J. Math. Cryptol. **4** (2010), 57–93
DOI 10.1515/JMC.2010.003

Common modulus attacks on small private exponent RSA and some fast variants (in practice)

M. Jason Hinek and Charles C. Y. Lam

Communicated by Kaoru Kurosawa

Abstract. In this work we re-examine two common modulus attacks on RSA. First, we show that Guo's continued fraction attack works much better in practice than previously expected. Given three instances of RSA with a common modulus N and private exponents each smaller than $N^{0.33}$, the attack can factor the modulus about 93% of the time in practice. The success rate of the attack can be increased up to almost 100% by including a relatively small exhaustive search. Next, we consider Howgrave-Graham and Seifert's lattice-based attack and show that a second necessary condition for the attack exists that limits the bounds (beyond the original bounds) once $n \geq 7$ instances of RSA are used. In particular, by construction, the attack is limited to private exponents at most $N^{0.5-\epsilon}$, given sufficiently many instances, instead of the original bound of $N^{1-\epsilon}$.

In addition, we also consider the effectiveness of the attacks when mounted against multi-prime RSA and Takagi's variant of RSA. For multi-prime RSA, we show three (or more) instances with a common modulus and private exponents smaller than $N^{1/3-\epsilon}$ is unsafe. For Takagi's scheme, we show that three or more instances with a common modulus $N = p^t q$ is unsafe when all the private exponents are smaller than $N^{2/(3(t+1))-\epsilon}$. The results, for both variants, is obtained using Guo's method and are almost always successful with the inclusion of a small exhaustive search. When only two instances are available, Howgrave-Graham and Seifert's attack can be successfully mounted on multi-prime RSA, with r primes in the modulus, when the private exponents are both smaller than $N^{(3+r)/7r-\epsilon}$.

Keywords. RSA, common modulus, multi-prime RSA, Takagi's scheme, small private exponent, lattice reduction, continued fractions.

2010 Mathematics Subject Classification. 94A60, 11T71.

1 Introduction

The RSA cryptosystem [20] is the most widely known and widely used public-key cryptosystem in the world today. It is well known, however, that RSA is insecure when the private exponent is too small. Wiener's famous continued fraction attack [24] can be used to efficiently factor the modulus when the private exponent is smaller than $N^{1/4-\epsilon}$, where N is the RSA modulus, and Boneh and Durfee's lattice-based attack [2] shows that private exponents up to $N^{0.2929-\epsilon}$ should be

considered unsafe also. The latter result is an asymptotic bound, but experiments have broken instances of RSA with private exponents up to about $N^{0.28}$ and recent work [21] shows that private exponents up to $N^{0.3}$ are vulnerable (for 1024-bit N) given some exhaustive search.

The bounds on the private exponent can be increased considerably when there are two or more instances RSA, having the same modulus, with small private exponents. An unpublished attack by Guo (described in [9]) can be used to efficiently factor the (common) modulus, with some non-negligible probability, when three instances of RSA, each with private exponents smaller than $N^{1/3-\epsilon}$, are given. A stronger attack, by Howgrave-Graham and Seifert [9], can be used when given two or more instances of RSA with a common modulus. For example, the attack works for private exponents up to $N^{0.357-\epsilon}$, given only two instances and for private exponents up to $N^{0.4-\epsilon}$ given three instances. Howgrave-Graham and Seifert's attack is a heuristic lattice-based attack, relying on an assumption about the lattices used, but is observed to work well in practice.

In this work we re-examine both Guo's and Howgrave-Graham and Seifert's common modulus attacks. Our original intent was simply to extend these attacks to multi-prime RSA and Takagi's scheme, however, in doing so, we also made some interesting observations about the original attacks on RSA. First, we noticed that Guo's attack is expected to be much more successful than originally suggested. This follows by using all of the information available and including some small exhaustive searches. This improvement is shown to hold in practice as well. Next, we observe that the bounds in Howgrave-Graham and Seifert's attack are overly optimistic when there are more than six instances of RSA with a common modulus. In particular, we show that there is a second necessary condition that limits the attack when there are many instances. Finally, and achieving our original goal, we show that multi-prime RSA and Takagi's scheme are vulnerable to common modulus attacks as well. Somewhat surprising, we find that Guo's attack works much better for multi-prime RSA (given three or more instances) than Howgrave-Graham and Seifert's attack which is opposite to the case of RSA. In addition, the strength of the attack on multi-prime RSA (i.e., the bounds on the private exponent) are the same as that for RSA. This is in contrast to all known attacks on multi-prime RSA (except for factoring). Also, we find that only Guo's attack can be used to attack Takagi's scheme. The strength of the attack (in theory and practice) depend on the structure of the modulus and decrease with increasing multiplicity of the prime p.

1.1 Motivation

When using RSA for both an encryption scheme and for a digital signature scheme, it is suggested by Ferguson and Schneier [6, §13.4.2] that instead of having two

sets of keys[1], each with a different modulus, that a single modulus should be used for the two (distinct) sets of keys. In this way, only two primes need to be generated and only one modulus needs to be stored for both keys. The same idea can be generalized to (arbitrary) multiple instances of RSA with a common modulus when a user requires several encryption (or signature) schemes. For example, a user may wish to have both professional and personal encryption and signature schemes; four instances of RSA in total.

Based on the best known small private exponent attacks on (single instance) RSA, a user may be tempted to use private exponents that are immune to these attacks but still relatively small in order to minimize decryption (signature generation) times. Howgrave-Graham and Seifert have shown, however, that private exponents exceeding the bounds by Boneh and Durfee ($N^{0.292}$) are insecure when two or more instances share a common modulus. In fact, they have shown that private exponents up to $N^{0.357}$ are unsafe when there are two instances sharing a common modulus (and stronger results for more instances).

There has been no consideration of the common modulus setting for multi-prime RSA or for Takagi's scheme though. In order to better understand the security of these variants of RSA it is important to consider all possible attacks (just as with RSA).

1.2 Related work

This work is directly based on Guo's continued fraction attack and Howgrave-Graham and Seifert's lattice based attack on common modulus RSA as described in [9]. Common modulus attacks have not been, to our knowledge, considered in the context of variants of RSA before.

There are some earlier common modulus attacks on RSA, by Simmons [22] and DeLaurentis [5], but these attacks apply to the so-called common modulus protocol. In this early protocol many users share the same modulus and each user is not supposed to know the factorization. Since any user with a valid private exponent can compute the factorization of the modulus, however, this protocol is completely insecure. Here, we consider a single user who has two or more instances of RSA with a common modulus.

1.3 Contributions

The contributions of our work, in brief, are as follows.

(i) We show that Guo's continued fraction attack is much more effective in practice than previously thought.

[1] A single key pair should never be used for more than one specific purpose. For example, a single key should not be used for encryption and digital signatures.

(ii) We show that Guo's attack can be mounted on multi-prime (with the same strength as for RSA) and on Takagi's scheme with reduced bounds (depending on the form of the modulus).

(iii) We show that a second necessary condition for Howgrave-Graham and Seifert's lattice-based attack exists, which limits the strength of the attack for $n \geq 7$ instances of RSA to private exponents smaller than $N^{1/2}$.

(iv) We show that Howgrave-Graham and Seifert's attack can be mounted on multi-prime RSA (but not Takagi's scheme), but that once there are three instances of multi-prime RSA, Guo's attack is stronger.

1.4 Outline

The rest of the paper is as follows. In Section 2, we review the RSA cryptosystem and as well as the two fast variants we consider (multi-prime and Takagi's scheme). The tools needed for the attacks (continued fractions and lattices) are briefly reviewed in Section 3. We use Wiener's attack as an example to illustrate both techniques. In Section 4, we review Guo's continued fraction attack and present experimental data to show the effectiveness of the attack. In addition, we mount the attack on multi-prime RSA and Takagi's scheme. In Section 5, we review Howgrave-Graham and Seifert's lattice-based attack. We show that a secondary necessary condition for the attack exists that limits the effectiveness of the attack for $n \geq 7$ instances of RSA. We also show the effectiveness of the attack when mounted on multi-prime RSA. Finally, we conclude with Section 6.

2 RSA and some fast variants

The RSA cryptosystem [20] is the most widely known and most widely used public key cryptosystem in the world. Let $N = pq$ be the product of two large (distinct) primes and let e and d be inverses modulo $\lambda(N) = \text{lcm}(p-1, q-1)$. Thus, e and d satisfy the RSA key equation

$$ed = 1 + k\lambda(N),$$

where k is some positive integer. From this equation, notice that e and d are also inverses modulo the constant k so that $\gcd(d, k) = 1$. The value N is called the RSA modulus (or modulus for short), e is the public (encrypting) exponent and d is the private (decrypting) exponent. The public key is given by (e, N) and the private key is given by (d, p, q). The exponents can actually be defined as inverses modulo any multiple of $\lambda(N)$. In fact, $\phi(N) = (p-1)(q-1) = \gcd(p-1, q-1)\lambda(N)$ is the value used in the original presentation of RSA [20]

and is often used in the presentation of many attacks. The reason that $\phi(N)$ is desirable, from the point of view of an attacker, is that

$$\phi(N) = (p-1)(q-1) = N - p - q + 1,$$

can be approximated as N minus a small correction term ($s = p + q - 1$). When the primes are balanced, that is $1/2 < p/q < 2$, we then have

$$|s| = |N - \phi(N)| = |p + q - 1| < 3N^{1/2},$$

and so N is a good approximation of $\phi(N)$. We will only consider RSA with balanced primes.

Given a plaintext message $m \in \mathbb{Z}_N$, the ciphertext is computed as $c = m^e \bmod N$ and plaintext is recovered since $c^d \bmod N = m^{ed} \bmod N = m$. Thus encryption is simply a modular exponentiation of the plaintext with exponent e and decryption is a modular exponentiation of the ciphertext with exponent d. We will refer to decryption in this manner as standard decryption. To speed up decryption, however, we can first compute partial decryptions modulo p and modulo q and then combine them with the Chinese remainder theorem to recover the plaintext (see [19]). In particular, letting $d_p = d \bmod p - 1$ and $d_q = d \bmod q - 1$, we can first compute $m_p = c^{d_p} \bmod p$ and $m_q = c^{d_q} \bmod q$ and then combine m_p and m_q, using the Chinese remainder theorem, to recover the plaintext m. We will refer to this type of decryption as CRT-decryption and the exponents d_p and d_q as CRT-exponents.

Using simple quadratic complexity modular arithmetic and the square-and-multiply method for modular exponentiation, the expected number of binary operations for standard decryption is expected to be $T_{\text{RSA}} = \frac{3}{2}\log_2(d)\log_2^2(N)$. Here, we also assume that the binary representation of d has roughly an equal number of ones and zeros. When the private exponent is smaller than each of the primes (roughly $N^{1/2}$), it follows that $d_p = d_q = d$, and the expected number of binary operations for CRT-decryption, is reduced to

$$T_{\text{CRT}} = 2\,\frac{3}{2}\log_2(d)\,\log_2^2(p) = 2\,\frac{3}{2}\log_2(d)\,\frac{1}{4}\log_2^2(N) = \frac{1}{2}\,T_{\text{RSA}},$$

where the time for exponentiations dominate the time and we ignore the cost for the initial reductions and final combining stages. The runtime can be reduced by another factor of two if we assume parallel computations. Thus, in theory, using CRT-decryption gives a decrease in decryption time by a factor of four.

In a typical instance of RSA with a small exponent the public exponent is expected to be roughly the same size as $\lambda(N)$. For randomly chosen balanced primes, it is expected that $\lambda(N) \approx \phi(N)$ and so $e \approx N$. We will assume that all

public exponents for RSA (with small private exponent) satisfies this approximation.

