Dit proefschrift is goedgekeurd door de promotor:

prof.dr.ir. H.C.A. van Tilborg

Copromotor:
dr. B.M.M. de Weger

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Jochemsz, Ellen

Cryptanalysis of RSA variants using small roots of polynomials / door Ellen Jochemsz. –
Eindhoven : Technische Universiteit Eindhoven, 2007.
Proefschrift. – ISBN 978-90-386-1080-1
NUR 919
Subject headings : cryptology
2000 Mathematics Subject Classification: 94A60, 12Y05, 11T71.

Printed by Printservice Technische Universiteit Eindhoven.
Cover: Historical lattices. Design by Verspaget & Bruinink.

# Cryptanalysis of RSA variants using small roots of polynomials

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op donderdag 4 oktober 2007 om 16.00 uur

door

Ellen Jochemsz

geboren te Rijnsburg

# Contents

# 1 Introduction

## 1.1 Cryptology, public key cryptography, and RSA

*Cryptology*, derived from the Greek word kryptos for "hidden" and logos for "word", is often defined as the study of secret writing. It is divided into two branches.

The first is *cryptography* which is mainly concerned with designing cryptosystems. In history, cryptosystems were simply algorithms to encipher and decipher a message. In the current days where computers are a major part of our industry, cryptography also provides solutions to many other security issues, such as authentication (for instance via digital signatures). A quote by the famous cryptographer Ron Rivest is that "cryptography is about communication in the presence of adversaries".

The second branch, *cryptanalysis*, deals with codebreaking, where an adversary tries to decrypt intercepted ciphertexts, or tries to find out who sent an anonymous message, or tries to pose as someone else, or tries to do any other thing that violates the goal of a cryptosystem. Some might say that cryptanalysis is the "dark side" of cryptology, while others believe that researchers should try their best in breaking existing cryptosystems to prevent the real villains from finding the attacks first.

Methods to send messages in a secret way have been around since people started writing, and at the same time, people have been interested in deciphering secret messages. The development of cryptography and cryptanalysis go hand in hand, and many times the course of history has been influenced by whether a cryptosystem used by a specific person or country was breakable or not [41, 67].

Up to 1976, almost all cryptosystems required that the sender and receiver shared a secret key. In these so-called *symmetric* cryptosystems, a message is encrypted by an encryption algorithm, which has both the message and the secret key as input. The output is a ciphertext, that can be decrypted by a decryption algorithm, which has the ciphertext and the same secret key as input.

In 1976, Whitfield Diffie and Martin Hellman introduced[1] the concept of *asymmetric* cryptography in their groundbreaking paper "New Directions in Cryptography" [23]. In

[1] In 1997 it became known that mathematicians of the British intelligence agency GCHQ had invented asymmetric cryptography (James Ellis in 1970), a key exchange protocol analogous to Diffie and Hellman's (Malcolm Williamson in 1974) and an asymmetric cryptosystem analogous to RSA (Clifford Cocks in 1973).

this paper a key exchange protocol is introduced, which can be used by two people to agree on a common secret key over an insecure channel. The paper also inspired the crypto community to look for an asymmetric cryptosystem. That is, an encryption algorithm where the encryption key and the decryption key are not the same, and where the encryption key can be published without giving away the information needed for decryption. In 1978, Ron Rivest, Adi Shamir, and Len Adleman introduced the first asymmetric cryptosystem, the RSA scheme [64]. Almost thirty years after its introduction, RSA is still the most used asymmetric cryptosystem in practice. The RSA scheme and its vulnerabilities are the main topic of this thesis.

In the setup phase of the RSA scheme in its simplest form, a person (whom we shall call Alice from here on) chooses two large prime numbers $p$ and $q$ and computes their product $N = pq$. Moreover, she selects a pair of integers $(e, d)$ such that

$$ed \equiv 1 \mod \phi(N),$$

where $\phi(N) := (p-1)(q-1)$ is Euler's totient function. Alice publishes $(N, e)$ and keeps $(p, q, d)$ private.

Another person (whom we shall call Bob) wishes to send a message $m$ to Alice, where $m$ is represented as an element of $\mathbb{Z}_N^* = (\mathbb{Z}/N\mathbb{Z})^*$. He knows $e$ and $N$ and can therefore encrypt $m$ by computing

$$c \equiv m^e \mod N.$$

Bob sends the ciphertext $c$ to Alice. Alice decrypts $c$ by computing

$$c^d \equiv m^{ed} \equiv m^{1+k\phi(N)} \equiv m \mod N.$$

The last equivalence, valid for all $m$ that are coprime to $N$, is a result of Euler's Theorem (which in turn is a generalization of Fermat's Little Theorem). One can see that indeed, Alice recovers the message $m$ using her secret decryption exponent $d$.

Since the introduction of RSA, it has become the most popular asymmetric cryptosystem, and therefore much research has been done on its vulnerabilities (see for instance [8]). It is essential that the secret information $(p, q, d)$ cannot be extracted efficiently from the public information $(N, e)$, since otherwise anyone could decrypt a message meant for Alice. If the primes $p$ and $q$ are known, then $\phi(N)$ is known and it is easy to compute $d$ using the Extended Euclidean Algorithm. Therefore, it is crucial that factoring the modulus $N$ is hard, and many researchers are trying to factor large RSA moduli $N$. It is not known if there exist integer factorization algorithms whose running time is polynomial in the bitsize of $N$. Currently, the best factoring algorithm, the Number Field Sieve, has running time subexponential in the bitsize of $N$.

However, we may be able to factor an RSA modulus $N$ if we can use additional information. In this thesis, we focus on breaking RSA and RSA variants when an attacker has some additional knowledge on the secret RSA parameters $(p, q, d)$. For instance, an attacker could know that $d$ is not chosen randomly, but selected in a special way. Or an attacker could know that there is a special relation between $p$ and $q$. Such special design criteria are not uncommon, since they can lead to a more efficient decryption or encryption.

Alternatively, an attacker could have obtained a part of the bits of the decryption exponent $d$ by using a so-called *side channel attack*. A side channel attack is a physical attack on an RSA implementation, where an attacker tries to get information on $d$ by connecting a device to the computer or smartcard that is performing the RSA decryption (and measuring the time it takes to run the decryption, or measuring the power consumption during the decryption process).

The results in this thesis show that one must be very careful when using RSA or RSA variants when the RSA parameters have special properties, or when part of the bits of $d$ have leaked. In essence, we show that in some of these variants, a multivariate polynomial appears which has a small root. Finding this small root means that secret RSA parameters of the system are fully exposed, and the factorization of $N$ can be found.

This brings us to the theory (and practice) of finding small roots of polynomials. In 1996, Don Coppersmith [14, 15, 16] introduced methods of finding small modular roots $x_0$ of univariate polynomials $f_N(x)$ modulo some composite integer $N$, and finding small integer roots $(x_0, y_0)$ of bivariate integer polynomials $f(x, y)$. For the polynomials that appear in the cryptanalysis of RSA variants, we sometimes need extensions of Coppersmith's methods to more variables. These extensions are heuristic, in the sense that they rely on an assumption, which turns out to work well in practice, but which must always be tested for specific attack scenarios. The methods enable us to analyze for each polynomial that occurs in an RSA variant, how small the root should be such that it can be found in time polynomial in the bitsize of $N$. In this way, we can see in which cases the additional information that the attacker has is enough to obtain the factorization of $N$ in polynomial time.

## 1.2 Overview of this work

Now that we have introduced the topic of the thesis, we are ready to state our main goal, and give an overview of the thesis and our contributions.

**Research goal:**

> We aim to design new attacks on RSA and RSA variants, which allow us to factor the RSA modulus $N$ in time polynomial in the bitsize of $N$.
>
> In the design of these new attacks, the assumption is made that an attacker has some information on the secret RSA parameters of the system, either obtained from e.g. a side channel attack, or by knowing special design criteria of the RSA variant.

**Motivation of the research:**

Let us briefly discuss the motivation for exploring attacks on RSA in which an attacker has some extra information on the RSA parameters.

As we shall see later, side channel attacks are a serious threat to RSA implementations. These attacks exploit physical characteristics such as the running time or the power consumption of a decryption procedure, to draw conclusions about the bits of the secret key $d$. This may lead to partial leakage of the bits of $d$, and raises the question whether or not this information is enough for an attacker to recover the rest of $d$.

The other attacks that we deal with, namely attacks on RSA variants that have special design criteria, are motivated by the various proposals to speed up RSA. Since RSA is a popular system, researchers are always interested in achieving a more efficient encryption or decryption procedure by choosing the RSA parameters in a special way.

**Organization of this thesis and our main contributions:**

Chapter 2: The RSA cryptosystem

In Chapter 2, we outline the RSA scheme and variants on RSA, and introduce different ways to mount attacks. Moreover, we show that in some of the cryptanalytic results on RSA, polynomials with small roots are used. We end this chapter by introducing two important attacks on RSA with small decryption exponent $d$, by Wiener [75] and Boneh and Durfee [10].

Chapter 3: Small roots of polynomials

Since polynomials with small roots play an important role in the cryptanalysis of RSA, we deal with this topic in a separate chapter (Chapter 3). We introduce Coppersmith's methods of finding small modular roots of univariate polynomials, and of finding small integer roots of bivariate polynomials [14, 15, 16]. We explain how the methods can be extended to more variables, although the techniques will become heuristic in these cases. We will also use the works of Howgrave-Graham [35] and Coron [18], who revisited Coppersmith's methods for small modular roots and small integer roots respectively.

One difficulty in applying a Coppersmith method to a new multivariate polynomial $f$ that appears in cryptanalysis, is the choice of the so-called shift polynomials. This choice depends on the monomials that appear in $f$. We present a general strategy that can be followed, which prescribes which shift polynomials can be used. The general strategy can also be used to give an easy answer to the question: How small should the root be such that it can be found in polynomial time? For small integer roots, this strategy is a generalization of a strategy by Blömer and May for bivariate polynomials [7].

*Parts of this chapter are based on [37], presented at Asiacrypt 2006, which is joint work with Alexander May.*

Chapter 4: Partial key exposure attacks on RSA

In Chapter 4, we introduce the concept of partial key exposure, which is the situation that an attacker has obtained a part of the bits of the secret exponent $d$. We describe the known partial key exposure attacks on RSA by Boneh, Durfee, and Frankel [12] and by Blömer and May [6], and present new ones.

The first new attack deals with the case that $d$ is chosen to be small. In the nineties, it was shown by Wiener [75] that a private exponent $d$ can be found if $d < N^{\frac{1}{4}}$. Verheul and van Tilborg [72] show that Wiener's attack can be extended slightly to $d < N^{\frac{1}{4}+\epsilon}$ for a small $\epsilon$, but at the cost of a workload of $O(N^{2\epsilon})$. Our new partial key exposure attack uses a 2-dimensional lattice and is an extension of the attacks by Wiener and Verheul/van Tilborg to the case of partial key exposure.

The attacks that we describe next show that partial key exposure attacks on RSA exist whenever either $d$ or $e$ is chosen to be significantly smaller than $\phi(N)$.

*Parts of this chapter are based on [36], presented at ISC 2006, which is joint work with Benne de Weger. Other parts of the chapter are based on [25], presented at Eurocrypt 2005, a joint paper with Matthias Ernst, Alexander May, and Benne de Weger.*

Chapter 5: Attacks on RSA-CRT variants

The most popular variant of RSA is called RSA-CRT, which only involves a small change in the decryption process. Instead of computing $c^d \bmod N$ directly, one could use so-called CRT-exponents $d_p \equiv d \mod (p-1)$ and $d_q \equiv d \mod (q-1)$. Then by combining

$$m_p \equiv c^{d_p} \mod p \quad \text{and} \quad m_q \equiv c^{d_q} \mod q$$

using the Chinese Remainder Theorem (CRT), one can also find the original message $m$. A gain in efficiency is caused by the fact that the private CRT-exponents $d_p$ and $d_q$ (and all the other numbers that occur in the process of the modular exponentiation) are the size of $p$ and $q$ instead of the size of $N$.

Moreover, it may be tempting to choose $d_p$ and $d_q$ significantly smaller than $p$ and $q$ to obtain an even faster decryption phase. Wiener suggested to use these small private CRT-exponents instead of small private exponents in the paper where he attacked $d < N^{\frac{1}{4}}$. It has been an open question since whether or not there exist polynomial time attacks on RSA-CRT with small CRT-exponents. In Chapter 5, we answer this question and show that there is an attack on RSA-CRT if $d_p$ and $d_q$ are smaller than $N^{0.073}$.

In a variant on RSA-CRT it is proposed to choose $d_q = d_p - 2$, such that a user only has to store one of the private CRT-exponents. We show that this is unsafe if $d_p < N^{0.099}$.

These new attacks are an addition to the known attacks on RSA-CRT variants by May [52], Bleichenbacher and May [4], Galbraith, Heneghan, and McKee [27] and Sun, Hinek, and Wu [69].

*Parts of this chapter are based on [38], presented at Crypto 2007, others are based on [37], presented at Asiacrypt 2006. Both papers are joint work with Alexander May.*

Chapter 6: Attacks on Common Prime RSA

Another variant on RSA, called Common Prime RSA, is the topic of Chapter 6. In this variant, the primes $p$ and $q$ satisfy a special relation, which causes Wiener's attack to work less well. Therefore, by choosing the primes in this special way, one is able to use a decryption exponent $d$ smaller than $N^{\frac{1}{4}}$. We discuss the known attacks collected in a paper by Hinek [32]. We show a new attack on Common Prime RSA which significantly restricts the number of safe choices for $d$.

*Parts of this chapter are based on [37], presented at Asiacrypt 2006, which is joint work with Alexander May.*

Chapter 7: Conclusions & open questions

Finally, we conclude with an overview of the security of RSA and its variants, and provide guidelines for designers and implementors of new RSA variants. Moreover, we address a number of open questions, either related to finding small roots of polynomials, or to RSA cryptanalysis.

# 2 The RSA cryptosystem

In this chapter, we introduce the RSA cryptosystem and some variants on it. Most of these variants were proposed in attempts to speed up either the encryption phase or the decryption phase of RSA. We describe the various ways to attack RSA schemes in the section on cryptanalysis of RSA.

## 2.1 Basics

The standard RSA scheme [64] is built up as follows.

Let $n$ be a security parameter, usually called the modulus length. Let $p$ and $q$ be two randomly generated primes of about $\frac{1}{2}n$ bits. Take $N := pq$ to be the $n$-bit RSA modulus. Typically, $n = 1024$, although $n = 2048$ is used in practice by more conservative users.

Next, one generates two integers $e$ and $d$ which are each other's inverse modulo $\phi(N) = (p-1)(q-1)$. This can be done by choosing either $e$ or $d$ at random, coprime to $\phi(N)$, and computing the other integer using the *Extended Euclidean Algorithm*. This algorithm, when performed on an integer pair $(a_1, a_2)$, finds integers $b_1$ and $b_2$ such that

$$a_1 b_1 + a_2 b_2 = \gcd(a_1, a_2).$$

So, when for instance $(e, \phi(N))$ is taken as an input, one finds $b_1$ and $b_2$ such that

$$eb_1 + \phi(N)b_2 = 1.$$

Take $b_1 = d$ and $b_2 = -k$. Then an integer pair $(e, d)$ can be found such that

$$ed = 1 + k\phi(N).$$

When Alice has generated the parameters of her RSA system, then the public key cryptosystem works as follows. Alice publishes $(e, N)$ and keeps $(p, q, d)$ secret. If Bob wants to send a message to Alice, he represents it as an integer $m \in (1, N)$ that is coprime to $N$. He then encrypts $m$ by computing

$$c \equiv m^e \mod N,$$

which is possible since he knows both $e$ and $N$.

Alice can decrypt the message by computing

$$c^d \equiv m^{ed} \equiv m^{1+k\phi(N)} \equiv m \mod N.$$

Aside from being an encryption scheme, RSA can also be used as a digital signature scheme. This is important when authentication of the sender of a message is requested. Suppose Alice wants to prove that a certain message is written by her. She can then compute

$$s \equiv m^d \mod N$$

using her private exponent $d$, and send the signature $s$, together with the message $m$. Then, anyone can compute

$$s^e \equiv m^{ed} \equiv m^{1+k\phi(N)} \equiv m \mod N,$$

and check if the output of this computation corresponds to the message $m$ accompanying the signature. This is called the verification of the signature $s$.

In practice, it is often not the message itself that gets signed, but a condensed version of $m$, the outcome of a hash function $H$ when performed on $m$. Naturally, anyone can verify that $H(m) = s^e \bmod N$ is the hash value of the corresponding $m$ if the hash function is publicly known.

## 2.2 RSA variants

Since its introduction in 1978, RSA has become a widely used cryptosystem. Therefore, many people have tried to speed up the process of encrypting/verifying or decrypting/signing a message.

Let us first examine the efficiency of these processes in the case of standard RSA. A standard way to perform this modular exponentiation is by the '*Square-and-Multiply Method*', also known as the 'Repeated Squaring Method'. In order to compute $m^e \bmod N$, one first looks at the bit representation of $e$, that is $(e_{n-1}, \ldots, e_0)$, for $e = \sum_{i=0}^{n-1} e_i 2^i$. So,

$$m^e = m^{e_0+2(e_1+2(e_2+2(e_3+\ldots)))}.$$

One can compute $m^e \bmod N$ in an iterative way by setting initial values $x = 1$, $y = m$, and for $i = 0, 1, \ldots, n-1$:

- if $e_i = 1$ then put $x = x \cdot y \mod N$,
- if $i < n-1$, then put $y = y^2 \mod N$.

After these iterations, output $x = m^e \bmod N$. In total, this so-called 'right-to-left' method involves as many squarings as the bitsize of $e$ and as many multiplications as there are ones in the bit representation of $e$.

Each multiplication or squaring can be performed in time $c \cdot n^2$, where $c$ is a constant. We conclude that the total method takes at most time $2 \cdot \text{bitsize}(e) \cdot cn^2 \leq 2cn^3$. Equivalently, the time for the decryption process is at most $2 \cdot \text{bitsize}(d) \cdot cn^2 \leq 2cn^3$.

Next, we will describe a number of proposed variants on RSA. Most of them focus on speeding up either the encryption (or signature verification) phase or the decryption (or signing) phase of the cryptosystem.

RSA with small public exponent $e$ ("Small-$e$"):

RSA with a small encryption exponent $e$ occurs often in practice. Since the encryption exponent is public, one can choose it to be for instance $e = 3$ or $e = 2^{16} + 1$, which are the most common choices. For $e = 3$, only two multiplications and one squaring are required for the exponentiation in the encryption phase. For $e = 2^{16} + 1$, the number of multiplications and squarings are 2 and 16 respectively. In the case of RSA-Small-$e$, a small $e$ is first fixed in the key generation (after the random choices of $p$ and $q$ have been made). Then, the Extended Euclidean Algorithm computes the corresponding $d$, which will be 'full size' (that is, about the same bitsize as $\phi(N)$) in general.

RSA with small private exponent $d$ ("Small-$d$"):

If a fast decryption/signing phase is needed, for instance on constrained devices like smartcards, one could use RSA with a small private exponent $d$. In that case, first $d$ can be chosen to be of a certain size, after which the corresponding $e$ is determined by the Extended Euclidean Algorithm. In general, $e$ will be 'full size' in this case. As we shall see in the next section on cryptanalysis of RSA, Wiener [75] showed in 1990 that there are polynomial time attacks on RSA with small $d$. Namely, he showed that $d$ can be found in time polynomial in $n = \text{bitsize}(N)$ if $d < N^{\frac{1}{4}}$. Boneh and Durfee extended this attack bound to $d < N^{0.292}$ in 2000 [10]. With $n = 1024$, this is already a much stronger attack than the brute force attack and meet-in-the-middle attack that will be described in Section 2.3.

Standard RSA-CRT ("CRT-Standard") :

A standard way to speed up RSA decryption, as proposed by Quisquater and Couvreur [61], is by splitting up the exponentiation in the decryption phase and using the *Chinese Remainder Theorem*. That theorem says that if two integers $r$ and $s$ are coprime, and we know integers $a_1, a_2$ such that

$$x \equiv a_1 \mod r, \quad x \equiv a_2 \mod s,$$

then the unique $x < rs$ that satisfies both equations can be constructed efficiently.

Instead of computing $c^d \bmod N$ directly, one could use so-called private CRT-exponents $d_p \equiv d \mod (p-1)$ and $d_q \equiv d \mod (q-1)$. Then by combining

$$m_p \equiv c^{d_p} \mod p \quad \text{and} \quad m_q \equiv c^{d_q} \mod q$$

using the Chinese Remainder Theorem, one can also find the original message $m$.

Since $\text{bitsize}(d_p) = \text{bitsize}(d_q) = \frac{1}{2}n$, each of these exponentiations will involve $\frac{1}{2}n$ squarings and at most $\frac{1}{2}n$ exponentiations, so the total number of squarings and multiplications stays the same as in the standard case (but the operands are shorter). A squaring or multiplication modulo $p$ takes time at most $c \cdot (\text{bitsize}(p))^2 = \frac{1}{4}cn^2$. Therefore, if one neglects the time used for the Chinese Remainder Theorem, then it can be concluded that decryption with the Quisquater/Couvreur method is four times faster than the decryption in standard RSA.

RSA-CRT with small public exponent $e$ ("CRT-Small-$e$") :

As in standard RSA, it is possible to choose a small $e$, after which the Extended Euclidean Algorithm finds the corresponding $d$, which in turn is split up in $d_p$ and $d_q$. This is probably the most popular variant of RSA in practice. If one chooses $p$ and $q$ at random, and then a small $e$, then the private CRT-exponents $d_p$ and $d_q$ will be about as long as $p$ and $q$ in general.

RSA-CRT with small private CRT-exponents $d_p, d_q$ ("CRT-Small-$d_p, d_q$") :

Similarly, it is also possible to choose small private CRT-exponents $d_p$ and $d_q$. After fixing $d_p$ and $d_q$, one needs to compute $d$ smaller than $\phi(N) = (p-1)(q-1)$ such that

$$d \equiv d_p \mod (p-1) \quad \text{and} \quad d \equiv d_q \mod (q-1).$$

Since $p-1$ and $q-1$ are not coprime, one cannot use the Chinese Remainder Theorem directly. However, one could do the following.

Pick random primes $p$, $q$ of the same bitlength such that $\gcd(p-1, q-1) = 2$. Choose random odd integers $d_p$, $d_q$ of the same small size, that is $d_p, d_q < N^\beta$ for some $\beta \in (0, \frac{1}{2})$. Use the Chinese Remainder Theorem to compute the unique $x < \frac{(p-1)(q-1)}{4}$ that satisfies

$$x \equiv \frac{d_p - 1}{2} \mod \frac{p-1}{2} \quad \text{and} \quad x \equiv \frac{d_q - 1}{2} \mod \frac{q-1}{2}.$$

Then for $d = 2x + 1$, it holds that $d \equiv d_p \mod (p-1)$ and $d \equiv d_q \mod (q-1)$. From this $d$, one can compute the corresponding $e$ as usual.

Wiener suggested to use these small private CRT-exponents instead of small private exponents in the paper where he attacked $d < N^{\frac{1}{4}}$. It has been an open question since whether or not there exist polynomial time attacks on RSA-CRT with small CRT-exponents.

RSA-CRT with unbalanced primes ("CRT-UnbalancedPrimes") :

In a study of RSA-CRT cases that can be broken, May designed the concept of RSA-CRT with unbalanced primes [52]. We know that in RSA-CRT, the equations

$$ed_p \equiv 1 \mod (p-1), \quad ed_q \equiv 1 \mod (q-1)$$

hold. Suppose that $q = N^\beta$ for some $\beta \in (0, \frac{1}{2})$ is the smaller of the two prime factors of $N$. This implies that the exponentiation modulo $q$ is relatively fast. Now one could also make the exponentiation modulo $p$ fast by choosing $d_p$ to be small. This is basically the variant that May proposes:

- have a modulus that is the product of unbalanced primes $p$, $q$ with $p > q$,
- choose $d_q$ randomly (so $d_q$ will be about as large as the 'small' prime $q$), and
- choose a small $d_p$ to speed up the exponentiation modulo the 'large' prime $p$.

RSA-CRT with small $e$ *and* small $d_p$ and $d_q$ ("CRT-BalancedExponents") :

In 2005, two independent papers [28, 70] proposed the same RSA variant, namely one that uses RSA-CRT in which both $e$ and $d_p$ and $d_q$ are smaller than standard. Let us show how this can be achieved in the key generation phase. We want $e, d_p, d_q, p, q, k_p, k_q$ such that

$$ed_p = 1 + k_p(p-1), \quad ed_q = 1 + k_q(q-1),$$

with $e$ of bitsize $\alpha n$, $d_p$ and $d_q$ of bitsize $\beta n$, and $p$ and $q$ of bitsize $\frac{1}{2}n$. Here, $\alpha \in (0,1)$ and $\beta \in (0, \frac{1}{2})$. By this construction, $k_p$ and $k_q$ are of bitsize $(\alpha + \beta - \frac{1}{2})n$.

First, one chooses random $d_p$, $d_q$ of the right bitsize, and $k_p, k_q$ of the right bitsize satisfying $\gcd(d_p, k_p) = \gcd(d_q, k_q) = \gcd(k_p, k_q) = 1$. Next, one computes $e'$ using CRT such that

$$e' \equiv d_p^{-1} \mod k_p \quad \text{and} \quad e' \equiv d_q^{-1} \mod k_q.$$

Since $e'$ is now smaller than $k_p k_q$ which is $(2\alpha + 2\beta - 1)n$ bits, compute $e := e' + c \cdot k_p k_q$ for some $c$ of bitsize $(1 - \alpha - 2\beta)n$. Finally, put $p := \frac{ed_p - 1}{k_p}$ and $q := \frac{ed_q - 1}{k_q}$, and check if $p$ and $q$ are both prime. If not, repeat the whole procedure until the $p$ and $q$ that are obtained are both prime.

Note that $c$ must be positive, so it is needed that $\alpha < 1 - 2\beta$. This key generation algorithm is a slight variation to the one proposed by Galbraith/Heneghan/McKee [28]. In their algorithm, they first choose $e$, $k_p$, and $k_q$, and then compute $d_p$ and $d_q$ as the inverses of $e$ mod $k_p$ and mod $k_q$. This requires $d_p > k_p$ and $d_q > k_q$, and thus $\alpha < \frac{1}{2}$, which is quite restrictive. However, if $\alpha < \frac{1}{2}$, then the method of [28] should be preferred, since one can generate $p$ and $q$ separately, and the method does not rely on two integers that have to be prime at the same time.

The key thing to note in the generation of these balanced exponents is the fact that $p$ and $q$ are generated last (and then tested for primality). Hence, the modulus $N$ is a product of two special primes instead of two randomly chosen ones, and therefore the number of possible $N$ is less than usual, but it is unknown whether this can be exploited.

RSA-CRT with small difference $d_p - d_q$ ("CRT-Qiao&Lam") :

In a proposal to save on both memory and decryption time on constrained devices like smartcards, Qiao and Lam [60] proposed to use RSA-CRT with small $d_p$, *and* to use $d_q = d_p - 2$. In this way, one profits from the fast decryption method of CRT-Small-$d_p, d_q$, while one has to store only one of the two private CRT-exponents.

RSA with special $p, q$ ("Small Prime Difference", "Common Prime", etc.):

Some settings of RSA have primes $p$ and $q$ that are generated in a special (non-random) way, either because of a faulty implementation, or on purpose. Examples include:

- A small prime difference $p - q$:
  It is known that it is unsafe to use primes with a small prime difference. Besides Fermat's factoring attack (see for instance [74]), there are attacks on the RSA setting with small $d$ and a small prime difference by de Weger [74].

- Primes $p$, $q$ such that $\gcd(p-1, q-1) = 2g$, for $g$ a large prime:
  As we shall see in Chapter 6, the attacks by Wiener and Boneh/Durfee on small $d$ work less well for this variant called Common Prime RSA. Therefore, it might be possible to use a $d < N^{\frac{1}{4}}$ if $g$, the prime factor that $p-1$ and $q-1$ are sharing, is large enough (though not too large, to avoid other attacks).

- Primes $p$ and $q$ that share a block of least significant bits:
  As we shall see in Chapter 4, this variant makes a partial key exposure attack by Boneh, Durfee, and Frankel harder to perform, and was therefore proposed as an interesting RSA variant by Steinfeld and Zheng [68].

- Partial knowledge of the primes $p$ and $q$:
  A consequence of the variant by Steinfeld and Zheng that we have just sketched is that an attacker can find the least significant bits that $p$ and $q$ share. In any variant where an attacker knows a set of either most significant bits (MSBs) or least significant bits (LSBs) of one of the secret primes $p$ and $q$, one must beware of an important result by Coppersmith [16]. This result states that $N$ can be factored efficiently if the known MSB or LSB part of $p$ is at least as big as $N^{\frac{1}{4}}$. This result will be discussed in Section 4 (in Theorem 4.1), since it is also the basis of the first partial key exposure attacks on RSA by Boneh, Durfee, and Frankel [12].

RSA with moduli $N = p_1 \cdot \ldots \cdot p_r$ or $N = p^r q$ ("Multi-prime"/"Takagi"):

We have already mentioned that the decryption/signing process in RSA can be made more efficient by performing exponentiations modulo the prime factors of $N$, and then combining these with the Chinese Remainder Theorem. It follows easily that using RSA with more, and smaller prime factors should improve the efficiency of this decryption phase even more.

Therefore, RSA variants have been proposed that use $N = p_1 \cdot \ldots \cdot p_r$, where the $p_i$ are distinct primes of equal bitsize, or $N = p^r q$, for primes $p$ and $q$ of equal bitsize and $r$ a small integer. The first variant is called "Multi-prime RSA", the second "Takagi's RSA" (also known as "Multi-power RSA") since it was proposed by Takagi [71]. Obviously, one necessary (though not sufficient) condition is that the prime factors are large enough to avoid attacks using the factorization methods.

## 2.3 Cryptanalysis

As there are many different ways to attack RSA, we divide the attacks into the following categories:

1. *Factoring $N$*:
   Given an RSA modulus $N$ of bitsize $n$, the goal is to find its prime factorization. In these attacks, an adversary gets no public exponent $e$, no ciphertext $c$, only the composite integer $N$. Currently, the best (general) factorization method, the Number Field Sieve [47], has a number of bit operations that is bounded by

   $$\exp\left((1.902 + o(1))\ln(N)^{\frac{1}{3}}(\ln(\ln(N)))^{\frac{2}{3}}\right)$$

   for $N \to \infty$. In special cases, namely when one is looking for a small prime factor of a number $N$, the Elliptic Curve Method (ECM) [49] could be used. Currently, the largest factor found by the ECM has 222 bits. We refer to [46] for more details on integer factorization.

2. *Brute force and meet-in-the-middle attacks on $d$ or $d_p, d_q$*:
   The brute force and meet-in-the-middle attacks on $d$ (or, in the RSA-CRT case, $d_p$ and $d_q$) show that one should choose these secret values large enough such that an attacker is not able to find them by simply trying all possibilities. If a message $m$ is very small, then one might try to encrypt all possible plaintexts $m$ with the public exponent $e$, and check if the result is an intercepted ciphertext. Similarly, if one knows that $d$ is chosen small, one might try all possibilities for $d$ to decrypt the ciphertext. All of these attacks are so-called *brute force attacks*, and are simply attacks using exhaustive search. To say that a certain parameter choice is safe against brute force attacks, we need to quantify the maximal amount of operations that an attacker is able to perform. A usual choice for this is $2^{80}$ (see for instance report on the hardness of computational problems in cryptography [24], where it is said that an exhaustive search of 80 bits is on the edge of what is not doable today). Hence, all secret RSA parameters should be at least 80 bits long to avoid brute force attacks.

   A more advanced category of attacks is called *meet-in-the-middle attacks*. In these attacks, there is a trade-off between storage and running time. For a detailed description of meet-in-the middle attacks on RSA and RSA-CRT, we refer to [51]. Here, we show how it works on a small $d$. Given a message-ciphertext pair $(m, c)$, such that $c \equiv m^e \mod N$, assume that $e$'s inverse $d$ has an upper bound $D$. Then, $d$ is built up as

   $$d = \lceil\sqrt{D}\rceil \cdot d_0 + d_1.$$

   Make a sorted list of all couples $(d_0, c^{\lceil\sqrt{D}\rceil \cdot d_0})$ for $d_0 < \sqrt{D}$. For $d_1$ running from 0 to $\sqrt{D}$, compute $m \cdot (c^{-1})^{d_1}$, and check if $(d_0, m \cdot (c^{-1})^{d_1})$ in the list. If so, then output $d = \lceil\sqrt{D}\rceil \cdot d_0 + d_1$. Hence, at the cost of a list with about $\sqrt{d}$ entries, the number of tries to find $d$ can be reduced to about $\sqrt{d}$.

3. *Attacks that involve plaintext-ciphertext pairs* $(m, c)$:
   These attacks use the knowledge that ciphertexts $c$ are computed as the $e$-th power of an unknown message $m$ modulo $N$. Therefore, they fall into the category of so-called "message recovery attacks", instead of the "key-recovery attacks" that are the main topic of this thesis. Examples include

   - Håstad attack [31]: It is unsafe to send the same message $m$ to more recipients that all use RSA with $e = 3$ (although they have different moduli).
   - Franklin-Reiter attack [17]: With $e = 3$, it is unsafe to send related messages $m_1$, $m_2$, with $m_1 \equiv f(m_2) \bmod N$ for some linear polynomial $f$.
   - Coppersmith short pad attack [16]: Suppose two messages $m_1$ and $m_2$ are essentially the same message $m$, but concatenated with different (unknown) paddings $r_1$ and $r_2$. If the paddings are at most $b = \lfloor n/e^2 \rfloor$ bits long, and the original message $m$ is at most $n - b$ bits, then an attacker can find $m$ from the ciphertexts, $e$ and $N$.

   All of the above attacks are described in Boneh's survey paper [8].

4. *Implementation attacks*:
   Even if a cryptosystem is flawless in theory, vulnerabilities can arise when it is implemented.

   - Side channel attacks: Side channel attacks take advantage of implementation-specific characteristics to recover the secret exponent $d$ that is involved in the computation of a decryption or signature. This characteristic information can be extracted by timing the decryption/signing process, by examining the power consumption of the process, etc. In the case of RSA, this could mean that an attacker examines the power consumption of a device that is applying the Square-and-Multiply Method for the decryption, and tries to distinguish per iteration if a squaring and a multiplication occur ($d_i = 1$) or if only a squaring occurs ($d_i = 0$).

     These types of attacks have become an important part of the research on RSA security in practice, since Kocher introduced his timing attacks [42] in 1996. Other main results in the area include the introduction of simple and differential power analysis [43] by Kocher, Jaffe, and Jun, and the introduction of attacking implementations by inducing faults in specific iterations by Boneh, DeMillo, and Lipton [9]. For an overview on side channel attacks, we refer to [62, 39]. For the motivation of the research in Chapter 4, on partial key exposure attacks, it is important to remark that some side channel attacks are able to reveal only a part of the secret exponent $d$ [22].

- Bleichenbacher's first attack on PKCS-1: In an old version of PKCS-1 (Public Key Cryptography Standard 1), an encryption of a message $m$ was in fact an encryption of the following data string:

| 02 | random padding | 00 | $m$ |
|---|---|---|---|

  In [3], Bleichenbacher shows that this causes problems when a protocol decrypting a ciphertext $c$ outputs an error message when the initial block does not consist of the bytes 0 and 2. Basically, an attacker can intercept a ciphertext $c$, and send $c' \equiv rc \bmod N$ to be decrypted, for some random $r$.

  Now the attacker will learn whether or not the 16 most significant bits are equal to 02. Hence, the attacker has a method that tells him if the decryption of a chosen ciphertext has the correct initial block. Bleichenbacher shows that this is enough to decrypt $c$.
- Bleichenbacher's second attack on PKCS-1: At the rump session of Crypto'06, Bleichenbacher showed another attack on an implementation of PKCS-1, which shows that in some cases, an RSA signature can be forged if a public exponent $e = 3$ is used. In order for an RSA signature to be accepted, it must look like

| standard PKCS-1 padding | bytes in ASN.1 format | hash of the signed data |
|---|---|---|

  after the cube root of the signature is taken. The second block indicates which hash algorithm is used, and how long the hash value is. Now the attack applies to implementations of this signature verification that fail to check if there are bits in the data string after the hash. If one can submit a signature of which the cube root is

| standard PKCS-1 padding | bytes in ASN.1 format | hash | extra bits |
|---|---|---|---|

  then the signature is accepted as valid. An attacker can choose the extra bits freely in order to create a perfect cube.

5. *Factoring $N$ with extra information on the RSA parameters*:
   These attacks are the main topic of this thesis. We have seen in the previous section that many RSA variants have special design criteria, that can help an adversary perform an attack. Also, as a result of the side channel attacks mentioned above, it is possible that an adversary has learned a part of the bits of $d$. Then, we could ask in which cases this so-called partial key exposure is enough to retrieve the rest of $d$. As opposed to the attacks in category 2 of this list, we do not use encryptions $c$ of messages $m$. Instead, we focus on the known relations between the RSA parameters, such as the so-called RSA key equation

$$ed = 1 + k\phi(N), \text{ or equivalently: } \quad ed = 1 + k(N + 1 - (p + q)).$$

In the chapters that follow, many known attacks will be described on the RSA variants, preceding the description of the new attacks that we have found on these variants. The new attacks include partial key exposure attacks on RSA-Small-$e$ and RSA-Small-$d$, the first polynomial time attack on RSA-CRT-Small-$d_p, d_q$, and new attacks on RSA-CRT-Qiao&Lam and Common Prime RSA.

Since this work does not contain new attacks on RSA with small prime difference, Multi-prime RSA or Takagi's RSA, we refer to [74], [33], and [13, 55] respectively for recent attacks on these variants.

We now proceed by introducing the two most important known attacks in this area, namely the attacks by Wiener [75] and Boneh and Durfee [10] on RSA-Small-$d$.

Wiener's Attack:

In 1990, Wiener showed the following result.

**Theorem 2.1 (Wiener, [75])**
*Let $N = pq$ be an RSA modulus, with $q < p < 2q$, and let $d < \frac{1}{3}N^{\frac{1}{4}}$. Given $N$, and $e$ such that $ed = 1 \bmod \phi(N)$, one can recover $d$ in time polynomial in the bitsize of $N$.*

**Proof.**
There exists an integer $k$ such that

$$ed = 1 + k\phi(N).$$

Therefore,

$$\frac{e}{\phi(N)} - \frac{k}{d} = \frac{1}{d\phi(N)},$$

which means that $\frac{k}{d}$ is a good approximation of $\frac{e}{\phi(N)}$. Although the value of $\phi(N)$ is unknown, it is known that

$$\phi(N) = (p-1)(q-1) = N + 1 - (p+q), \text{ so } |N - \phi(N)| < 3N^{\frac{1}{2}}.$$

Therefore, one would like to find $\frac{k}{d}$ as an approximation of the known fraction $\frac{e}{N}$. Then,

$$\begin{aligned}\left|\frac{e}{N} - \frac{k}{d}\right| &= \left|\frac{ed - kN}{dN}\right| = \left|\frac{(ed - k\phi(N)) + k(\phi(N) - N)}{dN}\right| = \left|\frac{1 - k(N - \phi(N))}{dN}\right| \\ &\leq \left|\frac{3kN^{\frac{1}{2}}}{dN}\right| = \left|\frac{3k}{dN^{\frac{1}{2}}}\right|.\end{aligned}$$

Since $k\phi(N) < ed$ and $e < \phi(N)$, it follows that $k < d < \frac{1}{3}N^{\frac{1}{4}}$. So,

$$\left|\frac{e}{N} - \frac{k}{d}\right| \leq \left|\frac{3k}{dN^{\frac{1}{2}}}\right| < \left|\frac{1}{dN^{\frac{1}{4}}}\right| < \frac{1}{3d^2}.$$

At this point, we need to recall some facts from the theory of continued fractions. The *continued fraction representation* of a real number $x$ is $[a_0, a_1, \ldots]$ for

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \ldots}}.$$

The fraction

$$\frac{p_i}{q_i} = [a_0, \ldots, a_i] = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ldots + a_i}}}$$

is called the $i$th *convergent* of $x$.

A classical theorem by Legendre [45] (for a recent reference, see [30, Theorem 184]) states that all fractions $\frac{a}{b}$ such that

$$\left| x - \frac{a}{b} \right| < \frac{1}{2b^2}$$

are obtained as convergents of $x$. Since $k$ and $d$ are coprime, $\frac{k}{d}$ can be found as one of the at most $n = \text{bitsize}(N)$ convergents of $\frac{e}{N}$.

□

In Section 1.1, we have mentioned that we focus on attacks on RSA which factor $N$ in polynomial time when we are given extra information on the secret parameters. However, we have only shown so far that one can recover $d$ in polynomial time. After applying Wiener's attack, finding the factorization of $N$ is easy, since we know both $d$ and $k$, so we can find the correct value of $\phi(N) = N + 1 - (p + q)$ after which we can solve $p$, $q$ from

$$\begin{cases} \phi &= N + 1 - (p + q), \\ N &= pq. \end{cases}$$

Suppose we could only retrieve $d$ and not $k$ in polynomial time. A proof of the following theorem, stating that knowledge of $d$ also allows one to factor $N$ in polynomial time, can be found in [8].

**Theorem 2.2**
*Let $N = pq$ be an RSA modulus. Suppose integers $e, d > 1$ are known such that $ed \equiv 1 \mod \phi(N)$. Then $N$ can be factored in probabilistic polynomial time.*

Here, probabilistic polynomial time means that the method involves a random $g \in \mathbb{Z}_N^*$, and succeeds in finding the factorization of $N$ (in polynomial time) with probability at least $\frac{1}{2}$. If the factorization fails, one simply has to try other choices for $g$.

Recently, May showed that this equivalence of finding $d$ and factoring $N$ is deterministic polynomial time for balanced primes [54], which was extended to the case of unbalanced primes by Coron and May in [20].

Boneh/Durfee's Attack:

Ten years after the publication of Wiener's attack, Boneh and Durfee showed an improvement on the attack bound to $d < N^{0.292}$ [10].

