $\frac{1}{u}$ to about $\frac{1}{u} + \frac{1}{2u^2}$ using the lattice based attack.

6 Experimental Results

In order to test the effectiveness of the lattice based attack, we carried out numerous experiments. Since the theoretical bounds for the lattice based

$\log_2 N$	u	v	Direct Computation	Lattice Theory $\alpha = 1.0$	Lattice Theory $\alpha = 0.99$
1024	3	1	0.33138	0.40677	0.39995
2048	3	1	0.33236	0.40758	0.40075
	4	1	0.24902	0.28357	0.27736
	4	2	0.49854	0.48147	0.47426
4096	3	1	0.33284	0.40798	0.40115
	4	1	0.24951	0.28416	0.27795
	4	2	0.49927	0.48194	0.47472
	5	1	0.19951	0.21945	0.21354
	5	2	0.39927	0.35521	0.34866
	5	3	0.59902	0.53280	0.52531

Table 1: Theoretical bounds on δ ($d = N^{\delta}$) for public exponents sized $e = N^{\alpha}$ for $\alpha = 1.0$ and $\alpha = 0.99$. Here u is the number of primes in the modulus, v is the number of known primes and $\log_2 N$ is the bitsize of the modulus.

attack are greater than those for the direct computation attack only when exactly one prime in the modulus is known ($v = 1$) we focus only on this case.

For each combination of modulus size and number of primes u, we considered many lattice parameters m and t. For each choice of lattice parameters, we performed several experiments to estimate the largest private exponent that is vulnerable to the attack. This was accomplished by carrying out a binary search on the private key size and terminating when the difference between the private key sizes of the current experiment and the previous is less than some threshold (we used 0.0005 as the cutoff difference). The largest private exponent for which the attack was successful is returned.

For each experiment, we generate new random primes and compute a valid private exponent of the desired size. Since the public exponent is defined by the private exponent ($e = d^{-1} \bmod \phi(N)$) the size of the public exponent varied from instance to instance. Once N, d and e are fixed, we carry out the attack from Theorem 2. Initially, the experiments for each lattice parameter was repeated 10 times to compute the average largest private exponent vulnerable to the attack for a given m and t. After may experiments, the number of trials was reduced to 5 and eventually down to 3 for some of the larger values of m and t as the variation between runs of the same lattice parameters was found to be negligible with respect to the

$\log_2 N$	u	v	δ	dim	(m,t)	LLL (hours)
1024	3	1	0.379	63	(8,2)	$\sim 1.5^*$
2048	3	1	0.379	52	(7,2)	~ 51
	4	1	0.263	52	(7,2)	~ 38
4096	3	1	0.380	52	(7,2)	$\sim 136^*$
	4	1	0.264	52	(7,2)	$\sim 97^*$
	5	1	0.202	52	(7,2)	$\sim 78^*$

Table 2: Experimental Bounds on δ. (*) denotes experiments done on the faster server.

accuracy of our results.

All work was done using Maple 9.5 [13] (resultant computations, root finding, polynomial manipulation) except for the lattice basis reduction which was carried out with NTL [15]. The experiments were carried out on either a Sun Fire V100 server with one UltraSPARC IIe processor with 2GB of memory running at 550 MHz, or on a Sun Fire V440 server with four UltraSPARC IIIi processors with 8 GB of memory each running at 1.062 GHz. Since there were seven V100 servers available and only one V440, most experiments were carried out on the slower machines. As such, all timing results given below are for the V100 server unless otherwise stated.

Table 2 shows the largest private exponent size that is vulnerable to the lattice based attack for each modulus size and number of primes in the modulus. The time needed to perform the lattice basis reduction is also included. In all cases we were able to recover private exponents that were larger than $N^{1/u}$, although it became more computationally expensive to reach this point when using more primes and when using larger moduli. In particular, when there are 5 primes in the modulus the time required for the lattice basis reduction was quite significant (78 hours) to simply exceed the bound for the direct attack.

For each of the cases considered the experimental bound reached roughly 93% of the theoretical bound with $\alpha = 1.0$ and 95% of the theoretical bound with $\alpha = 0.99$. Since the size of the public exponent in all of the experiments carried out had $0.99 < \alpha < 1.0$, we see that the attack comes fairly close to achieving the theoretical bounds. Of course, as mentioned above, the effort required to reach these bounds increases significantly when more primes are used and when larger moduli are used.