The strongest known small private exponent attack on (single instance) RSA is Boneh and Durfee's lattice-based attack [2]. The attack shows that private exponents smaller than $N^{0.2929-\epsilon}$ should be considered insecure. In practice, this bound can be increased with an additional exhaustive search. It was shown by Sarkar, Maitra and Sarkar [21], for example, that private exponents up to $N^{0.3}$ can feasibly be recovered.

2.1 Multi-prime RSA

Multi-prime RSA is a variant of RSA in which the modulus is the product of three or more (distinct) primes. When the modulus is the product of r primes, $N = p_1 \cdots p_r$, we call the system r-prime RSA. The public and private exponents are defined as inverses modulo $\phi(N) = (p_1 - 1) \cdots (p_r - 1)$. Encryption and standard decryption are exactly the same as with RSA (modular exponentiation with exponent e for encryption and d for decryption). CRT-decryption is a simple generalization of RSA. We compute the partial decryption modulo each of the r primes and then combine to recover the plaintext using the Chinese remainder theorem. For a fixed modulus size (bitlength), larger r implies smaller primes since each prime is roughly $N^{1/r}$ when the primes are balanced. For private exponents smaller than each of the primes the expected number of binary operations for CRT-decryption is given by

$$T_{\text{CRT}}^{r\text{-prime}} = r\,\frac{3}{2}\log_2(d)\,\log_2^2(p_i) = r\,\frac{3}{2}\log_2(d)\,\frac{1}{r^2}\log_2^2(N) = \frac{1}{r}\,T_{\text{RSA}}.$$

Here the runtime can be reduced by another factor of r if we assume parallel computations. Thus, in theory, CRT-decryption with r-prime RSA should give a decrease in decryption time by a factor of r^2 compared to standard decryption (using the same assumptions as above for RSA). Of course, using too many primes in the modulus makes the elliptic curve method for factoring more efficient so a trade-off must be made. Balancing the expected complexity of factoring the modulus with the number field sieve and the elliptic curve method, the suggested maximum number of primes for several common modulus bitlengths are given in the following.

Modulus size (bits)	1024	2048	4096	8192
Maximum number of primes	3	3	4	5

As soon as more than this number of primes are in the modulus, the elliptic curve method is expected to factor the modulus faster than the number field sieve. For more details, see Lenstra [12].

When all the primes in the modulus are pairwise balanced, which we will assume is always true, it can be shown (see [8]) that

$$|s| = |N - \phi(N)| = \left| \sum_i \frac{N}{p_i} - \sum_{i \neq j} \frac{N}{p_i p_j} + \cdots + (-1)^r \right| < (2r-1)N^{1-1/r}.$$

This value will be needed when extending Howgrave-Graham and Seifert's attack to multi-prime RSA.

In a typical instance of r-prime RSA with a small private exponent, the public exponent is expected to be roughly the same size as $\phi(N)$. Again, since $\phi(N) \approx N$, we will use the approximation that $e \approx N$ for all public exponents for r-prime RSA (with small private exponent).

The strongest known small private exponent attack on (single instance) r-prime RSA is, again, Boneh and Durfee's lattice-based attack (as applied to multi-prime RSA). The extension of the attack to multi-prime RSA, by Ciet et al. [3], shows that private exponents smaller than $N^{1-\sqrt{1-1/r}}$ should be considered insecure. For example, private exponents smaller than about $N^{0.1835}$ should be considered unsafe for 3-prime RSA, while private exponents smaller than about $N^{0.134}$ should be considered unsafe for 4-prime RSA. This trend, that attacks become weaker with increasing number of primes in the modulus, is common to all known attacks on multi-prime RSA except for factoring with the elliptic curve method (see Hinek [7] for more details about attacks on multi-prime RSA) and, as we shall see below, Guo's common modulus attack.

2.2 Takagi's scheme

Takagi's scheme [23] is another variant of RSA in which decryption costs are reduced. In this scheme, however, decryption is different from RSA (and even standard decryption does not apply). Here, the modulus has the form $N = p^t q$, for some positive integer $t > 1$, and the public and private exponents are defined modulo $\lambda'(N) = \text{lcm}(p-1, q-1)$. Notice that $\lambda'(N)$ is not a multiple of $\lambda(N) = p^{t-1}\text{lcm}(p-1, q-1)$. Encryption is the same as for RSA ($c = m^e \mod N$). For decryption, however, we first compute $m_p = c^{d_p} \mod p$. Using Hensel lifting, we then lift m_p (which is a partial decryption modulo p) to a partial solution modulo p^t. This is then combined with $m_q = c^{d_q} \mod q$ with the Chinese remainder theorem to recover the plaintext m. See Takagi [23] for full details.

The complexity of the Hensel lifting is dominated by the modular exponentiations, so (just considering the exponentiations), the expected number of binary

operations is

$$T_{\text{Takagi}} = 2\frac{3}{2}\log_2(d)\,\log_2^2(p)$$
$$= 2\frac{3}{2}\log_2(d)\,\frac{1}{(t+1)^2}\,\log_2^2(N) = \frac{2}{(t+1)^2}\,T_{\text{RSA}}.$$

Thus, when computing sequentially, decryption is faster than multi-prime RSA. In practice, Takagi has observed that decryption time for $t = 2$ with a 1024-modulus is about 42% faster than 3-prime RSA with a 1024-bit modulus. Just as with multi-prime RSA, the size of t must be balanced so that the modulus is not easily factored. The suggested maximum size of t is given by the suggested maximum size of r for multi-prime RSA letting $t + 1 = r$. If we assume parallel computations the decryption time will be essentially the same (when matching $r = t + 1$).

In a typical instance of Takagi's scheme with a small private exponent, the public exponent is expected to be roughly the same size as $\lambda'(N) = \text{lcm}(p-1, q-1)$. For randomly generated balanced primes, it is expected that $\text{lcm}(p-1, q-1) \approx pq \approx N^{2/(t+1)}$. We will use this approximation for all instances of Takagi's scheme with small private exponents.

The strongest known small private exponent attack on (a single instance of) Takagi's scheme is, yet again, a generalization of Boneh and Durfee's lattice-based attack on RSA. The generalization, due to Itoh, Kunihiro and Kurosawa [10], shows that private exponents smaller than $N^{(2-\sqrt{2})/(t+1)}$ should be considered insecure. For example, private exponents smaller than about $N^{0.1953}$ should be considered unsafe for moduli $N = p^2q$, while private exponents smaller than about $N^{0.1464}$ should be considered unsafe for moduli $N = p^3q$.

3 Continued fractions, lattices and Wiener's attack

In this section we review some of the mathematical results needed for the attacks. We assume the reader already has some familiarity with the topics and only review the needed results. To illustrate each topic (continued fractions and lattices) we briefly outline Wiener's small private exponent attack as implemented with each topic.

3.1 Continued fractions

We need only one main result from the theory of continued fractions (for more general information see Olds [18]). The result is restated in the following theorem.

Theorem 3.1 (Continued-Fractions). *Let a, b, c and d be integers satisfying*

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

where a/b and c/d are in lowest terms (i.e., $\gcd(a, b) = \gcd(c, d) = 1$*). Then c/d is one of the convergents in the continued fraction expansion of a/b. Further, the continued fraction expansion of a/b is finite with the total number of convergents being polynomial in* $\log(b)$.

Using this result, we review Wiener's small private exponent attack on RSA [24] (using Boneh's [1] approach). In order to simplify the presentation we will assume that the public and private exponents are defined modulo $\phi(N)$ instead of modulo $\lambda(N)$ as in Wiener's original work. Let (e, N) be an instance of RSA with balanced primes and let $d < \frac{1}{6}N^{1/4}$ be its corresponding private exponent. We start by substituting $\phi(N) = N - p - q + 1$ into the key equation giving

$$ed = 1 + k(N - p - q + 1),$$

and then dividing both sides by dN (and rearranging) to yield

$$\frac{e}{N} - \frac{k}{d} = \frac{1}{dN} - \frac{k(p + q - 1)}{dN}.$$

Since $|k| < |d| < \frac{1}{6}N^{1/4}$, and $|p + q - 1| < 3N^{1/2}$, notice that

$$\left|\frac{e}{N} - \frac{k}{d}\right| = \left|\frac{1}{dN} - \frac{k(p + q - 1)}{dN}\right| < \left|\frac{k(p + q - 1)}{dN}\right| < \frac{1}{2d^2}.$$

Therefore, from Theorem 3.1, it follows that $c = k/d$ is one of the convergents in the continued fraction expansion of e/N. Finding this convergent exposes $\phi(N)$ since $1/k < 1$ and

$$\left\lfloor \frac{e}{c} \right\rfloor = \left\lfloor \frac{ed}{k} \right\rfloor = \left\lfloor \frac{1}{k} + \phi(N) \right\rfloor = \phi(N).$$

Once $\phi(N) = (p - 1)(q - 1)$ is known the modulus is easily factored by solving the system $\phi(N) = (p-1)(q-1)$ and $N = pq$. Since we don't know the correct convergent, we can simply try each one (computing a candidate for $\phi(N)$) until the modulus is factored. Since the number of convergents is polynomial in $\log_2(N)$ and all computations can be done in time polynomial in $\log_2(N)$, it follows that when the private exponent is smaller than $\frac{1}{6}N^{1/4}$ the modulus can be factored in time polynomial in $\log_2(N)$.

Since the attack is guaranteed to work when $d < \frac{1}{6}N^{1/4}$, we know that the correct convergent in continued fraction expansion of e/N, call it c, should satisfy $|e/n - c| < 1/(2d^2) < 18N^{-1/2}$. Therefore, only the convergents satisfying this bound (or close to it) need be tested. This allows us to quickly eliminate many candidates.

3.2 Lattices

A lattice $\mathcal{L}$ is a discrete additive subgroup of $\mathbb{R}^n$. Given $m \leq n$ linearly independent vectors $b_1, \ldots, b_m \in \mathbb{R}^n$, the set $\mathcal{L} = \{\sum_{i=1}^{m} \alpha_i b_i \mid \alpha_i \in \mathbb{Z}\}$ is a lattice. The b_i are called basis vectors and $\mathcal{L}$ is the lattice generated by these basis vectors. Thus, $\mathcal{L}$ is the set of all integer linear combinations of the basis vectors. In addition, the volume of a lattice $\text{vol}(\mathcal{L})$ is the volume of the m-dimensional parallelepiped spanned by the b_i. The volume of a lattice is basis invariant (that is, it is a constant of the lattice). When $m = n$, the lattice is full rank and we can compute the volume as $\text{vol}(\mathcal{L}) = |\det(\mathcal{B})|$, where $\mathcal{B}$ is the matrix whose rows are the basis vectors.

The main result that we need for lattices gives a bound on the size of smallest vectors in a lattice (a non-zero smallest vector must exists since lattices are discrete). The result, due to Minkowski, is given in the following theorem. We use $\|x\|$ to denote the usual Euclidean norm of the vector x.

Theorem 3.2 (Minkowski). *Let $\mathcal{L}$ be an n-dimensional lattice with volume $vol(\mathcal{L})$. A smallest vector v in $\mathcal{L}$ satisfies*

$$\|v\| \leq \sqrt{n} \cdot vol(\mathcal{L})^{1/n}.$$

Using Theorem 3.2, we have a necessary condition for any vector x to be a smallest vector in a lattice. Notice that if x is a smallest vector in $\mathcal{L}$ then so is $-x$. To simplify the discussion, if $\pm x$ are the only two smallest vectors in $\mathcal{L}$ we will simply say that x is the smallest vector in $\mathcal{L}$.