Their method is not based on continued fractions, but is one of the first that is based on the theory of finding small roots of polynomials by Coppersmith. This method involves lattices and lattice basis reduction, and resultants or Gröbner bases. Often, the attacks that use the method are not provable but heuristic, although the heuristic seems to perform well in practice. Since Coppersmith's work on finding small roots of polynomials is the topic of the next chapter, we will treat Boneh and Durfee's attack in detail later.

For now, we shall only show which polynomial with a small root appears in the attack of Boneh and Durfee. It is derived directly from the RSA key equation, namely

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

Suppose one looks at this equation modulo $e$, and replaces $k$ and $p+q-1$ by the unknowns $x$ and $y$. Then, one would like to find the root $(x_0, y_0) = (k, p + q - 1)$ of the polynomial $f_e(x, y) = 1 + x(N - y)$ modulo $e$. The root can be considered 'small' since

$$|x_0| = |k| < d = N^{\beta} \text{ and } |y_0| = |p + q - 1| < 3N^{\frac{1}{2}}.$$

How small roots like this one can be found by a Coppersmith method, will be explained in the next chapter.

# 3
# Small roots of polynomials

In this chapter, we introduce the tools to solve the problem of finding small roots. In this thesis, 'finding small roots' means finding an explicit, numerical, and exact description of all roots of a polynomial that are bounded in size by some upper bound.

We start by describing the necessary preliminaries on lattices. After that, we describe Coppersmith's methods of finding small modular roots and small integer roots of polynomials. We end this chapter with a general strategy that can be applied on any given polynomial.

Parts of this chapter (and most of all the new general strategy in Section 3.3) are based on [37], which is joint work with Alexander May.

## 3.1 Preliminaries

Lattices:

Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_\omega \in \mathbb{R}^m$ be linearly independent (row) vectors, where $m$ and $\omega$ are integers such that $m \geq \omega$. A *lattice* $L$ is described as the set of vectors in $\mathbb{R}^m$ that are integer linear combinations of the *basis vectors* $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_\omega$. Formally,

$$L := \left\{ \mathbf{v} \in \mathbb{R}^m \,\middle|\, \mathbf{v} = \sum_{i=1}^{\omega} a_i \mathbf{b}_i, \text{ for } a_i \in \mathbb{Z} \right\}.$$

We usually say that $L$ is the lattice spanned by the rows of the matrix

$$\Gamma = \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_\omega \end{pmatrix}.$$

The dimension of $L$ is $\dim(L) := \omega$, and we call $L$ a full rank lattice if $m = \omega$. If $L$ has full rank, then the determinant of $L$ is $\det(L) := |\det(\Gamma)|$, and though there are infinitely many bases possible, the determinant is always the same.

We are interested in finding a basis of small, so-called *reduced* basis vectors. A small, or short, lattice vector is a vector $\mathbf{v}$ in $L$ such that its Euclidean norm $\|\mathbf{v}\|$ is relatively small.

The following theorem by Minkowski deals with the shortest nonzero vector in a lattice $L$. For details on the theorem, one could look at the survey of Nguyen and Stern [59].

**Theorem 3.1 (Minkowski, [57])**
*Every lattice $L$ of dimension $\omega$ contains a nonzero vector $\mathbf{v}$ that satisfies $\|\mathbf{v}\| \leq \sqrt{\omega}\det(L)^{\frac{1}{\omega}}$.*

Unfortunately, finding the shortest nonzero vector in a lattice is very hard in general. However, we can use *LLL reduction* designed by Lenstra, Lenstra, and Lovász [48] to find a whole basis of lattice vectors which are relatively small in norm.

The method of Lenstra, Lenstra, and Lovász is closely related to the Gram-Schmidt procedure of computing an orthogonal basis of the same determinant. Given a set of independent vectors $\{\mathbf{b}_1, \ldots, \mathbf{b}_\omega\}$, the Gram-Schmidt procedure constructs a set of orthogonal vectors $B^* = \{\mathbf{b}_1^*, \ldots, \mathbf{b}_\omega^*\}$, such that

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{ij}\mathbf{b}_j^*, \text{ with } \mu_{ij} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\|\mathbf{b}_j^*\|^2}.$$

For a (full rank) lattice $L$,

$$\det(L) = \prod_{i=1}^{\omega} \|\mathbf{b}_i^*\|,$$

however $B^*$ is typically not a basis of the same lattice anymore, and therefore an adaptation of the Gram-Schmidt procedure is needed.

When the LLL reduction algorithm is performed on an $\omega$-dimensional lattice $L$, it outputs (in time polynomial in $\omega$ and the bitsize of the entries of the basis matrix $\Gamma$) a basis $\{\mathbf{r}_1, \ldots, \mathbf{r}_\omega\}$ which is *LLL reduced*. This means that, if the Gram-Schmidt procedure is performed on the reduced basis, we get

$$||\mathbf{r}_i|| \leq 2^{\frac{j-1}{2}}||\mathbf{r}_j^*||, \text{ for } 1 \leq i \leq j \leq \omega. \tag{3.1}$$

It follows that

$$\prod_{i=1}^{\omega} ||\mathbf{r}_i|| \leq 2^{\frac{\omega(\omega-1)}{4}} \det(L).$$

Hence, if all reduced basis vectors would be approximately of equal length, then the norm of every $\mathbf{r}_i$ would be about the size of $\det(L)^{\frac{1}{\omega}}$. However, this is not always the case. A general result on the size of the individual reduced basis vectors, of which a proof can be found in [53], is stated in the following theorem.

**Theorem 3.2**
*Let $L$ be a lattice of dimension $\omega$. The reduced basis vectors $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_\omega\}$ that the LLL algorithm outputs satisfy*

$$||\mathbf{r}_1|| \leq ||\mathbf{r}_2|| \leq \ldots \leq ||\mathbf{r}_i|| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(L)^{\frac{1}{\omega+1-i}} \quad \textit{for all } 1 \leq i \leq \omega.$$

In the case that all reduced basis vectors are approximately of the same length, that is, if there are no exceptionally small lattice vectors, then we call such a lattice *balanced*. Let us formalize this property in the following assumption.

**Assumption 3.3 (Balancedness of a lattice)**
*The reduced basis vectors of a lattice $L$ have a norm of size* $\det(L)^{\frac{1}{\omega}}$.

As noted, this assumption holds for most (random) lattices, but special cases of lattices with 'extremely small' basis vectors certainly exist, and for these unbalanced lattices we are restricted to the general result of Theorem 3.2.

Special case; 2-dimensional lattices:

We define a 2-dimensional lattice $L$ as the set of all integer linear combinations of two linearly independent basis vectors $\{\mathbf{b}_1, \mathbf{b}_2\}$ (we represent all vectors as row vectors). To find a reduced basis $\{\mathbf{r}_1, \mathbf{r}_2\}$ one can use the *Lagrange reduction algorithm* [44] (for a more recent reference, see for instance [65, Chapter 4]), which is simply a generalization of Euclid's algorithm. The reduced basis found by Lagrange's algorithm is guaranteed to contain the smallest nonzero vector of the lattice.

We occasionally use the following notation for size-computations in this thesis. With $u \approx N^{\lambda}$, we mean that $u$ 'has the size of' $N^{\lambda}$, that is $|u| = C_u N^{\lambda}$ for some number $C_u$ that does not deviate much from 1 (relative to $N$). In other words, $C_u \in [N^{-\epsilon}, N^{\epsilon}]$ for some very small $\epsilon$. Naturally, $(v_1, v_2) \approx (N^{\lambda_1}, N^{\lambda_2})$ is a short notation for $v_1 \approx N^{\lambda_1}$ and $v_2 \approx N^{\lambda_2}$.

When we reduce the basis $\{\mathbf{b}_1, \mathbf{b}_2\}$ to $\{\mathbf{r}_1, \mathbf{r}_2\}$, with $\mathbf{r}_1$ the smaller reduced basis vector and $\mathbf{r}_2$ the larger reduced basis vector, it follows from (3.1) that

$$||\mathbf{r}_1|| \cdot ||\mathbf{r}_2|| \leq \sqrt{2} \cdot ||\mathbf{r}_1^*|| \cdot ||\mathbf{r}_2^*|| = \sqrt{2}\det(L).$$

So, we assume $||\mathbf{r}_1|| \approx a^{-1}\det(L)^{\frac{1}{2}}$ and $||\mathbf{r}_2|| \approx a\det(L)^{\frac{1}{2}}$ for some $a \geq 1$. Hence,

$$\Gamma_{\text{red}} = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \approx \det(L)^{\frac{1}{2}} \cdot \begin{pmatrix} a^{-1} & a^{-1} \\ a & a \end{pmatrix}.$$

If the two reduced basis vectors $\mathbf{r}_1$, $\mathbf{r}_2$ are 'nearly equal' in length, that is when $a$ does not deviate much from 1, then all elements of $\Gamma_{\text{red}}$ are of size $\det(L)^{\frac{1}{2}}$. However, it is also possible that there is one 'extremely small' basis vector, which makes the lattice 'unbalanced'.

When dealing with 2-dimensional lattices, we sometimes make the assumption that the lattice is 'balanced', which is true in most cases. If this assumption is made in a specific attack scenario, then we test its validity in experiments.

**Assumption 3.4 (Balancedness of 2-dimensional lattices)**
*The reduced basis vectors given by the rows of* $\Gamma_{red}$ *have a norm of size* $\det(L)^{\frac{1}{2}}$. *In other words, the parameter $a$ that describes the unbalancedness of the lattice is near to* 1.

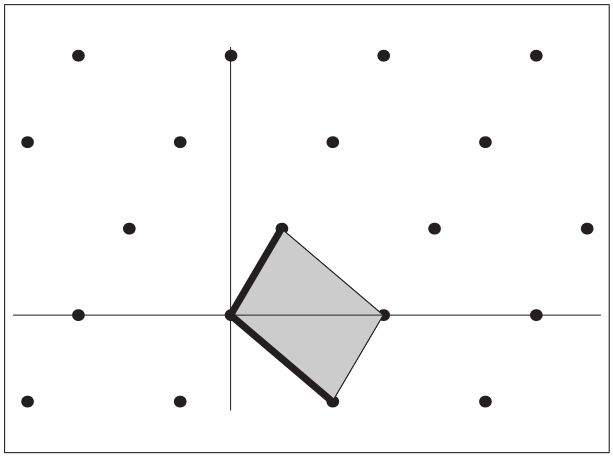

Figure 3.1: $a \approx 1$

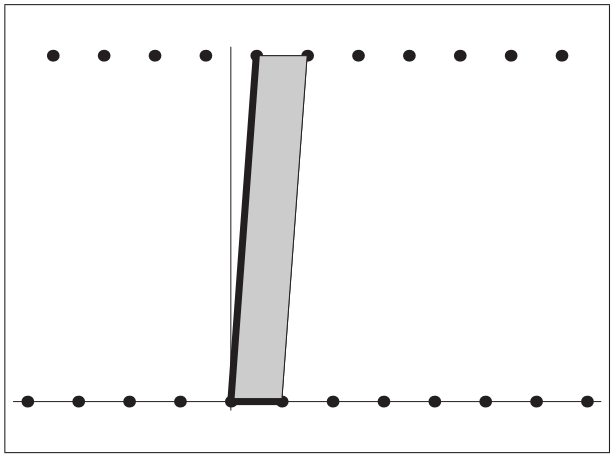

Figure 3.2: $a \gg 1$

## 3.2 Introduction to Coppersmith's method

In [14, 15, 16], Coppersmith describes rigorous techniques to find small modular roots of univariate polynomials and small integer roots of bivariate polynomials. The methods extend to more variables, however, this makes the methods heuristical as we shall see later.

Let us start with some helpful notations. Let $f_N(x) := \sum_i a_i x^i$ be a univariate polynomial with coefficients $a_i \in \mathbb{Z}_N$. Let $f(x,y) := \sum_{i,j} a_{ij} x^i y^j$ be a bivariate polynomial with coefficients $a_{ij} \in \mathbb{Z}$. The terms $x^i$ of $f_N$ and $x^i y^j$ of $f$ with nonzero coefficients are called monomials. The norm of a polynomial $f_N$ or $f$ is defined to be the Euclidean norm of the coefficient vector of the polynomial. Hence, $\|f_N\|^2 := \sum_i a_i^2$, and $\|f\|^2 := \sum_{i,j} a_{ij}^2$. The definitions for multivariate $f_N$ and $f$ are analogous, although we use the notation $f_N(x_1, \ldots, x_v)$ and $f(x_1, \ldots, x_v)$ if we have more than three variables.

We study the problems of

- finding a root $x_0$ of $f_N(x)$ modulo $N$, where $N$ has an unknown factorization and $x_0$ is known to be small: we know an upper bound $X$ such that $|x_0| < X$,
- finding an integer root $(x_0, y_0)$ of $f(x,y)$, where $(x_0, y_0)$ is known to be small: we know upper bounds $X$, $Y$ such that $|x_0| < X, |y_0| < Y$.

It is clear that in general, roots of $f_N(x)$ modulo $N$ or integer roots of $f(x,y)$ cannot always be found in polynomial time. For instance, if the root $m$ of $f_N(x) = x^e - c$ could be found in polynomial time, then one could efficiently decrypt RSA ciphertexts $c$. Equivalently, if one could find the integer root $(p,q)$ of $f(x,y) = xy - N$ in polynomial time, then one could factor efficiently. However, finding *small* roots may be possible in polynomial time, and finding the maximal $X$ for which this can be done for a specific polynomial $f_N(x)$ (or the maximal $X, Y$ for a polynomial $f(x,y)$) is the goal of the work originated by Coppersmith.

Small modular roots of univariate polynomials $f_N(x)$:

The idea behind Coppersmith's method for finding a small modular root $x_0$ of a polynomial $f_N(x)$ is to reduce this problem to finding the same small root $x_0$ of a polynomial $h(x)$ over the integers.

To construct this polynomial $h(x)$, we first fix an integer $m$ and construct a set of univariate polynomials $g_{jk}$:

$$g_{jk}(x) := x^j (f_N(x))^k N^{m-k}, \text{ for } k = 0, \ldots, m \text{ and some choice for } j.$$

It is important to note that all $g_{jk}$ share the root $x_0$ modulo $N^m$. Thus, an integer linear combination $h(x)$ of the different $g_{jk}$'s also has the root $x_0$ modulo $N^m$. Now suppose we know that $|h(x_0)| < N^m$. Then it follows that $h(x)$ must have the root $x_0$ over the integers.

Since an upper bound $X$ is known for $|x_0|$, the coefficients of the polynomial $h(xX) := \sum_i h_i X^i x^i$ can be used as an indication for the size of the terms in $h(x_0)$.

To see under which conditions we can conclude that the polynomial $h(x)$ has the root $x_0$ over the integers (instead of modulo $N^m$), we use a theorem by Howgrave-Graham, who reformulated Coppersmith's ideas of finding modular roots in [35].

**Theorem 3.5 (Howgrave-Graham, [35])**
*Let $h(x) \in \mathbb{Z}[x]$ be an integer polynomial consisting of at most $\omega$ monomials. Suppose that*

(1) $h(x_0) \equiv 0 \mod R$ *for some* $|x_0| < X$ *and some positive integer* $R$, *and*

(2) $||h(xX)|| < \frac{R}{\sqrt{\omega}}$.

*Then $h(x_0) = 0$ holds over the integers.*

**Proof.**
Let $h(x) := \sum_i b_i x^i$. Then,

$$|h(x_0)| = \left| \sum_i b_i x_0^i \right| \leq \sum_i |b_i x_0^i| \leq \sum_i |b_i X^i|.$$

Now, since $\|h(xX)\|$ is the Euclidean norm of the vector $(b_0, b_1 X, \ldots, b_n X^n)$, it holds that

$$\frac{R}{\sqrt{\omega}} > \|h(xX)\| = \sqrt{\sum_i (b_i X^i)^2}.$$

It can be concluded that

$$\sum_i |b_i X^i| \leq \sqrt{\omega} \cdot \sqrt{\sum_i (b_i X^i)^2} = \sqrt{\omega} \cdot \|h(xX)\| < R.$$

It follows that $|h(x_0)| < R$. But since $h(x_0) \equiv 0 \mod R$, it must hold that $h(x_0) = 0$.

□

Now suppose we can find a polynomial $h(x)$ as an integer linear combination of $g_{jk}$'s that satisfies

$$\|h(xX)\| < \frac{N^m}{\sqrt{\omega}},$$

where $\omega$ is the number of monomials of $h$. Then Theorem 3.5 tells us that we can find the root $x_0$ by solving $h(x) = 0$ over the integers. This leaves us the problem of finding $h$.

To find a polynomial $h(x)$ whose coefficients are small enough to satisfy Howgrave-Graham's bound, we use lattices and lattice basis reduction. We let the coefficient vectors of the polynomials $g_{jk}(xX)$ be the basis of a lattice $L$. After applying LLL reduction to the lattice basis of $L$, we obtain a set of small vectors that correspond to polynomials $r_1(xX), \ldots, r_\omega(xX)$, where $\omega$ is the dimension of the lattice. If $L$ has full rank, then the $r_i$ have at most $\omega$ monomials. From Theorem 3.2, we know that

$$\|r_1(xX)\| \leq 2^{\frac{\omega-1}{4}} \det(L)^{\frac{1}{\omega}}.$$

Thus, if $2^{\frac{\omega-1}{4}} \det(L)^{\frac{1}{\omega}} < \frac{N^m}{\sqrt{\omega}}$ holds, then the LLL reduction gives us a polynomial $h(x) = r_1(x)$ which satisfies Howgrave-Graham's bound.

It can be seen that, in order to satisfy the bound above, the main goal in choosing the $g_{jk}$ is to keep the determinant of the lattice $L$ low. Coppersmith described a way to choose these so-called shift polynomials $g_{jk}$ for several polynomials $f_N(x)$. Since we describe a general way of choosing the shift polynomials for multivariate polynomials $f_N(x_1, \ldots, x_v)$ in Section 3.3.1, we postpone the description of how to build the lattice $L$ to that section.

We note that Coppersmith's modular method for one variable is a provable method. As we shall see next, it can be extended to the multivariate case, but only when we introduce an assumption, which makes the method heuristic.

Small modular roots of multivariate polynomials $f_N(x_1, \ldots, x_v)$ :

As Coppersmith remarked in his work [16], the method sketched above can be extended to small roots $(x_1^{(0)}, \ldots, x_v^{(0)})$ of multivariate polynomials $f_N(x_1, \ldots, x_v)$. Analogous to the univariate method, one could construct a lattice $L$ with shift polynomials

$$g_{i_1 \ldots i_v k}(x_1, \ldots, x_v) := x_1^{i_1} \cdot \ldots \cdot x_v^{i_v} (f_N(x_1, \ldots, x_v))^k N^{m-k},$$

for fixed $m$, $k = 0, \ldots, m$ and some choice of $i_1, \ldots, i_v$.

From Theorem 3.2, we know that the first $v$ reduced basis vectors $\mathbf{r}_1, \ldots, \mathbf{r}_v$ of the lattice satisfy

$$||\mathbf{r}_1|| \leq ||\mathbf{r}_2|| \leq \ldots \leq ||\mathbf{r}_v|| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-v)}} \det(L)^{\frac{1}{\omega+1-v}}.$$

A typical case is $\|\mathbf{r}_i\| \approx \det(L)^{\frac{1}{\omega}}$, but if the lattice is unbalanced, we can only use the bound above.

Theorem 3.5 can easily be adapted as follows for multivariate polynomials.

**Theorem 3.6 (Howgrave-Graham, [35])**
*Let* $h(x_1, \ldots, x_v) \in \mathbb{Z}[x_1, \ldots, x_v]$ *be an integer polynomial that consists of at most* $\omega$ *monomials. Suppose that*

(1) $h(x_1^{(0)}, \ldots, x_v^{(0)}) \equiv 0 \bmod R$ *for some* $|x_1^{(0)}| < X_1, \ldots, |x_v^{(0)}| < X_v$ *and some positive integer* $R$*, and*

(2) $||h(x_1X_1, \ldots, x_vX_v)|| < \frac{R}{\sqrt{\omega}}$.

*Then* $h(x_1^{(0)}, \ldots, x_v^{(0)}) = 0$ *holds over the integers.*

Therefore, we have that if

$$2^{\frac{\omega(\omega-1)}{4(\omega+1-v)}} \det(L)^{\frac{1}{\omega+1-v}} < \frac{N^m}{\sqrt{\omega}}$$

is satisfied, then we find $v$ polynomials $r_i(x_1, \ldots, x_v)$ that have the root $(x_1^{(0)}, \ldots, x_v^{(0)})$ over the integers.

A common root of $v$ polynomials in $v$ variables can be extracted efficiently if the $v$ polynomials are *algebraically independent*. Polynomials $r_1, \ldots, r_v$ are said to be algebraically independent if and only if $P(r_1, \ldots, r_v) = 0$ implies $P = 0$ for a polynomial $P$ defined over $\mathbb{Q}[x_1, \ldots, x_v]$. A recent paper by Bauer and Joux [2] treats the independence issue in Coppersmith methods in detail.

If our polynomials $r_1, \ldots, r_v$ are indeed independent, then we can find the common root by using *resultants* (see [21, Section 3] for an introduction on the theory of resultants). For our purpose, it is enough to know that a resultant $r(x_1, \ldots, x_{v-1}) = \mathrm{Res}_{x_v}(r_1, r_2)$ of two polynomials $r_1(x_1, \ldots, x_v), r_2(x_1, \ldots, x_v)$ has the following properties.

- The resultant $r(x_1, \ldots, x_{v-1})$ of $r_1$ and $r_2$ with respect to $x_v$ can be computed efficiently as the determinant of a Sylvester matrix that consists of columns containing shifted versions of the coefficient vectors of $r_1$ and $r_2$.
- If $r_1$ and $r_2$ share a root $(y_1, \ldots, y_{v-1}, y_v)$ for some $y_v$, then $r = \mathrm{Res}_{x_v}(r_1, r_2)$ has the root $(y_1, \ldots, y_{v-1})$. Hence, the resultant can be used to eliminate a variable $x_v$.
- The resultant $r(x_1, \ldots, x_{v-1}) = 0$ if and only if $r_1$ and $r_2$ share a common factor which has a positive degree in $x_v$. Thus, if $r_1$ and $r_2$ are algebraically dependent, then the elimination fails because the resultant is the zero function.

Hence, the polynomials $r_i(x_1, \ldots, x_v)$ are algebraically independent if and only if the following scheme produces a common root:

$$\begin{gathered} res_1 := \mathrm{Res}_{x_v}(r_1, r_2),\ res_2 := \mathrm{Res}_{x_v}(r_2, r_3),\ \ldots,\ res_{v-1} := \mathrm{Res}_{x_v}(r_{v-1}, r_v), \\ res_v := \mathrm{Res}_{x_{v-1}}(res_1, res_2),\ \ldots,\ res_{2v-3} := \mathrm{Res}_{x_{v-1}}(res_{v-2}, res_{v-1}), \\ \vdots \\ res_{end} := \mathrm{Res}_{x_2}(res_{end-2}, res_{end-1}). \end{gathered}$$

Since $res_{end}(x_1^{(0)}) = 0$, we know we can find $x_1^{(0)}$. The other entries of the root can be found by back substitution.

We have sketched how Coppersmith's method is applied to polynomials with more variables, and we have encountered the following heuristic.

**Assumption 3.7 (Independent $r_i$ in multivariate Coppersmith methods)** *The polynomials $r_i$ that are derived from the reduced basis of the lattice in the Coppersmith method are algebraically independent. Equivalently, the resultant computations of the $r_i$ yield nonzero polynomials.*

In the examples we have tested, this heuristic often works perfectly. However, one should always perform experiments to check whether Assumption 3.7 holds in a specific attack scenario where a Coppersmith method is used on a multivariate polynomial.

Small integer roots of bivariate polynomials $f(x, y)$:

Coppersmith's second method was meant for finding small integer roots. In the sketch that we give in this section, we follow Coron's reformulation of Coppersmith's method [18]. Essentially, Coron picks a 'suitable' integer $R$ and transforms the situation into finding a small root modulo $R$, to which one can apply Howgrave-Graham's lemma.

The main goal is to construct a polynomial $h(x, y)$, which is independent from the original polynomial $f$, and which shares the integer root $(x_0, y_0)$ with $f$.

Before we introduce the method to obtain $h$, we need to make some definitions. Recall that $X$ and $Y$ are the known upper bounds on $|x_0|$ and $|y_0|$. Let $W$ be the coefficient of $f(xX, yY)$ that is largest in absolute value. That is, if $f(x, y) = \sum_{i,j} a_{ij} x^i y^j$ then $W := \max_{i,j} |a_{ij} X^i Y^j|$. Let $R := X^{l_1} Y^{l_2} W$ for some choice of $l_1, l_2$ that we specify later. To obtain $h$ we use the shift polynomials

$$g_{ij}(x, y) := x^i y^j f(x, y) \cdot \frac{R}{W X^i Y^j} \quad \text{and} \quad g'_{ij}(x, y) := x^i y^j R,$$

where the sets of combinations $(i, j)$ for $g$ and $g'$ are discussed in Section 3.3.2. In order to let all $g_{ij}$ be integer polynomials, $l_1$ is defined as the largest degree of $x$ and $l_2$ as the largest degree of $y$ in these $g_{ij}$ (given a choice of combinations $(i, j)$ that are used).

Obviously, all polynomials $g_{ij}$ and $g'_{ij}$ share the root $(x_0, y_0)$ modulo $R$. We let the coefficient vectors of the polynomials $g_{ij}(xX, yY)$ and $g'_{ij}(xX, yY)$ be the basis of a lattice $L$. After applying LLL lattice basis reduction, we obtain a set of small vectors that correspond to polynomials $r_1(xX), \ldots, r_\omega(xX)$, where $\omega$ is the dimension of the lattice. If $L$ has full rank, then the $r_i$ have at most $\omega$ monomials. From Theorem 3.2, we know that

$$\|r_1(xX, yY)\| \leq 2^{\frac{\omega-1}{4}} \det(L)^{\frac{1}{\omega}}.$$

Thus, if $2^{\frac{\omega-1}{4}} \det(L)^{\frac{1}{\omega}} < \frac{R}{\sqrt{\omega}}$ holds, then the LLL reduction gives us a polynomial $h(x, y) = r_1(x, y)$ which satisfies Howgrave-Graham's bound. The choice of $R$ ensures that $h(x, y)$ is independent of $f$. This is because $h$ is divisible by $X^{l_1} Y^{l_2}$.

**Theorem 3.8 (Coron, [18])**
*A multiple $h(x,y)$ of $f(x,y)$ that is divisible by $X^{l_1}Y^{l_2}$ has norm at least*

$$2^{-(\rho+1)^2+1}X^{l_1}Y^{l_2}W,$$

*where $\rho$ is the maximum degree of the polynomials $f, h$ in each variable separately.*

Hence, if $||h(xX,yY)|| < 2^{-(\rho+1)^2+1}X^{l_1}Y^{l_2}W = 2^{-(\rho+1)^2+1}R$, then $h$ cannot be a multiple of $f$. We can assume that $f$ is irreducible, for otherwise, we could have made our problem easier by looking at the roots of the factors of $f$ (note that factoring a polynomial over $\mathbb{Z}$ can be done efficiently, see for instance [73]). Therefore, $h$ must be independent of $f$.
If we put the two bounds of this section next to each other as follows,

$$\begin{aligned} &||h(xX,yY)|| \leq 2^{\frac{\omega-1}{4}}\det(L)^{\frac{1}{\omega}} < \tfrac{R}{\sqrt{\omega}} && \text{(Howgrave-Graham's bound)} \\ &||h(xX,yY)|| \leq 2^{\frac{\omega-1}{4}}\det(L)^{\frac{1}{\omega}} < 2^{-(\rho+1)^2+1}R && \text{(independency bound)} \end{aligned}$$

then we see that the difference between them is only in the terms that do not depend on $N$. In the analysis of the attacks that use Coppersmith methods, it is common to use an error term $\epsilon$ for these terms, and check only if

$$\det(L) < R^{\omega-\epsilon}.$$

This explains the choice of $R$ in Coron's work, since it implies that any polynomial that satisfies Howgrave-Graham's bound, is automatically also independent of $f$. As we have explained before, when we know that $f(x,y)$ and $h(x,y)$ are independent polynomials that share a root, we can use a resultant to extract the root.

Note that Coppersmith's method of finding integer roots of bivariate polynomials is a provable method. As we shall see next, it can be extended to multivariate polynomials, but only when we introduce an assumption, which makes the method heuristic.

Small integer roots of multivariate polynomials $f(x_1,\ldots,x_v)$:

As in the modular case, it was already known to Coppersmith that his method could be extended to the case of multivariate polynomials, at the cost of becoming heuristic. Analogously to the method for bivariate polynomials, we introduce an integer $W$ that is the coefficient of $f(x_1X_1,\ldots,x_vX_v)$ that is largest in absolute value. That is, if $f(x_1,\ldots,x_v) = \sum a_{i_1\ldots i_v}x_1^{i_1}\cdot\ldots\cdot x_v^{i_v}$ then $W := \max|a_{i_1\ldots i_v}X_1^{i_1}\cdot\ldots\cdot X_v^{i_v}|$. An alternative notation that we will sometimes use is $W = \|f(x_1X_1,\ldots,x_vX_v)\|_\infty$. Let $R := WX_1^{l_1}\cdot\ldots\cdot X_v^{l_v}$ for some choice of $l_i$ that we specify later.

Let

$$g_{i_1\ldots i_v}(x_1,\ldots,x_v) := x_1^{i_1}\cdot\ldots\cdot x_v^{i_v}f(x_1,\ldots,x_v)\cdot\tfrac{R}{WX_1^{i_1}\cdot\ldots\cdot X_v^{i_v}} \quad \text{and}$$

$$g'_{i_1\ldots i_v}(x_1,\ldots,x_v) := x_1^{i_1}\cdot\ldots\cdot x_v^{i_v}R,$$

for some sets of combinations $(i_1, \ldots, i_v)$ for $g$ and $g'$. In order to let all $g_{i_1 \ldots i_v}$ be integer polynomials, each $l_i$ is defined as the largest degree of $x_i$ in these $g_{i_1 \ldots i_v}$ (given a choice of the combinations $(i_1, \ldots, i_v)$ that are used).

We let the coefficient vectors of $g_{i_1 \ldots i_v}(x_1 X_1, \ldots, x_v X_v)$ and $g'_{i_1 \ldots i_v}(x_1 X_1, \ldots, x_v X_v)$ be the basis of a lattice $L$. After the LLL reduction the first $v-1$ vectors satisfy

$$||\mathbf{r}_1|| \leq ||\mathbf{r}_2|| \leq \ldots \leq ||\mathbf{r}_{v-1}|| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+2-v)}} \det(L)^{\frac{1}{\omega+2-v}}.$$

Together with Theorem 3.6, we can conclude that if $2^{\frac{\omega(\omega-1)}{4(\omega+2-v)}} \det(L)^{\frac{1}{\omega+2-v}} < \frac{R}{\sqrt{\omega}}$ is satisfied, then we find $v-1$ polynomials $r_i(x_1, \ldots, x_v)$ that have the root $(x_1^{(0)}, \ldots, x_v^{(0)})$ over the integers.

By the following generalization of Coron's theorem by Hinek and Stinson, we know that all these $r_i$ are independent of $f$.

**Theorem 3.9 (Hinek/Stinson, [34])**
*A multiple $h(x_1, \ldots, x_v)$ of $f(x_1, \ldots, x_v)$ that is divisible by $\prod_{j=1}^{v} X_j^{l_j}$ has norm at least*

$$2^{-(\rho+1)^v+1} \prod_{j=1}^{v} X_j^{l_j} W,$$

*where $\rho$ is the maximum degree of the polynomials $f, h$ in each variable separately.*

Analogously to the bivariate case, one can show that this bound is (up to some terms that do not depend on $N$) equivalent to Howgrave-Graham's bound. Therefore, if Howgrave-Graham's bound is satisfied, then the polynomials $r_i$ are certainly independent of $f$. Under Assumption 3.7, one can now use resultant methods to extract the root that $f$ and $r_1, \ldots, r_{v-1}$ share.

## 3.3 A general strategy for choosing the shifts

In Section 3.2, we have explained the general framework of Coppersmith methods, based on the works by Coppersmith, Howgrave-Graham, and Coron. One thing that we left open in the discussion of the methods is the choice of the shift polynomials that describe the lattice $L$.

This is because the choice of shifts depends heavily on the polynomial $f_N$ or $f$. For specific polynomials that were used in cryptanalytic situations, the choices of the shift polynomials have been described in the papers describing the attacks. So far, the only work on designing a general way to choose the shift polynomials has been by Blömer and May [7]. In their paper, they give a strategy for the choice of shifts in the case of small integer roots of bivariate polynomials $f(x, y)$.

In this section, we discuss a general strategy of choosing the shifts for multivariate polynomials, for both the modular and the integer case.

### 3.3.1 Small modular roots

Suppose we want to find a small root $(x_1^{(0)}, \ldots, x_v^{(0)})$ of a polynomial $f_N$ modulo a known composite integer $N$ of unknown factorization. We assume that we know an upper bound for the root, namely $|x_j^{(0)}| < X_j$ for some given $X_j$, for $j = 1, \ldots, v$.

Our goal in this section is to choose the shift polynomials

$$g_{i_1 \ldots i_v k}(x_1, \ldots, x_v) := x_1^{i_1} \cdot \ldots \cdot x_v^{i_v} (f_N(x_1, \ldots, x_v))^k N^{m-k},$$

that define the lattice $L$ in such a way that they produce a good bound

$$\det(L) < 2^{\frac{-\omega(\omega-1)}{4}} \cdot \left(\frac{1}{\sqrt{\omega}}\right)^{\omega+1-v} \cdot N^{m(\omega+1-v)}. \tag{3.2}$$

Remember that $\det(L)$ depends on the upper bounds $X_j$, and that we aim to find the maximal values of the $X_j$ for which the Coppersmith method succeeds in finding the root.

Suppose we are in an attack scenario where finding a small root means breaking a certain RSA instance (that is, being able to factor a modulus $N$ in polynomial time given $(N, e)$ if the RSA parameters satisfy special conditions). Then, we want to obtain a bound on how large the root can be such that it can still be found. To obtain a clean bound, it is common to introduce an error term $\epsilon$ for all the terms that do not depend on $N$. In this way, we get an asymptotic bound, since for $N \to \infty$, $\epsilon$ goes to 0. This means, that instead of checking (3.2), we simply use $\det(L) < N^{m(\omega+1-v)-\epsilon}$. If the number of variables $v$ is taken to be constant, we can further simplify the condition to

$$\det(L) < N^{m\omega-\epsilon}. \tag{3.3}$$

First of all, we describe a way to order the monomials. For an introduction on various monomial orderings, we refer to [21, Section 2.2]. In the case of modular roots of a polynomial $f_N$, we look at the Newton polygon $P$ of $f_N$. Suppose we represent every monomial $x_1^{i_1} \cdot \ldots \cdot x_v^{i_v}$ of $f_N$ with a tuple $(i_1, \ldots, i_v) \in \mathbb{Z}^v$. Then the Newton polygon $P$ of $f_N$ is defined as the convex hull of this set, that is

$$\text{conv}(\{(i_1, \ldots, i_v) \in \mathbb{Z}^v | x_1^{i_1} \cdot \ldots \cdot x_v^{i_v} \text{ is a monomial of } f_N\}).$$

Hence, all monomials of $f_N$ correspond to a point in $P \cap \mathbb{Z}^v$. For this Newton polygon $P$, we pick a positive weight vector that has a *unique* maximum in $P \cap \mathbb{Z}^v$. A positive weight vector is a vector $\mathbf{w} = (w_1, \ldots, w_v)$ with all $w_i \geq 0$, that assigns a weight to each point $(i_1, \ldots, i_v)$ in $P \cap \mathbb{Z}^v$. The weight of $(i_1, \ldots, i_v)$, that we shall sometimes also refer to as the weight of the corresponding monomial $x_1^{i_1} \cdot \ldots \cdot x_v^{i_v}$, is computed as $(w_1, \ldots, w_v) \cdot (i_1, \ldots, i_v)^T = \sum_{j=1}^{v} w_j \cdot i_j$.

As said, we choose a $\mathbf{w}$ such that one vertex of $P$, corresponding to a monomial $l$ of $f_N$, has maximal weight. As a result, there is no monomial in $f_N$ besides $l$ that is divisible by $l$. For a given weight vector, we can order the set of monomials of $f_N$ according to the weight of their corresponding point in $P \cap \mathbb{Z}^v$. If two monomials have equal weight, we can

order them according to the lexicographical ordering. The monomial in $f_N$ with maximal weight, $l$, is called the leading monomial of $f_N$, and we name its coefficient $a_l$. We can assume that $\gcd(N, a_l)$ is 1, or else we have found a factor of $N$. Therefore, we can use $f'_N = a_l^{-1} f_N \bmod N$.

We start by explaining the basic strategy for choosing the shifts, after which we extend it slightly to obtain the full strategy.

Basic Strategy:

Let $\epsilon > 0$ have any fixed small value. Depending on $\epsilon$, we fix an integer $m$. For $k \in \{0, \ldots, m\}$, we define the set $M_k$ of monomials by

$$M_k := \{x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} \mid \quad x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} \text{ is a monomial of } f_N^m \text{ and } \frac{x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v}}{l^k} \text{ is a monomial of } f_N^{m-k}\}.$$

Moreover, let $M_{m+1} := \emptyset$. In this definition of $M_k$ and throughout this thesis, we assume that the monomials of $f_N$ up to $f_N^{m-1}$ are all contained in the monomials of $f_N^m$. If this is not the case, the definition can be slightly changed such that $M_k$ contains all monomials $x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v}$ of $f_N^j$ for $j \in \{1, \ldots, m\}$ for which $\frac{x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v}}{l^k}$ is a monomial of $f_N^i$ for some $i \in \{0, \ldots, m-k\}$. Notice that by definition the set $M_0$ contains all the monomials in $f_N^m$. Next, we define the following shift polynomials:

$$g_{i_1 \ldots i_v k}(x_1, \ldots, x_v) := \frac{x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v}}{l^k} f'_N(x_1, \ldots, x_v)^k N^{m-k},$$

for $k = 0, \ldots, m$ and $x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} \in M_k \backslash M_{k+1}$.

All polynomials $g$ have the root $(x_1^{(0)}, \ldots, x_v^{(0)})$ modulo $N^m$. If we define a lattice $L$ by taking the coefficient vectors of $g(x_1 X_1, \ldots, x_v X_v)$ as a basis, we can force the matrix describing $L$ to be lower triangular. The diagonal elements are those corresponding to the monomial $l^k$ in $(f'_N)^k$ for each row. Therefore, the diagonal terms of the matrix are $X_1^{i_1} X_2^{i_2} \cdot \ldots \cdot X_v^{i_v} N^{m-k}$ for the given combinations of $k$ and $i_j$.

Let us try to get some intuition on the choice of the sets $M_k$. We aim to have a matrix with a low determinant. To keep the diagonal element corresponding to the monomial $x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v}$ of $f_N^m$ as small as possible, we use the largest possible powers of $f_N$ in the shifts. The condition that $\frac{x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v}}{l^k}$ is a monomial of $f_N^{m-k}$ ensures that no monomials appear that are not in $f_N^m$.

For a small example, consider the polynomial $f_N(x, y) = 1 + xy^2 + x^2 y$. Let us take $l = x^2 y$ as our leading term, and $m = 2$. We want to build a lattice whose columns correspond to the monomials $\{1, xy^2, x^2 y, x^2 y^4, x^3 y^3, x^4 y^2\}$ of $f_N^2$. The shifts given by our strategy are:

| | | | | | |
|---|---|---|---|---|---|
| for | $1 \in M_0 \backslash M_1$: | $N^2$, | for | $x^2 y \in M_1 \backslash M_2$: | $f_N N$, |
| for | $xy^2 \in M_0 \backslash M_1$: | $xy^2 N^2$, | for | $x^3 y^3 \in M_1 \backslash M_2$: | $xy^2 f_N N$, |
| for | $x^2 y^4 \in M_0 \backslash M_1$: | $x^2 y^4 N^2$, | for | $x^4 y^2 \in M_2 \backslash M_3$: | $f_N^2$. |

Note that the monomial $x^2y^4$ is not in $M_1$. Although $x^2y^4$ is divisible by $l = x^2y$ and therefore we could obtain $x^2y^4$ also by using the shift $y^3 f_N N$, the product $y^3 f_N$ would produce the new monomials $y^3$ and $xy^5$, which are not in $f_N^2$.

As one can see, the matrix describing the lattice of the above example is lower triangular. To get some intuition on the ordering that makes this possible, let us go back to the notation of the Newton polygon $P$. Recall that every monomial of $f_N^m$ corresponds to a point in $mP \cap \mathbb{Z}^v$. For every point $\mathbf{u} = (i_1, \ldots, i_v) \in \mathbb{Z}^v$ corresponding to a monomial $x_1^{i_1} \cdot \ldots \cdot x_v^{i_v}$, define

$$P_{\mathbf{u}} := \mathbf{u} - k\mathbf{l} + kP,$$

where $k$ is the largest integer possible such that $P_{\mathbf{u}} \subseteq mP$, and $\mathbf{l}$ is the point corresponding to the leading monomial $l$ of $f_N$. Note that

- $P_{\mathbf{u}}$ exists for every $\mathbf{u} \in mP \cap \mathbb{Z}^v$,
- $P_{\mathbf{u}}$ contains $\mathbf{u}$,
- $\mathbf{u}$ has maximal weight in $P_{\mathbf{u}}$, since $k\mathbf{l}$ is maximal in $kP$.