In Tables 3 and 4 we present all the data acquired. Table 3 gives the average largest private exponent vulnerable to the attack for each modulus

size and lattice parameters. The shaded values in both tables indicate when the experimental bound exceeded the direct computation bound. Table 4 gives the average time required for the lattice reduction for the largest private exponents. While the time required for the resultant computations are not given, they are comparable to the lattice reduction time for the larger lattice dimensions.

m, t	dim	3 1024	3 2048	3 4096	4 2048	4 4096	5 4096
1,1	5	0.331	0.332	0.332	0.248		
2,1	9	0.331	0.332	0.332	0.249	0.249	0.199
2,2	12	0.332	0.332	0.332	0.248		
3,1	14	0.356	0.357	0.358	0.249		
3,2	18	0.356	0.357	0.358	0.249		
3,3	22	0.356	0.357	0.358	0.249		
4,1	20	0.364	0.365	0.366	0.248	0.249	0.199
4,2	25	0.364	0.365	0.366	0.248	0.249	0.199
4,3	30	0.364					
4,4	35	0.364					
5,1	27	0.366	0.367	0.368	0.257	0.258	0.199
5,2	33	0.370	0.372	0.372	0.257	0.258	0.199
5,3	39	0.370					
5,4	45	0.371					
6,1	35	0.366	0.367	0.368	0.261	0.262	0.199
6,2	42	0.376	0.377	0.378	0.262	0.262	0.199
6,3	49	0.376					
6,4	56	0.375					
7,1	44	0.366					
7,2	52	0.378	0.379	0.380	0.263	0.264	0.203
7,3	60	0.378					
7,4	68	0.378					
8,1	54	0.368					
8,2	63	0.379					

Table 3: Experimental bounds on δ ($d = N^{\delta}$) for all experiments. Shaded values indicate smallest lattice dimension in which the experimental bound exceeds the direct computation bound.

m,t	dim	3 1024	3 2048	3 4096	4 2048	4 4096	5 4096
1,1	5	< 1s	< 1s	2s	< 1s		
2,1	9	3s	20s	120s	14s	60s	60s
2,2	12	10s	60s	300s	40s		
3,1	14	0.33m	2m	8m	3m		
3,2	18	2m	8m	50m	8m		
3,3	22	3m	18m	120m	16m		
4,1	20	0.03h	0.16h	1h	0.4h	1h*	1h*
4,2	25	0.15h	0.4h*	5h	1h	3h*	2h*
4,3	30	0.3h					
4,4	35	0.5h					
5,1	27	0.15h	1h	7h	36m	4h	9h
5,2	33	0.5h	3h	19h	3h	19h	25h
5,3	39	0.5h*					
5,4	45	2.3h					
6,1	35	0.6h	4h	28h	2h	13h	X
6,2	42	1h	13h	75h	12h	X	43h*
6,3	49	2h					
6,4	56	8.5h					
7,1	44	2.5h					
7,2	52	8h	51h	136h*	38h	97h*	78h*
7,3	60	16h					
7,4	68	11.5h*					
8,1	54	5h					
8,2	63	1.5h*					

Table 4: Lattice basis reduction times. Average times corresponding to the experimental bounds in Table 3, given in seconds (s), minutes (m) and hours (h). X denotes that times are not available. (*) denotes experiments done on the faster server. Shaded values indicate smallest lattice dimension in which the experimental bound exceeds the direct computation bound.

6.1 Algebraic Independence

In the lattice based attack, let $p_1(x, y), p_2(x, y), \ldots, p_w(x, y)$ denote the polynomials whose coefficient vectors (evaluated at (xX, yY)) are the vectors in the reduced lattice basis. Also, let (x_0, y_0) be a common root of each polynomial modulo M^m. Typically, when applying Coppersmith's methods to recover (x_0, y_0) we require $p_1(x, y)$ and $p_2(x, y)$ to be algebraically independent so that computing the resultant of the two polynomials (with respect to either x or y) is non-zero. In most cryptographic applications, it is assumed that this is always the case. However, this is in fact not always the case (see Blömer & May [2] and Hinek [8]). It is also not the case with this attack.