We now briefly describe Wiener's attack as a heuristic lattice-based attack. Again, let (e, N) be a valid public key with corresponding private exponent $d = N^{\delta}$. Letting $s = N - \phi(N)$, we begin by writing the key equation $ed = 1 + k(N - s)$ and the trivial equation $N^{0.5} = N^{0.5}$ as the vector-matrix equation

$$(k, d)\begin{bmatrix} N^{0.5} & -N \\ 0 & e \end{bmatrix} = (kN^{0.5}, 1 - ks).$$

Therefore, $v = (kN^{0.5}, 1 - ks)$ is an integer linear combination of the rows of the matrix $\mathcal{B}$. Letting $\mathcal{L}$ be the lattice generated by the rows of $\mathcal{B}$ we then know that

v is a vector in the lattice. Since the volume of the lattice is $\text{vol}(\mathcal{L}) = |\det(\mathcal{B})| = eN^{0.5} \approx N^{3/2}$, we know from Minkowski's theorem that a smallest vector in $\mathcal{L}$ must be bounded by $\sqrt{2}\ \text{vol}(\mathcal{L})^{1/2} \approx \sqrt{2}N^{3/4}$. Therefore, if the vector $v = (kN^{\delta}, 1-ks)$, which has size $\|v\| \approx N^{\delta+1/2}$, is a smallest vector in $\mathcal{L}$, it follows that $\delta + 1/2 < 3/4$ or more simply

$$\delta < \frac{1}{4} - \epsilon,$$

where $\epsilon > 0$ has been added to correct for small constants (that do not depend on N) that were ignored (for $\text{vol}(\mathcal{L})$ and $\|v\|$). Thus, when $d < N^{1/4}$, the vector v may be a smallest vector. If v is the smallest vector, then finding v will allow us to factor the modulus. Since $x\mathcal{B} = v$, we can solve for x which reveals d and k. Thus, we can compute $\phi(N) = (ed-1)/k$ and factor the modulus.

In order for this attack to succeed, we rely on the following assumption[2].

Assumption 3.3 (Small Vectors). For the lattices used here, a vector $v \in \mathcal{L}$ that satisfies Minkowski's bound (from Theorem 3.2) is likely a smallest vector in $\mathcal{L}$.

When this assumption holds for the given lattice, we can then use the above method to factor the modulus. Even if the assumption only holds a non-negligible fraction of the time, the attack is still a success.

Of course, we also need to be able to find the small vector v in the lattice. For this particular example, with a 2-dimensional lattice, a smallest vector in the lattice $\mathcal{L}$ can be found efficiently. In general though, we will be interested in finding a smallest vector in an n-dimensional (full rank) lattice $\mathcal{L}'$. In this case, the approach is to compute an LLL-reduced basis for the lattice which guarantees to contain a basis vector x' satisfying

$$\|x'\| \leq 2^{(n-1)/4}\text{vol}(\mathcal{L}')^{1/n}.$$

Even though the size of this small vector does not necessarily satisfy Minkowski's bound in Theorem 3.2, which means that x' is not necessarily a smallest vector, it is often the case in practice that this vector is a smallest vector in the lattice.

An LLL-reduced basis for an n-dimensional full rank lattice $\mathcal{L}$ can be computed with Nguyen and Stehlé's L^2 algorithm (see [17]) in time

$$O(n^3(n + \log B)\log B \cdot \mathcal{M}(n)),$$

where $\mathcal{M}(x)$ is the time needed to multiply two x-bit integers, and B is the size of the largest component in the original basis matrix.

[2] There is also a provable lattice-based version of Wiener's attack by May [13], but the attacks shown later are based on this heuristic attack.

3.3 Breaking RSA and its variants

We will consider an instance of RSA (or multi-prime RSA or Takagi's scheme) to be broken when the factorization of the modulus is known. There are several ways in which this can be accomplished.

First, given a multiple of $\lambda(N)$ (or $\phi(N)$) the factorization can be computed in probabilistic polynomial time using the results of Miller [15]. Miller's result is much more general. It also applies to a multiple of $\phi(N)$ for multi-prime RSA and for a multiple of $\mathrm{lcm}(p-1, q-1)$ for Takagi's scheme. If the exact value of $(p-1)(q-1)$ is known for RSA or for Takagi's variant, we can deterministically factor the modulus since $L = (p-1)(q-1)$ and $N = p^k q$ ($k = 1$ for RSA) give a system of two equations with two unknowns which we can easily solve.

Next, given the private exponent d of an instance of RSA, we can factor the modulus since $ed - 1 = k\lambda(N)$ reveals a multiple of $\lambda(N)$. Similarly, this also holds for multi-prime RSA and for Takagi's scheme (since with d known we can compute $ed - 1$ which is a multiple of $\mathrm{lcm}(p-1, q-1)$). For RSA and Takagi's scheme, there are also deterministic methods that can factor the modulus given the public key and the private exponent. See [14] for RSA and [11] for Takagi's scheme.

Finally, given the constant k in the RSA key equation we can also factor the modulus (assuming that the public exponent is roughly the same size as the modulus). Assuming that we know $g = \gcd(p-1, q-1)$, or more simply assuming that the exponents are defined modulo $\phi(N)$, we can compute $s = N - \phi(N) = p + q - 1$ given only e, N and k. Notice that reducing the key equation $ed = 1 + k(N - s)$ modulo the public exponent e yields

$$0 \equiv 1 + k(N - s) \pmod{e},$$

where s is the only unknown. Rearranging, we can compute s since

$$s \equiv N + k^{-1} \pmod{e},$$

where the inverse is well defined (as k and e must be relatively prime). Since the public exponent satisfies $e \approx N \gg s$, the value $(N + k^{-1}) \bmod e$ yields s. With s known, we also know $\phi(N) = N - s$ and can easily factor the modulus. This also holds for multi-prime RSA but not for Takagi's scheme.

4 Guo's common modulus attack

In [9], Howgrave-Graham and Seifert describe an unpublished attack by Guo[3] on common modulus RSA with small private exponents. Consider two instances of

[3] G. C. R. Guo, "An application of Diophantine approximation in computer security".

RSA with a common modulus N and key equations

$$e_1 d_1 = 1 + k_1 \lambda(N),$$
$$e_2 d_2 = 1 + k_2 \lambda(N).$$

Guo's main observation is that these equations can be combined to remove $\lambda(N)$. Indeed, multiplying the first equation by k_2, the second equation by k_1 and taking the difference yields

$$k_2 e_1 d_1 - k_1 e_2 d_2 = k_2 - k_1,$$

where all the unknowns are relatively small (when the private exponent is small). With this equation as a starting point, the attack then proceeds in a similar way as Wiener's continued fraction attack. Notice that dividing both sides of this equation by $e_2 k_2 d_1$ yields

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

which combined with Theorem 3.1, suggests that $k_1 d_2 / k_2 d_1$, in lowest terms, can be obtained from the continued fraction expansion of e_1/e_2 when the right-hand side of $(k_2 - k_1)/(e_2 k_2 d_1)$ is small enough. In particular, a sufficient condition for Theorem 3.1 to apply is given by

$$\left| \frac{k_2 - k_1}{e_2 k_2 k d_1} \right| < \frac{1}{2(k_2 d_1)^2}, \tag{4.1}$$

or more simply

$$d_1 < \frac{e_2}{2k_2 \, |k_2 - k_1|}. \tag{4.2}$$

When both private exponents are bounded by N^δ (i.e., $0 < d_1, d_2 < N^\delta$) and the public exponents are roughly the same size as the modulus (i.e., $e_1, e_2 \approx N$), it follows from the key equation that $0 < k_i < d_i < N^\delta$. Using these bounds for the k_i, d_i and e_i, a sufficient condition for inequality (4.2) to be satisfied is given by

$$\delta < \frac{1}{3} - \log_N(2) = \frac{1}{3} - \epsilon,$$

where $\epsilon > 0$ is small and decreases with increasing N. Therefore, when this inequality is satisfied, we know that we can compute $k_1 d_2 / k_2 d_1$ in lowest terms. However, knowing $k_1 d_2 / k_2 d_1$, in lowest terms, does not allow us to break either instance of RSA. Two problems are:

(i) Any common factors of $k_1 d_2$ and $k_2 d_1$ will be removed from the numerator and denominator of convergent $k_1 d_2 / k_2 d_1$.

(ii) Knowing $k_1 d_2$ or $k_2 d_1$ (or both) does not seem to help in factoring the modulus (or determining d_1 or d_2) without factoring $k_1 d_2$ or $k_2 d_1$.

As discussed in [9], Guo considers these two problems and offers the following solutions. The first problem is avoided by simply assuming that there are no common factors in the numerator and denominator of the correct convergent. That is, it is assumed that $\gcd(k_1 d_2, k_2 d_1) = 1$. Assuming that the d_i and k_i behave like randomly chosen numbers it is estimated that this will occur with probability $6/\pi^2 \simeq 0.61$.

For the second problem two solutions are given. Assuming that the above condition for common factors holds (i.e., $\gcd(k_1 d_2, k_2 d_1) = 1$), the solution offered are

• One could try to factor $k_1 d_2$ to determine d_2 (or factor $k_2 d_1$ to obtain d_1). With the bound $\delta < 1/3 - \epsilon$, the number $k_1 d_2$ is no larger than $N^{2/3}$ and is not expected to be of a difficult factorization shape. This is feasible for instances of RSA with a 1024-bit modulus but completely out of reach for instances with a 2048-bit modulus using the current state of the art in factoring techniques.

• As a second solution, and the one we will focus on, one can assume that a third instance of RSA with a small private exponent and the same modulus is available. Here, the continued fraction technique is repeated with a different pair of RSA instances. Assuming that all the k_i and d_i are pairwise relatively prime, one can determine $k_1 d_2$ from the continued fraction expansion of e_1/e_2 and $k_3 d_2$ from the continued fraction expansion of e_2/e_3, for example, which can then be used to compute

$$\gcd(k_1 d_2, k_3 d_2) = d_2 \gcd(k_1, k_3) = d_2.$$

With d_2 known, the modulus can be factored (see [4]). This method is expected to recover d_2 with probability about $(6/\pi^2)^3 \simeq 0.22$, under the assumption that the k_i and d_i behave as random numbers.

Once d_2 is known, we can factor the modulus since we know a multiple of $\phi(N)$. That is, $e_2 d_2 - 1 = k_2 \phi(N)$ is known. Thus, given three instances of RSA with a common modulus and private exponents each smaller than $N^{1/3-\epsilon}$, it is expected that Guo's attack will be successful with probability approximately 0.22. A sufficient condition for the attack to succeed is that the pairs $(k_1 d_2, k_2 d_1)$, $(k_2 d_3, k_3 d_2)$ and (k_1, k_3) are each relatively prime (and assuming all quantities are random integers this is expected to happen with probability about 0.22).