If $k$ is the integer that is used in the construction of $P_{\mathbf{u}}$ for some $\mathbf{u} = (i_1, \ldots, i_v)$, then the monomial $x_1^{i_1} \cdot \ldots \cdot x_v^{i_v}$ that corresponds to $\mathbf{u}$ is in $M_k \backslash M_{k+1}$. Hence, the monomials corresponding to $P_{\mathbf{u}}$ are exactly those that appear in the shift $g_{i_1 \ldots i_v k}$ of $x_1^{i_1} \cdot \ldots \cdot x_v^{i_v}$. In $P_{\mathbf{u}}$, $\mathbf{u}$ is the unique maximum with respect to the given weight vector. Therefore, if one orders the monomials of $f_N^m$ according to their weight, then one can be sure that any monomial corresponding to a point $\mathbf{u}'$ that is higher or equal in weight than $\mathbf{u}$ cannot occur in the shift of $\mathbf{u}$. Hence, if we order the matrix describing the lattice $L$ such that the columns correspond to the monomials of $f_N^m$ in increasing order, then we are certain that we get a lower triangular matrix.

For an example of the ordering, let us look at $f_N(x, y) = 1 + x + x^2 + y + xy + x^2y + xy^2$ (whose Newton polygon $P$ has the shape of a 'little house'). In Figure 3.3, the polygons $P$, $2P$, and $3P$ are shown, which means that we are looking at all monomials of $f_N^m$, for $m = 3$. For the weight vector $(0, 1)$, every monomial in $f_N$ has a weight that is equal to the exponent of $y$ in the monomial. Obviously, $l = xy^2$ is the unique monomial with maximal weight in $f_N$ (in other words: $\mathbf{l} = (1, 2)$ is the unique vertex of maximal weight in $P$).

The monomial $x^4y^5$ is in $f_N^3$, and therefore $\mathbf{u} = (4, 5)$ is in $3P \cap \mathbb{Z}^2$. For this $\mathbf{u}$ one can check that

$$P_{\mathbf{u}} = (4, 5) - 2(1, 2) + 2P.$$

Hence, the monomials in $g_{45} = \frac{x^4y^5}{l^2} f_N^2 N$ are those corresponding to the lattice points of $P_{\mathbf{u}}$. Since we have made sure that $(4, 5)$ is the unique vertex of maximal weight in $P_{\mathbf{u}}$, all other monomials of $g_{45}$ are strictly smaller in weight than $x^4y^5$. Therefore, the ordering corresponding to the weight vector ensures a triangular matrix.

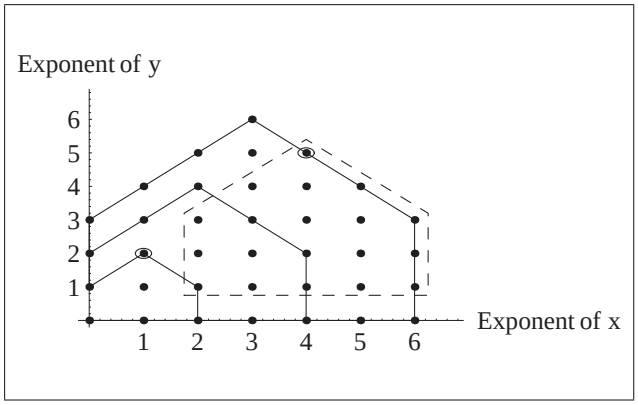


Figure 3.3: An example of the ordering

Now that we have discussed the triangular representation of the lattice, let us see which bound we can derive from our condition $\det(L) < N^{m\omega-\epsilon}$. We have that

$$\begin{aligned}\det(L) &= \prod_{k=0,\ldots,m \text{ and } x_1^{i_1}x_2^{i_2}\cdot\ldots\cdot x_v^{i_v}\in M_k\setminus M_{k+1}} X_1^{i_1}X_2^{i_2}\cdot\ldots\cdot X_v^{i_v}N^{m-k} \\ &= \prod_{j=1}^{v} X_j^{\sum_{k=0}^{m}\sum_{x_1^{i_1}x_2^{i_2}\cdot\ldots\cdot x_v^{i_v}\in M_k\setminus M_{k+1}} i_j} N^{\sum_{k=0}^{m}\sum_{x_1^{i_1}x_2^{i_2}\cdot\ldots\cdot x_v^{i_v}\in M_k\setminus M_{k+1}}(m-k)}.\end{aligned}$$

Moreover,

$$\dim(L) = \omega = \sum_{k=0}^{m} \sum_{x_1^{i_1}x_2^{i_2}\cdot\ldots\cdot x_v^{i_v}\in M_k\setminus M_{k+1}} 1.$$

We conclude that the condition $\det(L) < N^{m\omega-\epsilon}$ reduces to

$$\prod_{j=1}^{v} X_j^{s_j} < N^{s_N-\epsilon}, \text{ with } \begin{cases} s_j &= \sum_{x_1^{i_1}\ldots x_v^{i_v}\in M_0} i_j, \text{ for } 1\le j\le v, \text{ and} \\ s_N &= \sum_{k=0}^{m} k(|M_k|-|M_{k+1}|) = \sum_{k=1}^{m}|M_k|. \end{cases} \tag{3.4}$$

If we follow this procedure of choosing shifts for a given $f_N$, then (3.4) will give us an upper bound on the size of the root that we are trying to find. For $X_j$ and $N$ satisfying this bound we obtain $v$ polynomials $r_i$ such that $r_i(x_1^{(0)},\ldots,x_v^{(0)}) = 0$. If $v = 1$, then the method is provable. For $v \ge 2$, we rely on Assumption 3.7 to find $(x_1^{(0)},\ldots,x_v^{(0)})$.

Extended Strategy:

For many polynomials, it is profitable to use extra shifts of a certain variable (for instance, if one $X_i$ is significantly smaller than the other upper bounds $X_j$, $j \ne i$. If we use extra shifts of the variable $x_1$, then we can extend our basic strategy by using the following $M_k$.

$$M_k := \bigcup_{0\le j\le t} \{x_1^{i_1+j}x_2^{i_2}\cdot\ldots\cdot x_v^{i_v} \mid \quad x_1^{i_1}x_2^{i_2}\cdot\ldots\cdot x_v^{i_v} \text{ is a monomial of } f_N^m \text{ and } \frac{x_1^{i_1}x_2^{i_2}\cdot\ldots\cdot x_v^{i_v}}{l^k} \text{ is a monomial of } f_N^{m-k}\}.$$

Moreover, extra shifts of several variables, or combined shifts should be considered to obtain an optimal bound. The number of extra shifts can be described using a parameter $t$ that can be optimized. Using this new definition of $M_k$, the rest of the strategy conforms to the basic strategy as described before. Let us now give some examples, and show how the known results on small modular roots from [6, 10, 16] are all special cases of our basic or extended strategy.

Examples:

Here, we show that the known results of Boneh and Durfee, Blömer and May, and Coppersmith [10, 6, 16] in this field are special cases of our strategy. Although they were originally described in different ways, their results can also be obtained if one follows the basic or extended strategy described in this section.

'Boneh/Durfee attack' [10]:

Recall that in the attack by Boneh and Durfee [10], the goal was to find the small root $(d, p+q-1)$ of the polynomial $f_e(x,y) = 1 + x(N-y)$ modulo $e$.

Let us discuss in general how small a root $(x_0, y_0)$ of a polynomial

$$f_N(x,y) = a_0 + a_1x + a_2xy$$

modulo some $N$ should be such that it can be found by a Coppersmith method. The known bound that the upper bounds $X$ and $Y$ have to satisfy is

$$X^{2+3\tau}Y^{1+3\tau+3\tau^2} < N^{1+3\tau-\epsilon}.$$

Here and in the following attack bounds, $\tau > 0$ is always a parameter that can be optimized after plugging in the values for $X$, $Y$ and $N$.

The bound can be obtained by following our extended strategy with extra shifts of $y$. As we described above, the extended strategy then prescribes

$$M_k := \bigcup_{0\le j\le t} \{x^{i_1}y^{i_2+j} \,\Big|\, x^{i_1}y^{i_2} \text{ monomial of } f_N^m \text{ and } \frac{x^{i_1}y^{i_2}}{l^k} \text{ monomial of } f_N^{m-k}\}, \text{ for } l = xy.$$

Figures 3.4 and 3.5 show the monomials of $f_N^m$ (which would be used if one would only apply the basic strategy) and the new sets $M_k$ when including extra $y$-shifts. One can check that a correct description of the sets $M_k$ is

$$x^{i_1}y^{i_2} \in M_k \Leftrightarrow i_1 = k,\ldots,m \quad \text{and} \quad i_2 = k,\ldots,i_1+t,$$

for some $t = \tau m$ (in the figure, $t = 2$ is shown).

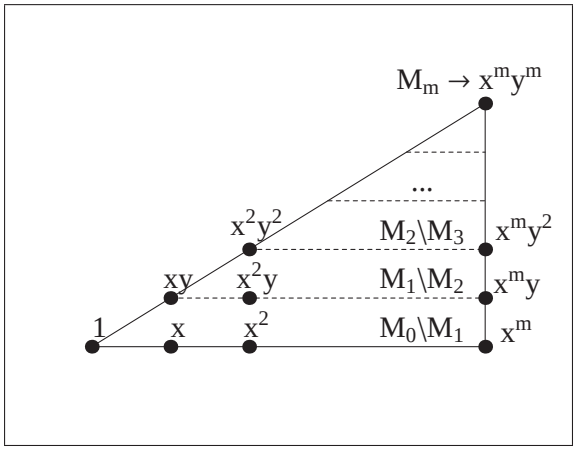


Figure 3.4: Shape of $f_N^m$

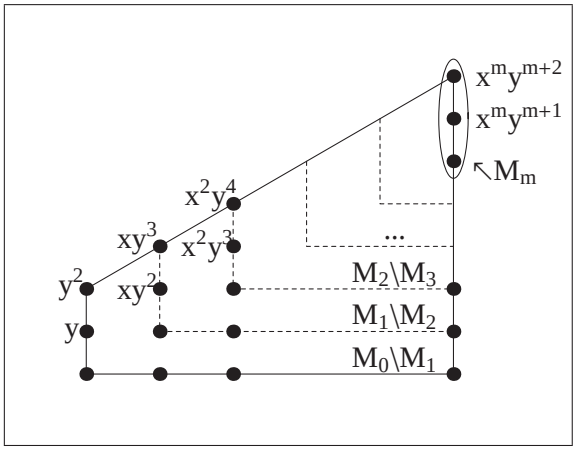


Figure 3.5: Extra $y$-shifts

Substituting this definition of $M_k$ in the bound of (3.4) leads to

$$X^{\frac{1}{6}(m^3(2+3\tau)+m^2(6+3\tau)+m\cdot 4)} Y^{\frac{1}{6}(m^3(1+3\tau+3\tau^2)+m^2(3+6\tau+3\tau^2)+m(2+3\tau))} < N^{\frac{1}{6}(m^3(1+3\tau)+m^2(3+3\tau)+m\cdot 2)-\epsilon}.$$

If we let all terms of order $o(m^3)$ contribute to $\epsilon$, then we see that for $m \to \infty$, this bound reduces to

$$X^{2+3\tau} Y^{1+3\tau+3\tau^2} < N^{1+3\tau-\epsilon}.$$

To illustrate the example, we show how the matrix describing the lattice $L$ looks for $m = 2$, $t = 1$. Since the Boneh/Durfee-attack is about finding a root modulo $e$, we will use the terminology of $f_e$ from here on instead of $f_N$.

$$\left(\begin{array}{cccc|ccc|cc}
e^2 & & & & & & & & \\
 & e^2X & & & & & & & \\
 & & e^2X^2 & & & & & & \\
 & & & e^2Y & & & & & \\
ea_0' & ea_1'X & & & eXY & & & & \\
 & ea_0'X & ea_1'X^2 & & & eX^2Y & & & \\
 & & & ea_0'Y & ea_1'XY & & eXY^2 & & \\
a_0'^2 & 2a_0'a_1'X & a_1'^2X^2 & & 2a_0'XY & & 2a_1'X^2Y & X^2Y^2 & \\
 & & & a_0'^2Y & 2a_0'a_1'XY & 2a_0'XY^2 & a_1'^2X^2Y & 2a_1'X^2Y^2 & X^2Y^3
\end{array}\right)$$

Figure 3.6: Example of a Boneh/Durfee-lattice: $m = 2$, $t = 1$

The columns of the matrix correspond to the monomials $1, x, x^2, y$ (of $M_0\backslash M_1$), $xy, x^2y, xy^2$ (of $M_1\backslash M_2$) and $x^2y^2, x^2y^3$ (of $M_2$) respectively. The rows represent the shifts $g_{i_1 i_2}(x,y) := x^{i_1} y^{i_2} (f_e'(x,y))^k e^{m-k}$, where $f_e'(x,y) = a_2^{-1} f_e(x,y) = a_0' + a_1'x + xy$.

In the case of the Boneh/Durfee-attack on RSA-Small-$d$, we know that the root $(x_0, y_0)$ we are looking for has upper bounds $X = N^\beta$ and $Y = 3N^{\frac{1}{2}}$. Since we put all terms that do not depend on $N$ in our error term $\epsilon$, we see that

$$\beta(2 + 3\tau) + \frac{1}{2}(1 + 3\tau + 3\tau^2) < 1 \cdot (1 + 3\tau)$$

is the asymptotical bound of the attack. This reduces to

$$3\tau^2 + 3\tau(2\beta - 1) + (4\beta - 1) < 0.$$

The left hand side polynomial has its maximum for $\tau = \frac{1}{2} - \beta$. Substituting this value, we obtain

$$-3\beta^2 + 7\beta - \frac{7}{4} < 0,$$

which reduces to $\beta < \frac{7}{6} - \frac{1}{3}\sqrt{7} \approx 0.284$.

In [10], Boneh and Durfee improve this bound a little more to $\beta < 1 - \frac{1}{2}\sqrt{2} \approx 0.292$. They improve the bound by looking at sublattices of the lattice described before. A complication with this technique is that the sublattices no longer have full rank, which makes the analysis harder. We go into more detail about sublattices in Section 7.2.

For completeness, Boneh/Durfee's result is stated in the following theorem.

**Theorem 3.10 (Boneh/Durfee, [10])**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists $n_0$ such that for every $n > n_0$, the following holds: Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$. Moreover, let $d < N^{0.292-\epsilon}$. Given $N$, and $e$ such that $ed = 1 \bmod \phi(N)$, one can recover $d$ in time polynomial in $n$.*

'Blömer/May attack' [6]:

In Section 4.2, we discuss the known partial key exposure attacks on RSA, among which are attacks by Blömer and May that use the following polynomial:

$$f_N(x, y, z) = a_0 + a_1 x + a_2 y + a_3 yz.$$

Here, we sketch how the known bound for this polynomial,

$$X^{1+4\tau} Y^{2+4\tau} Z^{1+4\tau+6\tau^2} < N^{1+4\tau-\epsilon},$$

can be obtained by following our extended strategy with extra shifts of $z$.

Suppose we choose $l = x$ as leading monomial. Then the set $M_k$ can be described as

$$M_k := \bigcup_{0 \le j \le t} \{x^{i_1} y^{i_2} z^{i_3+j} \mid x^{i_1} y^{i_2} z^{i_3} \text{ monomial of } f_N^m \text{ and } \tfrac{x^{i_1} y^{i_2} z^{i_3}}{l^k} \text{ monomial of } f_N^{m-k}\}.$$

We find that

$$x^{i_1}y^{i_2}z^{i_3} \in M_k \Leftrightarrow i_1 = k, \ldots, m \ ; \ i_2 = 0, \ldots, m - i_1 \ ; \ i_3 = 0, \ldots, i_2 + t,$$

for some $t = \tau m$. Substituting this new definition of $M_k$ in the bound (3.4) will result in Blömer/May's bound.

Generalized Rectangles and Generalized Lower Triangles [16]:

Now that we have used our extended strategy twice to obtain the known bounds of Boneh/Durfee and Blömer/May, we also mention two generalizations of known results that can be derived using only the basic strategy.

Suppose $f_N(x_1, \ldots, x_v)$ is a polynomial with the shape of a *generalized rectangle*, that is, the degree of $x_i$ is $\lambda_i D$. Then the bound (heuristically for $v \geq 2$) for which a root $(x_1^{(0)}, \ldots, x_v^{(0)})$ of $f_N$ modulo $N$ can be found, is given by

$$X_1^{\lambda_1} \cdot \ldots \cdot X_v^{\lambda_v} < N^{\frac{2}{(v+1)D} - \epsilon}.$$

This bound is a generalization of Coppersmith's bound $X < N^{\frac{1}{D}}$ [16] and the heuristic extension $XY < N^{\frac{2}{3D}}$. Nguyen and Stern [59] already mentioned the bound $X^{\lambda_1}Y^{\lambda_2} < N^{\frac{2}{3D}}$, for the bivariate case.

The general result can be obtained by following the basic strategy, with

$$l = x_1^{\lambda_1 D} \cdot x_2^{\lambda_2 D} \cdot \ldots \cdot x_v^{\lambda_v D},$$

and

$$x_1^{i_1}x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} \in M_k \ \Leftrightarrow \ i_j = \lambda_j Dk, \ldots, \lambda_j Dm \quad (j = 1, \ldots, v).$$

Suppose $f_N(x_1, \ldots, x_v)$ is a polynomial with the shape of a *generalized lower triangle*, that is, its monomials are $x_1^{i_1} \cdot \ldots \cdot x_v^{i_v}$ for

$$0 \leq i_1 \leq \lambda_1 D, \ 0 \leq i_2 \leq \lambda_2 D - \frac{\lambda_2}{\lambda_1} i_1, \ \ldots, \ 0 \leq i_v \leq \lambda_v D - \sum_{r=1}^{v-1} \frac{\lambda_v D}{\lambda_r} i_r.$$

Then the bound (heuristically for $v \geq 2$) is given by

$$X_1^{\lambda_1} \cdot \ldots \cdot X_v^{\lambda_v} < N^{\frac{1}{D} - \epsilon}.$$

The known special cases are those for $f_N(x)$ or $f_N(x, y)$ with *total degree* $D$ (in other words, $\lambda_j = 1$). Both $X < N^{\frac{1}{D}}$ and $XY < N^{\frac{1}{D}}$ appeared in [16].

The general bound can be easily derived by following the basic strategy with $l = x_1^{\lambda_1 D}$ and

$$\begin{aligned} x_1^{i_1}x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} \in M_k \Leftrightarrow \ & i_1 = \lambda_1 Dk, \ldots, \lambda_1 Dm \ ; \\ & i_j = 0, \ldots, \lambda_j Dm - \textstyle\sum_{r=1}^{j-1} \frac{\lambda_j}{\lambda_r} i_r \quad (j = 2, \ldots, v). \end{aligned}$$

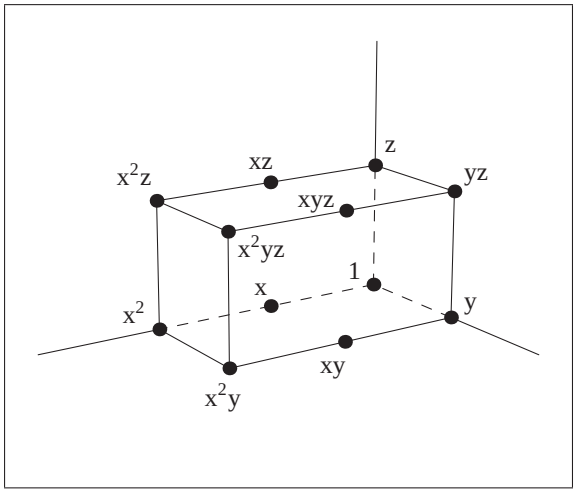


Figure 3.7: Generalized Rectangle

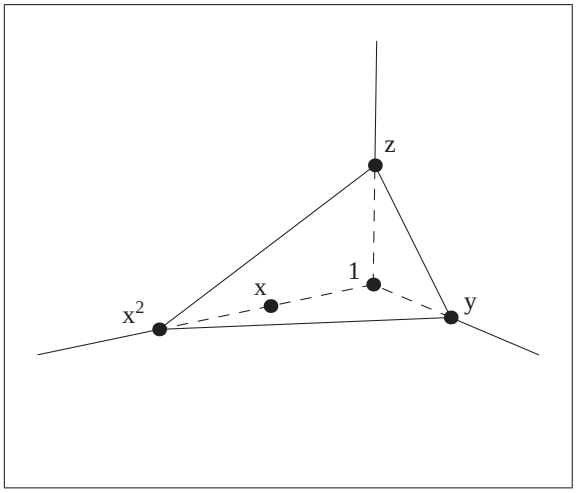


Figure 3.8: Generalized Lower Triangle

### 3.3.2 Small integer roots

Suppose we want to find a small integer root $(x_1^{(0)}, \ldots, x_v^{(0)})$ of an irreducible polynomial $f$. We know that the root is small in the sense that $|x_j^{(0)}| < X_j$, for $j = 1, \ldots, v$.

The goal in this section is to choose the shift polynomials

$$g_{i_1 \ldots i_v} := x_1^{i_1} \cdot \ldots \cdot x_v^{i_v} f(x_1, \ldots, x_v) \cdot \frac{R}{W X_1^{i_1} \cdot \ldots \cdot X_v^{i_v}} \quad \text{and}$$

$$g'_{i_1 \ldots i_v} := x_1^{i_1} \cdot \ldots \cdot x_v^{i_v} R,$$

that define the lattice $L$ in such a way that they produce a good bound

$$\det(L) < 2^{\frac{-\omega(\omega-1)}{4}} \cdot \left(\frac{1}{\sqrt{\omega}}\right)^{\omega+2-v} \cdot R^{\omega+2-v}. \tag{3.5}$$

To obtain a simple, asymptotic bound, we again introduce an error term $\epsilon$ for all the terms that do not depend on $N$. This means that instead of checking (3.5), we simply use $\det(L) < R^{\omega+2-v-\epsilon}$. If the number of variables $v$ is taken to be constant, we can further simplify the condition to

$$\det(L) < R^{\omega-\epsilon}. \tag{3.6}$$

Analogously to the modular case, we fix an integer $m$ depending on $\epsilon$. We call $d_j$ the maximal degree of $x_j$ in $f$, and $W$ the maximal absolute coefficient of $f(x_1 X_1, \ldots, x_v X_v)$. Moreover, we define $R = W \prod_{j=1}^{n} X_j^{d_j(m-1)}$. To work with a polynomial with constant term 1, we define $f' = a_0^{-1} f \bmod R$, where $a_0$ is the constant term of $f$. This means that we should have $a_0 \neq 0$ and $\gcd(a_0, R) = 1$. The latter is easy to achieve, analogous to [18, Appendix A], since any $X_j$ with $\gcd(a_0, X_j) \neq 1$ can be changed into an $X'_j$ such that $X_j < X'_j < 2X_j$ and $\gcd(a_0, X'_j) = 1$. The same holds for $W$.

Let us now consider the case $a_0 = 0$. In [18, Appendix A], Coron discussed this case for bivariate polynomials, and showed a simple way to transfer a polynomial $f$ with zero constant term into a polynomial $f^*$ with nonzero constant term.

A general way to do this for multivariate polynomials would be the following. First, we find a nonzero integer vector $(y_1, \ldots, y_v)$ such that $f(y_1, \ldots, y_v) \neq 0$. This can be constructed in polynomial time since there are only polynomially many roots within the given bounds. Then we define $f^*(x_1, \ldots, x_v) := f(x_1 + y_1, \ldots, x_v + y_v)$, and look for roots of $f^*$. Since $f^*(0, \ldots, 0) = f(y_1, \ldots, y_v)$, $f^*$ has a nonzero constant term.

We would like to point out that the switch to $f^*$ will affect the set of monomials, and new monomials may appear in $f^*$ that were not in $f$. This may affect the analysis and lead to a different Coppersmith-type bound. This issue already appears with bivariate polynomials, but it did not affect Coron's analysis since in his case the set of monomials stayed the same. From now on we will assume that $a_0 \neq 0$. In the unfortunate case that $a_0 = 0$ *and* switching to $f^*$ leads to a different monomial set, we suggest to use Coppersmith's original method[1] (as explained in Section 5.3.3).

Let us now describe our strategy for choosing the shifts in the case of integer roots. As before, we start with the basic strategy, that we extend later to obtain the full strategy.

Basic Strategy:

We define $S$ and $M$ as the sets of monomials of $f^{m-1}$ and $f^m$ respectively. We denote by $l_j$ the largest exponent of $x_j$ that appears in the monomials of $S$, i.e. $l_j = d_j(m-1)$.

Next, we define the following shift polynomials

$$g_{i_1 \ldots i_v} : x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} f'(x_1, \ldots, x_v) \prod_{j=1}^{n} X_j^{l_j - i_j}, \quad \text{for} \quad x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} \in S,$$

$$g'_{i_1 \ldots i_v} : x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} R, \quad \text{for} \quad x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} \in M \backslash S.$$

All $g_{i_1 \ldots i_v}$ and $g'_{i_1 \ldots i_v}$ have the root $(x_1^{(0)}, \ldots, x_v^{(0)})$ modulo $R$. The coefficient vectors of $g_{i_1 \ldots i_v}(x_1 X_1, \ldots x_v X_v)$ and $g'_{i_1 \ldots i_v}(x_1 X_1, \ldots x_v X_v)$ form a lattice basis of a lattice $L$.

Using the following ordering of the monomials of $S$, we can order the basis matrix such that it is upper triangular. We say that $x_1^{i_1} \cdot \ldots \cdot x_v^{i_v} < x_1^{i'_1} \cdot \ldots \cdot x_v^{i'_v}$ if $\sum i_j < \sum i'_j$. If $\sum i_j = \sum i'_j$, then we use the lexicographical ordering.

The diagonal elements of the rows of $g$ are those corresponding to the constant term in $f'$. Therefore, the diagonal entries of the matrix are $\prod_{j=1}^{n} X_j^{d_j(m-1)}$ for the polynomials $g$ and $W \prod_{j=1}^{n} X_j^{d_j(m-1)+i_j}$ for the polynomials $g'$.

For constant $v$, we know that the determinant condition $\det(L) < R^{\omega - \epsilon}$ ensures that the $v-1$ smallest vectors in an LLL reduced basis of $L$ correspond to $v-1$ polynomials $r_i(x_1, \ldots x_v)$ with $r_i(x_1^{(0)}, \ldots, x_v^{(0)}) = 0$.

[1] One could also use Coron's new method, as presented at Crypto'07 [19].

We find that the condition $\det(L) < R^{\omega-\epsilon}$ reduces to

$$\prod_{j=1}^{v} X_j^{s_j} < W^{s_W-\epsilon}, \text{ for } s_j = \sum_{x_1^{i_1}\ldots x_v^{i_v}\in M\setminus S} i_j \text{ , and } s_W = |S|. \tag{3.7}$$

So if the bound (3.7) holds, we obtain $v-1$ polynomials $r_i$ such that $r_i(x_1^{(0)},\ldots,x_v^{(0)}) = 0$. By Theorem 3.9, these polynomials are independent of $f$. So, under Assumption 3.7, the resultant computations of $f$ and $r_i$ (for $i = 1,\ldots,v-1$) will reveal the root.

Extended Strategy:

As in the modular case, our strategy is not finished before exploring the possibilities of extra shifts of a certain variable (or more variables). Suppose we use extra shifts of the variable $x_1$. Then, instead of $S = \{\text{monomials of } f^{m-1}\}$ and $M = \{\text{monomials of } f^m\}$, we use

$$\begin{aligned} S &= \bigcup_{0\le j\le t} \{x_1^{i_1+j}x_2^{i_2}\cdot\ldots\cdot x_v^{i_v} \mid x_1^{i_1}x_2^{i_2}\cdot\ldots\cdot x_v^{i_v} \text{ is a monomial of } f^{m-1}\},\\ M &= \{\text{monomials of } x_1^{i_1}x_2^{i_2}\cdot\ldots\cdot x_v^{i_v}\cdot f \mid x_1^{i_1}x_2^{i_2}\cdot\ldots\cdot x_v^{i_v}\in S\}. \end{aligned}$$

With the new definitions, the rest of the strategy conforms to the basic strategy, except for the value of $R$. It is necessary to change $R = W\prod_{j=1}^{v} X_j^{d_j(m-1)}$ into $R = W\prod_{j=1}^{v} X_j^{l_j}$, where $l_j$ is the largest exponent of $x_j$ that appears in the monomials of $S$.

Examples:

Here, we collect the known results by Blömer and May and Coppersmith [7, 16] for polynomials $f_N$ with a small integer root. These known results are again special cases of our extended or basic strategy. As our strategy for finding small integer roots is a generalization of the technique described in [7] to more variables, the first claim is not surprising. The examples we give in this section are very brief, since we will treat new polynomials for new attacks in more detail throughout this thesis.

'Blömer/May, upper triangle' [7]:

In [7], Blömer and May describe an alternative method to obtain the attack that Boneh, Durfee, and Howgrave-Graham published on Takagi's RSA variant [13]. For $N = p^r q$, and an approximation $\tilde{p}$ of $p$, one could try to find the root of the polynomial

$$f(x,y) = (\tilde{p}+x)^r y - N.$$

This is an example of what Blömer and May call a polynomial with the shape of an upper triangle. In general, we deal with a polynomial $f(x,y)$ with $x^{i_1}y^{i_2}$ for $i_2 = 0,\ldots,D$ and $i_1 = 0,\ldots,\lambda i_2$, for instance $f(x,y) = a_0 + a_1y + a_2xy + a_3x^2y$ (where $D = 1$ and $\lambda = 2$). The known bound is

$$X^{(\lambda+\tau)^2}Y^{2(\lambda+\tau)} < W^{\frac{1}{D}(\lambda+2\tau)-\epsilon}.$$

Figures 3.9 and 3.10 show the monomials of $f^m$ (which would be used if one would follow the basic strategy) and shapes of $S$ and $M$ if one uses extra shifts of $x$.

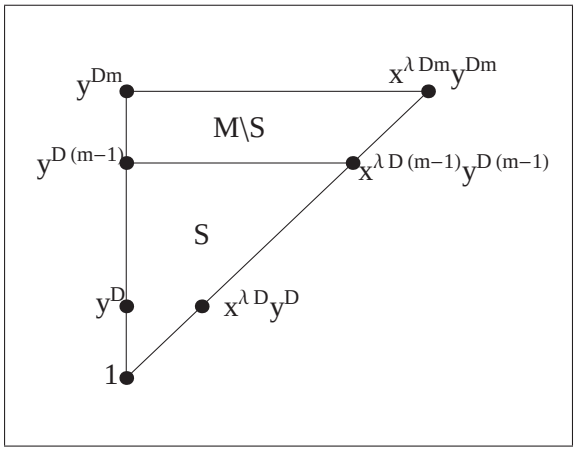


Figure 3.9: Shape of $f^m$

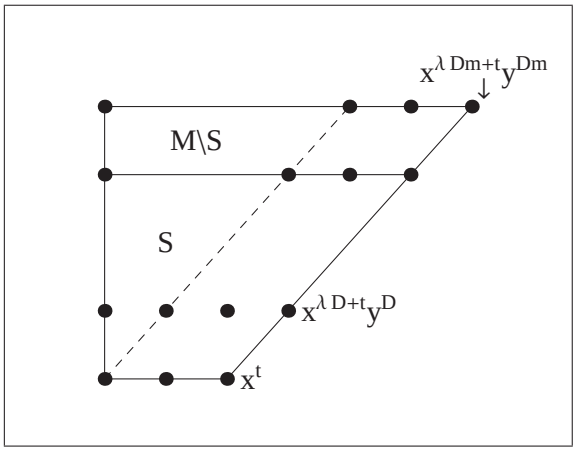


Figure 3.10: Extra $x$-shifts

One can check that the new definitions of $S$ and $M$ are

$$x^{i_1}y^{i_2} \in S \quad \Leftrightarrow i_2 = 0, \ldots, D(m-1)\ ;\ i_1 = 0, \ldots, \lambda i_2 + t.$$

$$x^{i_1}y^{i_2} \in M \quad \Leftrightarrow i_2 = 0, \ldots, Dm\ ;\ i_1 = 0, \ldots, \lambda i_2 + t.$$

for some $t = \tau Dm$.

Substituting this in inequality (3.7) will lead to the bound as given above.

'Blömer/May, extended rectangle' [7]:

Another type of polynomial that Blömer and May study in their paper about integer roots of polynomials, is the polynomial with the shape of an extended rectangle. That is, $f(x, y)$ with $x^{i_1}y^{i_2}$ for $i_2 = 0, \ldots, D$, $i_1 = 0, \ldots, \gamma D + \lambda(D - i_2)$, for instance $f(x, y) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4y + a_5xy$ (where $D = 1$ and $\gamma = 1$ and $\lambda = 2$). The known bound is

$$X^{\lambda^2+3\gamma\lambda+2\tau\lambda+4\tau\gamma+\tau^2+3\gamma^2}Y^{\lambda+3\gamma+2\tau} < W^{\frac{1}{D}(\lambda+2\gamma+2\tau)-\epsilon}.$$

Besides the shifts of the basic strategy (Figure 3.11), we use extra $x$-shifts (Figure 3.12) and work with

$$\begin{aligned} x^{i_1}y^{i_2} \in S \quad &\Leftrightarrow i_2 = 0, \ldots, D(m-1)\ ; \\ &\quad i_1 = 0, \ldots, \gamma D(m-1) + \lambda(D(m-1) - i_2) + t. \end{aligned}$$

$$\begin{aligned} x^{i_1}y^{i_2} \in M \quad &\Leftrightarrow i_2 = 0, \ldots, Dm\ ; \\ &\quad i_1 = 0, \ldots, \gamma Dm + \lambda(Dm - i_2) + t. \end{aligned}$$

for some $t = \tau Dm$. This leads to the bound as stated above.

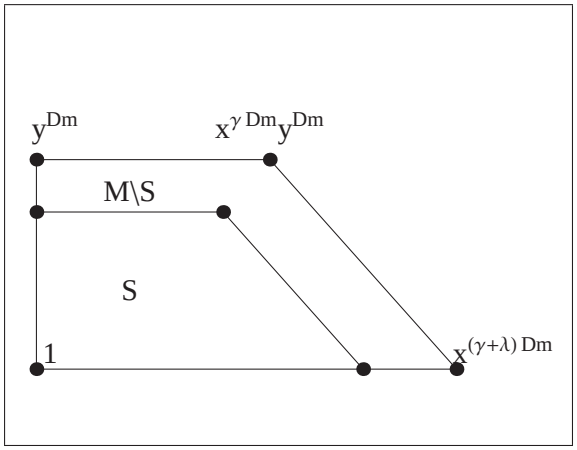


Figure 3.11: Shape of $f^m$

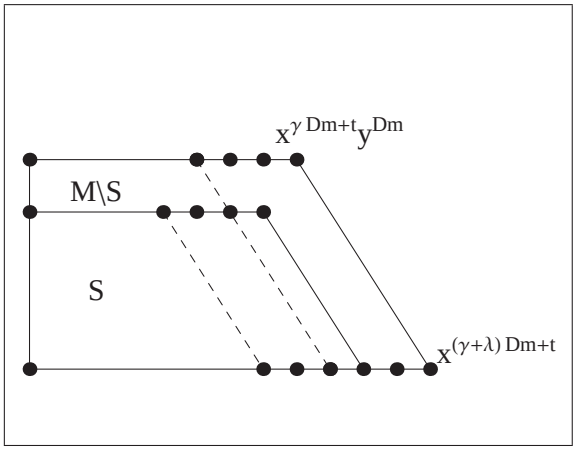


Figure 3.12: Extra $x$-shifts

Generalized Rectangles and Generalized Lower Triangles [16]

As in the modular case, we mention two generalizations of known results that can be derived using only the basic strategy.

Suppose $f(x_1, \ldots, x_v)$ is a polynomial with the shape of a *generalized rectangle*, that is, the degree of $x_i$ is $\lambda_i D$. Then the bound (heuristically for $v \geq 3$) is

$$X_1^{\lambda_1} \cdot \ldots \cdot X_v^{\lambda_v} < W^{\frac{2}{(v+1)D} - \epsilon}.$$

The first special case of this situation was again analyzed by Coppersmith in [16], finding the bound $XY < W^{\frac{2}{3D}}$ for polynomials $f(x, y)$ with degree $D$ per variable. In [7], Blömer and May generalized this for bivariate polynomials to the case of rectangles instead of squares. Our result is a generalization of the one described in [7] to polynomials with any number of variables.

The bound can easily be derived by following the basic strategy, with

$$x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} \in S \quad \Leftrightarrow \quad i_j = 0, \ldots, \lambda_j D(m-1) \quad (j = 1, \ldots, v)$$

$$x_1^{i_1} x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} \in M \quad \Leftrightarrow \quad i_j = 0, \ldots, \lambda_j Dm \quad (j = 1, \ldots, v)$$

Suppose $f(x_1, \ldots, x_v)$ is a polynomial with the shape of a *generalized lower triangle*, that is, its monomial are $x_1^{i_1} \cdot \ldots \cdot x_v^{i_v}$ for

$$0 \leq i_1 \leq \lambda_1 D,\ 0 \leq i_2 \leq \lambda_2 D - \frac{\lambda_2}{\lambda_1} i_1,\ \ldots,\ 0 \leq i_v \leq \lambda_v D - \sum_{r=1}^{v-1} \frac{\lambda_v}{\lambda_r} i_r.$$

Then the bound (heuristically for $v \geq 3$) is

$$X_1^{\lambda_1} \cdot \ldots \cdot X_v^{\lambda_v} < W^{\frac{1}{D} - \epsilon}.$$

The first known example is the polynomial $f(x, y)$ with *total degree* $D$, analyzed by Coppersmith [16]. He showed that the bound for this situation is $XY < W^{\frac{1}{D}}$. The generalization to lower triangles with unequal sides was again made by Blömer and May [7], for a bivariate polynomial $f(x, y)$ such that the monomials $x^{i_1}y^{i_2}$ appear for $0 \leq i_1 \leq D$ and $0 \leq i_2 \leq \lambda D - i_1$. Our result is an extension of their two-dimensional case.

To obtain the general bound, one can use

$$x_1^{i_1}x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} \in S \quad \Leftrightarrow \quad i_j = 0, \ldots, \lambda_j D(m-1) - \textstyle\sum_{r=1}^{j-1} \frac{\lambda_j}{\lambda_r} i_r \quad \text{for } j = 1, \ldots, v,$$

$$x_1^{i_1}x_2^{i_2} \cdot \ldots \cdot x_v^{i_v} \in M \quad \Leftrightarrow \quad i_j = 0, \ldots, \lambda_j Dm - \textstyle\sum_{r=1}^{j-1} \frac{\lambda_j}{\lambda_r} i_r \quad \text{for } j = 1, \ldots, v.$$

## 3.4 Tabular overview

In the following tables, we give an overview of all results concerning bounds for finding roots with Coppersmith methods. The first table includes the results for finding small modular roots, the second for finding small integer roots. In the second table, we have added some new results, corresponding to attacks that will be explained in detail in the following chapters.

| **Monomials of $f_N$** | **Bound** | **Reference** |
|---|---|---|
| $1, x, xy$ | $X^{2+3\tau}Y^{1+3\tau+3\tau^2} < N^{1+3\tau-\epsilon}$ | Boneh/Durfee [10] |
| $1, x, y, yz$ | $X^{1+4\tau}Y^{2+4\tau}Z^{1+4\tau+6\tau^2} < N^{1+4\tau-\epsilon}$ | Blömer/May [6] |
| generalized rectangle | $X_1^{\lambda_1} \cdot \ldots \cdot X_v^{\lambda_v} < N^{\frac{2}{(v+1)D}-\epsilon}$ | generalization of Coppersmith [16] |
| generalized lower triangle | $X_1^{\lambda_1} \cdot \ldots \cdot X_v^{\lambda_v} < N^{\frac{1}{D}-\epsilon}$ | generalization of Coppersmith [16] |

Table 3.1: Bounds for finding small modular roots of polynomials

| **Monomials of $f$** | **Bound** | **Reference** |
|---|---|---|
| upper triangle | $X^{(\lambda+\tau)^2}Y^{2(\lambda+\tau)} < W^{\frac{1}{D}(\lambda+2\tau)-\epsilon}$ | Blömer/May [7] |
| extended rectangle | $X^{\lambda^2+3\gamma\lambda+2\tau\lambda+4\tau\gamma+\tau^2+3\gamma^2}Y^{\lambda+3\gamma+2\tau} < W^{\frac{1}{D}(\lambda+2\gamma+2\tau)-\epsilon}$ | Blömer/May [7] |
| generalized rectangle | $X_1^{\lambda_1} \cdot \ldots \cdot X_v^{\lambda_v} < W^{\frac{2}{(v+1)D}-\epsilon}$ | generalization of Coppersmith [16] |
| generalized lower triangle | $X_1^{\lambda_1} \cdot \ldots \cdot X_v^{\lambda_v} < W^{\frac{1}{D}-\epsilon}$ | generalization of Coppersmith [16] |
| $1, x, y, yz$ | $X^{1+3\tau}Y^{2+3\tau}Z^{1+3\tau+3\tau^2} < W^{1+3\tau-\epsilon}$ | Section 4.4.2 |
| $1, x, y, z, yz$ | $X^{2+3\tau}Y^{3+6\tau+3\tau^2}Z^{3+3\tau} < W^{2+3\tau-\epsilon}$ | Section 4.4.2 |
| $1, x_1, x_2, x_3, x_4, x_1x_2, x_1x_4, x_2x_3, x_3x_4$ | $(X_1X_2)^{5+20\tau+27\tau^2+12\tau^3}(X_3X_4)^{5+20\tau+18\tau^2} < W^{3+12\tau+12\tau^2-\epsilon}$ | Section 5.3.1 |
| $1, x, x^2, y, z, xy, xz, yz$ | $X^{7+9\tau+3\tau^2}(YZ)^{5+\frac{9}{2}\tau} < W^{3+3\tau-\epsilon}$ | Section 5.4.1 |

Table 3.2: Bounds for finding small integer roots of polynomials

## 3.5 Complexity of attacks using Coppersmith's method

In the next chapters we will show many attacks on RSA variants that use Coppersmith methods. For those attacks, we claim that the factorization of the RSA modulus $N$ can be found in time polynomial in $n$, the bitsize of $N$. Therefore, we conclude this chapter by commenting on the complexity of Coppersmith methods.