Even when $p_1(x, y)$ and $p_2(x, y)$ are algebraically dependent it is sometimes still possible to recover (x_0, y_0) though. Indeed, if any two of the polynomials, $p_i(x, y)$ and $p_j(x, y)$ say, are algebraically independent and satisfy $p_i(x_0, y_0) = p_j(x_0, y_0) = 0$ then then they can be used to recover x_0 and y_0. In our experiments, for example, there were some instances when $p_1(x, y)$ and $p_3(x, y)$ could be used to recover d while $p_1(x, y)$ and $p_2(x, y)$ could not since they were algebraically dependent. Trying to increase the likelihood of success, when carrying out our experiments we tried to recover d using the polynomial $p_1(x, y)$ with $p_k(x, y)$ for $k = 2, 3, 4, 5$ rather than just using $p_1(x, y)$ and $p_2(x, y)$. Starting with $k = 2$ we try to compute d. If the two polynomials are algebraically dependent, we increment k and try again. Of more than two thousand successful attacks approximately 5.3% needed a polynomial other than $p_2(x, y)$ to recover d with $p_1(x, y)$ (ie, $k > 2$). The following table shows how many of the successful attacks were accomplished with each of the four polynomial combinations.

polynomials	p_1, p_2	p_1,p_3	p_1,p_4	p_1, p_5
%	94.7	3.8	1.3	0.2

The frequency of successful attacks in which the first two polynomials could not be used (as they were algebraically dependent) was seen to be greater when the size of the private exponent is close to the experimental bound. For example, we carried out some experiments for 3-prime RSA with a 1024-bit modulus using lattice parameters $m = t = 2$ (lattice dimension 12). Successful attacks used polynomials $p_1(x, y)$ and $p_j(x, y)$ for some $j \in [2, 5]$ (as discussed above). Figure 1 shows the value of j for each successful attack. It is clear that as the private exponent gets closer to the experimental bound ($\delta = 0.332$ from Table 3) that the likelihood that $p_1(x, y)$ and $p_2(x, y)$ are algebraically dependent increases. The figure also seems to indicate that the amount of algebraic dependence of the polynomials grows

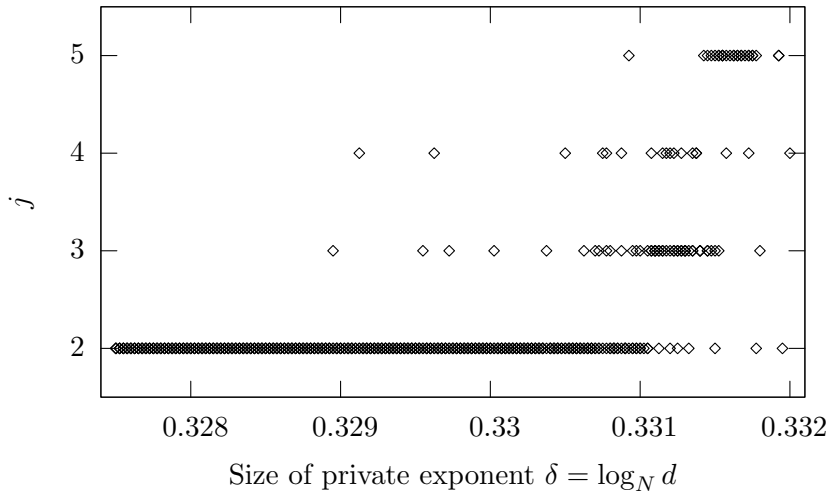

Figure 1: Successful attacks using polynomials $p_1(x, y)$ and $p_j(x, y)$ for 3-prime RSA with a 1024-bit modulus and lattice parameters $m = t = 2$.

as the private exponent approaches the bound (i.e., more of the polynomials are algebraically dependent). But, since we did not test algebraic dependence between all combinations of the polynomials considered, we feel more evidence is needed to back up this claim.

Frequency of algebraic dependence seemed to only depend on the size of the private exponent. There did not seem to be any correlation with any of the other parameters (lattice parameters, modulus size, number of primes, public exponent size).

7 Conclusion

As with all other non-factoring attacks on multiprime RSA (see Hinek, Low & Teske [9], Ciet, Koeune, Laguillaumie & Quisquater [4], and Hinek [8]), the attacks in this work become weaker with increasing modulus size and increasing number of primes in the modulus. Both the theoretical and experimental bounds on private exponents that are vulnerable to the attacks decrease while the computational costs increase with increasing N and u.

The experimental observations regarding algebraic dependence leaves some interesting open questions such as why the likelihood of algebraic dependence increases near the experimental bounds and is there any way of quantifying this phenomena. We are currently investigating this further.

A ANS.1 Key Syntax for PKCS #1 v2.1

Abstract Syntax Notation One (ASN.1) is a formal language for abstractly describing messages to be exchanged among a wide range of applications (including the Internet, cellular phones, ground-to-air communications, etc). For more information about ASN.1, see [1].