4.1 Guo's attack in practice

The success rate of Guo's attack, in practice, is actually much higher than the theoretical probability of 0.22, as described above. Looking at the attack more carefully, we derive a new sufficient condition for the attack to succeed. To this end, we consider the values used in the attack. From the continued fraction expansion of e_1/e_2, we compute the convergent $c_{12} = k_1 d_2 / k_2 d_1$ and from the continued fraction expansion of e_2/e_3, we compute the convergent $c_{23} = k_2 d_3 / k_3 d_2$. In trying to isolate the private exponent d_2, we use the numerator of c_{12} and the denominator of c_{23}, which are given by

$$\begin{aligned}
\operatorname{numer}(c_{12}) &= \frac{k_1 d_2}{\gcd(k_1 d_2, k_2 d_1)} \\
&= \frac{k_1 d_2}{\gcd(k_1, k_2)\gcd(d_1, d_2)} = \frac{k_1}{\gcd(k_1, k_2)} \frac{d_2}{\gcd(d_1, d_2)} \\
\operatorname{denom}(c_{23}) &= \frac{k_3 d_2}{\gcd(k_2 d_3, k_3 d_2)} \\
&= \frac{k_3 d_2}{\gcd(k_2, k_3)\gcd(d_2, d_3)} = \frac{k_3}{\gcd(k_2, k_3)} \frac{d_2}{\gcd(d_2, d_3)},
\end{aligned}$$

where the gcds in the denominators split because $\gcd(k_i, d_i) = 1$ for $i = 1, 2, 3$. (This follows since $e_i d_i = 1 + k_i \phi(N)$ for each i and so e_i and d_i are also inverses modulo k_i.) The candidate D_2 for d_2 is then computed as

$$\begin{aligned}
D_2 &= \gcd(\operatorname{numer}(c_{12}), \operatorname{denom}(c_{23})) \\
&= \gcd\left(\frac{k_1}{\gcd(k_1, k_2)} \frac{d_2}{\gcd(d_1, d_2)}, \frac{k_3}{\gcd(k_2, k_3)} \frac{d_2}{\gcd(d_2, d_3)} \right).
\end{aligned}$$

We can simplify this further by separating the common parts of d_2 with the remaining parts as

$$\begin{aligned}
D_2 &= \gcd\left(\frac{k_1}{\gcd(k_1, k_2)} \frac{d_2}{\gcd(d_1, d_2)}, \frac{k_3}{\gcd(k_2, k_3)} \frac{d_2}{\gcd(d_2, d_3)} \right), \\
&= \underbrace{\frac{d_2}{\operatorname{lcm}(\gcd(d_1, d_2), \gcd(d_2, d_3))}}_{d_2/d_2'} \underbrace{\gcd\left(\frac{k_1 d_{12}}{\gcd(k_1, k_2)}, \frac{k_3 d_{23}}{\gcd(k_2, k_3)} \right)}_{k_{13}'}, \\
&= d_2 \frac{k_{13}'}{d_2'},
\end{aligned}$$

where d_{12} and d_{23} are introduced to simplify the notation. Here d_2/d_2' is the common factors of d_2 and k_{13}' is the remaining part of the gcd.

The attack is successful (i.e., the computation reveals d_2) whenever $d_2' = k_{13}'$, thus a new sufficient condition is given by

$$d_2' = \operatorname{lcm}(\gcd(d_1, d_2), \gcd(d_2, d_3)) = \gcd\left(\frac{k_1 d_{12}}{\gcd(k_1, k_2)}, \frac{k_3 d_{23}}{\gcd(k_2, k_3)}\right) = k_{13}'. \tag{4.3}$$

In practice, we find that this condition is satisfied about 63% of the time. In particular, running 260,000 experiments (10,000 trials for 26 different values of $0.25 \le \delta \le 0.5$), we find that this condition is met (and the attack succeeds) with probability 0.629 ± 0.004 (where the 0.004 is one standard deviation) for 1024-bit moduli.

Guo's attack, as described above, however, does not use all the information available to it. We can improve the likelihood of success in two ways. First, notice that computing the continued fraction expansion of e_1/e_2, e_2/e_3 and e_3/e_1, we can obtain the convergents $c_{12} = k_1 d_2/k_2 d_1$, $c_{23} = k_2 d_3/k_3 d_2$ and $c_{31} = k_3 d_1/k_1 d_3$. Using all possible combinations of numerators and denominators, each from a different convergent, we can compute candidates for each of d_1, d_2, d_3, k_1, k_2 and k_3. For example, a candidate for d_2 was computed (as shown above) using $\operatorname{numer}(c_{12})$ and $\operatorname{denom}(c_{23})$, and

$$\begin{aligned} K_2 &= \gcd(\operatorname{denom}(c_{12}), \operatorname{numer}(c_{23})) \\ &= \gcd\left(\frac{k_2 d_1}{\gcd(k_1 d_2, k_2 d_1)}, \frac{k_2 d_3}{\gcd(k_2 d_3, k_3 d_2)}\right), \end{aligned}$$

gives a candidate for k_2. In fact, for each (i, j, ℓ) that is a permutation of $(1, 2, 3)$, it is easily shown that $D_j = \gcd(\operatorname{numer}(c_{ij}), \operatorname{denom}(c_{j\ell}))$ gives a candidate for d_j while a candidate for k_j is given by $K_j = \gcd(\operatorname{denom}(c_{ij}), \operatorname{numer}(c_{j\ell}))$.

In each case the candidate will be a rational multiple of the correct value (just as in the d_2 example) with a similar sufficient condition as given in inequality (4.3). If any of the candidates are equal to the correct value the modulus can be easily factored. Given any d_i, the modulus can be factored since a multiple of $\phi(N)$ is known. Given any k_i, the modulus can be factored if we assume the public exponents are full sized. Thus, trying each of the six candidates will further increase the likelihood that the attack will succeed. In practice, we observe that the attack is successful about 93% of the time when the private exponents are smaller than $N^{0.33}$. The attack continues to work with decreasing likelihood for larger private exponent sizes until the private exponent is slightly larger then $N^{1/3}$.

In addition to computing the six candidates instead of only one, we can also exploit the form of the candidates. Letting D_i and K_i be the candidates for d_i and k_i, and using the example for D_2 from above as a guide for notation, notice that each candidate can be written as

$$D_i = d_i \frac{k'_{jk}}{d'_i} \quad \text{or} \quad K_i = k_i \frac{d'_{jk}}{k'_i},$$

for each (i, j, k) that is a permutation of $(1, 2, 3)$. Simply rewriting these, we have

$$d_i = D_i \frac{d'_i}{k'_{jk}} \quad \text{or} \quad k_i = K_i \frac{k'_i}{d'_{jk}}.$$

In practice all of the primed values (which are integers) are expected to be small. If none of the candidates are correct, we can perform a small exhaustive search to determine the primed values and hence reveal one of the correct d_i or k_i. If we assume that the primed values are each bounded by 2^ℓ, for some positive integer ℓ, then a search space of $2^{2\ell}$ must be explored for each candidate.

We illustrate the effectiveness of Guo's attack, with the modifications mentioned above, in Table 1. For various sizes of private exponents (δ), we show the frequency of success of Guo's attack for RSA with $1024, 2048, 4096$ and 10,000-bit modulus sizes. For each δ, we ran 10,000 experiments when the modulus was 1024 and 2048 bits, 1,000 experiments for 4096-bit moduli and 100 experiments for the instances with 10,000-bit moduli. Three random instances of RSA with a common modulus and private exponents each of size at most δ were generated for each experiment. The data shows the frequency that the attack was successful when only one candidate (d_2) is computed, when all six candidates (d_i,k_i) are computed and when a small exhaustive search is allowed for each of the six candidates (denoted by "Guess"). Instead of actually performing the exhaustive search, we considered the sizes of the numerator and denominator (in lowest terms) of d'_i/d_i and k'_i/k_i. If both the numerator and denominator, for any one of the candidates, were no larger than 2^{10}, we considered the attack a success. In all of the experiments that we conducted, this additional search (of 2^{10} for numerator and denominator) was sufficient to recover the desired values. The failures occurred when the convergents in the continued fractions did not yield $k_i d_j / k_j d_i$ at all (and hence revealed no information). The bold data indicates the theoretical limits of the attack (and will indicate this in all subsequent data shown). As can be seen, the attack works quite well in practice. For private exponents smaller than about $N^{0.332}$, the attack (with modest exhaustive search) was always successful. The success rate quickly deteriorated as the size of the private exponents approaches or just exceeds the $1/3$ bound though.

	1024-bit N			2048-bit N			4096-bit N			10,000-bit N		
δ	D_2	D_i, K_i	Guess	D_2	D_i, K_i	Guess	D_2	D_i, K_i	Guess	D_2	D_i, K_i	Guess
0.25000	0.6338	0.9356	1.0000	0.6217	0.9342	1.0000	0.610	0.930	1.000	0.63	0.94	1.00
0.26000	0.6241	0.9326	1.0000	0.6317	0.9345	1.0000	0.610	0.931	1.000	0.63	0.94	1.00
0.27000	0.6263	0.9357	1.0000	0.6347	0.9339	1.0000	0.632	0.933	1.000	0.66	0.96	1.00
0.28000	0.6343	0.9394	1.0000	0.6261	0.9343	1.0000	0.634	0.933	1.000	0.63	0.90	1.00
0.29000	0.6283	0.9363	1.0000	0.6281	0.9320	1.0000	0.666	0.949	1.000	0.61	0.94	1.00
0.30000	0.6274	0.9356	1.0000	0.6289	0.9398	1.0000	0.627	0.929	1.000	0.67	0.96	1.00
0.31000	0.6314	0.9386	1.0000	0.6199	0.9355	1.0000	0.631	0.939	1.000	0.73	0.94	1.00
0.32000	0.6284	0.9339	1.0000	0.6278	0.9380	1.0000	0.650	0.942	1.000	0.60	0.95	1.00
0.33000	0.6247	0.9371	1.0000	0.6336	0.9361	1.0000	0.639	0.930	1.000	0.56	0.96	1.00
0.33100	0.6327	0.9364	1.0000	0.6274	0.9361	1.0000	0.614	0.931	1.000	0.67	0.91	1.00
0.33200	0.6294	0.9381	1.0000	0.6256	0.9367	1.0000	0.644	0.941	1.000	0.62	0.95	1.00
0.33300	0.5444	0.8240	0.8837	0.5386	0.8257	0.8841	0.514	0.767	0.834	0.40	0.62	0.71
0.33330	0.3404	0.5328	0.5779	0.2139	0.3480	0.3831	0.082	0.157	0.179	0.01	0.01	0.01
0.33333	**0.3250**	**0.5076**	**0.5535**	**0.1878**	**0.3117**	**0.3435**	**0.078**	**0.142**	**0.153**	**0.02**	**0.02**	**0.02**
0.33400	0.0839	0.1460	0.1666	0.0129	0.0244	0.0299	0.000	0.000	0.000	0.00	0.00	0.00
0.33500	0.0095	0.0210	0.0289	0.0003	0.0004	0.0006						
0.33600	0.0011	0.0029	0.0050	0.0000	0.0000	0.0000						
0.33700	0.0002	0.0003	0.0006									
0.33800	0.0000	0.0002	0.0002									
0.33900	0.0000	0.0000	0.0000									

Table 1. Guo's attack: Observed success rate when using only one candidate (D_2) and when using all six candidates (D_i, K_i) for various sizes of private exponents and modulus sizes.

Notice that in the above discussion we did not consider the problem of actually finding the correct convergent. In fact, in all of the experiments for the data collected in Table 1, we identified the correct convergent using the associated private information (d_i and k_i) to save time. In practice, however, we will only be given the public keys. Nonetheless, we can still narrow the search of potentially correct convergents to a small number by finding a good starting point (good starting convergent). When looking for the correct convergent of e_1/e_2, for example, from the description of Guo's attack, we know from Theorem 3.1 that the correct convergent c will most likely satisfy

$$\left|\frac{e_1}{e_2} - c\right| < \frac{1}{2(k_2 d_1)^2}.$$

The theorem is a sufficient condition and hence we cannot rule out that the correct convergent might not satisfy the bound. Since the theoretical bound for the attack is $\delta < 1/3 - \epsilon$ and each $k_i < d_i$, we then also expect that

$$\left|\frac{e_1}{e_2} - c\right| < \frac{1}{2(N^{1/3}N^{1/3})^2} = \frac{1}{2N^{4/3}}.$$

Thus, when computing the convergents in the continued fraction expansion of e_i/e_j, we can ignore all of the initial convergents that do not satisfy this bound. In practice, we find that the convergent immediately preceding the first convergent that satisfies this bound is a candidate for the correct convergent (i.e., it often is the correct convergent). We will refer to this particular convergent as the *good starting convergent*.

In Table 2, we show that, in practice, using this good starting convergent as our first candidate for the correct convergent works quite well.