The running time of a Coppersmith method is dominated by the lattice basis reduction. A reduction using the algorithm of Nguyen and Stehlé [58] can be performed in time $O(\omega^5(\omega+A)A)$, where $A$ is the maximal bitsize of an entry in the lattice $L$ of dimension $\omega$. Our lattice dimension $\omega$ depends on $\epsilon$ only, whereas the bitsize of the entries of the lattice in our attacks will be bounded by a polynomial in $n$, the bitsize of $N$. Therefore, the construction of $r_1, \ldots, r_v$ can be done in time polynomial in $n$. Moreover, the polynomials $r_1, \ldots, r_v$ have a fixed degree that only depends on $\epsilon$, and coefficients with bitsize polynomial in $n$.

If $r_1(x_1, \ldots, x_v)$, $r_2(x_1, \ldots, x_v)$ are two polynomials with $\deg_{x_1}(r_1) = d_1$, $\deg_{x_1}(r_2) = d_2$, then computing a resultant $\mathrm{Res}_{x_1}(r_1, r_2)$ consists of computing a determinant of a matrix of size $(d_1+d_2)\times(d_1+d_2)$. The total degree of the resultant (a polynomial in the variables $(x_2, \ldots, x_v)$) is polynomial in the total degree of $r_1$ and $r_2$. Equivalently, the size of the coefficients of the resultant is polynomial in the original coefficient sizes. To extract the root, we need to repeat this process of computing resultants a fixed number of times. Every time, computing the determinant of a matrix of size $D \times D$ can be done in a running time of order $D^3$. For details on resultant computations, we refer to [21, Chapter 3].

In Section 5.3 we use Gröbner bases as an alternative to resultants in the process of finding a common root from a set of polynomials. Computing a Gröbner basis can be done using the F4 algorithm [26] implemented in Magma. The complexity of computing a Gröbner basis for a system of equations with a finite number of solutions is polynomial in $D^v$, where $D$ is the maximal degree of the input polynomials (see for instance [29, Section 10.5]). However, in our attacks, both $D$ and $v$ will be fixed. To prevent the coefficients in the Gröbner basis from exploding, we could compute the Gröbner basis over $\mathbb{Z}_{p'}$, where $p'$ is a prime larger than the RSA modulus $N$.

Hence, for a fixed $\epsilon$, and a fixed number of variables, using Coppersmith's method to find a root can be done in polynomial time. However, note that in both the reduction phase and in the resultants/Gröbner phase, we can run into practical limitations. Although a lattice basis can be reduced in polynomial time, it could be that the reduction of a 300-dimensional lattice takes days or weeks on a computer. The same holds for the computation of resultants and Gröbner bases (especially for a larger number of variables). That is why experiments are needed to show the practical implications of theoretical attacks using Coppersmith methods.

# 4

# Partial key exposure attacks on RSA

In this chapter, we discuss the known attacks on RSA-Small-$e$ where a part of the bits of $d$ is known to an attacker. Moreover, we discuss some new partial key exposure attacks on RSA-Small-$d$ and RSA-Small-$e$.

Section 4.3 is based on [36], which is joint work with Benne de Weger. Section 4.4 is based on [25], a joint paper with Matthias Ernst, Alexander May, and Benne de Weger.

## 4.1 Introduction

The concept of partial key exposure attacks on RSA was introduced in 1997 by Boneh, Durfee and Frankel [12], and deals with the situation where an attacker has obtained some bits of the private exponent $d$. The main question is:

*How much information on the bits of $d$ is needed such that an attacker can efficiently reconstruct $d$, thereby breaking the RSA instance?*

The motivation for exploring partial key exposure attacks comes from side channel attacks such as power analysis, timing attacks, etc. Using a side channel, an attacker can expose a part of $d$, generally some MSBs (most significant bits) or LSBs (least significant bits). To see why an attacker usually gets either MSBs or LSBs of the secret exponent, we refer to [53].

In all subsequent papers about partial key exposure attacks on RSA, the assumption is made (besides knowledge of MSBs/LSBs of $d$) that one of the exponents $e, d$ (or, in RSA-CRT variants, $e$ or $d_p$ and $d_q$) is chosen to be small. This means that either $e$ or $d$ is at least significantly smaller than the modulus $N$. In Section 4.2 we will discuss these partial key exposure attacks on RSA-Small-$e$. The known partial key exposure attacks on RSA-CRT variants will be discussed in Chapter 5.

In Section 4.3 we discuss a simple but very efficient partial key exposure attack on RSA-Small-$d$ using a 2-dimensional lattice. As mentioned before, RSA-Small-$d$ is especially useful for signing operations on constrained devices such as smartcards. One could choose a private exponent $d$ that is large enough to counter the attacks by Wiener and Boneh/Durfee [75, 10]. However, since side channel attacks are often mounted on

smartcard-settings, exploring the possibilities of partial key exposure attacks on RSA-Small-$d$ is very important. The attack we discuss in Section 4.3 is a generalization of the attacks by Wiener [75] and Verheul/van Tilborg [72] to partial key exposure attacks.

In Section 4.4 we discuss new partial key exposure attacks on RSA-Small-$e$ and RSA-Small-$d$. In fact, we show that whenever one of the exponents $e$, $d$ is chosen significantly smaller than $N$, a partial key exposure attack exists.

## 4.2 Known attacks

Although Boneh, Durfee, and Frankel [12] introduced attacks that use partial knowledge of the secret exponent $d$, we shall start with a result of Coppersmith that the attacks in [12] are based on.

**Theorem 4.1 (Coppersmith, [16])**
*Let $N = pq$ be an $n$-bit RSA modulus, with $p$ and $q$ primes of bitsize $\frac{1}{2}n$. Suppose that either the high-order $\frac{1}{4}n$ bits of $p$ or the low-order $\frac{1}{4}n$ bits of $p$ are known. Then the factorization of $N$ can be found in time polynomial in $n$.*

**Proof.**
As Coppersmith showed in [16], this result can easily be derived from one of Coppersmith's bounds that we mentioned in Section 3.3.2. In [16], Coppersmith derives the bound

$$XY < W^{\frac{2}{3D}-\epsilon}$$

for polynomials $f(x, y)$ with degree per variable $D$. Suppose that an attacker knows an MSB part of $p$ of size at least $N^{\frac{1}{2}-\delta}$. That means that he knows an approximation $\tilde{p}$ of $p$ such that the unknown part $p_0 = p - \tilde{p}$ satisfies $|p_0| < N^{\delta}$. Additionally, he can also derive an approximation $\tilde{q}$ of $q$ such that $|q_0| = |q - \tilde{q}| < N^{\delta}$. The polynomial

$$f(x, y) = (\tilde{p} + x)(\tilde{q} + y) - N$$

has a root $(x_0, y_0) = (p_0, q_0)$. Since $f$ is a polynomial where the degree per variable is 1, it holds that if $XY < W^{\frac{2}{3}-\epsilon}$ is satisfied, then the unknown $p_0$ and $q_0$ can be found. Here, $X = Y = N^{\delta}$ and $W = \max\{\tilde{q}X, \tilde{p}Y, XY, |N - \tilde{p}\tilde{q}|\} = N^{\frac{1}{2}+\delta}$. Hence, the asymptotic bound is

$$2\delta < \frac{2}{3}\left(\frac{1}{2} + \delta\right)$$

which leads to $\delta < \frac{1}{4}$. The result for known LSBs of $p$ can be derived in a similar way.

□

The first partial key exposure attack on RSA-Small-$e$, by Boneh, Durfee, and Frankel, is summarized in the following theorem.

**Theorem 4.2 (Boneh/Durfee/Frankel (LSBs), [12])**
*Let $N = pq \equiv 3 \mod 4$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$. Let $e$, $d$ satisfy $ed \equiv 1 \mod \phi(N)$ with bitsize(e) $= \alpha n$ and bitsize(d) $= n$, for some $\alpha \in (0, \frac{1}{4})$. Suppose that an LSB part of $d$ is known of at least $\frac{1}{4}n$ bits. Then the factorization of $N$ can be found in time polynomial in $n$ and $e$.*

**Proof sketch.**
For a formal proof, we refer to [11]. However, we will sketch how this attack uses Coppersmith's theorem (Theorem 4.1). Suppose $d_0 \equiv d \mod 2^B$ is the known LSB part of $d$. Then the RSA key equation $ed = 1 + k(N + 1 - (p + q))$ implies that

$$ed_0 \equiv 1 + k(N + 1 - s) \mod 2^B, \text{ for } s = p + q.$$

If all possibilities for $k \in \{1, \ldots, e\}$ are tried then in one of these trials $s_0 \equiv p + q \mod 2^B$ can be found. Next, one tries to solve the equation

$$p^2 - s_0 p + N \equiv 0 \mod 2^B,$$

of which $p_0 \equiv p \mod 2^B$ is a solution. A solution $x = (p - \frac{s_0}{2})$ modulo $2^B$ of

$$x^2 \equiv \left(\frac{s_0}{2}\right)^2 - N \mod 2^B,$$

can easily be found if the right hand side is congruent to 1 modulo 8 (see [63, p.184] for a description of the method). Now,

$$\left(\frac{s_0}{2}\right)^2 - N \equiv \frac{(p-q)^2}{4} \mod 2^B$$

and $N \equiv 3 \mod 4$ ensures that the right hand side is congruent to 1 modulo 8. Hence, $p_0 \equiv p \mod 2^{\frac{n}{4}}$ can be found. With Theorem 4.1, this reveals the factorization of $N$.
□

Since RSA-Small-$e$ is a popular variant, the above attack for which only a quarter of the LSBs of $d$ are necessary is very serious. Steinfeld and Zheng [68] discovered that the above attack works less well if $p-q$ is a multiple of a large power of 2. Suppose that $p - q = 2^A \cdot y$ for some odd integer $y$. Then

$$\left(\frac{s_0}{2}\right)^2 - N \equiv \frac{(p-q)^2}{4} \equiv 2^{2A-2} y^2 \mod 2^B.$$

Therefore,

$$\left(\frac{x}{2^{A-1}}\right)^2 \equiv y^2 \mod 2^{B-2A+2}.$$

Hence, if $y^2 \equiv 1 \mod 8$, then one can obtain only $p \mod 2^{B-A+1}$ instead of $p \mod 2^B$.

Therefore, Steinfeld and Zheng [68] propose to choose $p$ and $q$ such that they share a number of LSBs, in order to diminish the power of the attack in [12]. Whether or not this specific design criterion for $p$ and $q$ gives rise to other attacks is still an open question.

The known attacks from [12] on RSA-Small-$e$ with known MSBs of $d$ are summarized in the following theorem.

**Theorem 4.3 (Boneh/Durfee/Frankel (MSBs), [12])**
*Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$. Let $e$, $d$ satisfy $ed \equiv 1 \mod \phi(N)$ with bitsize(e) $= \alpha n$ and bitsize(d) $= n$, for some $\alpha \in (0, \frac{1}{2})$. Suppose that an MSB part of $d$ is known of size at least $N^{1-\delta}$. Then the following statements are true.*

1. *If $e$ is prime and $\alpha \in (\frac{1}{4}, \frac{1}{2})$, then the factorization of $N$ can be found in time polynomial in $n$ if $\delta < 1 - \alpha$.*
2. *If $e$ is a product of $r$ distinct (known) factors and $\alpha \in (\frac{1}{4}, \frac{1}{2})$, then the factorization of $N$ can be found in time polynomial in $n$ and $2^r$ if $\delta < 1 - \alpha$.*
3. *If the factorization of $e$ is unknown and $\alpha \in (0, \frac{1}{2})$, then the factorization of $N$ can be found in time polynomial in $n$ and $\frac{N}{d}$ if $\delta < \alpha$.*
4. *If the factorization of $e$ is unknown and $\alpha \in (0, \frac{1}{2})$, then the factorization of $N$ can be found in time polynomial in $n$, $\frac{N}{d}$, and $\frac{\sqrt{N}}{p-q}$, if $\delta < \frac{1}{4}$.*

For the proofs, we refer to [12]. Most of the proofs, like the one for Theorem 4.2, use the information that is given to find either MSBs or LSBs of $p$, after which Theorem 4.1 is applied. Another fact that is used in some of the proofs is that for $e < N^{\frac{1}{2}}$, the value of $k := \frac{ed-1}{\phi(N)}$ is known up to some constant error if as many MSBs of $d$ as the bitsize of $e$ are given. Since this result influences our new attacks, we will discuss it briefly.

**Theorem 4.4 (Boneh/Durfee/Frankel, [12])**
*Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$. Let $e$, $d$ satisfy $ed \equiv 1 \mod \phi(N)$ with bitsize(e) $= \alpha n$ and bitsize(d) $= n$, for some $\alpha \in (0, \frac{1}{2})$. Suppose that an MSB part of $d$ is known of size at least $N^{\alpha}$. Then an approximation $\tilde{k}$ of $k = \frac{ed-1}{\phi(N)}$ can be computed such that the difference between $\tilde{k}$ and $k$ is bounded by a constant.*

**Proof.**
Suppose $d = \tilde{d} + d_0$, where $\tilde{d}$ is a known approximation of $d$ and $d_0$ is the unknown LSB part of $d$ of size $N^{\delta}$ for some $\delta < 1 - \alpha$. Then, for $\tilde{k} = \frac{e\tilde{d}-1}{N}$,

$$\begin{aligned} \left|k - \tilde{k}\right| &= \left|\frac{e(\tilde{d} + d_0) - 1}{\phi(N)} - \frac{e\tilde{d} - 1}{N}\right| = \left|\frac{(e\tilde{d} - 1)(N - \phi(N)) + ed_0N}{N\phi(N)}\right| \\ &< \frac{e}{\phi(N)}\left|\frac{\tilde{d} \cdot 3N^{\frac{1}{2}}}{N} + d_0\right| < \frac{N^{\alpha}}{\frac{1}{2}N}\left(3N^{\frac{1}{2}} + N^{\delta}\right) = 6N^{\alpha - \frac{1}{2}} + 2N^{\alpha+\delta-1} < 8. \end{aligned}$$

□

Boneh, Durfee and Frankel posed the open question whether or not there exist partial key exposure attacks for $e > N^{\frac{1}{2}}$. This question was answered affirmatively by Blömer and May, who showed other partial key exposure attacks. Their best results are summarized in the following theorems.

**Theorem 4.5 (Blömer/May (LSBs), [6])**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$ the following holds: Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$. Let $e$, $d$ satisfy $ed \equiv 1 \mod \phi(N)$ with bitsize($e$) $= \alpha n$ and bitsize($d$) $= n$, for some $\alpha \in (0, \frac{7}{8})$. Suppose that an LSB part of $d$ is known of size at least $N^{1-\delta}$, with $\delta \in (0,1)$, and*

$$\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1+6\alpha} - \epsilon.$$

*Then $N$ can be factored in time polynomial in $n$.*

**Proof.**
One can write $d$ as $d = d_0 + x2^{(1-\delta)n}$, where $x$ is the unknown MSB part of $d$ of size $N^\delta$. Hence,

$$e(d_0 + x2^{(1-\delta)n}) = 1 + k(N + 1 - (p+q)).$$

It follows that

$$f_{e2^{(1-\delta)n}}(y,z) = ed_0 - 1 - y(N+1-z)$$

has a small root $(y_0, z_0) = (k, p+q)$ modulo $e2^{(1-\delta)n}$.

From the "Boneh/Durfee" example in Section 3.3.1, one can see that for roots of a polynomial modulo $e2^{(1-\delta)n}$, with monomials $1, y, yz$, for some $\tau \geq 0$ that can be optimized later, the following bound holds:

$$X^{2+3\tau}Y^{1+3\tau+3\tau^2} < (e2^{(1-\delta)n})^{1+3\tau-\epsilon}.$$

Substituting $Y = N^\alpha$ and $Z = N^{\frac{1}{2}}$ leads to the asymptotical bound

$$\begin{aligned} \alpha(2+3\tau) + \frac{1}{2}(1+3\tau+3\tau^2) &< (\alpha+1-\delta)(1+3\tau), \text{ or} \\ 3\tau^2 + 3\tau(2\delta-1) + (2\alpha+2\delta-1) &< 0. \end{aligned}$$

The optimal $\tau = \frac{1}{2} - \delta$ gives

$$\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1+6\alpha}.$$

□

**Theorem 4.6 (Blömer/May (MSBs), [6])**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$ the following holds: Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$. Let $e$, $d$ satisfy $ed \equiv 1 \mod \phi(N)$ with bitsize($e$) $= \alpha n$ and bitsize($d$) $= n$, for some $\alpha \in (\frac{1}{2}, \frac{\sqrt{6}-1}{2})$. Suppose that an MSB part of $d$ is known of size at least $N^{1-\delta}$, with $\delta \in (0,1)$, and*

$$\delta < \frac{1}{8}(5 - 2\alpha - \sqrt{36\alpha^2 + 12\alpha - 15}) - \epsilon.$$

*Then $N$ can be factored in time polynomial in $n$.*

**Proof.**
One can write $d$ as $d = \tilde{d} + d_0$, where $d_0$ is the unknown LSB part of $d$ of size $N^\delta$. Knowledge of MSBs of $d$ also leads to knowledge of MSBs of $k$, since

$$\tilde{k} = \frac{e\tilde{d} - 1}{N + 1}$$

is a good approximation of $k$. Analogous to the proof of Theorem 4.4, one can check that $k_0$, the unknown LSB part of $k$, satisfies $|k_0| = |k - \tilde{k}| < 6N^{\alpha - \frac{1}{2}} + 2N^{\alpha+\delta-1} < 8N^{\alpha-\frac{1}{2}}$.

Hence, if one substitutes $x = d_0$, $y = k_0$, and $z = p + q - 1$ in the RSA key equation, then one obtains

$$e(\tilde{d} + x) = 1 + (\tilde{k} + y)(N - z).$$

It follows that

$$f_N(x, y, z) = ex + (\tilde{k} + y)z + (e\tilde{d} - 1)$$

has a small root $(x_0, y_0, z_0) = (d_0, k_0, p + q - 1)$ modulo $N$.

In Section 3.3, we have discussed the analysis for a polynomial modulo $N$, with monomials $1, x, z, yz$ (see "Blömer/May attack", one of the examples in Section 3.3.1). We saw that, for some $\tau \geq 0$ that can be optimized later, the following bound holds:

$$X^{1+4\tau}Y^{1+4\tau+6\tau^2}Z^{2+4\tau} < N^{1+4\tau-\epsilon}.$$

Substituting $X = N^\delta$, $Y = N^{\alpha-\frac{1}{2}}$, and $Z = N^{\frac{1}{2}}$, one finds the asymptotical bound

$$\begin{aligned} \delta(1 + 4\tau) + (\alpha - \frac{1}{2})(1 + 4\tau + 6\tau^2) + \frac{1}{2}(2 + 4\tau) &< (1 + 4\tau), \text{ or} \\ 6(2\alpha - 1)\tau^2 + 8\tau(\delta + \alpha - 1) + (2\alpha + 2\delta - 1) &< 0. \end{aligned}$$

The optimal $\tau = \dfrac{8(1 - \delta - \alpha)}{12(2\alpha - 1)}$ gives

$$\delta < \frac{1}{8}(5 - 2\alpha - \sqrt{36\alpha^2 + 12\alpha - 15}).$$

□

## 4.3 A new "2-dimensional" attack

In this section, we explore for which sizes of $d$ one can mount an attack in a few seconds with a very simple method using a 2-dimensional lattice. Our result is summarized in the following theorem.

**Theorem 4.7**
*Under Assumption 3.4, the following holds: Let $N = pq$ be an $n$-bit RSA modulus, and $p$ and $q$ primes of bitsize $\frac{1}{2}n$. Let $e$, $d$ satisfy $ed \equiv 1 \bmod \phi(N)$ with bitsize(e) $= n$ and bitsize(d) $= \beta n$ for some $0 < \beta < \frac{1}{2}$. Given a (total) amount of $(2\beta - \frac{1}{2})n$ MSBs and/or LSBs of $d$ (see Figure 4.1), $N$ can be factored in time polynomial in $n$, using a 2-dimensional lattice.*

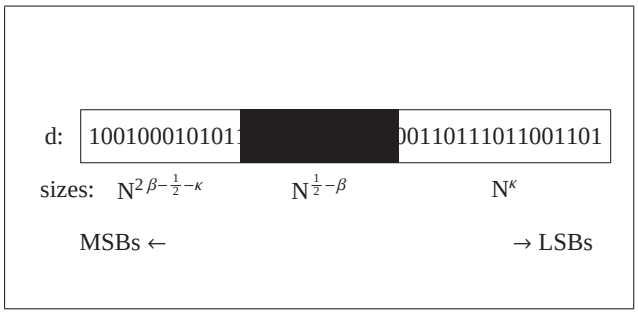


Figure 4.1: Partition of $d$ for small $d$

Besides showing how the result of Theorem 4.7 is obtained, we show that the results of Wiener [75] and Verheul/van Tilborg [72] can be obtained by our attack on small $d$ and are simply special (homogeneous and provable) cases.

### 4.3.1 Description of the new attack

Let $d = N^{\beta} < N^{\frac{1}{2}}$ and $e < \phi(N) < N$. Moreover, let $d_L$ be the known LSB part of $d$ of size $N^{\kappa}$, followed by an unknown middle part $x$ of size $N^{\delta}$, which itself is followed by a known MSB part $d_M$, of size $N^{\beta-\kappa-\delta}$. Hence, we can write

$$d = d_L + 2^{\lfloor \kappa n \rceil} x + 2^{\lfloor \kappa n \rceil + \lfloor \delta n \rceil} d_M,$$

where the notation $\lfloor\ \rceil$ means rounding to the nearest integer.

When we substitute the partition of $d$ in the RSA key equation $ed = 1 + k\phi(N)$, we obtain

$$e2^{\lfloor \kappa n \rceil} x + ed_L + e2^{\lfloor \kappa n \rceil + \lfloor \delta n \rceil} d_M - 1 = k(N - (p + q - 1)).$$

Therefore, we must find the solution $(x, y, z) = (x, k, p + q - 1)$ of the trivariate equation

$$e2^{\lfloor \kappa n \rceil} x - Ny + yz + R - 1 = 0, \text{ with } R = ed_L + e2^{\lfloor \kappa n \rceil + \lfloor \delta n \rceil} d_M.$$

The equation above implies that

$$\left|e2^{\lfloor \kappa n \rceil}x - Ny + R\right| = |1 - yz| \leq |k(p+q-1)| \leq |d(p+q)| \leq 3N^{\beta+\frac{1}{2}}.$$

This is an inhomogeneous diophantine approximation problem in the unknowns $x$ and $y$. To solve it, we define a lattice $L$ spanned by the rows of $\Gamma$, with

$$\Gamma = \begin{pmatrix} C & e2^{\lfloor \kappa n \rceil} \\ 0 & N \end{pmatrix} \quad \text{and a point } \mathbf{v} = (0, -R),$$

where $C$ is an integer of size $N^{\beta-\delta+\frac{1}{2}}$.

The lattice point $(x, -y) \cdot \Gamma$ is close to $\mathbf{v}$, since

$$(x, -y) \cdot \Gamma - \mathbf{v} = (Cx, e2^{\lfloor \kappa n \rceil}x - Ny + R) \approx (N^{\beta+\frac{1}{2}}, N^{\beta+\frac{1}{2}}).$$

Our strategy to find $x$ and $y$ is therefore to start with a lattice point $\mathbf{v}'$ close to $\mathbf{v}$, and add small multiples of the reduced basis vectors of the lattice $L$ until we get $\Gamma \begin{pmatrix} x \\ -y \end{pmatrix}$. To do so, we apply lattice basis reduction to the rows of $\Gamma$, and obtain a reduced matrix $\Gamma_{\text{red}}$, whose rows still span $L$. We aim to find an integer pair $(z_1, z_2)$ for which

$$(z_1, z_2) \cdot \Gamma_{\text{red}} = (x, -y) \cdot \Gamma - \left\lfloor \mathbf{v}\Gamma_{\text{red}}^{-1} \right\rceil \Gamma_{\text{red}},$$

where $\left\lfloor \mathbf{v}\Gamma_{\text{red}}^{-1} \right\rceil = \mathbf{v}'$ is the lattice point we get from rounding the elements of $\mathbf{v}\Gamma_{\text{red}}^{-1}$ to nearest integers. Alternatively, one could also solve the closest vector problem to obtain a lattice point $\mathbf{v}'$ to start with, but in this way, the closest vector will almost immediately appear as well.

It can be checked that

$$(z_1, z_2) \cdot \Gamma_{\text{red}} = ((x, -y) \cdot \Gamma - \mathbf{v}) - (\left\lfloor \mathbf{v}\Gamma_{\text{red}}^{-1} \right\rceil \Gamma_{\text{red}} - \mathbf{v}) \approx (N^{\beta+\frac{1}{2}}, N^{\beta+\frac{1}{2}}) + (\epsilon_1, \epsilon_2) \cdot \Gamma_{\text{red}},$$

with $|\epsilon_i| < \frac{1}{2}$. Therefore

$$(z_1, z_2) \approx (N^{\beta+\frac{1}{2}}, N^{\beta+\frac{1}{2}}) \cdot \Gamma_{\text{red}}^{-1} + (\epsilon_1, \epsilon_2).$$

We know that the reduced vectors $\mathbf{r}_1$, $\mathbf{r}_2$ of the reduced lattice basis represented by $\Gamma_{\text{red}}$ satisfy $||\mathbf{r}_1|| \approx a^{-1} \det(L)^{\frac{1}{2}}$ and $||\mathbf{r}_2|| \approx a \det(L)^{\frac{1}{2}}$ for some $a \geq 1$. Hence,

$$\begin{aligned} \Gamma_{\text{red}} &= \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \approx \det(L)^{\frac{1}{2}} \cdot \begin{pmatrix} a^{-1} & a^{-1} \\ a & a \end{pmatrix} \text{ and} \\ \Gamma_{\text{red}}^{-1} &= \frac{1}{\det(L)} \begin{pmatrix} r_{22} & -r_{12} \\ -r_{21} & r_{11} \end{pmatrix} \approx \det(L)^{-\frac{1}{2}} \cdot \begin{pmatrix} a & a^{-1} \\ a & a^{-1} \end{pmatrix}. \end{aligned}$$

Thus,

$$\begin{aligned} (z_1, z_2) &\approx (a \det(L)^{-\frac{1}{2}} N^{\beta+\frac{1}{2}} + \epsilon_1, a^{-1} \det(L)^{-\frac{1}{2}} N^{\beta+\frac{1}{2}} + \epsilon_2) \\ &\approx (aN^{\frac{1}{2}(\beta+\delta-\frac{1}{2})} + \epsilon_1, a^{-1} N^{\frac{1}{2}(\beta+\delta-\frac{1}{2})} + \epsilon_2). \end{aligned}$$

Each pair $(z_1, z_2)$ leads to a pair $(x, -y)$. If we try every $x$ as the unknown part of $d$, and every $y$ as $k$, we can find a $\phi$ that satisfies $ed - 1 = k\phi$. First we test whether $\phi$, computed as $\frac{ed-1}{k}$ is integral. For every possibility for $\phi$, we find the $p, q$ that satisfy both $N - \phi = p + q - 1$ and $N = pq$.

The number of pairs $(z_1, z_2)$ to check is of size

$$aN^{\frac{1}{2}(\beta+\delta-\frac{1}{2})} \cdot \max\{a^{-1}N^{\frac{1}{2}(\beta+\delta-\frac{1}{2})}, 1\}.$$

Hence, the number of pairs $(z_1, z_2)$ to try is either

- $O(N^{\beta+\delta-\frac{1}{2}})$, when $a < N^{\frac{1}{2}(\beta+\delta-\frac{1}{2})}$, or
- $O(aN^{\frac{1}{2}(\beta+\delta-\frac{1}{2})})$, when $a > N^{\frac{1}{2}(\beta+\delta-\frac{1}{2})}$.

Note that in the latter case, $z_2 = 0$, but we do have to check for all $z_1$ separately.

As we shall see later, the attacks of Wiener [75] and Verheul/van Tilborg [72] are special cases of this attack. For these situations, we show that the attacks are provable instead of heuristic, simply because $(x, -y) \cdot \Gamma$ is small enough to ensure that the search region does not depend on $a$.

However, if we are outside the range of Wiener's and Verheul/van Tilborg's attacks, it is highly unusual that the lattice involved contains an exceptionally small nonzero vector. By Assumption 3.4, we take $a$ to be close to 1. Under this heuristic, the number of pairs $(z_1, z_2)$ to try is $O(N^{\beta+\delta-\frac{1}{2}})$. We will comment later on how this assumption holds in the examples that we tested.

Under our assumption, and provided that $\delta$ is smaller than or at most only marginally larger than $\frac{1}{2} - \beta$, we can efficiently try all pairs $(z_1, z_2)$ and find the factorization of $N$.

One may note that by knowing MSBs of $d$, one can also obtain an MSB part of $k$. However, splitting $k$ into a known and an unknown part results in more combinations of variables, which we can only represent in a 3-dimensional lattice instead of a 2-dimensional one. The 3-dimensional lattice attack will give a poorer performance than the method described in this section. This is an example of a common phenomenon in lattice based cryptanalysis, namely that sometimes one can get better results by leaving out information that one knows, just because of the monomials of the equation involved.

Complexity of the attack:

The attack starts with one lattice basis reduction for a 2-dimensional lattice. This is just a Lagrange reduction, which takes at most $O((\log N)^3)$ bit operations.

Secondly, a number of $O(N^{\beta+\delta-\frac{1}{2}})$ pairs $(z_1, z_2)$ have to be checked for coming from a solution. For each vector this check takes $O((\log N)^2)$ bit operations.

It follows that the bit complexity of our attack is $O((\log N)^3)$ when $\delta \leq \frac{1}{2} - \beta$, which is polynomial. When $\delta = \frac{1}{2} - \beta + \epsilon$ the bit complexity becomes exponential, namely $O(N^\epsilon(\log N)^2)$. This results in an increased workload by a factor $N^\epsilon$.

In other words, for an additional amount of $r$ unknown bits, the complexity is equivalent to an exhaustive search over $r$ bits. Furthermore, if we let both $d$ and the unknown part of $d$ grow $r$ bits, such that the known part of $d$ stays of the same size, one can check that the extra workload will be an exhaustive search over $2r$ bits. This relates directly to a result of Verheul and van Tilborg [72], on which we shall comment later.

Examples:

We have done several experiments for this attack. A typical case is with 2048-bit $N$ and $\delta = 0.156$, $\beta = 0.350$ (e.g. $\epsilon = 0.006$), meaning that $d$ has about 717 bits, of which at most 320 of the least significant bits are unknown.

In this case, $N^{\frac{1}{2}(\delta+\beta-\frac{1}{2})} \approx 70$. We typically find a hit with $\|z\| \lessapprox 200$. A search area like this takes only a few seconds with Mathematica 5 on a 2GHz Pentium 4 PC. And with $\delta \leq \frac{1}{2} - \beta$ typically $\|z\| \approx 1$, and the computation time is only a fraction of a second.

Here's a baby example for $\{\delta = 0.156, \beta = 0.35\}$. Let the 128-bit public key be given by

$$N = 269866491905568049204176579604167754067,$$
$$e = 222981052634419442506270512141611354797.$$

Now suppose we know some MSBs of $d$, hence we know an approximation

$$\tilde{d} = 24584250313023$$

of $d$ for which $d_0 = d - \tilde{d}$ is $0.156 \cdot 128 \approx 20$ bits. We take

$$C = 2^{\lfloor 128 \cdot (0.35 - 0.156 + 0.5) \rceil} = 2^{89} \text{ and}$$
$$R = e\tilde{d} = 548182201302592421821865798975772347127175836262133 1,$$

and we know that we are looking for $\{d_0, k\}$ such that

$$(d_0, -k) \cdot \Gamma - \mathbf{v} = (d_0, -k) \cdot \begin{pmatrix} C & e \\ 0 & N \end{pmatrix} - (0, -R)$$

is small. Then $\Gamma_{\text{red}}$ is given by

$$\begin{pmatrix} 939237487206210868368 71453999104 & 223858603616044679201441362439981 \\ -645630915298759729739927100850176 & 239654325473299927083414831489037 \end{pmatrix}$$

and $\lfloor \mathbf{v}\Gamma_{\text{red}}^{-1} \rceil = (-21188034626414783992, -3082348742879388262)$.

We then enumerate the pairs $\{z_1, z_2\}$, for each value computing

$$(x, -y) = \left((z_1, z_2) \cdot \Gamma_{\text{red}} + \lfloor \mathbf{v}\Gamma_{\text{red}}^{-1} \rceil \Gamma_{\text{red}}\right) \Gamma^{-1}.$$

We try $d = \tilde{d} + x$ and $k = y$, and solve $N + 1 - \left(p + \frac{N}{p}\right) = \frac{ed-1}{k}$ to get a possible factor $p$. At $(z_1, z_2) = (-2, -1)$ we have a hit, namely $x = 1016998$, $y = 20313089635876$, so we find that $d = 24584251330021$ and $k = 20313089635876$.

It follows that $\phi(N) = 269866491905568049171299025219693706736$, and we obtain the factors

$$p = 15833051453602685849, \; q = 17044502930871361483.$$

### 4.3.2 Special cases: Wiener and Verheul/van Tilborg

Wiener showed that when $d < N^{\frac{1}{4}}$, it can be found in polynomial time [75]. Verheul and van Tilborg's extension of Wiener's result shows the price when $d$ is slightly larger than this [72]. Their attacks can be seen as homogeneous diophantine approximation problems, and continued fraction techniques are used to solve them.

Here, we will show that Wiener's and Verheul/van Tilborg's attacks are special cases of our method. Moreover, we will show that in these cases the method is provable, in other words, it does not depend on the size of $a$ (the parameter that describes the unbalancedness of the lattice).

As we explained in Section 2.3, Wiener bases his attack on the fact that $\frac{k}{d}$ can be found as a convergent of $\frac{e}{N}$ if

$$\left|\frac{e}{N} - \frac{k}{d}\right| < \frac{1}{2d^2}.$$

It is well known (see for instance [53]) that this can also be described using a 2-dimensional lattice. When we assume no part of $d$ is known ($d_M = d_L = 0$), it follows that $R = 0$ and

$$\Gamma = \begin{pmatrix} C & e \\ 0 & N \end{pmatrix}, \quad \mathbf{v} = (0, 0),$$

with $C$ of size $N^{\beta-\delta+\frac{1}{2}} = N^{\frac{1}{2}}$, will reproduce Wiener's result, namely that the method will work if $\beta < \frac{1}{4}$. Later we will show that the solution will be found by the shortest lattice vector only, making this case provable.

Verheul and van Tilborg [72] have given an extension of Wiener's attack, where $d$ is at most slightly larger than $N^{\frac{1}{4}}$ and no bits are known. To find $\frac{k}{d}$, they look not only at convergents of $\frac{e}{N}$, but also at 'linear combinations' of consecutive convergents, which, be it not the best, nevertheless are pretty good approximations. When $\frac{p_{i-1}}{q_{i-1}}, \frac{p_i}{q_i}$ are consecutive convergents, then they also look for approximations to $\frac{e}{N}$ of the form $\frac{\lambda p_i + \mu p_{i-1}}{\lambda q_i + \mu q_{i-1}}$ for parameters $\lambda, \mu \in \mathbb{N}$. Then they have a weaker inequality to satisfy, namely

$$\left|\frac{e}{N} - \frac{k}{d}\right| < \frac{c}{d^2},$$

where the exact value for $c$ depends on the search region for $\lambda$ and $\mu$. In this way they show that in order to extend Wiener's result for $d < N^{\frac{1}{4}}$ by $r$ bits, one has to do an additional computation with a complexity of an exhaustive search over $2r$ bits.

In the language of lattices this becomes immediately clear. With $\Gamma$ as above and $\mathbf{v} = \mathbf{0}$ (as we're still in the homogeneous case), we have seen that for $\delta = \beta = \frac{1}{4} + \epsilon$, the complexity of the attack is $O(N^{2\epsilon}(\log N)^2)$.

The example given in [72] will go as follows in our method. We start with the lattice

$$\Gamma = \begin{pmatrix} 2^{38} & e \\ 0 & N \end{pmatrix} = \begin{pmatrix} 2^{38} & 7115167804808765210427 \\ 0 & 31877667548624237348233 \end{pmatrix}$$

(note that in [72] the value of $e$ contains a misprint).

We compute the reduced basis

$$\Gamma_{\text{red}} = \begin{pmatrix} 42694311384449024 & 34997160860155755 \\ 87227281088446464 & -133735834148055649 \end{pmatrix}.$$

The lattice point we need is $(2d, -k) \cdot \Gamma = (3295186, -735493) \cdot \Gamma = (11, 5) \cdot \Gamma_{\text{red}}$. Here $2d$ appears instead of $d$ because in [72] $ed \equiv 1 \bmod \text{lcm}(p-1, q-1)$ is taken, and in this case $\gcd(p-1, q-1)$ appears to be equal to 2.

This shows that, at least in this example, the efficiency of our method is comparable to [72], since we had to search for the numbers 11 and 5 of respectively 3.5 and 2.3 bits, together less than 7 bits (rather than 6 bits, because we have to allow negative values for one of the coordinates).

The fact that Verheul and van Tilborg require a computation with a complexity of a $2r$-bit exhaustive search to allow $r$ unknown bits more than $\frac{1}{4}$th of $N$ for both $d$ and the unknown part of $d$ (which, in this case, are of course the same), corresponds to our complexity results. However, it does not directly imply that their method can be used in a partial key exposure setting. In that sense our result, with the homogeneous case being a special case of the general case, implies the result of [72], but not the other way around. We believe that the method of Verheul and van Tilborg can be combined with the method of Baker and Davenport [1], for solving inhomogeneous diophantine approximation problems, but we see no advantages above our uniform and clean lattice method.

Finally, we will show that the attacks of Wiener and Verheul/van Tilborg are *provable* cases of our method.

Recall that we look for a small pair $(d, k)$ such that

$$(d, -k) \cdot \begin{pmatrix} C & e \\ 0 & N \end{pmatrix} = (Cd, ed - kN) \approx (N^{\beta+\frac{1}{2}}, N^{\beta+\frac{1}{2}}).$$

We will argue that if $d < N^{\frac{1}{4}}$ (Wiener's case), this small vector is actually the smallest nonzero lattice vector. Since the Lagrange reduction always gives the shortest vector, we do not have to try different values for $z_1, z_2$.

Suppose it is not the smallest vector. Then the smallest vector cannot be linearly independent from it, for else the product of their sizes is smaller than $N^{2\beta+1} < N^{\frac{3}{2}}$, whereas the determinant of the lattice is $\det(L) = CN = N^{\frac{3}{2}}$. This is a contradiction. The other option when $(Cd, ed - kN)$ is not the smallest vector, is that the smallest vector is

$$(Cx, ex - yN) = \alpha(Cd, ed - kN), \text{ for some } \alpha \in [-1, 1].$$

It follows that $d = \frac{1}{\alpha}x$ and $k = \frac{1}{\alpha}y$, and since $ed - k\phi(N) = 1$, it must hold that

$$ex - y\phi(N) = \alpha.$$

Since the left hand side is an integer, $\alpha \neq 0$, and $\alpha \in [-1, 1]$, it follows that $|\alpha| = 1$. Therefore, $d = |x|$ and $k = |y|$. Hence, the shortest reduced basis vector immediately gives us $d$ and $k$. Thus, the method is clearly provable.

In the case of Verheul/van Tilborg's attack, $d = N^{\frac{1}{4}+\epsilon}$, so

$$(Cd, ed - kN) \approx (N^{\beta+\frac{1}{2}}, N^{\beta+\frac{1}{2}}) = (N^{\frac{3}{4}+\epsilon}, N^{\frac{3}{4}+\epsilon}),$$

so this vector is not the smallest reduced vector. However, one can see that the smallest vector must be linearly independent of it, so we know that

$$a^{-1} \det(L)^{\frac{1}{2}} \cdot N^{\frac{3}{4}+\epsilon} \geq \det(L).$$

It follows that $a < \det(L)^{-\frac{1}{2}} N^{\frac{3}{4}+\epsilon} = \det(L)^{-\frac{1}{2}} N^{\beta+\frac{1}{2}} = N^{\frac{1}{2}(\beta+\delta-\frac{1}{2})}$ and from the complexity computations in the previous section, we know that this means that the search area is $O(N^{\beta+\delta-\frac{1}{2}}) = O(N^{2\epsilon})$. So one can see that in this case, one also does not depend on Assumption 3.4.

### 4.3.3 Experiments for the new attack

In the following table, we show the running time of the 2-dimensional attack, for moduli $N$ of 2048 bits. This time includes the lattice basis reduction and trying all pairs $(z_1, z_2)$ to find $p$, $q$.

| $\beta$ | $\delta$ | Running time |
|---|---|---|
| 0.30 | 0.050 | 1 sec. |
| 0.30 | 0.100 | 1 sec. |
| 0.30 | 0.150 | 1 sec. |
| 0.30 | 0.200 | 1 sec. |
| 0.30 | 0.205 | 2 sec. |
| 0.30 | 0.210 | 21 min. |
| 0.35 | 0.050 | 1 sec. |
| 0.35 | 0.100 | 1 sec. |
| 0.35 | 0.150 | 1 sec. |
| 0.35 | 0.155 | 2 sec. |
| 0.35 | 0.160 | 21 min. |

| $\beta$ | $\delta$ | Running time |
|---|---|---|
| 0.40 | 0.050 | 1 sec. |
| 0.40 | 0.100 | 1 sec. |
| 0.40 | 0.105 | 2 sec. |
| 0.40 | 0.110 | 21 min. |
| 0.45 | 0.050 | 1 sec. |
| 0.45 | 0.055 | 2 sec. |
| 0.45 | 0.060 | 21 min. |

Table 4.1: Experimental results: 2-dimensional attack

The table shows that for $\beta = 0.30$ and $\delta = 0.205$, our attack works in approximately 2 seconds (this is an average over 50 experiments). For $\epsilon = 0.01$, the running time is still a reasonable 21 minutes. The experiments were performed using a simple Mathematica program that runs on a computer with Pentium III processor of 733 MHz.

Aside from the efficiency of the attack, we should also comment on the validity of our assumption. When we are outside the regions where the known continued fractions methods from Wiener and Verheul/van Tilborg apply, the attack depends on Assumption 3.4, namely that the elements of $\Gamma_{\text{red}}$ are all of size $\det(L)^{\frac{1}{2}}$. Here, we will comment on how this assumption holds in the examples that we tested.