Appendix A.1 of the PKCS #1 v2.1 specifications [14] define ASN.1 object identifiers and the types **RSAPublicKey** and **RSAPrivateKey** for RSA public and private keys, respectively. These types are given as below:

```
RSAPublicKey ::= SEQUENCE {
    modulus           INTEGER,  -- n
    publicExponent    INTEGER   -- e
}

RSAPrivateKey ::= SEQUENCE {
    version           Version,
    modulus           INTEGER,  -- n
    publicExponent    INTEGER,  -- e
    privateExponent   INTEGER,  -- d
    prime1            INTEGER,  -- p
    prime2            INTEGER,  -- q
    exponent1         INTEGER,  -- d mod (p-1)
    exponent2         INTEGER,  -- d mod (q-1)
    coefficient       INTEGER,  -- (inverse of q) mod p
    otherPrimeInfos   OtherPrimeInfos  OPTIONAL
}

OtherPrimeInfos ::= SEQUENCE SIZE(1..MAX) OF OtherPrimeInfo

OtherPrimeInfo ::= SEQUENCE {
    prime             INTEGER,  -- ri
    exponent          INTEGER,  -- di
    coefficient       INTEGER   -- ti
}
```

When there are $u > 2$ primes in the modulus, the type **OtherPrimeInfos** contains information about the primes $r_3, \ldots, r_u$. Here, r_i is the i^{th} prime in the modulus, $d_i = d \mod (r_i - 1)$ and $t_i = (r_1 \cdot r_2 \cdots r_{i-1})^{-1} \mod r_i$.

References

[1] Absract Syntax Notation One. ASN.1 information site. Available online at `http://asn1.elibel.tm.fr/en/index.htm`.

[2] J. Blömer and A. May. Low secret exponent RSA revisited. In J. H. Silverman, editor, *Cryptography and Lattices – Proceedings of CALC '01*, volume 2146 of *Lecture Notes in Computer Science*, pages 4–19. Springer-Verlag, 2001.

[3] D. Boneh and G. Durfee. Cryptanalysis of RSA with private key d less than $N^{0.292}$. In *Advances in Cryptology – Proceedings of EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 1–11. Springer-Verlag, 1999.

[4] M. Ciet, F. Koeune, F. Laguillaumie, and J.-J. Quisquater. Short private exponent attacks on fast variants of rsa. UCL Crypto Group Technical Report Series CG-2003/4, Université Catholique de Louvain, 2003. Available at `http://www.dice.ucl.ac.be./crypto/tech_reports/`.

[5] D. Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In U. Maurer, editor, *Advances in Cryptology – Proceedings of EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 178–189. Springer-Verlag, 1996.

[6] D. Coppersmith. Finding a small root of a univariate modular equation. In U. Maurer, editor, *Advances in Cryptology – Proceedings of EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 155–165. Springer-Verlag, 1996.

[7] J.-S. Coron. Finding small roots of bivariate integer polynomial equations revisited. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 492–505. Springer-Verlag, 2004.

[8] M. J. Hinek. New partial key exposure attacks on RSA revisited. Technical Report CACR 2004-2, Centre for Applied Cryptographic Research, University of Waterloo, 2004. Available online at `http://www.cacr.math.uwaterloo.ca/`.

[9] M. J. Hinek, M. K. Low, and E. Teske. On some attacks on multi-prime RSA. In K. Nyberg and H. M. Heys, editors, *Selected Areas in Cryptography 2002*, volume 2595 of *Lecture Notes in Computer Science*, pages 385–404. Springer-Verlag, 2003.

[10] N. A. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Cryptography and Coding*, volume 1355 of *Lecture Notes in Computer Science*, pages 131–142. Springer-Verlag, 1997.

[11] A. K. Lenstra. Unbelievable security : Matching AES security using public key systems. In *Advances in Cryptology – Proceedings of ASIACRYPT 2001*, volume 2248 of *Lecture Notes In Computer Science*, pages 67–86. Springer-Verlag, 2001.

[12] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[13] Maplesoft. Maple 9.5. `http://www.maplesoft.com`.

[14] RSA Laboratories. PKCS #1 v2.1: RSA cryptography standard. Available online at `http://www.rsasecurity.com/rsalabs/`, June 2001.

[15] V. Shoup. NTL: A library for doing number theory, version 5.3.1. Available online at `http://shoup.net/ntl/`.

[16] M. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory*, 36(3):553–558, 1990.