For each value of δ we show C_{ave}, $C_{\max}$ and $\#C$, where C_{ave} is the average distance (in absolute value) from the good starting convergent to the correct convergent taken over all successful trials (so $C_{\text{ave}} = 0$ corresponds to the good starting convergent being the correct convergent for each trial), $C_{\max}$ is the maximum distance (in absolute value) over all successful trials and $\#C$ is the (rounded) average number of total convergents for each continued fraction expansion. As can be seen, the good starting point was always correct for instances with private exponents $\delta \leq 0.31$ and this improves with increasing modulus size. For larger private exponent sizes some exhaustive search may be necessary. However, since the average distance is always less than 1.0, it is expected that only two convergents need to be tested for each continued fraction expansion. Since this additional (expected) complexity is so small, we simply assume that we can find the correct convergent (if it is present) with no extra costs.

	1024-bit N			2048-bit N			4096-bit N		
δ	C_{ave}	$C_{\max}$	$\#C$	C_{ave}	$C_{\max}$	$\#C$	C_{ave}	$C_{\max}$	$\#C$
0.25000	0.000	0	440	0.000	0	902	0.000	0	1820
0.26000	0.000	0	459	0.000	0	966	0.000	0	1875
0.27000	0.000	0	486	0.000	0	975	0.000	0	1903
0.28000	0.000	0	507	0.000	0	997	0.000	0	1991
0.29000	0.000	0	493	0.000	0	1046	0.000	0	2075
0.30000	0.000	0	531	0.000	0	1077	0.000	0	2140
0.31000	0.000	0	538	0.000	0	1116	0.000	0	2265
0.32000	0.000	1	585	0.000	0	1138	0.000	0	2308
0.33000	0.097	11	584	0.011	6	1197	0.000	0	2427
0.33100	0.177	10	585	0.043	8	1166	0.001	2	2364
0.33200	0.334	9	591	0.171	9	1199	0.033	4	2388
0.33300	0.453	12	587	0.422	10	1206	0.339	6	2420
0.33330	0.199	10	599	0.092	11	1193	0.021	7	2402
0.33333	0.187	8	592	0.076	8	1177	0.016	7	2387
0.33400	0.022	7	598	0.001	7	1190			
0.33500	0.001	4	602	0.000	3	1217			
0.33600	0.000	5	591						
0.33700	0.000	4	609						
0.33800	0.000	5	594						

Table 2. Guo's attack: Finding the correct convergent.

Efficiency

In Table 3, we illustrate the runtime needed to mount Guo's attack for 1024-bit RSA and various sizes of exhaustive searches. The attack was mounted on an AMD Opteron 850 server with quad 2.4 GHz processors and 16GB of RAM using Maple 12. The time needed to generate three random instances of RSA with small private exponents and compute the good starting convergents was approximately 0.15 seconds in all trials and is not included in the runtimes shown.

The data in Table 3 shows the average (observed) runtime for a partially optimized attack and also the full worst case time for Guo's attack. Each value of ℓ corresponds to an exhaustive search of 2^{ℓ} for both the numerator and denominator of the candidate for the private exponent. The data shows average times and one standard deviation taken over several trials (between 1,000 trials for short runtimes, 100 for intermediate runtimes and 10 for the largest runtimes). The partially optimized attack aborts the exhaustive search once the modulus is factored

for a given candidate but carries out the search for all three candidates of the private exponents. The full worst case data searches the entire $2^{2\ell}$ search space for each candidate. As is expected, the time for the attack increases by a factor of four when the value of ℓ is increased by one.

Notice that the data for the attack when $\ell = 10$, which corresponds to the data collected in Table 1, shows that the attack in practice (with a moderate exhaustive search) is both very successful and very efficient; taking approximately 10 seconds to factor the modulus.

Bound	Time T_ℓ (seconds)		Time T_ℓ (seconds)	
ℓ	$T_\ell \pm \sigma_\ell$	$T_\ell / T_{\ell-1}$	$T_\ell \pm \sigma_\ell$	$T_\ell / T_{\ell-1}$
3	0.08 ± 0.02	-	0.68 ± 0.04	-
4	0.15 ± 0.03	2.01	2.75 ± 0.10	4.05
5	0.30 ± 0.06	2.01	11.06 ± 0.29	4.02
6	0.62 ± 0.11	2.02	44.36 ± 0.89	4.01
7	1.24 ± 0.21	2.01	177.59 ± 3.12	4.00
8	2.53 ± 0.43	2.05	710.66 ± 11.54	4.00
9	5.24 ± 0.85	2.07	2835.36 ± 45.97	3.99
10	$\mathbf{11.14 \pm 1.71}$	**2.12**	$\mathbf{11365.47 \pm 171.12}$	**4.01**
11	25.02 ± 3.42	2.25		
12	61.02 ± 6.93	2.44		
13	165.42 ± 14.26	2.71		
14	503.20 ± 31.07	3.04		
15	1704.98 ± 67.82	3.39		
16	6180.87 ± 172.02	3.63		
	Partially Optimized Search		Full Worst Case	

Table 3. Worst case time needed for Guo's attack: RSA with 1024-bit modulus.

4.2 Multi-prime RSA

In the description of Guo's attack (and it's modifications) above, notice that once the equation

$$k_2 e_1 d_1 - k_1 e_2 d_2 = k_2 - k_1$$

is obtained, there is no way of knowing if the equation was derived using two key equations for RSA or using two key equations for multi-prime RSA. Thus, the

same attack and the same results hold for multi-prime RSA also. That is, we expect that the attack will be successful (with some non-negligible probability) when all three instances of multi-prime RSA have private exponents $d_i < N^{\delta}$ when

$$\delta < \frac{1}{3} - \epsilon,$$

where $\epsilon > 0$ is a small constant that is independent of N. Here, we again assume that each public exponent is full sized.

In practice, we have observed that the attack is actually slightly more successful for multi-prime RSA compared to RSA. In particular, the success rate for one candidate is about 0.65 and for any candidate is about 0.95 (compared to about 0.63 and 0.93 for RSA). We illustrate the effectiveness of the attack for small values of r and common modulus sizes in Table 4. Just as with the experiments for RSA, we average the success rates over $10,000$ trials for each private exponent size for the 1024- and 2048-bit modulus sizes and over $1,000$ trials when the bitlength of the modulus is 4096.

Similar to the RSA, the good starting convergent is, in practice, on average at most one convergent away from the correct convergent. For private exponents smaller than $N^{0.325}$ it is always the correct convergent. Since the data is very similar to that of RSA we omit the data here.

This attack on multi-prime RSA is actually quite remarkable since it is the first attack (other than factoring) that does not decrease with increasing number of primes in the modulus. This follows because the attack does not use the relation $|s| = |N - \phi(N)| < (2r-1)N^{1-1/r}$ which is used in all the other attacks. Since the size of s increases with increasing r, the other attacks become weaker.

4.3 Takagi's scheme

For Takagi's scheme, just as with multi-prime RSA, notice that Guo's attack is the same as for RSA. The only difference is that even when the public exponents are full sized, they are much smaller than the modulus N which is the case for RSA and multi-prime RSA. In particular, since the key equation is given by

$$ed = 1 + k\lambda'(N) = 1 + k\,\mathrm{lcm}(p-1, q-1),$$

it is expected, with high probability, that the public exponent will be roughly the same size as $\mathrm{lcm}(p-1, q-1)$ (when the private exponent is small). For randomly generated primes it is further expected that $\mathrm{lcm}(p-1, q-1)$ will be close to $(p-1)(q-1)$ and so a full sized public exponent will have size $N^{2/(t+1)}$ when the modulus is given by $N = p^t q$. Now, from the derivation of Guo's attack

	1024-bit N $r = 3$			2048-bit N $r = 3$			4096-bit N $r = 4$		
δ	D_2	D_i, K_i	Guess	D_2	D_i, K_i	Guess	D_2	D_i, K_i	Guess
0.25000	0.6527	0.9424	1.0000	0.6463	0.9473	1.0000	0.6370	0.9500	1.0000
0.26000	0.6518	0.9489	1.0000	0.6528	0.9504	1.0000	0.6630	0.9620	1.0000
0.27000	0.6531	0.9503	1.0000	0.6547	0.9519	1.0000	0.6440	0.9560	1.0000
0.28000	0.6535	0.9503	1.0000	0.6475	0.9470	1.0000	0.6730	0.9570	1.0000
0.29000	0.6519	0.9492	1.0000	0.6576	0.9453	1.0000	0.6400	0.9490	1.0000
0.30000	0.6554	0.9490	1.0000	0.6553	0.9464	1.0000	0.6800	0.9510	1.0000
0.31000	0.6556	0.9503	1.0000	0.6561	0.9470	1.0000	0.6520	0.9660	1.0000
0.32000	0.6505	0.9478	1.0000	0.6456	0.9481	1.0000	0.6610	0.9500	1.0000
0.32500	0.6575	0.9494	1.0000	0.6516	0.9489	1.0000	0.6550	0.9490	1.0000
0.33000	0.6485	0.9459	1.0000	0.6578	0.9524	1.0000	0.6810	0.9580	1.0000
0.33100	0.6466	0.9489	1.0000	0.6563	0.9485	1.0000	0.6770	0.9530	1.0000
0.33200	0.6469	0.9505	1.0000	0.6578	0.9469	1.0000	0.6800	0.9590	1.0000
0.33300	0.5564	0.8326	0.8785	0.5654	0.8282	0.8773	0.5300	0.7750	0.8320
0.33330	0.3496	0.5372	0.5731	0.2332	0.3614	0.3890	0.0890	0.1700	0.1820
0.33333	**0.3337**	**0.5138**	**0.5459**	**0.1987**	**0.3182**	**0.3455**	**0.0770**	**0.1190**	**0.1280**
0.33400	0.0854	0.1396	0.1585	0.0105	0.0230	0.0294	0.0000	0.0000	0.0000
0.33500	0.0094	0.0181	0.0229	0.0001	0.0001	0.0003			
0.33600	0.0011	0.0025	0.0034	0.0000	0.0000	0.0000			
0.33700	0.0002	0.0002	0.0004						
0.33800	0.0000	0.0000	0.0000						

Table 4. Guo's attack on multi-prime RSA: Empirical success rate.

earlier, recall that a sufficient condition to obtain the desired convergents in the continued fraction expansion of e_1/e_2 was given by (equation (4.2))

$$d_1 < \frac{e_2}{2k_2|k_2 - k_1|},$$

which can be rewritten as $2k_2\,|k_2 - k_1|\,d_1 < e_2$. Since $0 < k_1, k_2, d_1 < N^{\delta}$ and $e \approx N^{2/(t+1)}$, notice that a new sufficient condition is given by $2N^{3\delta} < N^{2/(t+1)}$, or more simply

$$\delta < \frac{2}{3(t+1)} - \log_N(2) = \frac{2}{3(t+1)} - \epsilon,$$

where $\epsilon > 0$ is small and becomes smaller with increasing N. Again, we used $k_i < d_i$, which follows from the key equation. The bound on δ shows that the

attack is expected to become weaker as the multiplicity of the prime p increases (the parameter t).

It should be pointed out that when mounting Guo's attack on Takagi's variant, we do not look for candidates for the k_i (constants in the key equations) since we do not have a method for factoring the modulus given one of the k_i. Thus, we only try to compute candidates for the three private exponents.

In practice, just as with RSA and multi-prime RSA, the attack works well up to the theoretical bound. We illustrate the effectiveness of the attack for small values of t and common modulus sizes in Tables 5 and 6. In particular, Table 5 shows the success rate for different sized moduli with $N = p^2q$. For moduli of this form, the bound in Guo's attack is $\delta < 2/9 \approx 0.2222$. For the 1024- and 2048-bit modulus sizes, we averaged the data over $10,000$ trials. For the 4096-bit modulus size, we used $1,000$ trials. As can be seen, the attack works quite well for private exponents approaching this bound.