Let $m$ be the maximal entry of $\Gamma_{\text{red}}$, and $m = a\det(L)^{\frac{1}{2}}$. We want to check that for the matrices involved in the attack, $a$ is close to 1. Therefore, we performed tests for the attacks for small $d$ in the following setup: $N$ is a 2048-bit modulus, $\beta \in [0.25, 0.5]$, $\epsilon \in [0, 0.1]$, and $\delta = \min\{\beta, \frac{1}{2} - \beta + \epsilon\}$.

For this case, the lattices behaved as expected. In 500 experiments, the average value of $a$ was approximately 1.9, and the maximal value of $a$ was approximately 39.

## 4.4 New attacks up to full size exponents

In this section we present partial key exposure attacks for full size public exponent that work up to full size private exponent. Additionally, we present a new partial key exposure attack for full size private exponent that works up to full size public exponent. This means that, as soon as either $e$ or $d$ is chosen to be small, an attacker needs only a part of $d$ to be able to factor $N$ in polynomial time. As opposed to the 2-dimensional attack in the previous section, these attacks use Coppersmith methods with bigger lattices.

Our new results on known MSBs of $d$ for small private exponent $d$ and full size public exponent $e$ are summarized in the following theorem.

**Theorem 4.8 (MSB small $d$)**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$. Let $0 < \delta < \beta < 1$. Furthermore, let $e$, $d$ satisfy $ed \equiv 1 \bmod \phi(N)$ with bitsize($e$) $= n$ and bitsize($d$) $= \beta n$. Given the $(\beta - \delta)n$ MSBs of $d$, $N$ can be factored in time polynomial in $n$ if:*

1. $\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1 + 6\beta} - \epsilon$, *or*
2. $\delta < \frac{3}{16} - \epsilon$ *and* $\beta \leq \frac{11}{16}$, *or*
3. $\delta < \frac{1}{3} + \frac{1}{3}\beta - \frac{1}{3}\sqrt{4\beta^2 + 2\beta - 2} - \epsilon$ *and* $\beta \geq \frac{11}{16}$.

The proof of this theorem can be found in Section 4.4.2.

In the case of known MSBs for full size $d$ and small $e$, we find an improvement of Theorem 4.3 and Theorem 4.6 for $e \in [N^{\frac{1}{2}}, N]$. Our result is stated in the next theorem.

**Theorem 4.9 (MSB small $e$)**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$. Let $0 < \delta < \frac{1}{2} < \alpha < 1$. Let $e$, $d$ satisfy $ed \equiv 1 \bmod \phi(N)$, with bitsize$(d) = n$ and bitsize$(e) = \alpha n$. Given the $(1-\delta)n$ MSBs of $d$, $N$ can be factored in time polynomial in $n$ if:*

$$\delta < \frac{1}{3} + \frac{1}{3}\alpha - \frac{1}{3}\sqrt{4\alpha^2 + 2\alpha - 2} - \epsilon.$$

The proof of this theorem can be found in Section 4.4.3.

In Figure 4.2 and 4.3 we illustrate our results on known MSBs of $d$. In Figure 4.2, the fraction of bits required for an attack is plotted as a function of the size of $d$. It shows the parts of the key space that are insecure by the attacks of Theorem 4.7 (the new 2-dimensional attack from Section 4.3), by the new attack of Theorem 4.8, and by the attacks of Wiener (Theorem 2.1) and Boneh/Durfee (Theorem 3.10). Figure 4.3 is a picture of the relation between the fraction of bits of $d$ required for an attack and the size of $e$, showing the results of Boneh/Durfee/Frankel (Theorem 4.3), Blömer/May (Theorem 4.6), and the new result of Theorem 4.9.

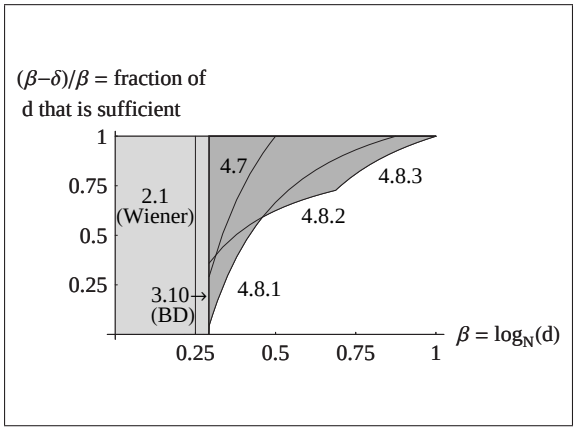


Figure 4.2: MSB attacks for small $d$

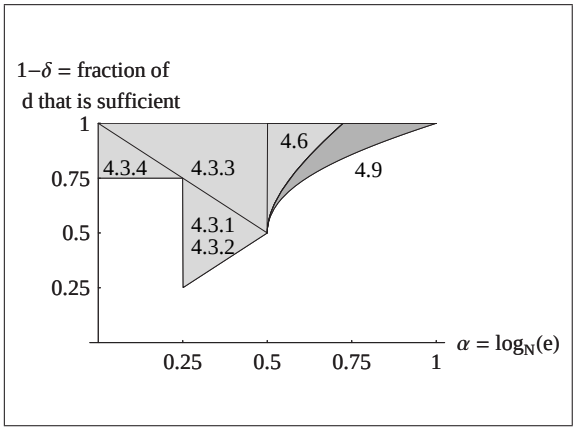


Figure 4.3: MSB attacks for small $e$

Our new result on known LSBs for relatively small $d$ and full size $e$ is as follows.

**Theorem 4.10 (LSB small $d$)**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$. Let $0 < \delta < \beta < 1$. Furthermore, let $e$, $d$ satisfy $ed \equiv 1 \bmod \phi(N)$ with bitsize$(e) = n$ and bitsize$(d) = \beta n$. Given the $(\beta - \delta)n$ LSBs of $d$, $N$ can be factored in time polynomial in $n$ when:*

$$\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1 + 6\beta} - \epsilon.$$

The proof of this theorem can be found in Section 4.4.4.

Figure 4.4 illustrates our result on known LSBs. The fraction of bits required for an attack is plotted as a function of the size of $d$. Figure 4.5 is a picture of the relation between the fraction of bits required for an attack, and the size of $e$, showing the results of Theorem 4.2 and 4.5. Analysis of our LSB method in the case where $e$ is small results in a bound equivalent to the best result of [6], as described in Theorem 4.5.

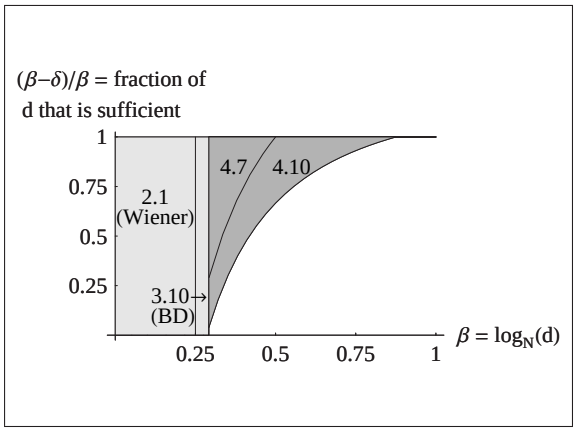


Figure 4.4: LSB attacks for small $d$

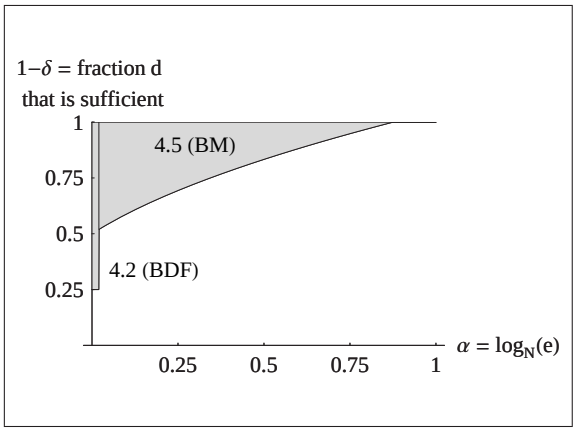


Figure 4.5: LSB attacks for small $e$

The results in this section can be viewed as evidence that side channel attacks are even more dangerous to RSA than we already knew. In essence, we show that there exist partial key exposure attacks up to full size exponents, hence if either $e$ or $d$ is chosen to be significantly smaller than $\phi(N)$, the system is vulnerable to this type of attacks. This can be understood as a warning to crypto-designers to choose both private and public exponent at random, or take sufficient countermeasures to prevent private key bits from leaking. Now, let us discuss the attacks that lead to the results mentioned in the above theorems.

### 4.4.1 Polynomials derived from the RSA key equation

Recall the RSA key equation

$$ed - 1 = k\phi(N), \quad \text{where } \phi(N) = (p-1)(q-1) = N - (p+q-1).$$

In our scenario, we assume that one of the exponents $e$ and $d$ is chosen to be small and the other one is of full size. Hence, either $e < N^\alpha$, $d < \phi(N)$, and $k < \frac{ed}{\phi(N)} < e < N^\alpha$, or $d < N^\beta$, $e < \phi(N)$, and $k < d < N^\beta$.

When MSBs of $d$ are known, we write $d = \tilde{d} + d_0$, where $\tilde{d}$ (representing the most significant bits of $d$) is known to the attacker, but $d_0$ (representing the least significant bits of $d$) is not. To make this precise, let $\delta$ be the parameter such that $|d_0| = |d - \tilde{d}| \leq N^\delta$. For the MSB case, we can thus rewrite the RSA key equation as

$$e(\tilde{d} + d_0) - 1 = k(N - (p+q-1)).$$

Hence, the polynomial

$$f(x, y, z) = ex - Ny + yz + (e\tilde{d} - 1)$$

has a root $(x_0, y_0, z_0) = (d_0, k, p + q - 1)$. Then the root is 'small' since $|x_0| < X$, $|y_0| < Y$, and $|z_0| < Z$ for some upper bounds $X, Y, Z$. In the case of RSA-Small-$e$, we have $X = N^{\delta}$, $Y = N^{\alpha}$, $Z = N^{\frac{1}{2}}$, neglecting the constants. For RSA-Small-$d$, the corresponding values are $X = N^{\delta}$, $Y = N^{\beta}$, and $Z = N^{\frac{1}{2}}$.

The attacker can also compute $\tilde{k} = \frac{e\tilde{d}-1}{N}$ as an approximation to $k$ and set $k_0 = k - \tilde{k}$ as the unknown part of $k$. It can be shown (analogous to the proof of Theorem 4.4) that

$$|k_0| < \frac{e}{\phi(N)}(\tilde{d} \cdot 3N^{-\frac{1}{2}} + d_0) < \begin{cases} 2N^{\alpha+\delta-1} + 6N^{\alpha-\frac{1}{2}} < 8N^{\gamma}, \text{ for } \gamma = \max\{\alpha + \delta - 1, \alpha - \frac{1}{2}\}, \\ N^{\delta} + 3N^{\beta-\frac{1}{2}} < 4N^{\gamma}, \text{ for } \gamma = \max\{\delta, \beta - \frac{1}{2}\}. \end{cases}$$

When we substitute the knowledge of the MSBs of $k$ into the RSA key equation, we obtain

$$e(\tilde{d} + d_0) - 1 = (\tilde{k} + k_0)(N - (p + q - 1)).$$

Hence,

$$f(x, y, z) = ex - Ny + yz + \tilde{k}z + (e\tilde{d} - 1 - \tilde{k}N)$$

has a root $(x_0, y_0, z_0) = (d_0, k_0, p + q - 1)$. With $X = N^{\delta}$, $Y = N^{\gamma}$, and $Z = N^{\frac{1}{2}}$, we have $|x_0| < X$, $|y_0| < Y$, and $|z_0| < Z$. Here, the value of $\gamma$ is different in the cases of RSA-Small-$e$ and RSA-Small-$d$, as we showed above.

When LSBs of $d$ are known, the attacker knows $\bar{d} \equiv d \bmod M$ for some $M$, and we write $d = \bar{d} + d_1 M$, where $\bar{d}$ and $M$ are known and $d_1$ is not. We assume that $d_1 \leq N^{\delta}$. We have no approximation of $k$ in this case, so we rewrite the RSA key equation as

$$e(d_1 M + \bar{d}) - 1 = k(N - (p + q - 1)).$$

Thus,

$$f(x, y, z) = eMx - Ny + yz + (e\bar{d} - 1)$$

has a root $(x_0, y_0, z_0) = (d_1, k, p + q - 1)$. Then $|x_0| < X$, $|y_0| < Y$, and $|z_0| < Z$, for some $X, Y, Z$. For RSA-Small-$e$, we use $X = N^{\delta}$, $Y = N^{\alpha}$, and $Z = N^{\frac{1}{2}}$, and for RSA-Small-$d$ these values are $X = N^{\delta}$, $Y = N^{\beta}$, and $Z = N^{\frac{1}{2}}$. As before, we neglected the constants in the definition of $Y$ and $Z$, and let these constants contribute to some error term $\epsilon$ later.

### 4.4.2 Attacks for known MSBs and small $d$

1. Attack using no partial knowledge of $k$: proof of Theorem 4.8.1

We start by describing a method that finds a small root $(x_0, y_0, z_0)$ of $f(x, y, z) = ex - Ny + yz + (e\tilde{d} - 1)$ over the integers, and prove the first result of Theorem 4.8, namely that we have a polynomial time MSB attack when

$$\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1 + 6\beta} - \epsilon.$$

Since we have not yet given a detailed example of the general strategy for choosing the shifts for polynomials with integer roots, let us do so now. We will show that the bound for which the roots of a polynomial $f(x, y, z) = a_0 + a_1x + a_2y + a_3yz$ can be found in polynomial time is

$$X^{1+3\tau}Y^{2+3\tau}Z^{1+3\tau+3\tau^2} < W^{1+3\tau-\epsilon},$$

for some $\tau \geq 0$ that can be optimized later.

We will use our (extended) strategy of Section 3.3.2, with extra $z$-shifts. First, we fix an integer $m$ depending on $\epsilon$ and a parameter $t$, that we will optimize later in terms of $m$.

We define the set $S$ as the monomials of $f^{m-1}$ plus extra shifts of $z$. The set $M$ is simply the set of all monomials that appear in $x^{i_1}y^{i_2}z^{i_3}f(x, y, z)$ with $x^{i_1}y^{i_2}z^{i_3} \in S$. One can check that

$$x^{i_1}y^{i_2}z^{i_3} \in S \quad \Leftrightarrow i_1 = 0, \ldots, m-1 \ ; \ i_2 = 0, \ldots, m-1-i_1 \ ; \ i_3 = 0, \ldots, i_2 + t,$$

$$x^{i_1}y^{i_2}z^{i_3} \in M \quad \Leftrightarrow i_1 = 0, \ldots, m \ ; \ i_2 = 0, \ldots, m-i_1 \ ; \ i_3 = 0, \ldots, i_2 + t,$$

defines such $S$ and $M$ as described above.

We denote by $l_j$ the largest exponent of $x_j$ that appears in the monomials of $S$, i.e. $l_1 = l_2 = m-1$ and $l_3 = m-1+t$. We have $W = \|f(xX, yY, zZ)\|_\infty$ and $R = (XY)^{m-1}Z^{m-1+t}W$.

In order to work with a polynomial with constant term 1, we define

$$f'(x, y, z) \equiv a_0^{-1}f(x, y, z) \bmod R = 1 + ax + by + cyz.$$

Next, we define the shift polynomials

$$g(x, y, z) \ := \ x^{i_1}y^{i_2}z^{i_3}f'(x, y, z)X^{m-i_1}Y^{m-i_2}Z^{m+t-i_3} \text{ , for } x^{i_1}y^{i_2}z^{i_3} \in S,$$

$$g'(x, y, z) \ := \ x^{i_1}y^{i_2}z^{i_3}R \text{ , for } x^{i_1}y^{i_2}z^{i_3} \in M \backslash S,$$

and let the coefficient vectors of $g(x_1X_1, \ldots x_vX_v)$ and $g'(x_1X_1, \ldots x_vX_v)$ form a lattice basis of a lattice $L$.

By the theory in Section 3.3.2, the following ordering of the monomials of $S$ gives us a triangular matrix describing the lattice. We say that

$$x_1^{i_1} \cdot \ldots \cdot x_v^{i_v} < x_1^{i'_1} \cdot \ldots \cdot x_v^{i'_v} \text{ if } \sum i_j < \sum i'_j.$$

If $\sum i_j = \sum i'_j$, then we use the lexicographical ordering. As a small example, we give the matrix corresponding to $m = 2, t = 1$. We have divided every row in the matrix by $XYZ^2$ to get a cleaner view. The result is in Figure 4.6.

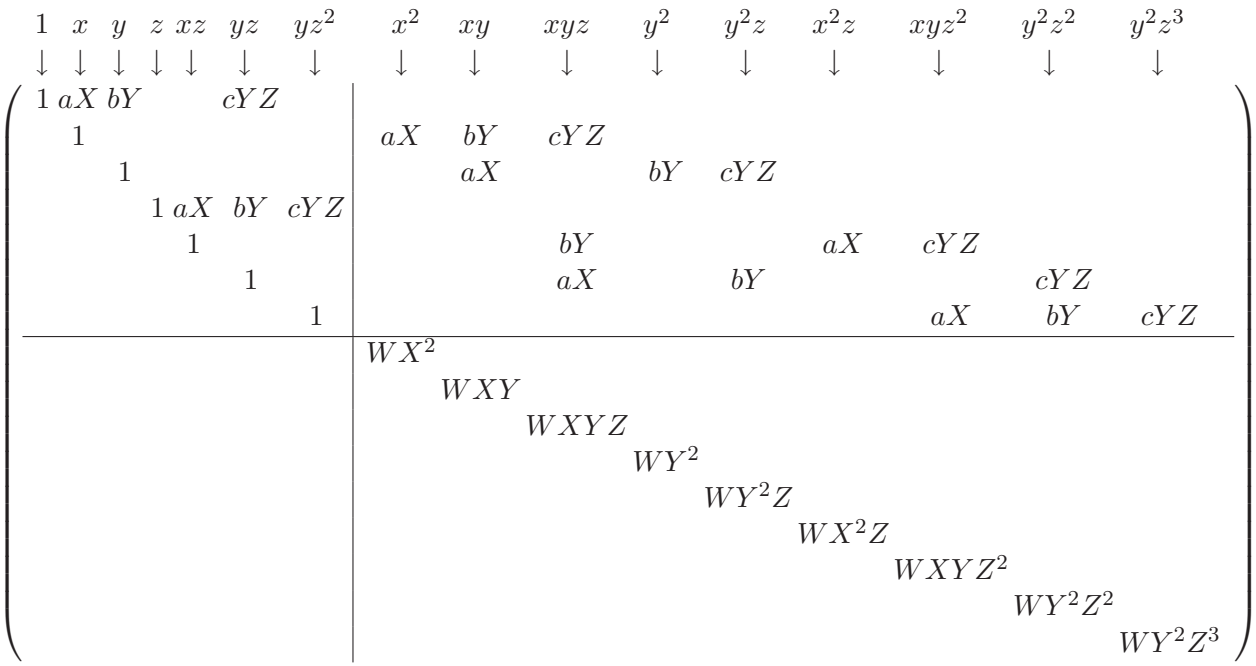


Figure 4.6: Example: Matrix for $m = 2$, $t = 1$

From Section 3.3.2, we know that

$$X^{s_1}Y^{s_2}Z^{s_3} < W^{s_W-\epsilon}, \text{ for } s_j = \sum_{x^{i_1}y^{i_2}z^{i_3}\in M\setminus S} i_j \text{ , and } s_W = |S|.$$

suffices for the polynomials $r_1, r_2$ corresponding to the smallest vectors in the reduced lattice basis of $L$ to satisfy $r_i(x_1^{(0)}, \ldots, x_v^{(0)}) = 0$. Under Assumption 3.7, the resultant computations of $f$, $r_1$, and $r_2$ will reveal the root.

With the given definitions for $S$ and $M$, we find the asymptotic bound

$$X^{\frac{1}{6}m^3(1+3\tau)+o(m^3)}Y^{\frac{1}{6}m^3(2+3\tau)+o(m^3)}Z^{\frac{1}{6}m^3(1+3\tau+3\tau^2)+o(m^3)} < W^{\frac{1}{6}m^3(1+3\tau)+o(m^3)},$$

which reduces to

$$X^{1+3\tau}Y^{2+3\tau}Z^{1+3\tau+3\tau^2} < W^{1+3\tau-\epsilon},$$

for $m \to \infty$, if we let all terms of order $o(m^3)$ contribute to $\epsilon$.

Finally, we come to the result of our partial key exposure attack on RSA-Small-$d$ using $f(x, y, z) = ex - Ny + yz + (e\tilde{d} - 1)$. In our case, $X = N^{\delta}$, $Y = N^{\beta}$, $Z = N^{\frac{1}{2}}$ and $W = \max\{eX, NY, YZ, R\} \geq NY = N^{1+\beta}$. Hence, the asymptotic bound to satisfy is

$$\delta(1+3\tau) + \beta(2+3\tau) + \frac{1}{2}(1+3\tau+3\tau^2) \;\; < \;\; (1+\beta)(1+3\tau),$$

or equivalently

$$3\tau^2 + 3\tau(2\delta - 1) + (2\delta + 4\beta - 1) \quad < \quad 0.$$

We find an optimal value $\tau = \frac{1}{2} - \delta$, which implies $\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1 + 6\beta}$. Thereby, we have derived the result of Theorem 4.8.1.

2. Attack using partial knowledge of $k$: proof of Theorem 4.8.2 and 4.8.3

We will now show how to obtain the second and third result mentioned in Theorem 4.8, namely that we have a polynomial time MSB attack whenever

$$\delta < \frac{3}{16} - \epsilon \text{ and } \beta \leq \frac{11}{16}, \quad \text{or} \quad \delta < \frac{1}{3} + \frac{1}{3}\beta - \frac{1}{3}\sqrt{4\beta^2 + 2\beta - 2} - \epsilon \text{ and } \beta \geq \frac{11}{16}.$$

For the situation where we use information on MSBs of $d$ to get an approximation $\tilde{k}$ of $k$, we want to find a small root $(x_0, y_0, z_0)$ of the polynomial

$$f(x, y, z) = ex - Ny + yz + \tilde{k}z + (e\tilde{d} - 1 - \tilde{k}N).$$

We will use the extended strategy as described in Section 3.3.2 with extra shifts of $y$. For a polynomial $f(x, y, z) = a_0 + a_1x + a_2y + a_3z + a_4yz$, the sets $S$ and $M$ can then be described by

$$\begin{aligned} x^{i_1}y^{i_2}z^{i_3} \in S \quad &\Leftrightarrow i_1 = 0, \ldots, m-1 \ ; \ i_2 = 0, \ldots, m - 1 - i_1 + t \ ; \\ &\qquad i_3 = 0, \ldots, m - 1 - i_1 \\ x^{i_1}y^{i_2}z^{i_3} \in M \quad &\Leftrightarrow i_1 = 0, \ldots, m \ ; \ i_2 = 0, \ldots, m - i_1 + t \ ; \ i_3 = 0, \ldots, m - i_1 \end{aligned}$$

for fixed $m$ (depending on $\epsilon$) and some $t = \tau m$ that will be optimized later.

Substituting this in bound (3.7), we obtain

$$X^{2+3\tau}Y^{3+6\tau+3\tau^2}Z^{3+3\tau} < W^{2+3\tau-\epsilon}.$$

In our case, we have $X = N^{\delta}$, $Y = N^{\gamma}$, with $\gamma = \max\{\delta, \beta - \frac{1}{2}\}$, and $Z = N^{\frac{1}{2}}$. Also, $W = \max\{eX, NY, YZ, \tilde{k}Z, R\} \geq NY = N^{1+\gamma}$. The optimal value $\tau = \frac{\frac{1}{2} - \delta - \gamma}{2\gamma}$ leads to the condition

$$\delta < \frac{1}{3}\gamma + \frac{1}{2} - \frac{1}{3}\sqrt{4\gamma^2 + 6\gamma}.$$

If $\gamma = \delta$, this implies

$$\delta < \frac{3}{16}.$$

Note that this bound is only valid if $\max\{\delta, \beta - \frac{1}{2}\} = \delta$, which implies that $\beta \leq \frac{11}{16}$.

If $\gamma = \beta - \frac{1}{2}$, we get

$$\delta < \frac{1}{3} + \frac{1}{3}\beta - \frac{1}{3}\sqrt{4\beta^2 + 2\beta - 2}\ , \text{ valid for } \beta \geq \frac{11}{16}.$$

This concludes the proof of Theorem 4.8.

### 4.4.3 Attacks for known MSBs and small $e$

Many users of RSA choose a small public exponent $e$. Therefore, we now let $e = N^\alpha$ and $d < \phi(N)$ and investigate the possibilities for new partial key exposure attacks for RSA-Small-$e$. The best result in this situation, as mentioned in Theorem 4.9, is that we obtain a polynomial time MSB attack whenever

$$\delta < \frac{1}{3} + \frac{1}{3}\alpha - \frac{1}{3}\sqrt{4\alpha^2 + 2\alpha - 2} - \epsilon \text{ for } \alpha > \frac{1}{2}.$$

We can again use the polynomial of the attack without partial knowledge of $k$, that is, $f(x, y, z) = ex - Ny + yz + (e\tilde{d} - 1)$, now with $X = N^\delta$, $Y = N^\alpha$ and $Z = N^{\frac{1}{2}}$. We will substitute all this, and $W = N^{1+\alpha}$ in the bound that we obtained for this polynomial, namely

$$X^{1+3\tau}Y^{2+3\tau}Z^{1+3\tau+3\tau^2} < W^{1+3\tau-\epsilon}.$$

As this case is completely similar to the case of small $d$, we find the bound

$$\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1 + 6\alpha} - \epsilon.$$

This result only holds for $\alpha > \frac{1}{2}$. In the case that $\alpha < \frac{1}{2}$, we know from Theorem 4.4 that we can assume that $k$ is known. Hence, the polynomial to be analyzed becomes bivariate.

Since our attack using no partial knowledge of $k$ obtains a bound inferior to the one using the approximation of $k$ that can be derived, it is not mentioned in Theorem 4.9. This brings us to the description of the attack using partial information on $k$.

Attack using partial knowledge of $k$: proof of Theorem 4.9

When we use partial information on $k$, where $k$ is partly unknown (so $\alpha > \frac{1}{2}$), we can use

$$f(x, y, z) = ex - Ny + yz + \tilde{k}z + (e\tilde{d} - 1 - \tilde{k}N).$$

We have $X = N^\delta$, $Y = N^\gamma$, with $\gamma = \max\{\alpha + \delta - 1, \alpha - \frac{1}{2}\}$, and $Z = N^{\frac{1}{2}}$. Using $W = N^{1+\gamma}$, we get the same condition as in the previous paragraph, namely

$$\delta < \frac{1}{3}\gamma + \frac{1}{2} - \frac{1}{3}\sqrt{4\gamma^2 + 6\gamma} - \epsilon.$$

We analyze this for two possibilities of $\gamma$.

If we substitute $\gamma = \alpha + \delta - 1$ (in other words, we assume $\delta > \frac{1}{2}$), we obtain the condition

$$\delta < \frac{3 + 4\alpha - 4\alpha^2}{16\alpha} - \epsilon.$$

However, for $\alpha > \frac{1}{2}$, $\delta < \frac{3+4\alpha-4\alpha^2}{16\alpha} < \frac{1}{2}$, so we get no result.

If $\gamma = \alpha - \frac{1}{2}$, we find

$$\delta < \frac{1}{3} + \frac{1}{3}\alpha - \frac{1}{3}\sqrt{4\alpha^2 + 2\alpha - 2} - \epsilon.$$

This concludes the proof of Theorem 4.9.

### 4.4.4 Attack for known LSBs and small $d$

In this section, we will show how to obtain the result of Theorem 4.10, namely that we have a polynomial time LSB attack whenever

$$\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1+6\beta} - \epsilon.$$

Attack description: proof of Theorem 4.10

Polynomial $f(x,y,z) = eMx - Ny + yz + (e\bar{d} - 1)$ has the same monomials as in the MSB-attack where no partial knowledge of $k$ was used. So, we can directly apply the known analysis for a polynomial with monomials $1, x, y, yz$. We use

$$X^{1+3\tau}Y^{2+3\tau}Z^{1+3\tau+3\tau^2} \leq W^{1+3\tau-\epsilon},$$

on $X = N^{\delta}$, $Y = N^{\beta}$, $Z = 3N^{\frac{1}{2}}$ and $W = \max\{eMX, NY, YZ, R\} \geq NY = N^{1+\beta}$. This implies

$$\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1+6\beta}.$$

This concludes the proof of Theorem 4.10.

If we adapt the LSB attack for the situation when $e$ is not of full size, we get exactly the result from Blömer and May in Theorem 4.5.

### 4.4.5 Experiments for the new attacks

We state some experimental results to give an idea of the performance of our methods. In the following three examples, $N \approx 2^{1024}$. The experiments are performed on a server containing two Pentium III processors of 1000 Mhz, and all the lattice basis reductions are done using Shoup's NTL [66].

For our attack on small $d$ without using partial knowledge of $k$, a typical case is $\beta = 0.3$ and $\delta = 0.21$ (e.g. 70% of $d$ is unknown). An attack using $m = 3$, $t = 1$ involved a 10 minute reduction of the 30-dimensional lattice.

We performed an attack on small $d$ using partial knowledge of $k$ for $\beta = 0.6$, $\delta = 0.13$ (e.g. 22% of $d$ is unknown), with $m = 3$, $t = 2$. The 50-dimensional lattice took $3\frac{1}{4}$ hours to reduce.

We performed the attack on small $e$ using partial knowledge of $k$ for $\alpha = 0.7$, $\delta = 0.08$ (e.g. 8% of $d$ is unknown), using $m = 3$, $t = 2$. The reduction of the 50-dimensional lattice took $2\frac{3}{4}$ hours.

More experimental results are included in the following tables, but now for a modulus length of 256 bits. As the bounds on $\delta$ stated in the theorems are asymptotic bounds, the goal of the tables is to provide some insight of what values of $\delta$ our attacks can realize in practice. Note that the LLL reduction time for a larger modulus length is longer, so the attacks will take longer when performed on $N \approx 2^{1024}$ or $N \approx 2^{2048}$.

| $\beta$ | $\delta$ | $m = 2$ | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|
| | asympt. | $t = 0$ | $t = 1$ | $t = 2$ | $t = 0$ | $t = 1$ | $t = 2$ |
| 0.30 | 0.28 | 0.19 | 0.19 | 0.19 | 0.19 | 0.21 | 0.21 |
| 0.35 | 0.25 | 0.13 | 0.14 | 0.14 | 0.14 | 0.16 | 0.16 |
| 0.40 | 0.22 | 0.09 | 0.11 | 0.11 | 0.09 | 0.14 | 0.15 |
| 0.45 | 0.19 | 0.04 | 0.10 | 0.10 | 0.05 | 0.12 | 0.12 |
| 0.50 | 0.17 | 0 | 0.08 | 0.09 | 0 | 0.10 | 0.11 |
| 0.55 | 0.14 | 0 | 0.08 | 0.08 | 0 | 0.09 | 0.11 |
| 0.60 | 0.12 | 0 | 0.04 | 0.04 | 0 | 0.06 | 0.10 |
| 0.65 | 0.10 | 0 | 0 | 0 | 0 | 0 | 0.06 |
| 0.70 | 0.07 | 0 | 0 | 0 | 0 | 0 | 0.01 |
| 0.75 | 0.05 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.80 | 0.03 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.85 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 |
| Dimension: | | 10 | 16 | 22 | 20 | 30 | 40 |
| LLL (sec): | | 1 | 2 | 8 | 3 | 25 | 100 |

Table 4.2: Experiments for small $d$, not using partial knowledge of $k$

| $\beta$ | $\delta$ | $m = 2$ | | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|---|
| | asympt. | $t = 0$ | $t = 1$ | $t = 2$ | $t = 3$ | $t = 0$ | $t = 1$ | $t = 2$ |
| 0.30 | 0.19 | 0.19 | 0.20 | 0.20 | 0.20 | 0.19 | 0.19 | 0.19 |
| 0.35 | 0.19 | 0.15 | 0.16 | 0.16 | 0.16 | 0.16 | 0.16 | 0.16 |
| 0.40 | 0.19 | 0.12 | 0.12 | 0.12 | 0.12 | 0.14 | 0.15 | 0.15 |
| 0.45 | 0.19 | 0.10 | 0.11 | 0.12 | 0.12 | 0.12 | 0.13 | 0.13 |
| 0.50 | 0.19 | 0.08 | 0.11 | 0.12 | 0.12 | 0.12 | 0.13 | 0.13 |
| 0.55 | 0.19 | 0.08 | 0.11 | 0.12 | 0.12 | 0.11 | 0.13 | 0.13 |
| 0.60 | 0.19 | 0.05 | 0.11 | 0.11 | 0.11 | 0.11 | 0.12 | 0.13 |
| 0.65 | 0.19 | 0 | 0.05 | 0.06 | 0.06 | 0.05 | 0.08 | 0.10 |
| 0.70 | 0.18 | 0 | 0 | 0 | 0 | 0 | 0.04 | 0.05 |
| 0.75 | 0.14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.80 | 0.11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.85 | 0.08 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.90 | 0.05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.95 | 0.03 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Dimension: | | 14 | 20 | 26 | 32 | 30 | 40 | 50 |
| LLL (sec): | | 1 | 7 | 17 | 40 | 26 | 180 | 480 |

Table 4.3: Experiments for small $d$, using partial knowledge of $k$

When we are in the area of the key space where the 2-dimensional attack applies, this method is clearly faster, and should be preferred over the attacks using larger lattices.

| $\alpha$ | $\delta$ | $m=2$ | | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|---|
| | asymptotic | $t=0$ | $t=1$ | $t=2$ | $t=3$ | $t=0$ | $t=1$ | $t=2$ |
| 0.50 | 0.50 | 0.25 | 0.33 | 0.38 | 0.40 | 0.32 | 0.37 | 0.41 |
| 0.55 | 0.33 | 0.17 | 0.21 | 0.23 | 0.25 | 0.21 | 0.23 | 0.24 |
| 0.60 | 0.27 | 0.09 | 0.14 | 0.17 | 0.18 | 0.13 | 0.16 | 0.19 |
| 0.65 | 0.22 | 0.02 | 0.07 | 0.10 | 0.10 | 0.07 | 0.11 | 0.13 |
| 0.70 | 0.18 | 0 | 0.02 | 0.03 | 0.04 | 0.02 | 0.04 | 0.08 |
| 0.75 | 0.14 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0.02 |
| 0.80 | 0.11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.85 | 0.08 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.90 | 0.05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.95 | 0.03 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Dimension: | 14 | 20 | 26 | 32 | 30 | 40 | 50 |
| | LLL (sec): | 1 | 5 | 13 | 40 | 33 | 180 | 520 |

Table 4.4: Experiments for small $e$, using partial knowledge of $k$

Concerning the choice of $t$, recall that $t = \tau m$, and that we use $\tau = \frac{1}{2} - \delta$ to obtain the asymptotic result of our RSA-Small-$d$ attack using no partial knowledge of $k$. This explains that for $m = 2$ in Table 4.2, a value of $t$ larger than 1 gives no significant improvement, but for $m = 3$, $t = 2$ may give a better result when the bound on $\delta$ is 'low'. For the attacks using partial knowledge of $k$, the optimal value of $\tau$ is $\tau = \frac{\frac{1}{2}-\delta-\gamma}{2\gamma}$. This explains for example, that when $e = N^\alpha$ with $\alpha$ close to $\frac{1}{2}$, a larger value of $t$ gives a better bound on $\delta$ in the experiments (as can be seen in Table 4.4).

After these experiments, we are now ready to comment on Assumption 3.7. Let $r_1(x, y, z)$ and $r_2(x, y, z)$ be polynomials that correspond to the smallest LLL reduced vectors in our method. If we assume that Howgrave-Graham's bound is satisfied for both, then $r_1(x_0, y_0, z_0) = r_2(x_0, y_0, z_0) = 0$. Now Assumption 3.7 does not hold when the resultant computations with $r_1$ and $r_2$ yield the zero polynomial. Therefore, we performed some tests to see how often this occurs. We found that for small $\delta$, approximately 0.1% of pairs $(r_1, r_2)$ the heuristic failed. However, Bauer and Joux [2] recently showed that for larger $\delta$, the heuristic fails more often. For example, let us look at the attack on small $d$ without using knowledge of $k$ (Table 4.2). In the attack for $m = 2$, $t = 1$, $\beta = 0.35$ and $\delta = 0.10$, the heuristic fails in about 20% of the cases. This does not mean that the whole attack will fail in these cases. The method finds about 10 polynomials $r_i$ that are small enough to have the root over the integers, and if $r_1$ and $r_2$ are algebraically dependent, we just try other combinations until the resultant method succeeds. In Section 5.3, we show a relaxation of Assumption 3.7 with the same idea. For the attack to succeed, we only need two independent $r_i$'s that satisfy $r_i(x_0, y_0, z_0) = 0$, they do not have to be $r_1$ and $r_2$. For a *provable* way to find a third independent polynomial $r_2$ from $f$ and $r_1$ in the case of this attack, we refer to the paper of Bauer and Joux [2].

Experiments also show that the theoretical bound under which our methods works,

$$\det(L) \leq (2^{\frac{-\omega}{4}} \frac{1}{\sqrt{\omega}})^{\omega-1} R^{\omega-1},$$

is far too strict. It would imply that for $m \in \{2, 3\}$, the method will never work, which clearly contradicts the practice. This is both due to the term $(2^{\frac{-\omega}{4}} \frac{1}{\sqrt{\omega}})^{\omega-1}$, when it is known that LLL reduction achieves much better bounds in practice, and to the fact that we use the LLL bound for the second smallest reduced vector. In practice, we experienced that our method works until $\det(L)$ comes close to $R^{\omega}$ (the upper bound for the first reduced vector to be small enough, omitting the terms that do not depend on $N$).

Apparently, the lattice $L$ satisfies Assumption 3.3 in this case. Because the lattice is balanced, all reduced vectors (especially the smallest vectors) have a norm of approximately $\det(L)^{\frac{1}{\omega}}$ and the bound $\det(L)^{\frac{1}{\omega}} < R$ describes the cases for which the attack works in practice.

## 4.5 Tabular overview

The following tables include all known and new partial key exposure attacks on RSA-Small-$e$ and RSA-Small-$d$ respectively.

| **Type of attack** | **Attack bound** | **Reference** |
|---|---|---|
| known MSBs | $\delta < 1 - \alpha$ and $\alpha \in (\frac{1}{4}, \frac{1}{2})$ | Thm. 4.3.1/4.3.2 |
| | $\delta < \alpha$ and $\alpha \leq \frac{1}{2}$ | Thm. 4.3.3 |
| | $\delta < \frac{1}{4}$ and $\alpha \leq \frac{1}{2}$ | Thm. 4.3.4 |
| | $\delta < \frac{1}{8}(5 - 2\alpha - \sqrt{36\alpha^2 + 12\alpha - 15}) - \epsilon$ and $\alpha \in (\frac{1}{2}, \frac{\sqrt{6}-1}{2})$ | Thm. 4.6 |
| | $\delta < \frac{1}{3} + \frac{1}{3}\alpha - \frac{1}{3}\sqrt{4\alpha^2 + 2\alpha - 2} - \epsilon$ and $\alpha \in (\frac{1}{2}, 1)$ | Thm. 4.9 |
| known LSBs | $\delta < \frac{3}{4}$ and $e = \text{poly}(n)$ | Thm. 4.2 |
| | $\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1 + 6\alpha} - \epsilon$ and $\alpha \in (0, \frac{7}{8})$ | Thm. 4.5 |

Table 4.5: Partial key exposure attacks for RSA-Small-$e$

| **Type of attack** | **Attack bound** | **Reference** |
|---|---|---|
| no bits necessary | $\beta < 0.25$ | Wiener<br>Thm. 2.1 |
| | $\beta < 0.292 - \epsilon$ | Boneh/Durfee<br>Thm. 3.10 |
| known MSBs/LSBs | $\delta < \frac{1}{2} - \beta$ | Thm. 4.7 |
| known MSBs | $\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1+6\beta} - \epsilon$ | Thm. 4.8.1 |
| | $\delta < \frac{3}{16} - \epsilon$ and $\beta \leq \frac{11}{16}$ | Thm. 4.8.2 |
| | $\delta < \frac{1}{3} + \frac{1}{3}\beta - \frac{1}{3}\sqrt{4\beta^2 + 2\beta - 2} - \epsilon$ and $\beta > \frac{11}{16}$ | Thm. 4.8.3 |
| known LSBs | $\delta < \frac{5}{6} - \frac{1}{3}\sqrt{1+6\beta} - \epsilon$ | Thm. 4.10 |

Table 4.6: Partial key exposure attacks for RSA-Small-$d$

# 5 Attacks on RSA-CRT variants

In this chapter we discuss the known attacks on RSA-CRT variants. Moreover, we discuss a new attack on CRT-Small-$d_p, d_q$ (that also affects the security of CRT-BalancedExponents), and a new attack on CRT-Qiao&Lam.

Section 5.3 is based on [37] and Section 5.4 is based on [38], both of which are joint papers with Alexander May.

## 5.1 Introduction

As we have mentioned before, in many implementation proposals of the RSA cryptosystem, either the public exponent $e$ or the private exponent $d$ is chosen to be small. Since the attacks of Wiener [75] and Boneh and Durfee [10] on RSA-Small-$d$, we know that choosing a small $d$ can be dangerous. As an alternative approach, Wiener proposed to use the Chinese Remainder Theorem (CRT) for decryption/signing, and 'small private CRT-exponents' instead of a small private exponent. The new decryption/signing process of RSA, as proposed by Quisquater and Couvreur in [61] then works in the following way:

- 'split' the private exponent $d$ into $d_p \equiv d \mod (p-1)$ and $d_q \equiv d \mod (q-1)$,
- decrypt/sign by first computing

  $$m_p \equiv c^{d_p} \mod p \quad \text{and} \quad m_q \equiv c^{d_q} \mod q,$$

  and then using CRT to compute the unique value $m$ modulo $N$ such that

  $$m \equiv m_p \mod p \quad \text{and} \quad m \equiv m_q \mod q.$$

RSA using CRT is specifically useful in time-critical applications, for instance for signing procedures on smartcards. As for the security of RSA variants using CRT, there exists a meet-in-the-middle attack enabling the adversary to factor $N$ in time and space $\tilde{O}(\min\{\sqrt{d_p}, \sqrt{d_q}\})$, which is exponential in the bitsize of the minimum of $d_p$ and $d_q$, see [8, 51]. We use the "soft-$O$" notation here, which ignores the logarithmic factors.