In Table 6, we show the success rates when mounting the attack on Takagi's scheme with 4096-bit moduli of the form $N = p^3q$. For moduli of this form, Guo's bound is $\delta < 1/6 \approx 0.1667$. Again, the data illustrates that the attacks work quite well up to private exponents approaching the theoretical bound. The data shown is averaged over $1,000$ trials for each private exponent size. In addition to the success rates, we also include the data showing that the good starting convergent is indeed a good starting convergent (just as with RSA and multi-prime RSA).

5 Howgrave-Graham and Seifert's attack

Howgrave-Graham and Seifert's small private exponent attack on common modulus RSA [9] improves upon Guo's attack in several ways. In particular, the attack can be mounted with only two instances of RSA (although it gets stronger with more), the problems associated with relatively prime quantities are not a concern and, most importantly, the attack (even with only two instances) is much stronger.

Even though the attack is a heuristic attack it has been shown to work well in practice when the number of instances of RSA and the modulus sizes are relatively small (see [9]). Given $n \leq 5$ instances of RSA with a common modulus, each having private exponent smaller than N^{δ_n}, the attack can factor the modulus when δ_n is smaller than given in Table 7.

When there is only one instance of RSA ($n = 1$), the attack is simply Wiener's attack when mounted as a heuristic latticed-based attack (as described earlier). With only two instances, the attack is already much stronger than Guo's attack and with six instances, the attack is expected to factor the modulus when the private exponents are approaching $N^{1/2}$.

	1024-bit N $N = p^2q$			2048-bit N $N = p^2q$			4096-bit N $N = p^2q$		
δ	D_2	D_i	Guess	D_2	D_i	Guess	D_2	D_i	Guess
0.15000	0.6193	0.9190	1.0000	0.6289	0.9207	1.0000	0.6400	0.9240	1.0000
0.16000	0.6252	0.9176	1.0000	0.6221	0.9218	1.0000	0.6150	0.9210	1.0000
0.17000	0.6307	0.9218	1.0000	0.6339	0.9239	1.0000	0.6130	0.9020	1.0000
0.18000	0.6287	0.9185	1.0000	0.6322	0.9238	1.0000	0.6160	0.9280	1.0000
0.19000	0.6322	0.9257	1.0000	0.6221	0.9266	1.0000	0.6480	0.9160	1.0000
0.20000	0.6312	0.9243	1.0000	0.6241	0.9234	1.0000	0.6080	0.8980	1.0000
0.21000	0.6384	0.9206	1.0000	0.6281	0.9181	1.0000	0.6350	0.9330	1.0000
0.21500	0.6205	0.9200	1.0000	0.6312	0.9242	1.0000	0.6350	0.9110	1.0000
0.22000	0.6356	0.9228	1.0000	0.6318	0.9234	1.0000	0.6280	0.9110	1.0000
0.22100	0.6336	0.9222	1.0000	0.6262	0.9209	1.0000	0.6340	0.9350	1.0000
0.22200	0.4737	0.7101	0.7803	0.4206	0.6470	0.7121	0.3420	0.5180	0.5730
0.22220	0.3276	0.5066	0.5638	0.2098	0.3289	0.3785	0.0760	0.1440	0.1700
0.22222	**0.3172**	**0.4847**	**0.5416**	**0.1902**	**0.3121**	**0.3577**	**0.0750**	**0.1240**	**0.1510**
0.22300	0.0667	0.1082	0.1387	0.0059	0.0108	0.0186	0.0000	0.0000	0.0000
0.22400	0.0065	0.0120	0.0194	0.0001	0.0002	0.0005			
0.22500	0.0013	0.0023	0.0040	0.0000	0.0000	0.0000			
0.22600	0.0000	0.0001	0.0004						
0.22700	0.0001	0.0001	0.0001						
0.22800	0.0000	0.0000	0.0000						

Table 5. Guo's attack on Takagi's scheme: Empirical success rate for $N = p^2q$.

When there are seven or more instances of RSA, however, we note that the bounds suggested in [9] are too optimistic. We (will) argue that the bound is $N^{1/2}$ for any number of instances beyond six. We will discuss this in more detail later, but now we show how the attack is mounted for two and three instances to give a flavor of the general approach. Since we also want to mount the attack on multi-prime RSA, we will re-derive the attack (for $n = 2, 3$) for multi-prime RSA. The attack is identical to Howgrave-Graham and Seifert's attack except that the bound for $s = N - \phi(N)$ is left as a function of the number of primes r.

Following Howgrave-Graham and Seifert, we let W_i denote the key equation

$$W_i : \; e_i d_i - k_i N = 1 - k_i s,$$

which is the basis for Wiener's attack, and let $G_{i,j}$ denote the equation

$$G_{i,j} : \; k_i d_j e_j - k_j d_i e_i = k_i - k_j,$$

4096-bit $N = p^3q$						
δ	D_2	D_i	Guess	C_{ave}	$C_{\max}$	#C
0.100	0.618	0.919	1.000	0	0	729
0.110	0.616	0.929	1.000	0	0	793
0.120	0.637	0.925	1.000	0	0	869
0.130	0.648	0.916	1.000	0	0	923
0.140	0.620	0.898	1.000	0	0	1012
0.150	0.624	0.922	1.000	0	0	1060
0.160	0.632	0.916	1.000	0	0	1159
0.161	0.623	0.915	1.000	0	0	1166
0.162	0.633	0.918	1.000	0	0	1151
0.163	0.616	0.919	1.000	0	0	1161
0.164	0.623	0.916	1.000	0	1	1182
0.165	0.655	0.927	1.000	0.02	5	1186
0.166	**0.678**	**0.923**	**1.000**	**0.22**	**10**	**1195**
0.167	0.006	0.010	0.015	0.87	3	1179
0.168	0.000	0.000	0.000			

Table 6. Guo's attack on Takagi's scheme: Empirical success rate for $N = p^3q$.

n	1	2	3	4	5	6
δ_n	0.250	0.357	0.400	0.441	0.468	0.493

Table 7. Common modulus attack with small private exponent.

which is the basis for Guo's attack. Recall that for multi-prime RSA, the quantity $s = N - \phi(N)$ satisfies $|s| < (2r-1)N^{1-1/r} \approx N^{1-1/r}$ when there are r primes in the modulus. This inequality also holds for $r = 2$ (RSA).

First consider two instances of multi-prime RSA. Let (e_1, N) and (e_2, N) be two valid multi-prime RSA public keys (with $e_1 \neq e_2$), each having their private exponent smaller than N^{δ_2}. Thus, the constants in the key equations satisfy $k_1, k_2 < N^{\delta_2}$. Using the equations k_2W_1, $G_{1,2}$, W_1W_2 and the trivial equation $I_2 : k_1k_2 = k_1k_2$, we construct a lattice with a known small vector. In particular, notice that the equations

$$\begin{aligned}
I_2 &: k_1k_2 = k_1k_2 \\
k_2W_1 &: k_2d_1e_1 - k_2k_1N = k_2(1 - k_1s) \\
G_{1,2} &: k_1d_2e_2 - k_2d_1e_1 = k_1 - k_2 \\
W_1W_2 &: d_1d_2e_1e_2 - d_1k_2e_1N - d_2k_1e_2N + k_1k_2N^2 = (1 - k_1s)(1 - k_2s),
\end{aligned}$$

can be written as the vector-matrix equation $x_2\mathcal{B}_2 = v_2$, where

$$x_2 = (k_1k_2, k_2d_1, k_1d_2, d_1d_2)$$

$$\mathcal{B}_2 = \begin{bmatrix} 1 & -N & 0 & N^2 \\ & e_1 & -e_1 & -e_1N \\ & & e_2 & -e_2N \\ & & & e_1e_1 \end{bmatrix}$$

$$v_2 = (k_1k_2, k_2(1-k_1s), k_1-k_2, (1-k_1s)(1-k_2s)).$$

The vector v_2 is an integer linear combination of the rows in $\mathcal{B}_2$, and is therefore a vector in the lattice $\mathcal{L}_2$ generated by the rows in $\mathcal{B}_2$. If the vector v_2 is a smallest vector in $\mathcal{L}_2$ then recovering v_2 will allow us to factor the modulus. Indeed, given $\mathcal{B}_2$ and v_2, we can compute x_2 whose first two components k_1k_2 and d_1k_2 yield k_1/d_1. Just as in Wiener's attack, this allows us to compute

$$\phi(N) = \frac{e_1d_1 - 1}{k_1} = \left\lfloor e_1\left(\frac{d_1}{k_1}\right)\right\rfloor,$$

which then allows us to factor the modulus (deterministically for RSA and probabilistically for multi-prime RSA with $r > 2$). Since the components of v_2 are not balanced, we can modify the equation by multiplying it by the diagonal matrix $\mathcal{D}_2 = \operatorname{diag}(N^{2(1-1/r)}, N^{1-1/r}, N^{\delta_2+2(1-1/r)}, 1)$, and considering the new vector-matrix equation $x_2\mathcal{B}_2' = v_2'$, where

$$x_2 = (k_1k_2, k_2d_1, k_1d_2, d_1d_2)$$

$$\mathcal{B}_2' = \mathcal{B}_2\mathcal{D}_2 = \begin{bmatrix} N^{2(1-1/r)} & -N^{2-1/r} & 0 & N^2 \\ & e_1N^{1-1/r} & -e_1N^{\delta_2+2(1-1/r)} & -e_1N \\ & & e_2N^{\delta_2+2(1-1/r)} & -e_2N \\ & & & e_1e_1 \end{bmatrix}$$

$$\begin{aligned} v_2' &= v_2\mathcal{D}_2 \\ &= \big(k_1k_2N^{2(1-\frac{1}{r})}, k_2(1-k_1s)N^{1-\frac{1}{r}}, \\ &\qquad (k_1-k_2)N^{\delta_2+2(1-\frac{1}{r})}, (1-k_1s)(1-k_2s)\big). \end{aligned}$$

Notice that the target vector v_2' is a vector in the lattice, call it $\mathcal{L}_2'$, that is generated by the rows of $\mathcal{B}_2'$ and that the components of v_2' are balanced (up to multiplicative constants that do not depend on N). Now, the target vector has size

$$\|v_2'\| \approx N^{2\delta_2+2(1-1/r)},$$

and the lattice $\mathcal{L}_2'$ has volume

$$\mathrm{vol}(\mathcal{L}_2') = |\det(\mathcal{B}_2')| = e_1^2 e_2^2 N^{\delta_2+5(1-1/r)} \approx N^{\delta_2+9-5/r},$$

when the public exponents are full sized. From Theorem 3.2 (Minkowski), we know that a shortest vector in $\mathcal{L}_2'$ will have norm at most $2\mathrm{vol}(\mathcal{L}_2')^{1/4}$. Therefore, a necessary condition for v_2' to be a shortest vector in $\mathcal{L}_2'$ is given by $\|v_2\| \leq 2\mathrm{vol}(\mathcal{L}_2')^{1/4}$, or, looking at the exponents of N

$$2\delta_2 + 2(1-1/r) \leq \frac{1}{4}(\delta_2 + 9 - 5/r),$$

where we have ignored all constants not depending on N. This is further simplified as

$$\delta_2 < \frac{3+r}{7r} - \epsilon,$$

where $\epsilon > 0$ is added to account for the ignored constants. Letting $r = 2$, we recover Howgrave-Graham and Seifert's original bound of $\delta_2 < 5/14 - \epsilon \approx 0.357 - \epsilon$. If both private exponents satisfy this bound and if v_2' is a smallest vector in $\mathcal{L}_2'$ and if we can find a smallest vector in the lattice (which we can efficiently do for a 4-dimensional lattice using Nguyen and Stehlé's greedy algorithm [16]) then we can factor the modulus. Thus, if Assumption 3.3 holds we can factor the modulus. Again, solving for the vector x_2 reveals k_2/d_2 which can be used to compute $\phi(N)$.

Now consider three instances of multi-prime RSA with a common modulus. Let (e_1, N), (e_2, N), (e_3, N) be three valid multi-prime RSA public keys (with $e_i \neq e_j$), each having their private exponent smaller than N^{δ_3}. In this case, a lattice is constructed with the eight equations: $k_1k_2k_3 = k_1k_2k_3$, $k_2k_3W_1$, $k_3G_{1,2}$, $k_3W_1W_2$, $k_2G_{1,3}$, $W_1G_{2,3}$, $W_2G_{1,3}$, and $W_1W_2W_3$. In particular, these equations

can be written as the vector-matrix equation $x_3\mathcal{B}_3 = v_3$, where

$$x_3 = (k_1k_2k_3, d_1k_2k_3, k_1d_2k_3, d_1d_2k_3, k_1k_2d_3, d_1k_2d_3, k_1d_2d_3, d_1d_2d_3)$$

$$\mathcal{B}_3 = \begin{bmatrix} 1 & -N & 0 & N^2 & 0 & 0 & 0 & -N^3 \\ & e_1 & -e_1 & -e_1N & -e_1 & 0 & e_1N & e_1N^2 \\ & & e_2 & -e_2N & 0 & e_2N & 0 & e_2N^2 \\ & & & e_1e_2 & 0 & -e_1e_2 & -e_1e_2 & -e_1e_2N \\ & & & & e_3 & -e_3N & -e_3N & e_3N^2 \\ & & & & & e_1e_3 & 0 & -e_1e_3N \\ & & & & & & e_2e_3 & -e_2e_3N \\ & & & & & & & e_1e_2e_3 \end{bmatrix}$$

$$v_3 = \big(k_1k_2k_3, k_2k_3(1-k_1s), k_3(k_1-k_2), k_3(1-k_1s)(1-k_2s), \\ k_2(k_1-k_3), (1-k_1s)(k_2-k_3), (1-k_2s)(k_1-k_3), \prod_{i=1,2,3}(1-k_is)\big).$$

As in the $n = 2$ case, the components of v_3 are not balanced. Multiplying the equation by the diagonal matrix $\mathcal{D}_3$ given by

$$\operatorname{diag}\big(N^{3(1-\frac{1}{r})}, N^{2(1-\frac{1}{r})}, N^{\delta_3+3(1-\frac{1}{r})}, N^{1-\frac{1}{r}}, \\ N^{\delta_3+3(1-\frac{1}{r})}, N^{\delta_3+2(1-\frac{1}{r})}, N^{\delta_3+2(1-\frac{1}{r})}, 1\big),$$

we obtain a new vector-matrix equation $x_3\mathcal{B}_3' = x_3\mathcal{B}_3\mathcal{D}_3 = v_3\mathcal{D}_3 = v_3'$. Here the new target vector v_3' is a vector in the lattice $\mathcal{L}_3'$ (generated by the rows in $\mathcal{B}_3'$) and has balanced components. The new target vector has size

$$\|v_3'\| \approx N^{3(\delta_3+1-1/r)},$$

and the new lattice has volume

$$\operatorname{vol}(\mathcal{L}_3') = |\det(\mathcal{B}_3')| = e_1^4e_2^4e_3^4N^{4\delta_3+16(1-1/r)} \approx N^{4\delta+28-16/r}.$$

From Theorem 3.2, we then know that a necessary condition for v_3' to be a smallest vector in $\mathcal{L}_3'$ is given by $\|v_3'\| \le \sqrt{8}\operatorname{vol}(\mathcal{L}_3')^{1/8}$. Ignoring constants that do not depend on N, this is satisfied when

$$3\delta_3 + 3 - 3/r \le \frac{1}{8}(4\delta_3 + 28 - 16/r),$$

or more simply

$$\delta_3 < \frac{2+r}{5r} - \epsilon,$$

where $\epsilon > 0$ has been added to account for the ignored constants. If Assumption 3.3 holds and v_3' is a smallest vector in $\mathcal{L}_3'$ and we can compute a smallest vector in $\mathcal{L}_3'$ (by computing an LLL-reduced basis for the lattice) then computing v_3' allows us to factor the modulus. Just as with $n = 2$, we can recover the vector x_3 whose first two components $k_1k_2k_3$ and $d_1k_2k_3$ allow us to compute k_1/d_1, and hence $\phi(N)$.

In the general case, when there are n instances of multi-prime RSA with a common modulus, a vector-matrix equation $x_n\mathcal{B}_n = v_n$ is constructed with 2^n equations. The first equation is the trivial equation $k_1 \cdots k_n = k_1 \cdots k_n$, the final equation is the product of all the Wiener equations $W_1 \cdots W_n$, and the remaining equations are products of various Wiener and Guo equations (W_i and $G_{i,j}$), chosen so that the basis matrix $\mathcal{B}_n$ is triangular. However, the choice of the middle $2^n - 2$ equations to ensure a triangular basis matrix is not unique and this determination becomes an optimization problem to maximize the volume of the lattice (see [9] for more detail on this equation determination). The last component of v_n (coming from the right hand side of the equation $W_1 \cdots W_n$) will dominate the components of v_n with size $N^{n(\delta_n+1-1/r)}$. Multiplying the vector-matrix equation by an appropriate diagonal matrix we construct a new vector-matrix equation $x_n\mathcal{B}_n' = v_n'$, where the new target vector has balanced components. Just as in the $n = 2, 3$ cases, the diagonal matrix will leave the last row of $\mathcal{B}_n$ unchanged (and so the final row of the new basis matrix will still correspond to $W_1 \cdots W_n$). Using Theorem 3.2, a necessary condition for δ_n can be determined so that v_3' is a smallest vector in $\mathcal{L}_n'$ (the lattice generated by the rows of $\mathcal{B}_n'$). Following the general bounds determination from Howgrave-Graham and Seifert, it can be shown that if δ_n satisfies

$$\delta_n < \frac{n2^n + (2^n - (2n+1)\binom{n}{n/2})(1-\frac{1}{r})}{n2^n - \left(2^n - (2n+1)\binom{n}{n/2}\right)}, \tag{5.1}$$

when n is even or

$$\delta_n < \frac{n2^n + (2^n - 4n\binom{n-1}{(n-1)/2})(1-\frac{1}{r})}{n2^n - \left(2^n - 4n\binom{n-1}{(n-1)/2}\right)}, \tag{5.2}$$

when n is odd, then the target vector v_n' will satisfy Minkowski's bound (Theorem 3.2) for the lattice $\mathcal{L}_n'$. Letting $r = 2$ recovers Howgrave-Graham and Seifert's bounds.

However, the bounds are not a sufficient condition for v'_n to be a smallest vector. In fact, based on the structure of the basis matrix $\mathcal{B}'_n$, we can construct another necessary condition that requires δ_n to be much smaller for large values of n. Consider the description of the construction of the basis matrix $\mathcal{B}'_n$ given above. It is always the case that the last column of the basis matrix will always correspond to the product of all the Wiener equations $W_1 \cdots W_n$, that this column is not modified when balancing the components of the target vector (since the right hand side of this equation is always the dominant component of the target vector) and that the basis matrix is always (upper) triangular. Looking at the left hand side of the product of all the Wiener equations $W_1 \cdots W_n$:

$$(e_1 d_1 - k_1 N)(e_2 d_2 - k_2 N) \cdots (e_n d_n - k_n N),$$

each term in the expansion of this product is of the form

$$\prod_{i \in S} e_i d_i \prod_{j \in \{1,\ldots,n\} \setminus S} k_j N,$$

where $S \subseteq \{1, \ldots, n\}$. Thus, each component in the column of the basis matrix corresponding to this equation is of the form

$$N^{n-|S|} \prod_{i \in S} e_i.$$

Since all of the public exponents satisfy $e_i \approx N$, it follows that each component has (approximate) size N^n. The size of the last row vector in the basis matrix, which has only one non-zero component since the matrix is (upper) triangular, then also has (approximate) size N^n. This vector is clearly a vector in the lattice since it is a basis vector (being a vector in the basis matrix). Thus, if the target vector v'_n is to be a smallest vector in the lattice, it must also be smaller than this vector. Since v'_n has size $N^{n(\delta_n+1-1/r)}$, it follows that another necessary condition (for v'_n to be a smallest vector) is given by

$$n(\delta_n + 1 - 1/r) < n,$$

or more simply

$$\delta_n < \frac{1}{r}. \tag{5.3}$$

Therefore, the size of the private exponents must satisfy inequality (5.3) in addition to inequalities (5.1) and (5.2) if it is to be a smallest vector. When all of these inequalities hold, and when v'_n actually is a smallest vector, finding v'_n allows

us to factor the modulus in the same way as illustrated in the $n = 2, 3$ cases. In particular, the components of the vector x_n will have the form $(h_1, \ldots, h_n)$ where $h_i \in \{k_i, d_i\}$. Since all 2^n possible combinations will be present, we know that $k_1 \cdots k_n$ and $d_1 k_2 \cdots k_n$ will be present (and defined by the structure of $\mathcal{B}_n$). Thus, the value k_1/d_1 can be found and used to compute $\phi(N)$ as described above.

For RSA ($r = 2$), notice that inequality (5.3) implies that Howgrave-Graham and Seifert's attack cannot break instances of RSA with private exponents greater than $N^{1/2}$ (regardless of the number of instances present). Since the bounds imposed by inequalities (5.1) and (5.2) exceed $N^{1/2}$ when $n \geq 7$, the bounds originally suggested by Howgrave-Graham and Seifert are overly optimistic in this range. Thus, for any $n \geq 7$, we should have $\delta_n < 1/2 - \epsilon$ as the bound. The bounds for $n \leq 6$ remain as originally stated. In fact, the experiments in [9] verified the practical effectiveness of the attacks for $2 \leq n \leq 5$. Unfortunately, since the lattice dimension is exponential in n, mounting the attack for $n \geq 6$ becomes computationally expensive (and hence was not done) and so the $N^{1/2}$ ceiling was not observed (experimentally) by Howgrave-Graham and Seifert or here.

For multi-prime RSA ($n > 2$), the bound from inequality (5.3) dominates the attack for almost all parameter choices except $r = 3$ with two instances, which has a bound $\delta_2 < 6/21 \approx 0.286 < 1/3$, $r = 4$ with two instances, where the bounds match at $1/4$, and $r = 3$ with three instances where the bounds match at $1/3$.

We did not extend Howgrave-Graham and Seifert's attack to Takagi's variant. Our attempts only led to non-attacks (i.e., the bounds on δ are always negative). For simplicity, let the public/private exponents be defined $(p-1)(q-1)$ instead of modulo $\lambda'(N) = \mathrm{lcm}(p-1, q-1)$. The obvious attempt is to multiply the key equation by p^{t-1} to obtain an equation

$$edp^{t-1} = p^{t-1} + k\phi(N) = p^{t-1} + k(N - s),$$

where $s \approx N^{t/(t+1)}$. Using this for the W_i equations, we can follow the derivation (for $n = 2, 3$ as above for example). Working through the details, the attack fails because each public exponent is of size (roughly) $N^{2/(t+1)}$, which reduces the volume of the basis matrix considerably. Matching the size of the target vector to the volume of the lattice (by Minkowski's theorem) we find that $\delta < 0$ is a necessary condition for the target vector to be a smallest vector in the lattice.

5.1 Practical effectiveness

In Table 8, we illustrate the practical effectiveness of Howgrave-Graham and Seifert's attack when mounted against RSA and multi-prime RSA for up to $n = 4$ instances.