Let us briefly recall some RSA-CRT variants, for which we shall discuss the known attacks in Section 5.2, and new attacks on some of these variants in Section 5.3 and 5.4.

Standard RSA-CRT ("CRT-Standard"):

In all RSA-CRT variants, the RSA-CRT parameters $e$, $d_p$, $d_q$, $p$, $q$ satisfy

$$ed_p \equiv 1 \mod (p-1) \quad \text{and} \quad ed_q \equiv 1 \mod (q-1).$$

If all parameters are chosen generically, then we have

$$ed_p = 1 + k_p(p-1) \quad \text{and} \quad ed_q = 1 + k_q(q-1),$$

for $e \approx N, p \approx q \approx d_p \approx d_q \approx N^{\frac{1}{2}}, k_p \approx k_q \approx N$.

RSA-CRT with small public exponent $e$ ("CRT-Small-$e$") :

As in standard RSA, it is possible to choose a small $e$, after which the Extended Euclidean Algorithm finds the corresponding $d$, which in turn is then split up in $d_p$ and $d_q$. If one chooses $p$ and $q$ first at random, and then a small $e$, then the private CRT-exponents $d_p$ and $d_q$ will be about as long as $p$ and $q$ in general. So, we have

$$e \approx N^{\alpha}, p \approx q \approx d_p \approx d_q \approx N^{\frac{1}{2}}, k_p \approx k_q \approx N^{\alpha}, \text{ for } \alpha < 1.$$

RSA-CRT with small private CRT-exponents $d_p, d_q$ ("CRT-Small-$d_p, d_q$") :

This variant was proposed by Wiener in [75] as a possible alternative to using a small decryption exponent $d$. In this case, one assumes that the decryption/signing time is critical. First one picks $p$ and $q$ as random primes of bitsize $\frac{1}{2}n$, and $d_p$ and $d_q$ as random integers of a certain small bitsize such that they are coprime to $p-1$ and $q-1$. Then one can compute $d$ smaller than $\phi(N)$ such that $d \equiv d_p \bmod (p-1)$ and $d \equiv d_q \bmod (q-1)$. From this $d$, one can compute the corresponding $e$ as usual.

Hence, the setting is: $e$ is full size, $p$ and $q$ are balanced, and $d_p$ and $d_q$ are balanced but significantly smaller than $p$ and $q$. So, we have

$$e \approx N,\ p \approx q \approx N^{\frac{1}{2}}, d_p \approx d_q \approx N^{\beta}, k_p \approx k_q \approx N^{\beta+\frac{1}{2}}, \text{ for } \beta < \frac{1}{2}.$$

It has been an open question since Wiener's work whether or not there exist polynomial time attacks on RSA-CRT with small CRT-exponents. In Section 5.3, we give an affirmative answer to this question, when we describe a new attack that works for $\beta < 0.0734$.

RSA-CRT with unbalanced primes ("CRT-UnbalancedPrimes") :

This variant was studied by May [52], who found the first polynomial time attacks on an RSA-CRT case. In this scenario, the idea is to minimize the cost of the decryption mod $q$ by taking a small $q$, and to minimize the cost of the decryption mod $p$ by taking a small $d_p$. Hence, the setting is: $e$ is full size, $p$ and $q$ are unbalanced (where $q$ is the smaller prime), and $d_p$ is significantly smaller than $p$. So,

$$e \approx N,\ p \approx N^{1-\gamma},\ q \approx N^{\gamma},\ d_p \approx N^{\beta}, d_q < N^{\gamma}, k_p \approx N^{\beta+\gamma}, k_q \approx N,$$

for $\gamma < \frac{1}{2}$ and $\beta < 1 - \gamma$.

RSA-CRT with small $e$ *and* small $d_p$ and $d_q$ ("CRT-BalancedExponents") :

This variant, proposed by Galbraith/Heneghan/McKee [28] and Sun/Wu [70], balances the cost of decryption and encryption, by enabling an implementor of the key generation to choose $e$ *and* $d_p$ and $d_q$ smaller than standard. In Section 2.2, we discussed a possible key generation method for this variant.

The general setting is: $e$ is relatively small, $p$ and $q$ are balanced, and $d_p$ and $d_q$ are balanced but smaller than $p$ and $q$. So,

$$e \approx N^{\alpha},\ p \approx q \approx N^{\frac{1}{2}}, d_p \approx d_q \approx N^{\beta}, k_p \approx k_q \approx N^{\alpha+\beta-\frac{1}{2}}, \text{ for } \alpha < 1 \text{ and } \beta < \frac{1}{2}.$$

In the papers [28] and [70], the authors describe various attacks, and propose parameters for which the mentioned attacks do not work. A new attack by Bleichenbacher and May [4] made the authors of the proposals change their parameter choices [27, 69]. Our new attack, described in Section 5.3, leads to another part of the key space that should be considered unsafe.

RSA-CRT with small difference $d_p - d_q$ ("CRT-Qiao&Lam") :

Qiao and Lam [60] proposed to use CRT-Small-$d_p, d_q$ *and* to use $d_q = d_p - 2$. In this way, one profits from the fast decryption method of CRT-Small-$d_p, d_q$, while one only has to store one of the two private CRT-exponents. In Section 5.4 we describe a new attack on this variant.

If we now summarize the five possible ways to speed up either encryption/verifying or decryption/signing (or both), the general setting is as follows.

$$\begin{cases} ed_p = 1 + k_p(p-1), \\ ed_q = 1 + k_q(q-1), \end{cases} \quad \text{with} \quad \boxed{\begin{array}{lll} e & \approx N^{\alpha} & \text{for } 0 < \alpha \leq 1 \\ p & \approx N^{1-\gamma} & \text{for } 0 < \gamma \leq \frac{1}{2} \\ q & \approx N^{\gamma} & \\ d_p & \approx N^{\beta} & \text{for } 0 < \beta \leq 1-\gamma \\ d_q & \approx N^{\kappa} & \text{for } \kappa = \begin{cases} \beta, \text{ if } \gamma = \frac{1}{2} \\ \gamma, \text{ if } \gamma < \frac{1}{2} \end{cases} \\ k_p & \approx N^{\alpha+\beta+\gamma-1} & \\ k_q & \approx N^{\alpha+\kappa-\gamma} & \end{array}}$$

The default values for the parameters are $\alpha = 1$, $\gamma = \beta = \kappa = \frac{1}{2}$, and for special RSA-CRT settings described above, the parameters are as below.

| | |
|---|---|
| CRT-Small-$e$: | $\alpha < 1$, $\gamma = \beta = \kappa = \frac{1}{2}$. |
| CRT-Small-$d_p, d_q$: | $\alpha = 1$, $\gamma = \frac{1}{2}$, $\beta = \kappa < \frac{1}{2}$. |
| CRT-UnbalancedPrimes: | $\alpha = 1$, $\gamma = \kappa < \frac{1}{2}$, $\beta < 1 - \gamma$. |
| CRT-BalancedExponents: | $\alpha < 1$, $\gamma = \frac{1}{2}$, $\beta = \kappa < \frac{1}{2}$. |
| CRT-Qiao&Lam: | see CRT-Small-$d_p, d_q$. |

In the next section we will look at the known attacks on the RSA-CRT variants. We will give short descriptions of each attack in the most general setting possible, and explore for which values of the parameters (and therefore for which variants) the attacks work.

## 5.2 Known attacks

In this section, we briefly discuss the attacks proposed in the literature on the subject. We give the theorems in the general setting in order to see if an attack that is designed for one CRT variant may be applicable to another CRT variant.

It is important to note that $\alpha + \beta + \gamma - 1 \geq 0$ and $\alpha + \kappa - \gamma \geq 0$, since $k_p$ and $k_q$ are positive integers. In the CRT-BalancedPrimes variant, it may happen that $\alpha + \beta - \frac{1}{2} \approx 0$, which means that $k_p$ and $k_q$ are very small, and must be assumed to be public knowledge. Since it makes no sense to treat $k_p$ and $k_q$ as variables in an equation if they are known, we make a distinction between attacks with unknown $k_p$, $k_q$, and attacks with known $k_p, k_q$.

First, we start with the attacks on RSA-CRT variants where $k_p$ and $k_q$ are not known. To introduce the notation for all of these attacks, we use the following condition.

**Condition (*) (Notation for the known attacks on RSA-CRT variants)**
*Let $N = pq$ be an $n$-bit RSA modulus. Let $0 < \alpha \leq 1$, $0 < \gamma \leq \frac{1}{2}$, $0 < \beta \leq \frac{1}{2}$, and $0 < \kappa \leq \frac{1}{2}$ such that $\kappa = \gamma$ if $\gamma < \frac{1}{2}$ and $\kappa = \beta$ if $\gamma = \frac{1}{2}$. Let $p$, $q$ be primes of bitsize $(1-\gamma)n$ and $\gamma n$. Furthermore, let $e$, $d_p$, $d_q$ satisfy $ed_p \equiv 1 \mod (p-1)$ and $ed_q \equiv 1 \mod (q-1)$ with bitsize($e$) $= \alpha n$, bitsize($d_p$) $= \beta n$, and bitsize($d_q$) $= \kappa n$.*

The following attack was originally meant for the CRT-UnbalancedPrimes setting.

**Theorem 5.1 (May, [52])**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Consider an RSA-CRT instance satisfying (*). Then $N$ can be factored in time polynomial in $n$ when:*

$$\beta < \frac{1}{2}(2 - 3\gamma + \gamma^2 - \alpha) - \epsilon.$$

**Proof.**
For the attack, May notices that $ed_p + (k_p - 1) \equiv 0 \mod p$, hence

$$f_p(x, y) = ex - y \text{ has a small root } (d_p, k_p - 1) \text{ modulo } p.$$

In this thesis, we have not discussed Coppersmith methods for small roots modulo an *unknown* modulus $p$, of which one knows a multiple $N$. However, these methods exist (see [13, 54]), and May shows that using the shifts

$$g_{jk}(x, y) = x^j N^{\max\{0,\tau m-k\}} f_p^k(x, y), \text{ for } k = 0, \ldots, m-1, j = m-k-1,$$

one finds the bound

$$(XY)^{\frac{1}{2}} < p^\tau N^{-\frac{1}{2}\tau^2-\epsilon}.$$

For $p = N^{1-\gamma}$, $X = N^\beta$, $Y = N^{\alpha+\beta+\gamma-1}$ the optimal value of $\tau$ is $\tau = 1-\gamma$. It follows that

$$\beta < \frac{1}{2}(2 - 3\gamma + \gamma^2 - \alpha) - \epsilon.$$

□

For CRT-UnbalancedPrimes, the bound of Theorem 5.1 reduces to May's result [52]

$$\beta < \frac{1}{2}(1 - 3\gamma + \gamma^2) - \epsilon.$$

For CRT-BalancedExponents, it follows that

$$\beta < \frac{3}{8} - \frac{1}{2}\alpha - \epsilon$$

is unsafe. CRT-Small-$e$ and CRT-Small-$d_p, d_q$ are not affected.

The second attack by May, a result described in the following theorem, was also originally meant for the setting of CRT-UnbalancedPrimes.

**Theorem 5.2 (May, [52])**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Consider an RSA-CRT instance satisfying (*). Then $N$ can be factored in time polynomial in $n$ when:*

$$\beta < 1 - \frac{2}{3}\gamma - \frac{2}{3}\sqrt{\gamma^2 + 3\gamma\alpha} - \epsilon.$$

**Proof.**
To obtain this result, May looks at the equation $ed_p = 1 + k_p(p-1)$ modulo $e$. Hence,

$$k_p(p-1) + 1 \equiv 0 \mod e.$$

If this is multiplied by $q$, then the result is

$$\begin{aligned} k_p(N-q)+q &\equiv 0 \mod e, \text{ or equivalently} \\ (k_p-1)N-(k_p-1)q+N &\equiv 0 \mod e. \end{aligned}$$

Thus, $f_e(x,y) = x(N-y)+N$ has a small root $(x_0,y_0) = (k_p-1, q)$ modulo $e$. This is a polynomial à la Boneh/Durfee (see Section 3.3.1), and the known bound is

$$X^{2+3\tau}Y^{1+3\tau+3\tau^2} < e^{1+3\tau-\epsilon}.$$

For $X = N^{\alpha+\beta+\gamma-1}$, $Y = N^{\gamma}$, the optimal $\tau = \frac{1-2\gamma-\beta}{2\gamma}$ leads to the bound

$$\beta < 1 - \frac{2}{3}\gamma - \frac{2}{3}\sqrt{\gamma^2+3\gamma\alpha} - \epsilon.$$

□

For CRT-UnbalancedPrimes, the bound of Theorem 5.2 reduces to

$$\beta < 1 - \frac{2}{3}\gamma - \frac{2}{3}\sqrt{\gamma^2+3\gamma} - \epsilon.$$

For the setting of CRT-BalancedExponents, it leads to the bound

$$\beta < \frac{2}{3} - \frac{1}{3}\sqrt{1+6\alpha} - \epsilon.$$

This last bound implies that $\alpha+\beta < \frac{1}{2}$, so therefore it has no consequences for the security of CRT-BalancedExponents.

The last attack that was proposed for the CRT-UnbalancedPrimes case is by Bleichenbacher and May.

**Theorem 5.3 (Bleichenbacher/May, [4])**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Consider an RSA-CRT instance satisfying (*). Then $N$ can be factored in time polynomial in $n$ when:*

$$\beta < \frac{1}{3}(3 - 2\gamma - \gamma^2 - \sqrt{12\alpha\gamma - 12\alpha\gamma^2 + 4\gamma^2 - 5\gamma^3 + \gamma^4}) - \epsilon.$$

**Proof sketch.**
Bleichenbacher and May improved the attack of Theorem 5.2 in their paper [4]. They observe that the shifts that are used for polynomial $f_e(x,y) = x(N-y)+N$, namely

$$g_{ijk}(x,y) = x^i y^j f^k(x,y) e^{m-k},$$

contains many powers of $y$. The $y_0$ that one is looking for, namely $y_0 = q$, satisfies $y_0 \cdot p = N$. Therefore, Bleichenbacher and May introduce a new variable $z$ for $p$, and use the shifts

$$g_{ijk}(x,y,z) = x^i y^j z^s f^k(x,y) e^{m-k},$$

for a value of $s$ that has to be optimized. Moreover, since $y_0 \cdot z_0 = N$, they replace every occurrence of $yz$ by $N$. For a detailed analysis of the determinant of the lattice that follows, and a proof of this theorem, we refer to [4].

□

For CRT-UnbalancedPrimes, the above result reduces to

$$\beta < \frac{1}{3}(3 - 2\gamma - \gamma^2 - \sqrt{12\gamma - 8\gamma^2 - 5\gamma^3 + \gamma^4}) - \epsilon.$$

For CRT-BalancedExponents, it leads to an attack bound

$$\beta < \frac{7}{12} - \frac{1}{12}\sqrt{7 + 48\alpha} - \epsilon.$$

Let us now move on to the known attacks that were designed for CRT-BalancedExponents. The following attack features in both proposals of the CRT-BalancedExponents case.

**Theorem 5.4 (GHM and SW, [28, 70])**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Consider an RSA-CRT instance satisfying (*). Then $N$ can be factored in time polynomial in $n$ when:*

$$\beta < 1 + \gamma - 2\kappa - 2\alpha - \epsilon.$$

**Proof.**
For this attack, the equations

$$k_p p \equiv k_p - 1 \mod e \quad \text{and} \quad k_q q \equiv k_q - 1 \mod e$$

are multiplied with each other. This results in

$$k_p k_q (N - 1) + k_p + k_q - 1 \equiv 0 \mod e,$$

which means that the polynomial $f_e(u, v) = u(N-1)+v$ has a small root $(k_p k_q, k_p + k_q - 1)$ modulo $e$. This is a simple case of a polynomial where the total degree is 1 (a specific case of the "generalized lower triangle", treated in Section 3.3.1). The bound that can be derived from the discussion of generalized lower triangles in Section 3.3.1 is

$$UV < e^{1-\epsilon}.$$

Substituting $U = N^{2\alpha+\beta+\kappa-1}$ and $V = N^{\alpha+\kappa-\gamma}$, one obtains

$$\beta < 1 + \gamma - 2\kappa - 2\alpha - \epsilon.$$

□

For $\{\gamma = \frac{1}{2},\ \kappa = \beta\}$, it follows that the above attack works for

$$\beta < \frac{1}{2} - \frac{2}{3}\alpha - \epsilon$$

which is only applicable in the CRT-BalancedExponents setting. For $\{\gamma < \frac{1}{2},\ \kappa = \gamma\}$, it gives

$$\beta < 1 - 2\alpha - \gamma - \epsilon,$$

which doesn't affect CRT-UnbalancedPrimes.

The attack by Bleichenbacher and May originally meant for CRT-BalancedExponents is described in the following theorem.

**Theorem 5.5 (Bleichenbacher/May, [4])**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Consider an RSA-CRT instance satisfying (*). Then $N$ can be factored in time polynomial in $n$ when:*

$$\beta < \frac{1}{2}(1 + 2\gamma - 2\alpha - 3\kappa) - \epsilon.$$

**Proof.**
In the latest result on CRT-BalancedExponents, Bleichenbacher and May follow the strategy of Theorem 5.4, except that they do not look at the equations modulo $e$. That is, they multiply the equations

$$ed_p + k_p - 1 = k_p p \quad \text{and} \quad ed_q + k_q - 1 = k_q q$$

with each other, which results in the equation

$$e^2 d_p d_q + e d_p(k_q - 1) + e d_q(k_p - 1) - (N-1)k_p k_q - (k_p + k_q - 1) = 0.$$

Then, $(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)}) = (d_p d_q, d_p(k_q - 1) + d_q(k_p - 1), k_p k_q, k_p + k_q - 1)$ is a root of

$$f(x_1, x_2, x_3, x_4) = e^2 x_1 + e x_2 - (N-1)x_3 - x_4 = 0.$$

This is simply a linear polynomial, whose known bound is

$$X_1 X_2 X_3 X_4 < W^{1-\epsilon},$$

for $W = N^{2\alpha+\beta+\kappa}, X_1 = N^{\beta+\kappa}, X_2 = N^{\alpha+\beta+\kappa-\gamma}, X_3 = N^{2\alpha+\beta+\kappa-1}, X_4 = N^{\alpha+\kappa-\gamma}$. This immediately leads to the bound given in this theorem.

□

For $\{\gamma = \frac{1}{2},\ \kappa = \beta\}$, it follows that the attack works for

$$\beta < \frac{2}{5} - \frac{2}{5}\alpha - \epsilon$$

which is only applicable in the CRT-BalancedExponents setting. For $\{\gamma < \frac{1}{2},\ \kappa = \gamma\}$, it gives

$$2\alpha + 2\beta + \gamma < 1,$$

which doesn't affect the CRT-UnbalancedPrimes case.

Having discussed many attacks for the RSA-CRT cases with unknown $k_p$, $k_q$, we now summarize the best known result for the case that $k_p$ and $k_q$ are known.

**Theorem 5.6 (GHM, [28])**
*Let $N = pq$ be an $n$-bit RSA modulus, where $p$ and $q$ are primes of equal bitsize. Let $0 < \alpha < 1$ and $0 < \beta \leq \frac{1}{2}$. Furthermore, let $e$, $d_p$, $d_q$ satisfy $ed_p = 1 + k_p(p-1)$ and $ed_q = 1 + k_q \mod (q-1)$ with bitsize$(e) = \alpha n$, and bitsize$(d_p)$ = bitsize$(d_q) = \beta n$. Suppose that $k_p$ and $k_q$ are very small, that is $\alpha + \beta - \frac{1}{2} \approx 0$. Then $N$ can be factored in time polynomial in $n$ when:*

$$\alpha > \frac{1}{4}.$$

**Proof.**
The attack originates from Coppersmith's attack that uses partial knowledge of $p$. Namely, when $k_p$ and $k_q$ are known, then one knows

$$p \equiv k_p^{-1}(k_p - 1) \mod e \quad \text{and} \quad q \equiv k_q^{-1}(k_q - 1) \mod e.$$

When $e > N^{\frac{1}{4}}$, then Theorem 4.1 says that one can find $p$.

□

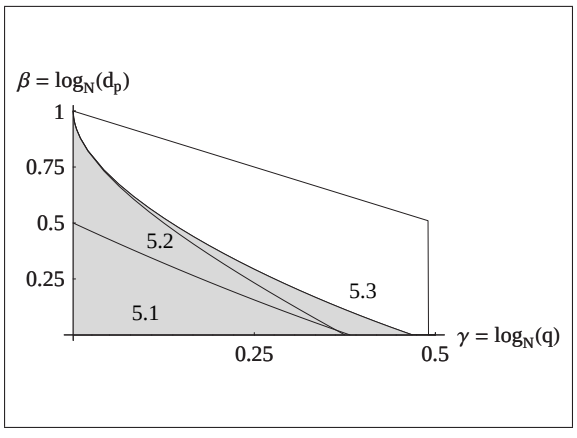


Figure 5.1: CRT-UnbalancedPrimes

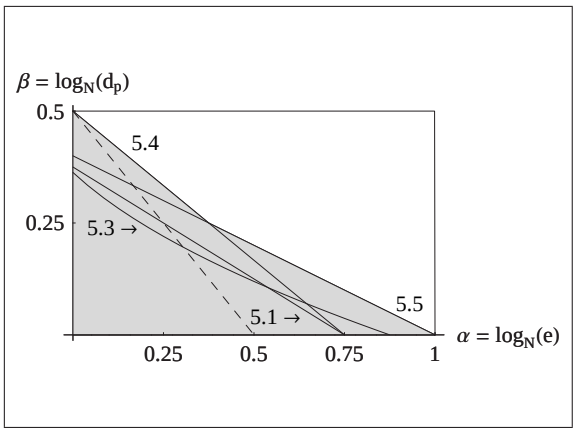


Figure 5.2: CRT-BalancedExponents

This concludes the discussion of all known polynomial time attacks on CRT variants. Note that up to now, there only exist polynomial time attacks on CRT-UnbalancedPrimes and CRT-BalancedExponents. Figure 5.1 and 5.2 give an overview of the attacks on these variants. The numbers in the figures refer to the respective theorems.

Besides the 'normal' polynomial time attacks on RSA-CRT variants, there also exist partial key exposure attacks. The known partial key exposure attacks from Blömer and May [6] on CRT-Small-$e$ are summarized in the next theorem. For a proof of this theorem, we refer to [6].

**Theorem 5.7 (Blömer/May, [6])**
*For every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Let $N = pq$ be an $n$-bit RSA modulus, where $p$ and $q$ are primes of equal bitsize. Let $0 < \alpha < 1$, $0 < \beta \leq \frac{1}{2}$, and $0 < \delta < \beta$. Furthermore, let $e$, $d_p$, $d_q$ satisfy $ed_p = 1 + k_p(p-1)$ and $ed_q = 1 + k_q \mod (q-1)$ with bitsize$(e) = \alpha n$.*

1. *Suppose an attacker knows MSBs of $d_p$, and suppose the unknown LSB-part of $d_p$ is of size $N^\delta$. Then $N$ can be factored in time polynomial in $n$ when:*

$$\delta < \frac{1}{4} - \alpha - \epsilon.$$

2. *Suppose an attacker knows LSBs of $d_p$, and suppose the unknown MSB-part of $d_p$ is of size $N^\delta$. Moreover, suppose that $e$ is of size poly$(n)$ such that all possible $k_p$ can be tried out in polynomial time. Then $N$ can be factored in time polynomial in $n$ when:*

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

## 5.3 A new attack on CRT-Small-$d_p, d_q$

In this section, we discuss a new attack on CRT-Small-$d_p, d_q$. Recall that all attacks in the previous section were not applicable in the standard RSA case with small CRT-exponents $d_p$ and $d_q$, that is, when $p$ and $q$ are balanced and $e$ is full size. Here, we describe a way to extend one of the attacks of Bleichenbacher/May [4] such that it also works in the case of CRT-Small-$d_p, d_q$. This leads to the first polynomial time attack on standard RSA with small private CRT-exponents.

The new attack involves a small root $(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)})$ of a polynomial $f$. As opposed to the other attacks in this thesis, we will use an alternative way of extracting a common root of a set of polynomials here. We use Gröbner bases instead of resultants. Because this alternative method also involves a new assumption, we shall explain the details first before we state the result of the new attack.

Extracting the common root:

Assume that we want to retrieve a common root from four polynomials $f$, $r_1$, $r_2$, $r_3$ in four variables. Usually, one uses resultants to eliminate variables one by one until one obtains a univariate polynomial $res_6(x_1)$ that has $x_1^{(0)}$ as a root.

$$\begin{aligned} res_1 &= \mathrm{Res}_{x_4}(f, r_1) & & \\ & & res_4 &= \mathrm{Res}_{x_3}(res_1, res_2) & \\ res_2 &= \mathrm{Res}_{x_4}(f, r_2) & & & res_6 = \mathrm{Res}_{x_2}(res_4, res_5) \\ & & res_5 &= \mathrm{Res}_{x_3}(res_2, res_3) & \\ res_3 &= \mathrm{Res}_{x_4}(f, r_3) & & & \end{aligned}$$

However, this method only works under the assumption that the polynomials are algebraically independent. If, for example, $res_4(x_1, x_2)$ and $res_5(x_1, x_2)$ share a common polynomial factor $g(x_1, x_2)$ that contains the root, then $res_6(x_1) = \mathrm{Res}_{x_2}(res_4, res_5)$ will be the zero polynomial and $x_1^{(0)}$ cannot be determined.

Unfortunately, one cannot easily use more than three candidates $r_j$, besides repeating the scheme for different combinations. Moreover, the last resultant computation can take a significant amount of time and memory, since the degrees of the polynomials that are the outcome of the resultants grow fast. This is why we use *Gröbner bases* instead of resultant methods to extract the root. For a detailed introduction of Gröbner bases, we refer to [21, Chapter 2].

Suppose we have a set of polynomials $\{f, r_1, \ldots, r_\ell\}$ and suppose that these polynomials have the small root $(x_1^{(0)}, \ldots, x_v^{(0)})$ in common. Then a Gröbner basis $G := \{g_1, \ldots, g_t\}$ is a set of polynomials of which one of the properties is that it preserves the set of common roots of $\{f, r_1, \ldots, r_\ell\}$. In other words, the variety of the ideal $I$ generated by $\{g_1, \ldots, g_t\}$ is the same as the variety of the ideal generated by $\{f, r_1, \ldots, r_\ell\}$. The advantage of having a Gröbner bases is that the $g_i$ can be computed with respect to some ordering that eliminates the variables. Having such an elimination ordering, it is easy to extract the desired root.

In our experiments we usually find much more polynomials $r_1, \ldots, r_\ell$ than the required amount of $\ell = 3$. Therefore, we have two advantages of Gröbner basis in comparison with resultant methods. First, in contrast to resultants the computation time of a Gröbner basis usually benefits from more overdefined systems. This lowers the time for extracting the root. Secondly, we do not have to search over all subsets of three polynomials until we find an algebraically independent one. Instead, we simply put all the polynomials in our Gröbner basis computation. This computation can only fail if the variety $\mathbf{V}(I)$ defined by the ideal $I$ which is generated by $\{f, r_1, \ldots, r_\ell\}$ is not zero-dimensional. Therefore, we make the following heuristic assumption for our attack.

**Assumption 5.8 (Relaxation of Assumption 3.7)**
*Let $f, r_1, \ldots, r_\ell$ be the polynomials in our attack that share the root $(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)})$ over the integers, for some $\ell \geq 3$. We assume that the variety $\boldsymbol{V}(I)$ of the ideal $I$ generated by $\{f, r_1, \ldots, r_\ell\}$ is zero-dimensional.*

Now that we have defined the assumption upon which our new result will be based, we are ready to state it.

**Theorem 5.9 (RSA-CRT with Small $d_p$, $d_q$)**
*Under Assumption 5.8, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$. Let $e < \phi(N)$, $d_p < p - 1$, and $d_q < q - 1$ be the public exponent and private CRT-exponents, satisfying $ed_p \equiv 1 \mod (p-1)$ and $ed_q \equiv 1 \mod (q-1)$. Let $bitsize(d_p), bitsize(d_q) \leq \beta n$. Then $N$ can be factored in time polynomial in $n$ provided that*

$$\beta < 0.0734 - \epsilon.$$

The rest of Section 5.3 will be devoted to the details of this new result. We will show the polynomial with a small root from which the attack is derived, prove the result of Theorem 5.9, comment on how the attack can be implemented, and discuss the results of the experiments.

### 5.3.1 A bound for a specific polynomial $f$ with a small root

Let us take another look at the attack of Bleichenbacher and May from Theorem 5.5.

Bleichenbacher and May show that multiplying the two RSA-CRT equations $ed_p = 1 + k_p(p-1)$ and $ed_q = 1 + k_q(q-1)$ with each other in a clever way yields the linear equation

$$e^2x_1 + ex_2 - (N-1)x_3 - x_4 = 0,$$

if we substitute $x_1 = d_pd_q$, $x_2 = d_p(k_q - 1) + d_q(k_p - 1)$, $x_3 = k_pk_q$, $x_4 = k_p + k_q - 1$.

Although linearization of an equation makes the analysis easier and keeps the lattice dimension small, better results can sometimes be obtained by using a nonlinear polynomial equation directly. The equation

$$e^2d_pd_q + ed_p(k_q - 1) + ed_q(k_p - 1) - (N-1)k_pk_q - (k_p + k_q - 1) = 0$$

yields a polynomial

$$f(x_1, x_2, x_3, x_4) = e^2x_1x_2 + ex_1x_4 - ex_1 + ex_2x_3 - ex_2 - (N-1)x_3x_4 - x_3 - x_4 + 1$$

with monomials $1, x_1, x_2, x_3, x_4, x_1x_2, x_1x_4, x_2x_3, x_3x_4$ and a small root

$$(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)}) = (d_p, d_q, k_p, k_q), \text{ with } \begin{cases} |x_1^{(0)}| < X_1 & = N^{\beta}, \\ |x_2^{(0)}| < X_2 & = N^{\beta}, \\ |x_3^{(0)}| < X_3 & = N^{\alpha+\beta-\frac{1}{2}}, \\ |x_4^{(0)}| < X_4 & = N^{\alpha+\beta-\frac{1}{2}}. \end{cases}$$

We will follow the strategy for finding small integer roots of Section 3.3.2 to analyze which attack bound corresponds to this polynomial $f$.

In the basic strategy, the set $S$ that describes which monomials $x_1^{i_1}x_2^{i_2}x_3^{i_3}x_4^{i_4}$ are used for the shift polynomials, is simply the set that contains all monomials of $f^{m-1}$ for a given integer $m$. The set $M$ contains all monomials that appear in $x_1^{i_1}x_2^{i_2}x_3^{i_3}x_4^{i_4}f(x_1,x_2,x_3,x_4)$, with $x_1^{i_1}x_2^{i_2}x_3^{i_3}x_4^{i_4} \in S$. More precisely, $S$ and $M$ can be described as

$$x_1^{i_1}x_2^{i_2}x_3^{i_3}x_4^{i_4} \in S \Leftrightarrow \begin{cases} i_1 = 0,\ldots,m-1-i_3, \\ i_2 = 0,\ldots,m-1-i_4, \\ i_3 = 0,\ldots,m-1, \\ i_4 = 0,\ldots,m-1, \end{cases} \quad x_1^{i_1}x_2^{i_2}x_3^{i_3}x_4^{i_4} \in M \Leftrightarrow \begin{cases} i_1 = 0,\ldots,m-i_3, \\ i_2 = 0,\ldots,m-i_4, \\ i_3 = 0,\ldots,m, \\ i_4 = 0,\ldots,m. \end{cases}$$

However, in Section 3.3.2 it is also advised to explore the possibility of extra shifts of one or more variables. Since $X_1$ and $X_2$ are significantly smaller than $X_3$ and $X_4$ for $\alpha > \frac{1}{2}$, we find that the attack bound is superior for $\alpha = 1$ if we use extra shifts of $x_1$ and $x_2$.

Therefore, we take

$$x_1^{i_1}x_2^{i_2}x_3^{i_3}x_4^{i_4} \in S \Leftrightarrow \begin{cases} i_1 = 0,\ldots,m-1-i_3+t, \\ i_2 = 0,\ldots,m-1-i_4+t, \\ i_3 = 0,\ldots,m-1, \\ i_4 = 0,\ldots,m-1, \end{cases} \quad x_1^{i_1}x_2^{i_2}x_3^{i_3}x_4^{i_4} \in M \Leftrightarrow \begin{cases} i_1 = 0,\ldots,m-i_3+t, \\ i_2 = 0,\ldots,m-i_4+t, \\ i_3 = 0,\ldots,m, \\ i_4 = 0,\ldots,m, \end{cases}$$

for some $t$ that has to be optimized as a function of $m$ and $\alpha$.

Our goal is to find at least three polynomials $r_1$, $r_2$, $r_3$ with the root $(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)})$ over the integers. From Section 3.3.2 we know that these polynomials can be computed by lattice basis reduction techniques as long as

$$X_1^{s_1}X_2^{s_2}X_3^{s_3}X_4^{s_4} < W^{s_W}, \text{ for } s_j = \sum_{x_1^{i_1}x_2^{i_2}x_3^{i_3}x_4^{i_4} \in M \setminus S} i_j \quad \text{and } s_W = |S|. \tag{5.1}$$

For a given integer $m$ and $t = \tau m$, our last definition of $S$ and $M$ yields the bound

$$(X_1X_2)^{(\frac{5}{12}+\frac{5}{3}\tau+\frac{9}{4}\tau^2+\tau^3)m^4+o(m^4)}(X_3X_4)^{(\frac{5}{12}+\frac{5}{3}\tau+\frac{3}{2}\tau^2)m^4+o(m^4)} < W^{(\frac{1}{4}+\tau+\tau^2)m^4+o(m^4)}.$$

### 5.3.2 Description of the new attack

We use the bound derived in the previous section to prove Theorem 5.9. Moreover, we show how to use the new attack for the case of CRT-BalancedExponents instead of CRT-Small-$d_p, d_q$.

Proof of Theorem 5.9:

Let us continue with the polynomial

$$f(x_1,x_2,x_3,x_4) = e^2x_1x_2 + ex_1x_4 - ex_1 + ex_2x_3 - ex_2 - (N-1)x_3x_4 - x_3 - x_4 + 1$$

from the previous section. In the inequality

$$(X_1X_2)^{(\frac{5}{12}+\frac{5}{3}\tau+\frac{9}{4}\tau^2+\tau^3)m^4+o(m^4)}(X_3X_4)^{(\frac{5}{12}+\frac{5}{3}\tau+\frac{3}{2}\tau^2)m^4+o(m^4)} < W^{(\frac{1}{4}+\tau+\tau^2)m^4+o(m^4)},$$

the values $X_1 = X_2 = N^{\beta}$, $X_3 = X_4 = N^{\alpha+\beta-\frac{1}{2}}$, and $W = \max\{e^2X_1X_2, (N-1)X_3X_4)\} = N^{2\alpha+2\beta}$ can be substituted. We let $m$ grow to infinity and let all terms of order $o(m^4)$ contribute to some error term $\epsilon$, and obtain the asymptotic bound

$$\left(\frac{5}{12}+\frac{5}{3}\tau+\frac{9}{4}\tau^2+\tau^3\right)\cdot 2\beta+\left(\frac{5}{12}+\frac{5}{3}\tau+\frac{3}{2}\tau^2\right)\cdot(2\alpha+2\beta-1) < \left(\frac{1}{4}+\tau+\tau^2\right)\cdot(2\alpha+2\beta),$$

which leads to

$$\beta < \frac{5-4\alpha+20\tau-16\alpha\tau+18\tau^2-12\alpha\tau^2}{14+56\tau+66\tau^2+24\tau^3} - \epsilon.$$

For $\alpha = 1$, we find an optimal value of $\tau \approx 0.381788$ and we get

$$\beta < 0.0734 - \epsilon.$$

Hence, for a 1024-bit modulus, $d_p$ and $d_q$ are in the asymptotical attack space if they are less then 75 bits. Analogously, for a 2048-bit modulus, $d_p$ and $d_q$ are in the asymptotical attack space if they are at most 150 bits.

The new attack for smaller $\alpha$:

For other $\alpha$ between $\frac{1}{2}$ and 1, we can use the same bound

$$\beta < \frac{5-4\alpha+20\tau-16\alpha\tau+18\tau^2-12\alpha\tau^2}{14+56\tau+66\tau^2+24\tau^3} - \epsilon$$

to optimize $\tau$, and find the corresponding bound for $\beta$.

In order to decide which shift polynomials to use, we assumed that $x_1^{(0)}, x_2^{(0)}$ are smaller than $x_3^{(0)}, x_4^{(0)}$, that is $\alpha \geq \frac{1}{2}$. For $\alpha < \frac{1}{2}$, one uses extra $x_3$ and $x_4$-shifts instead of extra $x_1$ and $x_2$-shifts. Because of the symmetry in the monomial set of $f$, one can immediately see that the attack bound is

$$(X_1X_2)^{(\frac{5}{12}+\frac{5}{3}\tau+\frac{3}{2}\tau^2)m^4+o(m^4)}(X_3X_4)^{(\frac{5}{12}+\frac{5}{3}\tau+\frac{9}{4}\tau^2+\tau^3)m^4+o(m^4)} < W^{(\frac{1}{4}+\tau+\tau^2)m^4+o(m^4)}.$$

The above bound leads to

$$\beta < \frac{5-4\alpha+20\tau-16\alpha\tau+27\tau^2-30\alpha\tau^2+12\tau^3-24\alpha\tau^3}{14+56\tau+66\tau^2+24\tau^3} - \epsilon.$$

Note that this bound only holds for $\alpha+\beta > \frac{1}{2}$, since we assume that the values of $k_p$ and $k_q$ are unknown to the attacker. Both conditions are only met if $\alpha \geq \frac{1}{6}$. However, in Section 5.3.4 we provide experimental evidence that our heuristic attack is successful only when $\alpha \geq \frac{1}{4}$.

The following table shows the asymptotic bound of this attack for several sizes of $\alpha$, and the optimal $\tau$ that is used to obtain the bound.

| $\alpha$ | optimal $\tau$ | bound |
|---|---|---|
| 0.25 | 0.382 | $\beta < 0.2867$ |
| 0.30 | 0.033 | $\beta < 0.2714$ |
| 0.4 | 0 | $\beta < 0.2428$ |
| 0.5 | 0 | $\beta < 0.2142$ |
| 0.6 | 0 | $\beta < 0.1857$ |

| $\alpha$ | optimal $\tau$ | bound |
|---|---|---|
| 0.7 | 0 | $\beta < 0.1571$ |
| 0.8 | 0 | $\beta < 0.1285$ |
| 0.9 | 0.151 | $\beta < 0.1002$ |
| 0.95 | 0.256 | $\beta < 0.0865$ |
| 1.0 | 0.382 | $\beta < 0.0734$ |

Table 5.1: Bounds for different choices of $\alpha$

In the revised paper by Sun/Hinek/Wu [69], the authors propose as new parameters $\{\alpha = 0.577, \beta = 0.186\}$. For this choice, we find the bound $\beta < 0.192$, which breaks the new proposal in polynomial time. In the following figure, we show the new attack area.

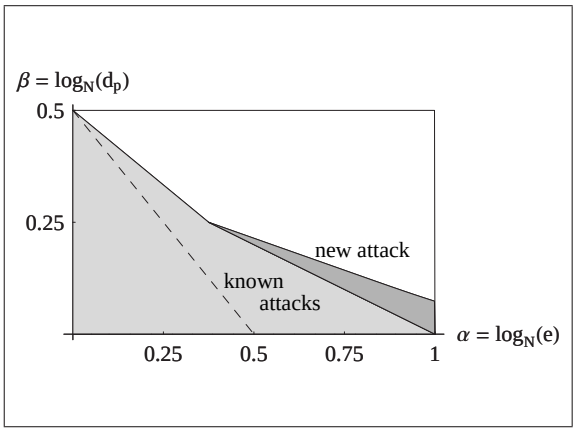


Figure 5.3: CRT-BalancedExponents

### 5.3.3 Implementation of the new attack

Although we have derived our attack bound directly from the strategy of Section 3.3.2, we deviate from this strategy for the implementation of the attack. Basically, we make use of Coppersmith's original technique [16] instead of Coron's reformulation [18]. This does not change the asymptotic bound of the attack, but it has a major practical advantage. Namely, the lattices used in the attacks are high-dimensional, and Coppersmith's original method requires only the reduction of a lower-dimensional sublattice[1]. Since the LLL process is the most costly factor in our attack, this leads to a significant improvement in practice. Furthermore, we slightly adapt Coppersmith's original method such that we directly obtain triangular lattice bases, which in turn simplifies the determinant calculations.

[1]In the Crypto'07 proceedings, a new article by Coron [19] shows how to adapt his method such that it also requires only the reduction of a sublattice instead of the reduction of the full lattice, and hence his new technique could be applied here, too.

So, let us first explain how to apply Coppersmith's original technique for our attack. We introduce the shift polynomials

$$g_{i_1 i_2 i_3 i_4}(x_1, x_2, x_3, x_4) = x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} f(x_1, x_2, x_3, x_4),$$

for $x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in S$ for a set of monomials $S$, as specified in Section 5.3.1.

As before, we define the set $M$ as the set of all monomials that appear in the shift polynomials. We use the notation $s = |S|$ for the total number of shifts and $d = |M| - |S|$ for the difference of the number of monomials and the number of shifts. Notice that the maximal coefficient of $f(x_1X_1, x_2X_2, x_3X_3, x_4X_4)$ is $e^2X_1X_2$, and the monomial corresponding to it is $x_1x_2$. We define $S'$ as the set of monomials $x_1^{i_1+1} x_2^{i_2+1} x_3^{i_3} x_4^{i_4}$, for $x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in S$. Naturally, $|S'| = |S| = s$. We now build a $(d+s) \times (d+s)$ matrix $B_1$ as follows.