While the attack is only a heuristic, it works extremely well when mounted against RSA in practice. In [9, Figures 2–7], Howgrave-Graham and Seifert show experimental results (success rates and runtimes) for up to $n = 5$ instances of RSA (with modulus lengths ranging from 200 to 700 bits). In Table 8, we illustrate the effectiveness of the attack against RSA and multi-prime RSA ($r = 3, 4$) when there are up to four instances ($n = 2, 3, 4$) with modulus lengths of 1024 or 2048 bits. For each value of δ in the table, *Success* is the number of successful attacks out of 100 trials, unless otherwise noted. For each grouping of number of instances, the theoretical bound is listed in the final row (along with an indication if the attack can achieve this bound in practice in a random sampling of 100 trials).

From the data in the table it is clear that the attack works quite well against RSA ($r = 2$), which was already shown by Howgrave-Graham and Seifert in [9]. The attack almost always succeeds as the size of the private exponents approach the theoretic bound at which point the success rate quickly deteriorates to zero. The attack is successful (albeit with small probability) right up to the theoretical bound. When the attack is mounted against multi-prime RSA, however, the experimental limits of the attack do not reach the theoretical limits and this discrepancy seems to grow with increasing number of primes in the modulus (based on the small sample set of $r = 2, 3, 4$ only). Given two instances of multi-prime RSA, the attack is still a great improvement over single instance small private attacks (e.g., Boneh and Durfee's attack) though. As soon as three instances are known, however, Guo's attack is stronger. For $r = 3$, the theoretical bounds are actually the same but, in practice, Guo's attack is successful right up to the $N^{1/3}$ bound, whereas Howgrave-Graham and Seifert's attack works for private exponents smaller than about $N^{0.278}$. For larger values of r, the theoretical bound ($\delta < 1/r$) is always smaller than Guo's bound $N^{1/3}$. Thus, when there are at least three instances available, Guo's is stronger in practice.

The data also indicates that the attack becomes less effective in practice as the size of the modulus increases.

Efficiency

The dominant cost of Howgrave-Graham and Seifert's attack is computing a small vector which is hopefully the target vector. For n instances of RSA, the lattice in the attack has dimension $\dim(\mathcal{L}) = 2^n$ and the complexity of computing an LLL-reduced basis using Nguyen and Stehlé's L^2 algorithm is

$$O(2^{3n}(2^n + \log B)\log B \cdot \mathcal{M}(2^n)),$$

where B is the size of the largest component in the basis matrix (see [17]). the attack is only feasible when there are relatively few instances of RSA (small n) since the complexity is exponential in this parameter.

	1024-bit N $N = pq$		1024-bit N $N = p_1p_2p_3$		2048-bit N $N = p_1p_2p_3p_4$	
	δ	Success	δ	Success	δ	Success
$n = 2$ instances	0.351	100	0.243	100	0.181	100
	0.352	100	0.244	100	0.182	100
	0.353	100	0.245	100	0.183	100
	0.354	100	0.246	100	0.184	100
	0.355	97	0.247	88	0.185	100
	0.356	75	0.248	52	0.186	77
	0.357	6	0.249	4	0.187	2
	0.358	0	0.250	0	0.188	0
	0.357	✓	0.286	✗	0.250	✗
$n = 3$ instances	0.394	100	0.271	100	0.185	100
	0.395	100	0.272	100	0.190	100
	0.396	100	0.273	100	0.195	100
	0.397	100	0.274	100	0.200	100
	0.398	100	0.275	100	0.205	100
	0.399	74	0.276	71	0.206	100
	0.400	3	0.277	5	0.207	93
	0.410	0	0.278	0	0.208	0
	0.400	✓	0.333	✗	0.300	✗
$n = 4$ instances	0.415	100	0.285	100	0.212	100
	0.420	100	0.286	100	0.213	100
	0.425	100	0.287	100	0.214	100
	0.430	100	0.288	100	0.215	100
	0.435	100	0.289	99	0.216	100
	0.436	100	0.290	83	0.217	100
	0.437	11	0.291	1	0.218	21
	0.438	0	0.292	0	0.219	0
	0.441	✗	0.379	✗	0.348	✗

Table 8. Howgrave-Graham and Seifert's attack in practice. Number of successes for 100 trials for each value of δ shown for $n = 2, 3, 4$ instances of RSA and multi-prime RSA with 3 or 4 primes in the modulus. Last row for a given number of instances shows theoretical bound.

When the number of instances of RSA is fixed it can be shown that the attack is polynomial in $\log(N)$. This follows from the construction of the lattices that are used which imply that $B < N^{2n}$. In practice, of course, the constants in the complexity estimate arising from the fixed n limit the feasibility of the attack to a small number of instances. We illustrate this in Table 9, where we show the runtime of mounting the attack for up to $n = 4$ instances of RSA with a common modulus for various modulus sizes. Again, the attack was mounted on an AMD Opteron 850 server with quad 2.4 GHz processors and 16GB of RAM. The LLL-reduced basis was computed using Victor Shoup's NTL c++ library. The values in the table are averages of 100 trials for $n = 2, 3$ and for 10 trials for $n = 4$. As can be seen, the attack is very feasible for small values of n but the runtime quickly increases (as is expected).

Modulus (bitlength)	Attack Time		
	$n = 2$	$n = 3$	$n = 4$
1024	0.1 sec	0.3 min	0.3 hours
2048	0.6 sec	1.7 min	1.7 hours
4096	3.5 sec	9.9 min	9.4 hours
10000	34.7 sec	93.0 min	72.5 hours

Table 9. Howgrave-Graham and Seifert's attack: Efficiency of the attack for various modulus sizes and number of instances of RSA.

6 Conclusions

In this work, we re-examined Guo's continued fraction and Howgrave-Graham and Seifert's lattice-based attacks on small private exponent RSA with a common modulus. We have shown that Guo's attack is actually quite effective in practice when a modest exhaustive search is allowed (2^{20} bits in total). We have also shown that the theoretical bounds of Howgrave-Graham and Seifert's attack is $N^{1/2}$ once there are seven or more instances of RSA. This corrects the original bounds proposed in the attack. The bounds for $n \leq 6$ instances remains the same as originally given.

The correction to the bound in Howgrave-Graham's bound arises from the details of the basis construction as given in [9]. In particular, the equation $W_1 \cdots W_n$ leads to the second necessary condition $\delta < 1/2$. Removing this equation (and possibly others) may still lead to an attack for private exponents greater than the $N^{1/2}$ bound. We are currently investigating this.

In addition, we have also mounted the attacks on two fast variants of RSA: multi-prime RSA and Takagi's variant. For multi-prime RSA, we find that in prac-

tice, Guo's attack is the stronger of the two attacks as soon as three instances are available. For Takagi's scheme, only Guo's attack can be applied. Thus, there is no attack on Takagi's scheme when only two instances are available. It is an open question if such an attack exists.

Bibliography

[1] D. Boneh, Twenty years of attacks on the RSA cryptosystem, *Not. Amer. Math. Soc.* **46** (1999), 203–213.

[2] D. Boneh and G. Durfee, Cryptanalysis of RSA with private key d less than $N^{0.292}$, *IEEE Trans. Inf. Theory* **46** (2000), 1339–1349.

[3] M. Ciet, F. Koeune, F. Laguillaumie and J.-J. Quisquater, *Short Private Exponent Attacks on Fast Variants of RSA*, Université Catholique de Louvain, UCL Crypto Group Technical Report Series no. CG-2002/4, 2002.

[4] J.-S. Coron and A. May, Deterministic polynomial time equivalent of computing the RSA secret key and factoring, *J. Cryptol.* **20** (2007), 39–50.

[5] J. M. DeLaurentis, A further weakness in the common modulus protocol for the RSA cryptoalgorithm, *Cryptologia* **8** (1984), 253–259.

[6] N. Ferguson and B. Schneier, *Practical Cryptography*, Wiley Publishing, Inc., Indianapolis, Indiana, 2003.

[7] M. J. Hinek, On the security of multi-prime RSA, *J. Math. Cryptol.* **2** (2008), 117–147.

[8] M. J. Hinek, M. K. Low and E. Teske, On some attacks on multi-prime RSA, in: *Selected Areas in Cryptography – SAC 2002*, Lecture Notes in Computer Science 2595, pp. 385–404, Springer-Verlag, 2003.

[9] N. Howgrave-Graham and J.-P. Seifert, Extending Wiener's attack in the presence of many decrypting exponents, in: *Secure Networking – CQRE (Secure) '99*, Lecture Notes in Computer Science 1740, pp. 153–166, Springer-Verlag, 1999.

[10] K. Itoh, N. Kunihiro and K. Kurosawa, Small secret key attack on a variant of RSA (due to Takagi), in: *CT-RSA 2008* (T. Malkin, ed.), Lecture Notes in Computer Science 4964, pp. 387–406, 2008.

[11] N. Kunihiro and K. Kurosawa, Deterministic polynomial time equivalence between factoring and key-recovery attack on Takagi's RSA, in: *PKC 2007* (T. Okamoto and X. Wang, eds.), Lecture Notes in Computer Science 4450, pp. 412–425, Springer-Verlag, 2007.

[12] A. K. Lenstra, Unbelievable security. Matching AES security using public key systems, in: *Advances in Cryptology – ASIACRYPT 2001*, Lecture Notes in Computer Science 2248, pp. 67–86, 2001.

[13] A. May, *New RSA vulnerabilities using lattice reduction methods*, Ph. D. thesis, University of Paderborn, 2003.

[14] A. May, Computing the RSA secret key is deterministic polynomial time equivalent to factoring, in: *Advances in Cryptology – CRYPTO 2004*, Lecture Notes in Computer Science 3152, pp. 213–219, Springer-Verlag, 2004.

[15] G. L. Miller, Riemann's hypothesis and tests for primality, *J. Computer System Sci.* **13** (1976), 300–317.

[16] P. Q. Nguyen and D. Stehlé, Low-Dimensional lattice basis reduction revisited, in: *Algorithmic Number Theory: 6th International Symposium, ANTS-VI*, Lecture Notes in Computer Science 3076, pp. 338–357, Springer-Verlag, 2004.

[17] P. Q. Nguyen and D. Stehlé, An LLL algorithm with quadratic complexity, *SIAM J. Comput.* **39** (2009), 874–903.

[18] C. D. Olds, *Contiued Fractions*, Random House, Inc., 1963.

[19] J.-J. Quisquater and C. Couvreur, Fast decipherment algorithm for RSA public key cryptosystem, *Electron. Lett.* **18** (1982), 905–907.

[20] R. L. Rivest, A. Shamir and L. M. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Commun. ACM* **21** (1978), 120–126.

[21] Sa. Sarkar, S. Maitra and Su. Sarkar, *RSA Cryptanalysis with Increased Bounds on the Secret Exponent using Less Lattice Dimension*, Cryptology ePrint Archive, Report 2008/315, 2008, `http://eprint.iacr.org/`.

[22] G. J. Simmons, A "weak" privacy protocol using the RSA crypto algorithm, *Cryptologia* **7** (1983), 180–182.

[23] T. Takagi, A fast RSA-type public-key primitive modulo $p^k q$ using Hensel lifting, *IEICE Trans.* **87-A** (2004), 94–101.

[24] M. J. Wiener, Cryptanalysis of short RSA secret exponents, *IEEE Trans. Inf. Theory* **36** (1990), 553–558.

Received June 10, 2009; revised March 18, 2010.

Author information

M. Jason Hinek, *i*CORE Information Security Lab, Department of Computer Science, University of Calgary, Calgary, AB, T2N 1N4, Canada.
E-mail: `mjhinek+ucalgary@gmail.com`

Charles C. Y. Lam, Department of Mathematics, California State University, Bakersfield, Bakersfield, CA. 93311-1022, USA.
E-mail: `clam@csub.edu`