The upper left $d \times d$ block is diagonal, where the rows of the block represent the monomials $x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in M \backslash S'$. The diagonal entry of the row corresponding to $x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4}$ is $(X_1^{i_1} X_2^{i_2} X_3^{i_3} X_4^{i_4})^{-1}$. The lower left $s \times d$ block contains only zeros. The last $s$ columns of the matrix $B_1$ represent the shift polynomials $g_{i_1 i_2 i_3 i_4} = x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} f$, for $x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in S$. The first $d$ rows correspond to the monomials in $M \backslash S'$, and the last $s$ rows to the monomials of $S'$. The entry in the column corresponding to $g_{i_1 i_2 i_3 i_4}$ is the coefficient of the monomial in $g_{i_1 i_2 i_3 i_4}$.

We illustrate the description with a simple example. Let us use the set $S$ as described in Section 5.3.1 with $m = 1$ and $t = 0$, which results in the lattice basis $B_1$ given in Figure 5.4. We only use the polynomial $f(x_1, x_2, x_3, x_4)$ itself as a shift polynomial. Therefore, $s = 1$ and we have $d + s = 9$ monomials. The rows represent the monomials $1, x_1, x_2, x_3, x_4, x_3x_4, x_2x_3, x_1x_4, x_1x_2$ and the last column corresponds to the coefficients of these monomials in $f$.

$$\left(\begin{array}{cccccccc|c}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & \frac{1}{X_1} & 0 & 0 & 0 & 0 & 0 & 0 & -e \\
0 & 0 & \frac{1}{X_2} & 0 & 0 & 0 & 0 & 0 & -e \\
0 & 0 & 0 & \frac{1}{X_3} & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & \frac{1}{X_4} & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{X_3X_4} & 0 & 0 & 1-N \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{X_2X_3} & 0 & e \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{X_1X_4} & e \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^2
\end{array}\right)$$

Figure 5.4: Matrix $B_1$ for the case $m = 1$, $t = 0$

In general, the determinant of matrix $B_1$ is

$$\det(B_1) = \left( \prod_{x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in M \backslash S'} (X_1^{i_1} X_2^{i_2} X_3^{i_3} X_4^{i_4})^{-1} \right) \cdot (e^2)^s.$$

Let us get back to our example, and consider the vector

$$\mathbf{v}(x_1, x_2, x_3, x_4) = (1, x_1, x_2, x_3, x_4, x_3x_4, x_2x_3, x_1x_4, x_1x_2)\,.$$

Note that

$$\mathbf{v}(x_1, x_2, x_3, x_4) \cdot B_1 = \left(1, \frac{x_1}{X_1}, \frac{x_2}{X_2}, \frac{x_3}{X_3}, \frac{x_4}{X_4}, \frac{x_3x_4}{X_3X_4}, \frac{x_2x_3}{X_2X_3}, \frac{x_1x_4}{X_1X_4}, f(x_1, x_2, x_3, x_4)\right).$$

So,

$$\|\mathbf{v}(d_p, d_q, k_p, k_q) \cdot B_1\| = \left\|\left(1, \frac{d_p}{X_1}, \frac{d_q}{X_2}, \frac{k_p}{X_3}, \frac{k_q}{X_4}, \frac{k_pk_q}{X_3X_4}, \frac{d_qk_p}{X_2X_3}, \frac{d_pk_q}{X_1X_4}, 0\right)\right\| \leq \sqrt{d}.$$

Since the $X_j$ determine the upper bound of the root, there is always such a vector $\mathbf{v}$ which, if one substitutes the unknowns $\{d_p, d_q, k_p, k_q\}$ for the variables $\{x_1, x_2, x_3, x_4\}$, becomes a vector with Euclidean norm smaller than $\sqrt{d}$ after multiplication with the matrix $B_1$.

Let us perform a unimodular transformation $U_1$ on $B_1$ to create a matrix $B_2$ such that

$$B_2 = U_1 \cdot B_1 = \left(\begin{array}{c|c} A_{d\times d} & 0_{d\times s} \\ \hline A'_{s\times d} & I_{s\times s} \end{array}\right).$$

Now if the rows of $B_1$ form a basis of a lattice $L$, then the rows of $B_2$ form a basis of the same lattice. Moreover, the rows of

$$B_3 = \left(\begin{array}{c|c} A_{d\times d} & 0_{d\times s} \end{array}\right)$$

are a basis of the sublattice $L_0$ of $L$ which has zeros in the last $s$ entries. Notice that $\det(L_0) = \det(L)$. Clearly, $\mathbf{v}(d_p, d_q, k_p, k_q) \cdot B_1$ is in the lattice $L_0$ spanned by the rows of $B_3$. Since

$$\mathbf{v}(d_p, d_q, k_p, k_q) \cdot B_1 = \mathbf{v}(d_p, d_q, k_p, k_q)U_1^{-1}B_2,$$

this means that the last $s$ entries of $\mathbf{v}(d_p, d_q, k_p, k_q)U_1^{-1}$ must be zero. We use the notation $\lfloor \mathbf{v} \rfloor_{\text{sh}}$ for the vector $\mathbf{v}$ 'shortened' to its first $d$ entries. Then,

$$\lfloor \mathbf{v}(d_p, d_q, k_p, k_q) \cdot B_1 \rfloor_{\text{sh}} = \lfloor \mathbf{v}(d_p, d_q, k_p, k_q)U_1^{-1}B_2 \rfloor_{\text{sh}} = \lfloor \mathbf{v}(d_p, d_q, k_p, k_q)U_1^{-1} \rfloor_{\text{sh}} A.$$

Next, we reduce $A$ using lattice basis reduction to a basis $B = U_2A$. It follows that

$$\lfloor \mathbf{v}(d_p, d_q, k_p, k_q) \cdot B_1 \rfloor_{\text{sh}} = \lfloor \mathbf{v}(d_p, d_q, k_p, k_q)U_1^{-1} \rfloor_{\text{sh}} U_2^{-1}B.$$

We use the notation $\mathbf{v}'(d_p, d_q, k_p, k_q)$ for the vector $\lfloor \mathbf{v}(d_p, d_q, k_p, k_q)U_1^{-1} \rfloor_{\text{sh}} U_2^{-1}$, and $B^*$ (with row vectors $\mathbf{b}_i^*$) for the basis after applying Gram-Schmidt orthogonalization to $B$. Now we can make three observations. Firstly, the vector $\mathbf{v}'$ is integral. This is because both matrices $U_1$ and $U_2$ have integer entries. Secondly, $\|\mathbf{v}(d_p, d_q, k_p, k_q) \cdot B_1\| < \sqrt{d}$. Thirdly, it is known [48] that the Gram-Schmidt orthogonalization of the LLL reduced basis satisfies

$$\|\mathbf{b}_d^*\| \geq 2^{\frac{-(d-1)}{4}} \det(L)^{\frac{1}{d}}.$$

So, if we combine these three facts, we obtain that

$$\begin{aligned}\sqrt{d} \geq \ & \|\mathbf{v}(d_p, d_q, k_p, k_q) \cdot B_1\| = \| \ \lfloor \mathbf{v}(d_p, d_q, k_p, k_q) \cdot B_1 \rfloor_{\text{sh}} \ \| = \| \ \mathbf{v}'(d_p, d_q, k_p, k_q) B \ \| \\ \geq \ & \mid \mathbf{v}'(d_p, d_q, k_p, k_q)_d \mid \ \cdot \|\mathbf{b}_d^*\| \geq \mid \mathbf{v}'(d_p, d_q, k_p, k_q)_d \mid \cdot 2^{\frac{-(d-1)}{4}} \det(L)^{\frac{1}{d}}.\end{aligned}$$

Since the terms $2^{\frac{-(d-1)}{4}}$ and $\sqrt{d}$ do not depend on $N$, we let them contribute to an error term $\epsilon$. Thus, whenever

$$\det(L)^{\frac{1}{d}} > 1,$$

we must have $\mid \mathbf{v}'(d_p, d_q, k_p, k_q)_d \mid = 0$.

Hence, the polynomial $r_1$ corresponding to the coefficient vector $\mathbf{v}'(x_1, x_2, x_3, x_4)_d$ has the root $(d_p, d_q, k_p, k_q)$ over the integers.

We shall now show that the bound $\det(L)^{\frac{1}{d}} > 1$ is equivalent to the bound (5.1) that was given in Section 5.3.1. One can check that

$$\det(L)^{\frac{1}{d}} \ = \ \det(B_1)^{\frac{1}{d}} = \left( \prod_{x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in M \setminus S'} (X_1^{i_1} X_2^{i_2} X_3^{i_3} X_4^{i_4})^{-1} \right)^{\frac{1}{d}} \cdot (e^2)^{\frac{s}{d}}.$$

So the bound $\det(L)^{\frac{1}{d}} > 1$ implies that

$$\left( \prod_{x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in M \setminus S'} (X_1^{i_1} X_2^{i_2} X_3^{i_3} X_4^{i_4}) \right) < (e^2)^s. \tag{5.2}$$

Replace $e^2$ by $\frac{W}{X_1 X_2}$. We observe that the difference between the monomials of $M \setminus S'$ and $M \setminus S$ is $s$ times the monomial $x_1 x_2$. Multiplying both sides by $(X_1 X_2)^s$ yields

$$X_1^{s_1} X_2^{s_2} X_3^{s_3} X_4^{s_4} < W^s, \text{ for } s_j = \sum_{x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in M \setminus S} i_j \quad \text{and } s = \sum_{x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in S} 1.$$

It follows that if this bound holds, then applying Coppersmith's method gives us a polynomial $r_1(x_1, x_2, x_3, x_4)$ from the coefficient vector $\mathbf{v}'(x_1, x_2, x_3, x_4)_d$, such that $r_1$ has the desired root $(d_p, d_q, k_p, k_q)$ over the integers. But in order to extract the root, we have to construct at least two more polynomials which share the same root.

It is always possible to construct a constant number of polynomials with the same common root provided that condition (5.2) is satisfied, at the cost of a slightly larger error term $\epsilon$ in the construction. To show this, we use a theorem of Jutla [40], which gives us a lower bound for the length of any Gram-Schmidt vector in an LLL reduced basis. Namely,

$$\|\mathbf{b}_i^*\| \geq 2^{\frac{-(i-1)}{4}} \left( \frac{\det(L)}{b_{\max}^{m-i}} \right)^{\frac{1}{i}} \qquad \text{for } i = 1 \ldots d,$$

where $b_{\max}$ is the maximal length of the Gram-Schmidt orthogonalization of the matrix $A$ (the matrix before starting the LLL reduction process). Following the analysis of [40], it can be checked that in our attack, $b_{\max} = e^2$. Therefore, $\|\mathbf{b}_i^*\| > 1$ reduces to

$$2^{\frac{-(i-1)}{4}} \left( \frac{\left( \prod_{x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in M \setminus S'} (X_1^{i_1} X_2^{i_2} X_3^{i_3} X_4^{i_4})^{-1} \right) \cdot (e^2)^s}{(e^2)^{d-i}} \right)^{\frac{1}{i}} > 1.$$

Since $2^{\frac{-(i-1)}{4}}$ does not depend on $N$, we let it contribute to an error term $\epsilon$. This simplifies our condition to

$$\prod_{x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \in M \setminus S'} (X_1^{i_1} X_2^{i_2} X_3^{i_3} X_4^{i_4}) < (e^2)^{s-(d-i)},$$

Notice that for $i = d$, we obtain the same bound as in (5.2). In Section 5.3.1, we have seen that $s = (\frac{1}{4} + \tau + \tau^2)m^4 + o(m^4)$. Hence, as long as $d - i = o(m^4)$, the asymptotic bound does not change and we get just another error term that contributes to $\epsilon$. This is clearly satisfied if $d - i$ is a small constant. Thus, all polynomials $r_1, \ldots, r_\ell$ corresponding to the coefficient vectors $\mathbf{v}'(x_1, x_2, x_3, x_4)_{d+1-i}$, $i = 1 \ldots \ell$, share the common root $(d_p, d_q, k_p, k_q)$.

As we discussed at the beginning of Section 5.3, Gröbner bases are an excellent way to find a common root from the set $\{f, r_1, \ldots, r_\ell\}$. Hence, under Assumption 5.8, we can find the secret root $(d_p, d_q, k_p, k_q)$.

### 5.3.4 Experiments for the new attack

In the following experiments, we applied our attack for varying sizes of $e$ and $d_p$, $d_q$. The LLL reduction was carried out using a C-implementation of the provable $L^2$ reduction algorithm due to Nguyen and Stehlé [58]. The timings were performed on a 1GHz PC running Cygwin.

#### Experiments for small $e$

All experiments in this section were done for 1000-bit $N$. For every fixed $e$, we looked for the maximal bitsize for $d_p, d_q$ that gave us enough small vectors for recovering the secrets. In our experiments, we fixed the attack parameter $m = 2$ and tried different values of $t$.

In the table below, the third column provides the bound of Bleichenbacher and May (Theorem 5.5) which can be achieved using a 3-dimensional lattice. The fourth column provides the bound for the attack that appears in the papers of Galbraith, Heneghan, and McKee [28] and Sun and Wu [70], and that was described in Theorem 5.4. This attack, that we will call the GHM-attack from now on, is also related to the attack described in this section. If an entry in the GHM-column is negative for a specific choice of $\alpha$, then this means that the GHM-attack does not work for this $\alpha$ (since obviously, private CRT-exponents with negative bitsize are not possible).

The $\beta$-column gives the theoretical upper bound for the chosen parameters $m, t$ and $e$. If an entry in this column is negative, then it means that theoretically, the new attack should not work yet for this small lattice size. The 'asymp'-column gives the asymptotic bound which is reached when the lattice dimension goes to infinity.

| $e$ | $d_p, d_q$ | BM Thm.5.5 | GHM Thm.5.4 | $\beta$ | asymp | lattice parameters | LLL time |
|---|---|---|---|---|---|---|---|
| 250 bit | 332 bit | 0.250 | 0.333 | 0.227 | 0.287 | $m = 2, t = 0, \dim = 27$ | 2 sec |
| 300 bit | 299 bit | 0.250 | 0.300 | 0.209 | 0.271 | $m = 2, t = 0, \dim = 27$ | 2 sec |
| 400 bit | 239 bit | 0.240 | 0.233 | 0.173 | 0.243 | $m = 2, t = 0, \dim = 27$ | 2 sec |
| 500 bit | 199 bit | 0.200 | 0.167 | 0.136 | 0.214 | $m = 2, t = 0, \dim = 27$ | 2 sec |
| 577 bit | 168 bit | 0.169 | 0.115 | 0.108 | 0.192 | $m = 2, t = 0, \dim = 27$ | 2 sec |
| 700 bit | 119 bit | 0.120 | 0.033 | 0.064 | 0.157 | $m = 2, t = 0, \dim = 27$ | 2 sec |
| 800 bit | 79 bit | 0.080 | −0.033 | 0.027 | 0.128 | $m = 2, t = 0, \dim = 27$ | 2 sec |
| 900 bit | 38 bit | 0.040 | −0.100 | −0.009 | 0.100 | $m = 2, t = 0, \dim = 27$ | 2 sec |
| 900 bit | 40 bit | 0.040 | −0.100 | 0.013 | 0.100 | $m = 2, t = 1, \dim = 56$ | 93 sec |
| 925 bit | 29 bit | 0.030 | −0.117 | −0.018 | 0.093 | $m = 2, t = 0, \dim = 27$ | 2 sec |
| 925 bit | 31 bit | 0.030 | −0.117 | 0.006 | 0.093 | $m = 2, t = 1, \dim = 56$ | 87 sec |
| 950 bit | 19 bit | 0.020 | −0.133 | −0.027 | 0.087 | $m = 2, t = 0, \dim = 27$ | 2 sec |
| 950 bit | 23 bit | 0.020 | −0.133 | −0.001 | 0.087 | $m = 2, t = 1, \dim = 56$ | 80 sec |

Table 5.2: Experiments for the new attack: CRT-BalancedExponents

In all the experiments mentioned in Table 5.2, we were able to recover the factorization of $N$. Experimentally, we see that our attack is much better than theoretically predicted. The reason is that for these RSA parameter settings, the shortest vectors are linear combinations of certain subsets of the lattice basis. I.e., the shortest vectors belong to some specific sublattice and the determinant calculation of the full lattice in Section 5.3.1 does not accurately capture the optimal choice of basis vectors. However, to identify the optimal sublattice structure for every fixed size $e$ seems to be a difficult task.

Let us first comment on the results for 250-bit and 300-bit $e$. Recall the GHM-attack from Theorem 5.4, and note that it is closely related to our new attack. Basically, they use a Coppersmith method for finding modular roots, to find the small root $(k_p, k_q)$ of a polynomial $f_e$ modulo $e$. The polynomial $f_e$ is exactly our polynomial $f$ taken modulo $e$. Hence, the goal of their attack is to find the modular root $(k_p, k_q)$ of the polynomial

$$f_e(x_3, x_4) = (N-1)x_3x_4 + x_3 + x_4 - 1$$

modulo $e$. This polynomial $f_e$, with monomials $1, x_3, x_4, x_3x_4$ has the well-known ('generalized rectangle') bound

$$X_3X_4 < e^{\frac{2}{3}-\epsilon},$$

that specifies for which upper bounds $X_3, X_4$ of $x_3, x_4$ the root can be found in polynomial

time. Substituting $X_3 = X_4 = N^{\alpha+\beta-\frac{1}{2}}$, and $e = N^\alpha$, we find the attack bound

$$\beta < \frac{1}{2} - \frac{2}{3}\alpha - \epsilon.$$

Since for $\alpha = 0.25, \alpha = 0.3$, the bound of the GHM-attack is superior to our new attack bound, the GHM-attack should be used for these cases instead of the new attack. However, if one uses the new attack, the lattice basis reduction algorithm finds very short vectors that correspond to certain sublattices that still lead to the GHM-bound. This explains for these small values of $\alpha$, why the experimental results are better than expected.

These were the only instances that we discovered, where Assumption 5.8 failed. Since the reduced basis vectors corresponded to the underlying structure of the GHM-attack, we were not able to eliminate three variables. However, we always found a polynomial of the form $(k_p + k_q - 1)x_3x_4 - k_pk_q(x_3 + x_4 - 1)$ in the Gröbner basis, which directly yields $k_p$ and $k_q$. Knowledge of $k_p$ is sufficient to factor $N$ in polynomial time, provided that $e$ is large enough: Notice that

$$p = 1 - k_p^{-1} \mod e.$$

From the theorem of Coppersmith for factoring with high bits known (Theorem 4.1), it follows that we can find $p$ in polynomial time whenever $e \geq N^{\frac{1}{4}}$, which is satisfied in our experiments. We also made attacks for the case $e < N^{\frac{1}{4}}$, where we still got the secrets $k_p, k_q$. However, this information does not seem to be sufficient for factoring $N$ efficiently. This is consistent with the GHM-attack, where Galbraith, Heneghan, and McKee state that the attack only succeeds if the factorization of $N$ can be extracted in polynomial time from the knowledge of the exposed $k_p$, $k_q$.

For $\alpha \geq \frac{2}{5}$, i.e. $e$ of bitsize at least 400, Assumption 5.8 was always valid. In all experiments, the Gröbner basis corresponding to $\{f, r_1, \ldots, r_\ell\}$ yielded $(d_p, d_q, k_p, k_q)$ and therefore the factorization of $N$. The roots were found by using the F4 Gröbner basis algorithm implemented in Magma V2.11-14. We would like to note that, when we did not include all candidates $r_1, \ldots, r_\ell$ but used only a few, it sometimes happened that we could eliminate two variables only. In that case, we were still able to retrieve the secrets, since the Gröbner Basis, where $x_2$ and $x_4$ were eliminated, then contained a polynomial with the terms $(d_p + (k_p - 1)x_1 - d_px_3)$ and $(d_q + (k_q - 1)x_1 - d_qx_3)$ in its factorization.

For $e$ of bitsizes 400 up to 800, we actually rediscovered the bound $\frac{2}{5}(1-\alpha)$ by Bleichenbacher/May experimentally. Again the lattice basis reduction algorithm found certain sublattices which in this case lead to the BM-bound. Even a moderate increase of the lattice dimension did not give us any improvement in this range of $e$. Although our asymptotical bound always beats the BM-bound, we are not able to see this effect for small $e$, since going beyond the BM-bound requires high-dimensional lattice bases.

For $e$ larger than 900 bits we can for the first time see the effect of increasing the lattice dimension and we are able to go slightly beyond the BM-bound. This effect intensifies for full size $e$, where the BM-bound does not give any results at all.

**Experiments for full size $e$**

Here we describe the experiments for RSA with a standard key generation for small CRT-exponents, which usually yields full size $e$. Namely, the parameters $d_p$, $d_q$ are chosen for a fixed bitsize and $e$ is the unique integer modulo $\phi(N)$ which is the inverse of $d_p, d_q$ modulo $p-1$ and $q-1$, respectively.

Every experiment in Table 5.3 gave us sufficiently many polynomials with the desired roots over the integers, such that we could recover the factorization. The Gröbner basis computation never took more than 100 sec and consumed a maximum of 300 MB.

Notice that for 10000-bit $N$, we can recover $d_p, d_q$ of bitsize 140, which would not be possible by a meet-in-the-middle attack.

| $N$ | $d_p, d_q$ | $\beta$ | lattice parameters | LLL time |
|---|---|---|---|---|
| 1000 bit | 10 bit | $-0.015$ | $m = 2, t = 1, \dim = 56$ | 61 sec |
| 1000 bit | 13 bit | $-0.002$ | $m = 2, t = 2, \dim = 95$ | 1129 sec |
| 1000 bit | 15 bit | 0.002 | $m = 3, t = 1, \dim = 115$ | 13787 sec |
| 2000 bit | 20 bit | $-0.015$ | $m = 2, t = 1, \dim = 56$ | 255 sec |
| 2000 bit | 22 bit | $-0.002$ | $m = 2, t = 2, \dim = 95$ | 1432 sec |
| 2000 bit | 32 bit | 0.002 | $m = 3, t = 1, \dim = 115$ | 36652 sec |
| 5000 bit | 52 bit | $-0.015$ | $m = 2, t = 1, \dim = 56$ | 1510 sec |
| 5000 bit | 70 bit | $-0.002$ | $m = 2, t = 2, \dim = 95$ | 18032 sec |
| 10000 bit | 105 bit | $-0.015$ | $m = 2, t = 1, \dim = 56$ | 3826 sec |
| 10000 bit | 140 bit | $-0.002$ | $m = 2, t = 2, \dim = 95$ | 57606 sec |

Table 5.3: Experiments for the new attack: CRT-Small-$d_p, d_q$

As in the experiments before, the $\beta$-bound is very inaccurate. For lattice dimensions 56 and 95, we should not obtain any results at all, while experimentally we succeeded for $d$ with bitsizes roughly a 0.010-fraction respectively a 0.013-fraction of $n$. On the other hand, our asymptotical bound states that we could in theory go up to a 0.073-fraction. Unfortunately, we are a tad bit away from the theoretical bound, since currently the best LLL reductions only allow to reduce lattice bases of moderate size, when the base matrices have large entries. Let us give a numerical example. Theoretically, for $m = 10$ we find an optimal value of $t = 6$ which yields a bound of 0.063. However, this parameter choice results in a lattice dimension of 4200 which is clearly out of practical reach.

Our result guarantees that one can find the factorization of $N$ for a sufficiently large – but fixed – lattice dimension for CRT-exponents $d_p, d_q$ up to a 0.073-fraction. Moreover, it is does not rule out that one can go beyond this bound. Even with our approach, the experimental results seem to indicate that an analysis of sublattice structures could lead to a better theoretical bound (see Section 7.2). We hope that these open problems stimulate further research in the exciting areas of lattice-based cryptanalysis and fast practical lattice basis reduction algorithms.

## 5.4 A new attack on CRT-Qiao&Lam

Qiao and Lam [60] proposed to use $d_p$ and $d_q$ such that $d_p - d_q = 2$ in their method for fast signature generation on a low-cost smartcard. For the size of $d_p$ and $d_q$, they suggest 128 bits to counteract the meet-in-the-middle attack. Moreover, they state that 96 bits should be enough to counteract this attack in practice. In current proposals, a minimum of 160 bits is advised for the private exponents to counteract the meet-in-the-middle attack.

In this section, we explain how a small root of a polynomial $f(x, y, z) = a_0 + a_1x + a_2x^2 + a_3y + a_4z + a_5xy + a_6xz + a_7yz$ results in a new attack on CRT-Qiao&Lam. We show the following result.

**Theorem 5.10 (Attack on CRT-Qiao&Lam)**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$. Let $ed \equiv 1 \mod \phi(N)$, and $d_p$ and $d_q$ be such that $d_p \equiv d \mod (p-1)$ and $d_q \equiv d \mod (q-1)$. Assume that $d_p$ and $d_q$ are chosen such that $d_p = d_q + c$ for some known $c$ and for bitsize$(d_p)$, bitsize$(d_q) \leq \beta n$ for some $0 < \beta < \frac{1}{2}$. Then $N$ can be factored in time polynomial in $n$ provided that*

$$\beta < \frac{1}{4}(4 - \sqrt{13}) - \epsilon.$$

Notice that $\frac{1}{4}(4 - \sqrt{13}) \approx 0.099$. Hence, our attack applies whenever $d_p$ or $d_q$ is smaller than $N^{0.099-\epsilon}$ and the difference $c = d_p - d_q$ is known to an attacker.

### 5.4.1 A bound for a specific polynomial $f$ with a small root

In this section we will give a novel analysis for a trivariate polynomial that appears in the cryptanalysis of the CRT-Qiao&Lam variant (and, as we shall see in Chapter 6, also in the cryptanalysis of the Common Prime RSA variant).

Let $f(x, y, z) = a_0 + a_1x + a_2x^2 + a_3y + a_4z + a_5xy + a_6xz + a_7yz$ be a polynomial with a small root $(x^{(0)}, y^{(0)}, z^{(0)})$, with $|x^{(0)}| < X$, $|y^{(0)}| < Y$, $|z^{(0)}| < Z$. We show that under Assumption 3.7 for every fixed $\epsilon$, all sufficiently small roots can be found in time polynomial in the bitsize of $W$ provided that

$$X^{7+9\tau+3\tau^2}(YZ)^{5+\frac{9}{2}\tau} < W^{3+3\tau-\epsilon},$$

where we can optimize $\tau \geq 0$ after the substitution of values for $X, Y, Z$, and $W$.

Let us follow the extended strategy described in Section 3.3.2 to show how this bound can be obtained. Our goal is to construct two polynomials $r_1, r_2$ with root $(x^{(0)}, y^{(0)}, z^{(0)})$ that are not multiples of $f$. To do so, we fix an integer $m$ depending on $\epsilon$ and an integer $t = \tau m$ that describes the number of extra $x$-shifts. We define $R = WX^{2(m-1)+t}(YZ)^{m-1}$ and $f' = a_0^{-1}f \mod R$. The shift polynomials $g$ and $g'$ are given by:

$$\begin{aligned} g: &\quad x^{i_1}y^{i_2}z^{i_3}f'(x,y,z)X^{2(m-1)+t-i_1}Y^{m-1-i_2}Z^{m-1-i_3} &&\text{for} \quad x^{i_1}y^{i_2}z^{i_3} \in S, \\ g': &\quad Rx^{i_1}y^{i_2}z^{i_3} &&\text{for} \quad x^{i_1}y^{i_2}z^{i_3} \in M \backslash S, \end{aligned}$$

for

$$S = \bigcup_{0\le j\le t}\{x^{i_1+j}y^{i_2}z^{i_3} \mid x^{i_1}y^{i_2}z^{i_3} \text{ is a monomial of } f^{m-1}\},$$

$$M = \{\text{monomials of } x^{i_1}y^{i_2}z^{i_3}\cdot f \mid x^{i_1}y^{i_2}z^{i_3}\in S\}.$$

It follows that

$$\begin{aligned} x^{i_1}y^{i_2}z^{i_3}\in S \quad &\Leftrightarrow\ i_2=0,\dots,m-1\ ;\ i_3=0,\dots,m-1\ ;\\ &\qquad i_1=0,\dots,2(m-1)-(i_2+i_3)+t.\end{aligned}$$

$$x^{i_1}y^{i_2}z^{i_3}\in M \quad \Leftrightarrow\ i_2=0,\dots,m\ ;\ i_3=0,\dots,m\ ;\ i_1=0,\dots,2m-(i_2+i_3)+t.$$

All polynomials $g$ and $g'$ have the root $(x^{(0)},y^{(0)},z^{(0)})$ modulo $R$. Let $r_1$ and $r_2$ be linear combinations of the polynomials $g$ and $g'$. As was explained in Section 3.2, if $r_1$ and $r_2$ satisfy Howgrave-Graham's bound $||r_i(xX,yY,zZ)|| < \frac{R}{\sqrt{\omega}}$, then we can assume that $r_1$ and $r_2$ both have the root $(x^{(0)},y^{(0)},z^{(0)})$ over the integers, and also that they are algebraically independent of $f$.

Using the coefficient vectors of $g(xX,yY,zZ)$ and $g'(xX,yY,zZ)$ as a basis, we build a lattice $L$. We order the vectors such that the matrix is triangular, with the diagonal entries of $g$ equal to $X^{2(m-1)+t}(YZ)^{m-1}$, and those of $g'$ equal to $RX^{i_1}Y^{i_2}Z^{i_3} = X^{2(m-1)+t+i_1}Y^{m-1+i_2}Z^{m-1+i_3}W$.

Now by (3.7), provided that $\prod_{j=1}^{3}X_j^{s_j} < W^{|S|}$ with $s_j = \sum_{x_1^{i_1}\dots x_3^{i_3}\in M\setminus S} i_j$ holds, the polynomials $r_1$ and $r_2$ corresponding to the shortest two LLL reduced basis vectors satisfy Howgrave-Graham's bound. This bound reduces to

$$X^{(\frac{7}{3}+3\tau+\tau^2)m^3+o(m^3)}(YZ)^{(\frac{5}{3}+\frac{3}{2}\tau)m^3+o(m^3)} \le W^{(1+\tau)m^3+o(m^3)}.$$

If we let all terms of order $o(m^3)$ contribute to $\epsilon$, the condition simplifies to

$$X^{7+9\tau+3\tau^2}(YZ)^{5+\frac{9}{2}\tau} < W^{3+3\tau-\epsilon}.$$

### 5.4.2 Description of the new attack

We use the bound derived in the previous section to prove Theorem 5.10.

Proof of Theorem 5.10:

When $d_p - d_q = c$, the public and private variables of RSA-CRT satisfy the following relations.

$$\begin{cases} ed_p &= 1+k_p(p-1),\\ e(d_p-c) &= 1+k_q(q-1),\end{cases} \quad \text{or equivalently} \quad \begin{cases} ed_p-1+k_p &= k_p p,\\ ed_p-ce-1+k_q &= k_q q.\end{cases}$$

Multiplying the two equations results in

$$(1+ce)-(2e+ce^2)d_p+e^2d_p^2-(ce+1)k_p-k_q+ed_pk_p+ed_pk_q+(1-N)k_pk_q=0,$$

in which the unknowns are $d_p$, $k_p$, and $k_q$. We can extract from this equation that

$$f(x,y,z)=(1+ce)-(2e+ce^2)x+e^2x^2-(ce+1)y-z+exy+exz+(1-N)yz$$

has a small root $(d_p,k_p,k_q)$. From $(d_p,k_p,k_q)$, the factorization of $N$ can easily be found. Suppose $\max\{d_p,d_q\}$ is of size $N^\beta$ for some $\beta\in(0,\frac{1}{2})$. Then $k_p$ and $k_q$ are both bounded by $N^{\beta+\frac{1}{2}}$ (as usual, we omit constants and let these contribute to the error term $\epsilon$). Therefore, we put $X=N^\beta, Y=Z=N^{\beta+\frac{1}{2}}$, and $W=N^{2+2\beta}$.

In Section 5.4.1 we showed that for this polynomial, the asymptotic bound is

$$X^{7+9\tau+3\tau^2}(YZ)^{5+\frac{9}{2}\tau}<W^{3+3\tau},$$

where $\tau\geq 0$ can be optimized. Substituting the values for $X$, $Y$, $Z$, and $W$, we obtain

$$(7+9\tau+3\tau^2)\beta+(5+\tfrac{9}{2}\tau)(2\beta+1)-(3+3\tau)(2\beta+2)<0, \text{ or}$$

$$3\beta\tau^2+3(4\beta-\tfrac{1}{2})\tau+(11\beta-1)<0.$$

For the optimal value $\tau=\frac{\frac{1}{2}-4\beta}{2\beta}$, this reduces to

$$\beta<\frac{1}{4}(4-\sqrt{13})\approx 0.099.$$

Therefore, for a 1024 bit modulus $N$, the system should be considered unsafe when $d_p$ is at most $0.099\cdot 1024\approx 101$ bits. Theoretically, this breaks the system of Qiao and Lam for the proposed 96 bit exponents in time polynomial in the bitsize of $N$.

We can add an exhaustive search on the most significant bits of $d_p$ and try the attack for each candidate for $\tilde{d_p}$. Here, $d_p=\tilde{d_p}+d_0$, where the unknown part of $d_p$ is $d_0$. The corresponding polynomial $f$ will change, but it will still have the same monomials. Therefore, the analysis will follow easily. The proposal of Qiao and Lam to use 128 bit private exponents can also be considered unsafe when applying such an extra exhaustive search, although performing such an attack may be costly in practice.

### 5.4.3 Experiments for the new attack

We performed several experiments to test the validity of Assumption 3.7 and to show which results can be achieved with relatively small lattices. We implemented the new attacks on a 2.4GHz Pentium running Linux. The LLL lattice basis reduction was done using Shoup's NTL [66]. For the attack on RSA-CRT with known difference described in Theorem 5.10, the parameters $d_p, d_q$ were chosen with difference $d_p-d_q=2$ as suggested in the Qiao-Lam scheme. For $m=2$ the choice $t=8$ maximizes the size of the attackable $d_p$.

| $N$ | $d_p$ | lattice parameters | LLL time |
|---|---|---|---|
| 1000 bit | 10 bit | $m = 2, t = 3, \dim = 54$ | 32 min |
| 2000 bit | 22 bit | $m = 2, t = 3, \dim = 54$ | 175 min |
| 3000 bit | 42 bit | $m = 2, t = 3, \dim = 54$ | 487 min |
| 4000 bit | 60 bit | $m = 2, t = 3, \dim = 54$ | 1015 min |
| 5000 bit | 85 bit | $m = 2, t = 3, \dim = 54$ | 1803 min |
| 500 bit | 9 bit | $m = 2, t = 8, \dim = 99$ | 105 min |
| 1000 bit | 18 bit | $m = 2, t = 8, \dim = 99$ | 495 min |
| 500 bit | 13 bit | $m = 3, t = 3, \dim = 112$ | 397 min |

Table 5.4: Experiments attack CRT-Qiao&Lam

In each experiment we obtained two polynomials $r_1(x, y, z)$, $r_2(x, y, z)$ with the desired root $(x^{(0)}, y^{(0)}, z^{(0)})$. Solving $g(z) = \mathrm{Res}_y(\mathrm{Res}_x(r_1, f), \mathrm{Res}_x(r_2, f)) = 0$ yielded the unknown $z^{(0)}$. The parameters $y^{(0)}$ and $x^{(0)}$ could be obtained by back substitution. The resultant heuristic of Assumption 3.7 worked perfectly in practice. For every instance, we could recover the secrets and hence factor $N$.

One should note that our experiments are quite far from solving the proposed 96-bit $d_p, d_q$ instances of the Qiao-Lam scheme. Theoretically, the smallest $m$ for which one obtains the 96-bit bound is $m = 61$ together with $t = 36$, resulting in a lattice dimension of 376712. Reducing lattice bases in this dimension is clearly out of reach.

However, we would like to point out that we did not optimize the performance of our attack. For optimization of the running time, one should combine brute force guessing of most significant bits of $d_p$ with the described lattice attack. Moreover, one should apply faster lattice basis reduction methods like the recently proposed $L^2$-method of Nguyen, Stehlé [58]. Additionally, a significant practical improvement should be obtained by implementing Coppersmith's original method instead of Coron's method as we did in Section 5.3.

## 5.5 Tabular overview

The following table includes all known and new attacks on RSA-CRT variants that run in polynomial time.

| **Variant** | **Attack bound** | **Reference** |
|---|---|---|
| CRT-Small-$e$ | *known MSBs:* $\delta < \frac{1}{4} - \alpha - \epsilon$ | Thm. 5.7.1 |
| | *known LSBs:* $\delta < \frac{1}{4} - \epsilon$ and $k_p$ is known | Thm. 5.7.2 |
| CRT-Small-$d_p, d_q$ | $\beta < 0.0734 - \epsilon$ | Thm. 5.9 |
| CRT-UnbalancedPrimes | $\beta < \frac{1}{2}(1 - 3\gamma + \gamma^2) - \epsilon$ | Thm. 5.1 |
| | $\beta < 1 - \frac{2}{3}\gamma - \frac{2}{3}\sqrt{\gamma^2 + 3\gamma} - \epsilon$ | Thm. 5.2 |
| | $\beta < \frac{1}{3}(3 - 2\gamma - \gamma^2 - \sqrt{12\gamma - 8\gamma^2 - 5\gamma^3 + \gamma^4}) - \epsilon$ | Thm. 5.3 |
| CRT-BalancedExponents | $\beta < \frac{3}{8} - \frac{1}{2}\alpha - \epsilon$ | Thm. 5.1 |
| | $\beta < \frac{7}{12} - \frac{1}{12}\sqrt{7 + 48\alpha} - \epsilon$ | Thm. 5.3 |
| | $\beta < \frac{1}{2} - \frac{2}{3}\alpha - \epsilon$ | Thm. 5.4 |
| | $\beta < \frac{2}{5} - \frac{2}{5}\alpha - \epsilon$ | Thm. 5.5 |
| | $\beta < \frac{5-4\alpha+20\tau-16\alpha\tau+18\tau^2-12\alpha\tau^2}{14+56\tau+66\tau^2+24\tau^3} - \epsilon$, for $\alpha \geq \frac{1}{2}$, and $\tau \geq 0$ to be optimized | Sect. 5.3.2 |
| | $\beta < \frac{5-4\alpha+20\tau-16\alpha\tau+27\tau^2-30\alpha\tau^2+12\tau^3-24\alpha\tau^3}{14+56\tau+66\tau^2+24\tau^3} - \epsilon$ for $\frac{1}{4} \leq \alpha \leq \frac{1}{2}$, and $\tau \geq 0$ to be optimized | Sect. 5.3.2 |
| | $\alpha > \frac{1}{4}$ and $k_p$ is known | Thm. 5.6 |
| CRT-Qiao&Lam | $\beta < \frac{1}{4}(4 - \sqrt{13}) - \epsilon$ | Thm. 5.10 |

Table 5.5: Polynomial time attacks on RSA-CRT variants

# 6 Attacks on Common Prime RSA

In this chapter we discuss the known attacks on Common Prime RSA, and show a new attack on this variant. Section 6.3 is based on [37], a joint paper with Alexander May.

## 6.1 Introduction

In Chapter 5, we mentioned that RSA-CRT is often used when efficient decryption is needed, since using RSA-Small-$d$ can be unsafe because of the attacks of Wiener and Boneh/Durfee [75, 10]. However, there is also a possibility to choose $d < N^{\frac{1}{4}}$ in RSA while avoiding Wiener's attack. There is a variant of RSA where Wiener's attack works less well, as was already shown by Wiener, namely when $\gcd(p-1, q-1)$ has a large prime factor. Lim and Lee used this fact in a proposal [50], which was attacked a few years later by McKee and Pinch [56]. Recently Hinek [32] revisited this variant, calling it Common Prime RSA, and investigated its potential and its weaknesses.

In Common Prime RSA, we have $N = pq$ for primes $p$ and $q$ such that

$$p = 2ga + 1 \text{ and } q = 2gb + 1,$$

for $g$ a large (and possibly unknown) prime, and $a, b$ coprime integers. The exponents $e$ and $d$ are mutually inverse modulo $\operatorname{lcm}(p-1, q-1) = 2gab$:

$$ed = 1 + k \cdot 2gab, \text{ with } 0 < e, d < 2gab.$$

The goal is to safely choose an exponent $d < N^{\frac{1}{4}}$, which enables a fast RSA decryption process. We set $g = N^{\gamma}$ and $d = N^{\beta}$ for some $0 \leq \gamma < \frac{1}{2}$, $0 < \beta < 1 - \gamma$. Then, $e$ is of size $N^{1-\gamma}$, $k$ is of size $N^{\beta}$, and $a$ and $b$ are both of size $N^{\frac{1}{2}-\gamma}$.

## 6.2 Known attacks

We give a short description of the known attacks on instances of Common Prime RSA, in which we focus only on polynomial time attacks. Naturally, there exist non-polynomial time attacks for this specific RSA variant, such as a method for factoring these special primes $p$ and $q$ other than the standard factorization methods (see [56, 32]).

**Theorem 6.1 (Hinek, [32]: Known $g$)**
*Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$ such that $p - 1 = 2ga$ and $q - 1 = 2gb$, for some prime $g$ of bitsize $\gamma n$, with $0 \leq \gamma < \frac{1}{2}$. Let $ed \equiv 1 \mod 2gab$, with bitsize$(e) = (1-\gamma)n$ and bitsize$(d) = \beta n$, with $0 < \beta < 1 - \gamma$. Finally, suppose that the value of $g$ is known. Then $N$ can be factored in time polynomial in $n$ when:*

$$\gamma \geq \frac{1}{4}.$$

**Proof.**
As Hinek remarks in [32], the attack for $\gamma \geq \frac{1}{4}$ is very easy to perform. Recall that $p - 1 = 2ga$ and $q - 1 = 2gb$, so $a$ and $b$ are of size $N^{\frac{1}{2}-\gamma}$. When $g \geq a + b$, or equivalently: $\gamma \geq \frac{1}{4}$, then an attacker knows $c = a + b$ from

$$\frac{N-1}{2g} = 2gab + a + b \equiv a + b \mod g.$$

Then one can solve $a$ from $\frac{N-1}{2g} = 2ga(c - a) + c$.

□

When $g < a+b$, McKee/Pinch have an attack with expected running time $O(N^{\frac{1}{4}-\gamma})$, which implies a running time of at most $O(N^{\epsilon})$ for $\gamma > \frac{1}{4} - \epsilon$. Their attack for known $g$ is based on the Baby-Step-Giant-Step algorithm, see [56] for details.

Another attack on Common Prime RSA with known $g$ was given by Hinek in [32], and is summarized in the following theorem.

**Theorem 6.2 (Hinek, [32]: Known $g$ and small $d$)**
*For every $\epsilon > 0$, there exists an integer $n_0$ such that for every $n > n_0$, the following holds: Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$ such that $p - 1 = 2ga$ and $q - 1 = 2gb$, for some prime $g$ of bitsize $\gamma n$, with $0 \leq \gamma < \frac{1}{4}$. Let $ed \equiv 1 \mod 2gab$, with bitsize$(e) = (1-\gamma)n$ and bitsize$(d) = \beta n$, with $0 < \beta < 1 - \gamma$. Finally, suppose that the value of $g$ is known. Then $N$ can be factored in time polynomial in $n$ when:*

$$\beta < \frac{7}{6} - \frac{5}{3}\gamma - \frac{1}{3}\sqrt{(7 - 10\gamma)(1 - 4\gamma)} - \epsilon.$$

**Proof.**
The bound in the above theorem is not explicitly mentioned in [32], but it follows from the attack described there. Basically, when $g$ is known, $A = \lfloor \frac{N-1}{4g^2} \rceil$ is a good approximation of $ab$ since

$$\frac{N-1}{4g^2} = ab + \frac{a+b}{2g}.$$

Therefore, one can write $ab = A + \alpha$, with $\alpha \approx N^{\frac{1}{2}-2\gamma}$. From

$$ed - 1 = k2g(A + \alpha),$$

it follows that $f(x,y,z) = ex - 2gAy - 2gyz - 1$ has the root $(d,k,\alpha)$ over the integers, with $X = N^{\beta}$, $Y = N^{\beta}$, and $Z = N^{\frac{1}{2}-2\gamma}$. Note that this polynomial has the same monomials as the polynomial used for an attack in Section 4.4.2, namely the attack on RSA-Small-$d$, with knowledge of MSBs of $d$, *without* using the extra knowledge on $k$. It can be concluded that the attack will work for

$$X^{1+3\tau}Y^{2+3\tau}Z^{1+3\tau+3\tau^2} < W^{1+3\tau-\epsilon},$$

with $W = \|f(xX, yY, zZ)\| = N^{1+\beta-\gamma}$. Substituting the parameters, one gets the asymptotical bound

$$3\tau^2(\frac{1}{2} - 2\gamma) + 3\tau(\beta - \frac{1}{2} - \gamma) + (2\beta - \frac{1}{2} - \gamma) < 0,$$

and it follows that the optimal choice for $\tau$ is $\tau = \frac{\frac{1}{2}+\gamma-\beta}{1-4\gamma}$. One can check that this leads to the bound

$$\beta < \frac{7}{6} - \frac{5}{3}\gamma - \frac{1}{3}\sqrt{(7-10\gamma)(1-4\gamma)}, \text{ provided that } \gamma < \tfrac{1}{4}.$$

□

Note that this result can easily be extended with an exhaustive search on the most significant bits of $d$. Suppose that $d = \tilde{d} + d_0$, where $d_0$ is the unknown part of $d$. Replacing $d$ by $\tilde{d} + d_0$ will give rise to a similar polynomial $f$ (only the constant term changes), and therefore the analysis follows easily.

**Theorem 6.3 (Wiener and Hinek, [75, 32]: Small $d$ and unknown $g$)**
*Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$ such that $p - 1 = 2ga$ and $q - 1 = 2gb$, for some prime $g$ of bitsize $\gamma n$, with $0 \leq \gamma < \frac{1}{2}$. Let $ed \equiv 1 \mod 2gab$, with bitsize$(e) = (1-\gamma)n$ and bitsize$(d) = \beta n$, with $0 < \beta < 1-\gamma$. Then $N$ can be factored in time polynomial in $n$ if*

1. *the bound $\beta < \frac{1}{4} - \frac{1}{2}\gamma$ holds, or*
2. *the bound $\beta < \gamma^2 - \epsilon$ holds, or*
3. *the bound $\beta < \frac{2}{5}\gamma - \epsilon$ holds.*

*The bounds 2. and 3. hold for any $\epsilon > 0$ and sufficiently large $n$.*

**Proof.**
The first bound is the generalization of Wiener's attack to the case of Common Prime RSA. Let us discuss how Wiener's attack works in this case.

It holds that

$$ed = 1 + k \cdot 2gab, \quad \text{or:} \quad ed = 1 + k \cdot \frac{(p-1)(q-1)}{2g}.$$

Therefore,

$$\frac{e}{(p-1)(q-1)} - \frac{k}{d \cdot 2g} = \frac{1}{d(p-1)(q-1)},$$

which means that $\frac{k}{d \cdot 2g}$ is a good approximation of $\frac{e}{\phi(N)}$. Since $|N - \phi(N)| < 3N^{\frac{1}{2}}$, it follows that

$$\begin{aligned}
\left|\frac{e}{N} - \frac{k}{d \cdot 2g}\right| &= \left|\frac{ed \cdot 2g - kN}{Nd \cdot 2g}\right| = \left|\frac{(ed \cdot 2g - k\phi(N)) + k(\phi(N) - N)}{Nd \cdot 2g}\right| \\
&= \left|\frac{2g - k(N - \phi(N))}{Nd \cdot 2g}\right| = \left|\frac{1}{Nd} - \frac{k(N - \phi(N))}{Nd \cdot 2g}\right| \\
&\leq \left|\frac{3kN^{\frac{1}{2}}}{Nd \cdot 2g}\right| = \left|\frac{3k}{N^{\frac{1}{2}}d \cdot 2g}\right| < \left|\frac{3}{2N^{\frac{1}{2}}g}\right|.
\end{aligned}$$

Recall that $\frac{k}{d \cdot 2g}$ can be found using the continued fraction expansion of $\frac{e}{\phi(N)}$ when

$$\left|\frac{e}{N} - \frac{k}{d \cdot 2g}\right| < \frac{1}{(d \cdot 2g)^2}.$$

Ignoring the terms that do not depend on $N$, this results in the bound

$$2\beta + 2\gamma < \frac{1}{2} + \gamma, \text{ which reduces to } \beta < \frac{1}{4} - \frac{1}{2}\gamma.$$

With the knowledge of $dg$ and $k$, one can easily find $g$ as the remainder of the division of $edg$ by $k$. If one has $d$, $k$, and $g$, then $p$ and $q$ can be derived easily.

The second bound in this theorem arises from the following observations. Note that

$$\frac{N-1}{2} = g \cdot (2gab + a + b).$$

Let $h := 2gab + a + b$. Now, if the key equation

$$ed = 1 + k2gab$$

is multiplied by the inverse $\hat{e}$ of $e$ modulo $gh$ (so $\hat{e}e = 1 + \hat{\alpha}gh$), then

$$d - \hat{e} = (2\hat{e}kab - \hat{\alpha}dh)g.$$

So, $f_g(x) = x - \hat{e}$ has the small root $d$ modulo $g$. The upper bound on the root that we want to find is $X = N^{\beta}$. In this thesis, we do not discuss Coppersmith methods for finding small roots modulo an unknown prime of which a multiple is known, however these methods exist (see [13, 53]). By a known result described in [53, Theorem 6], we can find this root when

$$X < (gh)^{\gamma^2 - \epsilon},$$

which gives us the asymptotic bound $\beta < \gamma^2$. As before, this can be extended with an exhaustive search on the most significant bits of $d$, since the shape of the polynomial $f_g$ does not change if we replace $d$ by $\tilde{d} + d_0$ of which $d_0$ is the unknown LSB part of $d$.

The third attack of this theorem starts by multiplying the equations:

$$ed = 1 + k(p-1)b, \quad \text{and} \quad ed = 1 + k(q-1)a,$$

which can be written out as

$$e^2d^2 + ed(ka + kb - 2) - (N-1)k^2ab - (ka + kb - 1) = 0.$$

Now $f(x, y, z, u) = e^2x + ey - (N-1)z - u$ has the small root

$$(x_0, y_0, z_0, u_0) = (d^2, d(ka + kb - 2), k^2ab, (ka + kb - 1)),$$

with $X = N^{2\beta}, Y = N^{2\beta+\frac{1}{2}-\gamma}, Z = N^{2\beta+1-2\gamma}, U = N^{\frac{1}{2}-\gamma}$, and $W = \|f(xX, yY, zZ, uU)\|_\infty = N^{2\beta+2-2\gamma}$. The bound

$$XYZU < W^{1-\epsilon},$$

for a linear four-variate polynomial gives the asymptotic attack bound $\beta < \frac{2}{5}\gamma$. Adding an exhaustive search on the bits of $d$ is now a problem, since the shape of the polynomial changes drastically when replacing $d$ by $\tilde{d}+d_0$ and the analysis will not extend in this case. □

As a visualization, Figure 6.1 shows the attacks of Theorem 6.3, that is, all known polynomial time attacks on Common Prime RSA if we suppose that $g$ is unknown.

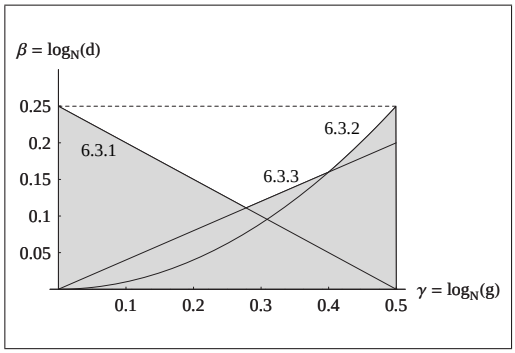


Figure 6.1: Common Prime RSA

## 6.3 A new attack on Common Prime RSA

In this section, we explain how a small root of a polynomial $f(x, y, z) = a_0 + a_1x + a_2x^2 + a_3y + a_4z + a_5xy + a_6xz + a_7yz$ results in a new attack on a variant of RSA called Common Prime RSA. We show the following result.

**Theorem 6.4 (Attack on Common Prime RSA)**
*Under Assumption 3.7, for every $\epsilon > 0$, there exists $n_0$ such that for every $n > n_0$, the following holds: Let $N = pq$ be an $n$-bit RSA modulus, and $p$, $q$ primes of bitsize $\frac{1}{2}n$ such that $p - 1 = 2ga$ and $q - 1 = 2gb$, for some prime $g$ of bitsize $\gamma n$, with $0 \leq \gamma < \frac{1}{2}$. Let $ed \equiv 1 \mod 2gab$, with bitsize$(e) = (1-\gamma)n$ and bitsize$(d) = \beta n$, with $0 < \beta < 1 - \gamma$. Then $d$ can be found in time polynomial in $n$ provided that*

$$\beta < \frac{1}{4}(4 + 4\gamma - \sqrt{13 + 20\gamma + 4\gamma^2}) - \epsilon.$$

### 6.3.1 Description of the new attack

The new attack can be obtained by treating the equation in Hinek's second lattice attack (Theorem 6.3.3) in a different way. We will now show how to modify Hinek's attack to obtain the result of Theorem 6.4.

Proof of Theorem 6.4:

Let us take another look at the equation

$$e^2d^2 + ed(ka + kb - 2) - (ka + kb - 1) - (N - 1)k^2ab = 0,$$

in which the unknowns are $d$, $k$, $a$ and $b$. We can conclude from this equation that the polynomial $f(x, y, z) = e^2x^2 + ex(y + z - 2) - (y + z - 1) - (N - 1)yz$ has a small root $(d, ka, kb)$ with $X = N^\beta, Y = N^{\beta+\frac{1}{2}-\gamma}, Z = N^{\beta+\frac{1}{2}-\gamma}$. Moreover, $W = N^{2+2\beta-2\gamma}$.

Note that the polynomial involved in this attack has the same set of monomials as the polynomial of the attack on CRT-Qiao&Lam (see Section 5.4.1). Therefore, we can use the asymptotical bound

$$X^{7+9\tau+3\tau^2}(YZ)^{5+\frac{9}{2}\tau} < W^{3+3\tau}$$

which yields

$$3\beta\tau^2 + 3(4\beta - \frac{1}{2} - \gamma)\tau + (11\beta - 1 - 4\gamma) < 0.$$

For the optimal value of $\tau$, namely $\tau = \frac{\frac{1}{2}+\gamma-4\beta}{2\beta}$, this reduces to

$$\beta < \frac{1}{4}(4 + 4\gamma - \sqrt{13 + 20\gamma + 4\gamma^2}).$$

Figure 6.2 shows the new attack region as well as the known attacks, for any size of modulus $N$. Combinations of $d$ and $g$ that should be considered unsafe by the new attack are in the dark shaded area, whereas the lighter shaded area was already unsafe by the known attacks. It can be seen that the number of 'safe' combinations $\{d, g\}$ with $d < N^{\frac{1}{4}}$ has significantly decreased.

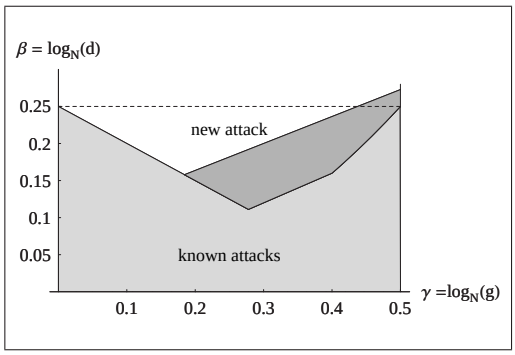


Figure 6.2: New attack region

We note that for 'small' $N$ (such as the regular 1024 bits), other attacks such as factoring attacks may apply, see [32]. Also, depending on the size of $N$, the attacks in the figure could be extended by an additional exhaustive search.

### 6.3.2 Experiments for the new attack

We performed experiments to check the validity of Assumption 3.7 and to demonstrate the practicality of our attack. We have implemented the new attack for the parameter setting $m = 2$, $t = 0$ (without the possible additional exhaustive search), to give an impression on what a realistic bound is for the smallest lattice possible. Of course, extending to $m = 3$, $m = 4$, etc. and using $x$-shifts will give results closer to the theoretical attack bound

$$\beta < \frac{1}{4}(4 + 4\gamma - \sqrt{13 + 20\gamma + 4\gamma^2}),$$

but will also result in a longer time needed for the lattice basis reduction. For $m = 2$, $t = 0$ the reduction time (the longest part of the attack) is about one minute. The following table summarizes the experimental results performed for $n = 1024$. As one can see, the results are already outside the *asymptotical* range of the known attacks.

| $\gamma$ | maximal $\beta$ (asymptotic) new attack | obtained $\beta$ $(m = 2, t = 0)$ new attack | maximal $\beta$ (asymptotic) known attacks |
|---|---|---|---|
| 0.10 | 0.130 | 0.07 | 0.20 |
| 0.20 | 0.164 | 0.10 | 0.15 |
| 0.30 | 0.200 | 0.13 $(*)$ | 0.12 |
| 0.40 | 0.237 | 0.17 $(*)$ | 0.16 |
| 0.50 | 0.275 | 0.2 | 0.25 |

Table 6.1: Experiments for the new attack on Common Prime RSA

The resultant heuristic of Assumption 3.7 worked perfectly in most cases. However, in the rare situation that both $\beta$ and $\gamma$ were very small (e.g. $\gamma = 0.1$ and $\beta = 0.05$), we encountered cases where some of the polynomials $r_i$ were algebraically dependent. In these cases, we could still recover the secret information in two different ways. One way was to use combinations of $r_1$ and the somewhat 'larger' $r_i$ for $i > 2$, instead of only $r_1$ and $r_2$. The other way was by examining the cause of the zero resultant. In essence, $\text{Res}_y(\text{Res}_x(r_1, f), \text{Res}_x(r_2, f)) = 0$ because $\text{Res}_x(r_1, f)$ and $\text{Res}_x(r_2, f)$ have a common polynomial factor, whose coefficients immediately reveal the secrets.

## 6.4 Tabular overview

The following table shows the known polynomial time attacks on Common Prime RSA.

| **Type of attack** | **Attack bound** | **Reference** |
|---|---|---|
| known $g$ | $\gamma \geq \frac{1}{4}$ | Thm. 6.1 |
| known $g$ and small $d$ | $\beta < \frac{7}{6} - \frac{5}{3}\gamma - \frac{1}{3}\sqrt{(7-10\gamma)(1-4\gamma)} - \epsilon$ and $\gamma < \frac{1}{4}$ | Thm. 6.2 |
| unknown $g$ and small $d$ | $\beta < \frac{1}{4} - \frac{1}{2}\gamma$ | Thm. 6.3.1 |
| | $\beta < \gamma^2 - \epsilon$ | Thm. 6.3.2 |
| | $\beta < \frac{2}{5}\gamma - \epsilon$ | Thm. 6.3.3 |
| | $\beta < \frac{1}{4}(4 + 4\gamma - \sqrt{13 + 20\gamma + 4\gamma^2}) - \epsilon$ | Thm. 6.4 |

Table 6.2: Polynomial time attacks on Common Prime RSA

# 7 Conclusion & open questions

## 7.1 The security of RSA: Advice for implementors

In this thesis we have presented many variants on RSA, and discussed a variety of attacks on them. In this section we give a conclusion on the security of RSA, and on the possibilities to speed up RSA encryption or decryption by choosing special parameters.

First of all, the main lesson of this thesis is:

*One should be very careful in choosing special relations between RSA parameters.*

Although these special relations may lead to more efficient RSA variants, they also give more possibilities to the attacker. Hence, the safest thing to do is to follow a standard implementation (for an $n$-bit modulus $N$ that is the product of two primes):

- Choose a parameter $n$ such that you are certain that factoring an $n$-bit RSA modulus cannot be performed.
- Choose random $\frac{1}{2}n$-bit primes $p$ and $q$ independently.
  If $p$ and $q$ are truly chosen at random, then one can expect that $p - q$ is not small enough to apply one of the attacks described by de Weger [74], but a check can always be performed. Similarly, one could check that $\gcd(p-1,\ q-1)$ is not too large, to prevent attacks on the Common Prime RSA variant.
- Choose a random integer $e$ smaller than, and coprime to $\phi(N) = (p-1)(q-1)$.
  If $e$ is chosen at random then one can expect with overwhelming probability that it has the same bitsize as $\phi(N)$ (or only a few bits smaller), so attacks on small $e$ will not apply.
- Determine $d$ by applying the Extended Euclidean Algorithm on $e$ and $\phi(N)$.
  Again, one can expect that $d$ is 'full size', hence attacks on small $d$ do not apply.
- In order to use Quisquater/Couvreur's decryption method using the Chinese Remainder Theorem [61], compute $d_p \equiv d \mod (p-1)$ and $d_q \equiv d \mod (q-1)$.
  In this way, $d_p$ and $d_q$ will have bitsize $\frac{1}{2}n$, so attacks on small $d_p$, $d_q$ will not apply.

In the above steps it is very easy to perform tests to check if $e$, $d$, and $d_p$, $d_q$ are indeed of the expected bitsize.

Specific applications may ask for more efficient variants, so let us discuss the possibilities for speeding up RSA that have been treated in this thesis.

RSA-Small-$e$ and RSA-CRT-Small-$e$:

One of the most popular RSA variants in practice is RSA with a small public exponent $e$ (or RSA-CRT with a small $e$). The choice $e = 3$ should be considered unsafe nowadays. For $e = 3$, there are simply too many things that can go wrong, like:

- attacks involving specific intercepted ciphertexts (Section 2.3),
- attacks related to the implementation (Section 2.3),
- partial key exposure attacks (Section 4.2 and 5.2).

For larger $e$, like the popular $e = 65537$, the only concern is partial key exposure attacks. As has been known for some time (see Section 4.2), if $e$ is this small, then only $\frac{1}{4}n$ LSBs of the decryption exponent $d$ are sufficient to retrieve the factorization of $N$. Similarly, only $\frac{1}{4}n$ LSBs or $\frac{1}{4}n$ MSBs of $d_p$ are sufficient if RSA-CRT-Small-$e$ is used for a very small value of $e$.

Even for larger $e$, one must always be aware that partial key exposure attacks exist (see Chapter 4) and that it is essential to keep all bits of $d$ (or $d_p, d_q$) secret, e.g. take countermeasures against side channel attacks.

RSA-Small-$d$ and RSA-CRT-Small-$d_p, d_q$:

If one wants to decrease the cost for RSA decryption (or RSA signature generation), then the main possibilities are to use RSA-Small-$d$ or RSA-CRT-Small-$d_p, d_q$. Since the attacks of Wiener [75] and Boneh/Durfee [10] we know that there exists a polynomial time attack whenever $d < N^{0.292}$.

In their paper, Boneh and Durfee explain that for $d < N^{0.5}$, the equation

$$x(N + 1 - y) \equiv 1 \mod e$$

is likely to have a unique solution $x = k$, $y = p + q$. Therefore, they propose to choose $d > N^{0.5}$. In all parameter proposals, it is wise to choose your parameters 'at a safe distance' from the current attacks. Having a list of attacks simply does not guarantee the non-existence of other attacks.

When choosing a small $d$, one again has to keep in mind that partial key exposure attacks exist whenever $d$ is less than $n$ bits long. Hence, sufficient countermeasures must be taken such that parts of the secret key cannot leak.

For the case of small $d_p$, $d_q$, note that in this thesis, the first polynomial time attack on RSA-Small-$d_p, d_q$ is discussed. However, the bound $\{d_p, d_q\} < N^{0.0734}$ is hard to achieve in practice, and we think that $\{d_p, d_q\} > N^{0.25}$ is still a very safe choice that can be made.

RSA-CRT-UnbalancedPrimes and RSA-CRT-BalancedExponents:

We doubt that the CRT-UnbalancedPrimes and CRT-BalancedExponents variants are used in practice. Especially the CRT-UnbalancedPrimes setting was mainly proposed in [52] to investigate which CRT-variants could be attacked in polynomial time. The CRT-BalancedExponents variant has a better motivation, but seems to give rise to many attacks. If one wants to use one of these variants, then we recommend to choose the parameters 'at a safe distance' from the current attacks (see Figure 5.1 and 5.3). For instance a suitable choice for CRT-BalancedExponents could be $e \approx N^{\alpha}$ and $\{d_p, d_q\} \approx N^{\beta}$ with $\alpha > \frac{1}{2}$ and $\beta > \frac{3}{4} - \frac{1}{2}\alpha$. However, since we do not know how to efficiently generate key pairs that satisfy this constraint (see Section 2.2), the relevance of this variant is truly questionable.

Other variants:

We do not recommend to use Common Prime RSA. Apart from the polynomial time attacks in Section 6 there are other, non-polynomial time time attacks that are already pretty powerful for $n = 1024$ or $n = 2048$ (see [32]). For these choices of $n$, there is simply not enough room to choose $d < N^{0.25}$ at some safe distance from the known attacks.

At this moment CRT-Qiao&Lam seems to be secure for $d_p > N^{0.25}$, but (as is the case for all variants) other attacks may arise that improve the current attack bound.

Naturally, we have not discussed all RSA variants in detail (although we have covered the most popular variants). Since Multi-prime RSA and Takagi's RSA were not discussed in this work, we refer to [33] and [13, 55] for the known attacks on these variants.

## 7.2 Open questions

In this section, we discuss some interesting open questions related to the work in this thesis. First, we will describe two open questions related to finding small roots of polynomials. Afterwards, we discuss two open questions related to cryptanalysis.

**Open questions related to finding small roots:**

Refinements of the (extended) strategy to choose shift polynomials:

In Section 3.3, we discussed a general strategy to choose the shift polynomials used in a Coppersmith method given the monomials that appear in a polynomial $f_N$ or $f$. For the extended strategy, we noted that in some cases, it is profitable to use extra shifts of a certain variable (or certain variables). Hence, the current way of choosing the shifts is to use the basic strategy, and then trying out some possibilities for using extra shifts.

*Open question: Can we find a refinement of the way to choose the shift polynomials that guarantees an optimal bound for any polynomial $f$ that the strategy is used on?*

Investigating the heuristic involved when dealing with multivariate polynomials:

Recall that in Coppersmith methods for finding roots of multivariate polynomials, we often have to deal with a heuristic assumption. If we want to find a modular root of a polynomial $f_N$ in $v$ variables ($v \geq 2$), and the LLL reduction gives us polynomials $r_1, r_2, \ldots, r_v$ that share the desired root over the integers, then we can extract the root *assuming* that the $r_i$ are algebraically independent. In the case of finding a small integer root of a polynomial $f$ in $v$ variables ($v \geq 3$), if the LLL reduction gives us polynomials $r_1, r_2, \ldots, r_{v-1}$ that share the desired root over the integers, then we can extract the root *assuming* that the polynomials of the set $\{f, r_1, \ldots, r_{v-1}\}$ are algebraically independent. The validation of such an assumption is done using experiments, and since usually no problems occur, most (practically oriented) researchers see no problems in using a heuristic attack. However, for the theory of Coppersmith methods it is important to study this heuristic in more detail.

*Open question: Can we define in which cases a Coppersmith method for a multivariate polynomial is provable, in other words, does not rely on a heuristic assumption?*

Recently, Bauer and Joux [2] made some important progress in this area. For the case of finding small roots of integer polynomials $f(x, y, z)$ they use the traditional Coppersmith method to find a polynomial $r_1(x, y, z)$ independent of $f$ that also has the desired root. Next, they construct a new lattice similar to Coppersmith's one that produces a third polynomial $r_2(x, y, z)$ that contains the root and that is independent from $\{f, r_1\}$. Bauer and Joux show that if a certain criterion on the Gröbner basis of the ideal defined by $\{f, r_1\}$ is satisfied, their method for finding a third independent polynomial $r_2$ will always succeed.

**Open questions related to cryptanalysis:**

Extending attacks with an exhaustive search:

On the one hand, we have seen in this thesis that for a given attack using a Coppersmith method, it can be costly to perform an attack in practice for parameters that are close to the attack bound. On the other hand, in most cases an attack bound can be extended a little bit by adding an exhaustive search.

Let us discuss a small example. Recall the new attack on Common Prime RSA of Section 6.3.1. We obtained the asymptotical bound

$$X^{7+9\tau+3\tau^2}(YZ)^{5+\frac{9}{2}\tau} < W^{3+3\tau}, \text{ for } \begin{cases} X = |d| = N^{\beta}, \\ Y = |ka| = N^{\beta+\frac{1}{2}-\gamma}, \\ Z = |kb| = N^{\beta+\frac{1}{2}-\gamma}, \\ W = \max\{e^2X^2, (N-1)YZ\} = N^{2+2\beta-2\gamma}. \end{cases}$$

Now suppose we guess a number of most significant bits of $d$ (say, a string of bitsize $\mu n$, for some $0 < \mu < \beta$). Then we can use the same asymptotical bound, but now with

$$X = N^{\beta-\mu},\ Y = Z = N^{\beta+\frac{1}{2}-\gamma},\ W = N^{2+2\beta-2\gamma}.$$

One can check that, instead of the asymptotical attack bound

$$\beta < \frac{1}{4}(4 + 4\gamma - \sqrt{13 + 20\gamma + 4\gamma^2})$$

we now obtain

$$\beta < \frac{1}{4}(4 + 4\gamma - \sqrt{13 + 20\gamma + 4\gamma^2 - 20\mu - 8\gamma\mu + 4\mu^2}).$$

A typical example would be $\gamma = 0.1, \mu = 0.02$ (think of $\mu = 0.02$ as guessing about 20 bits of $d$ if $N$ is a 1024-bit modulus). Then, the normal asymptotic bound of the attack is

$$\beta < 0.130464,$$

whereas the asymptotic bound *with* $\mu = 0.02$ is

$$\beta < 0.143914.$$

Hence, by taking $\mu = 0.02$ you achieve an improvement of the attack bound by 0.01345. It follows that one cannot directly translate the knowledge of 20 bits into an improvement of the attack bound by 20 bits. In other words, if one would perform this attack $2^{20}$ times for each possibility of the 20 MSBs of $d$, then one still would not be able to handle a $d$ that exceeds the asymptotical bound by 20 bits. This brings us to our open question.

*Open question: Is there a better way to combine an attack using a Coppersmith method with an exhaustive search?*

Imagine that if $d_M^{(1)}$ and $d_M^{(2)}$ are two possibilities for the MSBs of $d$, and that $d_M^{(1)}$ and $d_M^{(2)}$ are almost the same. Then is it really necessary to perform Coppersmith's method twice? The lattices involved will be very similar and it seems like a waste of computing time to perform the LLL reduction of these slightly different lattices twice. Is it possible to reuse a method performed for one candidate $d_M$ for a number of other candidates $d'_M$ that are 'close enough' to $d_M$?

Refining attack bounds using sublattices:

An important facet of Coppersmith attacks on RSA that we have not yet discussed in this thesis, is the use of sublattices. Boneh and Durfee [10] were the first to use sublattices to improve an attack bound. In Section 3.3.1 we explained that using the extended strategy for finding small modular roots, one can find the factorization of $N$ in polynomial time if $d < N^{0.284}$. However, the improved attack bound that Boneh and Durfee give is $d < N^{0.292}$. Let us briefly sketch how the use of sublattices helps here.

Recall the example given in Section 3.3.1 for the Boneh/Durfee-attack for the case $\{m = 2, t = 1\}$, where the lattice $L$ was spanned by the rows of a $9 \times 9$ matrix. Now imagine that one looks at a lattice $L'$ that is spanned by a subset of these nine shift polynomials. That means that the lattice $L'$ does not have full rank anymore, so calculating

its determinant (or, as it should be called in this case, its volume) takes more effort. However, "throwing away" some of the rows can result in a better bound, as shown in [10].

Often, the experiments of an attack will indicate that using a sublattice can be profitable. In practice, the smallest lattice vectors of the LLL reduced basis turn out to be an integer linear combination of only a specific subset of the original shift polynomials. In other words, the LLL algorithm does not use all vectors to approximate the shortest lattice vectors. This suggests that analyzing the attack for the lattice $L'$ containing only this subset of the shift polynomials can lead to an improved attack bound. For more details, we refer to the papers of Boneh/Durfee and Blömer/May on this topic [10, 5].

In our new attacks on RSA variants, we occasionally encountered this phenomenon. Recall the result of Theorem 5.9, our new attack on CRT-Small-$d_p, d_q$. As we mentioned in Section 5.3.3, we used Coppersmith's original method for the experiments, to profit from the fact that the lattices that need to be reduced in Coppersmith's original method have smaller dimension than the ones in Coron's reformulation [18]. However, with Coppersmith's original approach it is harder to analyze if sublattices can improve the attack bound. Recall that in Coron's method, the shortest vectors in the reduced method immediately correspond to the small polynomials $r_i$. Hence the transformation matrix of the LLL reduction can tell us which shift polynomials have contributed to the smallest polynomials $r_i$. We have programmed our new attack on RSA-CRT using Coron's method for $\{m = 2, t = 1\}$ (a 81-dim. lattice) to see what would happen in the LLL process. It turns out that the polynomials $r_i(x_1, x_2, x_3, x_4)$ corresponding to the smallest vectors in the LLL reduced basis all have a special structure.

In the following description of these polynomials $r_i$ we have replaced $X_2$ by $X_1$ and $X_4$ by $X_3$, since the upper bounds that we have for $x_1^{(0)} = d_p$ and $x_2^{(0)} = d_q$ are the same, as are the upper bounds for $x_3^{(0)} = k_p$ and $x_4^{(0)} = k_q$.

$$
\begin{aligned}
r_i(x_1, x_2, x_3, x_4) := {} & a_1 X_1^4 X_3^2 \cdot f \;+\; a_2 X_1^3 X_3^2 \cdot (x_1 + x_2) \cdot f \;+\; a_3 X_1^4 X_3 \cdot (x_3 + x_4) \cdot f \\
& + a_4 X_1^3 X_3 \cdot (x_1 x_4 + x_2 x_3) \cdot f \;+\; a_5 X_1^2 X_3^2 \cdot x_1 x_2 \cdot f \;+ \\
& + a_6 X_1^4 \cdot x_3 x_4 \cdot f \;+\; a_7 X_1^2 X_3^2 \cdot (x_1^2 + x_2^2) \cdot f \;+ \\
& + a_8 X_1^3 X_3 \cdot (x_1 x_3 + x_2 x_4) \cdot f \;+\; a_9 X_1^2 X_3 \cdot (x_1^2 x_4 + x_2^2 x_3) \cdot f \\
& + a_{10} X_1^2 X_3 \cdot (x_1 x_2 x_3 + x_1 x_2 x_4) \cdot f \;+\; a_{11} X_1 X_3^2 \cdot (x_1^2 x_2 + x_1 x_2^2) \cdot f \\
& + a_{12} X_1^3 \cdot (x_1 x_3 x_4 + x_2 x_3 x_4) \cdot f \;+\; a_{13} X_1^2 \cdot x_1 x_2 x_3 x_4 \cdot f \\
& + a_{14} X_1 X_3 \cdot (x_1^2 x_2 x_4 + x_1 x_2^2 x_3) \cdot f \;+\; a_{15} X_3^2 \cdot (x_1^2 x_2^2) \cdot f \\
& - \left\lfloor \frac{\text{sum of coefficients of } x_3^2 \text{ in the above polynomials}}{W X_1^4 X_3^2} \right\rfloor W X_1^4 X_3^2 \cdot x_3^2 \\
& - \ldots \\
& - \left\lfloor \frac{\text{sum of coefficients of } x_1^3 x_2^3 \text{ in the above polynomials}}{W X_1^4 X_3^2} \right\rfloor W X_1^4 X_3^2 \cdot x_1^3 x_2^3 .
\end{aligned}
$$

Two plain observations are:

- Due to the symmetry of

$$f(x_1, x_2, x_3, x_4) := (ex_1 + x_3 - 1)(ex_2 + x_4 - 1) - Nx_3x_4$$
$$= e^2x_1x_2 + e(x_1x_4 + x_2x_3 - x_1 - x_2) + (1 - N)x_3x_4 - x_3 - x_4 + 1,$$

  the polynomials corresponding to the smallest vectors are also very symmetrical.

- The rows corresponding to the polynomials

$$Rx_3^2,\ Rx_2x_3^2,\ Rx_3^2x_4,\ Rx_4^2,\ Rx_1x_4^2,\ Rx_3x_4^2,\ Rx_1^2x_4^2,\ Rx_2^2x_3^2,\ Rx_1x_3^2,\ Rx_2x_4^2,$$

  are *not* used in the smallest vectors. This is because the term

$$\left\lfloor \frac{\text{sum of coefficients of } x_3^2 \text{ in the above polynomials}}{WX_1^4X_3^2} \right\rceil$$

  is zero, and equivalently for $x_2x_3^2, \ldots, x_2x_4^2$.

Whether or not these observations can be used to obtain a better attack bound, is the open question with which we conclude this thesis.

*Open question: Can we use the fact that the 'smallest' polynomials are symmetrical, and that a few rows of the original matrix are not necessary for LLL to create these 'small' polynomials, to improve the attack bound on CRT-Small-$d_p, d_q$?*

A possible direction to take in order to answer this question is to look at the following sublattice of $L$. Replace the two rows that represent the shifts polynomials

$$x_1 \cdot f \cdot X_1^{(m-1)+t-1}X_2^{(m-1)+t}X_3^{m-1}X_4^{m-1} \quad \text{and} \quad x_2 \cdot f \cdot X_1^{(m-1)+t}X_2^{(m-1)+t-1}X_3^{m-1}X_4^{m-1}$$

by one row that contains their sum. Since we know that the shortest vectors in the LLL reduced basis take the shifts of $x_1$ and $x_2$ with the same coefficient, the sum of the two shifts can be used instead of two single shifts. The same holds for the shifts of $x_3$ and $x_4$, the shifts of $x_1x_4$ and $x_2x_3$, etc. All of these couples of shifts can be replaced by their sum.

In the example we gave for the new RSA-CRT-Small-$d_p, d_q$ attack (for $m = 2, t = 1$), our sublattice $L'$ looks like:

$$\begin{aligned}
&\text{shift } 1: && f \cdot X_1^4X_3^2,\\
&\text{shift } 2: && (x_1 + x_2) \cdot f \cdot X_1^3X_3^2,\\
&\text{shift } 3: && (x_3 + x_4) \cdot f \cdot X_1^4X_3^2,\\
&\ldots\\
&\text{shift } 14: && (x_1^2x_2x_4 + x_1x_2^2x_3) \cdot f \cdot X_1X_3,\\
&\text{shift } 15: && x_1^2x_2^2 \cdot f \cdot X_3^2,
\end{aligned}$$

together with the rows for the monomials that are in $M \backslash S$:

$$\begin{aligned}
&\text{shift } 16: \quad (x_3^2 + x_4^2)R, \\
&\text{shift } 17: \quad (x_2x_3^2 + x_1x_4^2)R, \\
&\text{shift } 18: \quad (x_3^2x_4 + x_3x_4^2)R, \\
&\ldots \\
&\text{shift } 44: \quad (x_1^3x_2^2x_4 + x_1^2x_2^3x_3)R, \\
&\text{shift } 45: \quad x_1^3x_2^3R.
\end{aligned}$$

In this example, the lattice $L'$ is defined by 45 polynomials with in total 81 monomials. The short vectors of $L$ that were found by the LLL reduction are also in $L'$. In this specific case, we even know that we can eliminate five rows more because we know that the shifts $(x_3^2 + x_4^2)RX_1^4$, etc. do not appear in the smallest reduced lattice vectors of $L$. The determinant of the 45-dimensional $L'$ of the example is given by

$$\det(L') = 2^{18}(X_1X_2)^{128}(X_3X_4)^{74}W^{30}, \text{ so } (\det(L'))^{\frac{1}{45}} \approx (X_1X_2)^{2.844}(X_3X_4)^{1.644}W^{0.667}.$$

This is only slightly smaller than the value of

$$(\det(L))^{\frac{1}{81}} = (X_1X_2)^{\frac{232}{81}}(X_3X_4)^{\frac{134}{81}}W^{\frac{56}{81}} \approx (X_1X_2)^{2.864}(X_3X_4)^{1.654}W^{0.691}.$$

Substituting the values of the $X_i$, $W$, and $R$ in the respective inequalities $\det(L')^{\frac{1}{45}} < R$ and $\det(L)^{\frac{1}{81}} < R$ yields the bounds $\beta < 0.0096$ and $\beta < -0.0153$.

The determinant of the 40-dimensional lattice $L''$ is a more complicated expression that involves many terms in the $X_i$, $W$, $e$ and $N$. When we substitute all known values in a simplified version of the expression $\det(L'')^{\frac{1}{40}} < R$, we find the bound $\beta < 0.0102$. This bound is supported by the experiments in Section 5.3.4, where we can see in Table 5.3 that for $\{m = 2, t = 1\}$, we can successfully mount an attack if the length of the private CRT-exponents is about $\frac{1}{100}n$.

At this moment, it is not yet clear if and how well this approach would work in general (especially for $m \to \infty$ with optimal $t$). We do not have a general analysis of the determinant of the lattice $L'$ (which does not have full rank anymore, so this determinant is the volume of the lattice). Trying this and other approaches to improve the attack bound of RSA-CRT-Small-$d_p, d_q$ will be one of the many challenges in the fascinating research area of cryptanalysis of RSA variants using small roots of polynomials.

# Bibliography


[1] A. Baker and H. Davenport. The equations $3x^2 - 2 = y^2$ and $8x^2 - 7 = z^2$. *Quarterly Journal of Mathematics (Oxford) (2)*, volume 20: pages 129–137, 1969.

[2] A. Bauer and A. Joux. Toward a rigorous variation of Coppersmith's algorithm on three variables. In *Proceedings of Eurocrypt'07*, volume 4515 of *Lecture Notes in Computer Science*, pages 361–378, 2007.

[3] D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *Proceedings of Crypto'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 1–12, 1998.

[4] D. Bleichenbacher and A. May. New attacks on RSA with small secret CRT-exponents. In *Proceedings of PKC'06*, volume 3958 of *Lecture Notes in Computer Science*, pages 1–13, 2006.

[5] J. Blömer and A. May. Low secret exponent RSA revisited. In *Proceedings of CaLC'01*, volume 2146 of *Lecture Notes in Computer Science*, pages 4–19, 2001.

[6] J. Blömer and A. May. New partial key exposure attacks on RSA. In *Proceedings of Crypto'03*, volume 2729 of *Lecture Notes in Computer Science*, pages 27–43, 2003.

[7] J. Blömer and A. May. A tool kit for finding small roots of bivariate polynomials over the integers. In *Proceedings of Eurocrypt'05*, volume 3494 of *Lecture Notes in Computer Science*, pages 251–267, 2005.

[8] D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, volume 46(2): pages 203–213, 1999.

[9] D. Boneh, R.A. DeMillo, and R.J. Lipton. On the importance of checking cryptographic protocols for faults. In *Proceedings of Eurocrypt'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51, 1997.

[10] D. Boneh and G. Durfee. Cryptanalysis of RSA with private key $d$ less than $N^{0.292}$. *IEEE Transactions on Information Theory*, volume 46(4): pages 1339–1349, 2000.

[11] D. Boneh, G. Durfee, and Y. Frankel. Exposing an RSA private key given a small fraction of its bits, full version of [12]. http://crypto.stanford.edu/∼dabo/abstracts/bits_of_d.html.

[12] D. Boneh, G. Durfee, and Y. Frankel. Exposing an RSA private key given a small fraction of its bits. In *Proceedings of Asiacrypt'98*, volume 1514 of *Lecture Notes in Computer Science*, pages 25–34, 1998.

[13] D. Boneh, G. Durfee, and N. Howgrave-Graham. Factoring $n = p^r q$ for large $r$. In *Proceedings of Crypto'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 326–337, 1999.

[14] D. Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In *Proceedings of Eurocrypt'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 178–189, 1996.

[15] D. Coppersmith. Finding a small root of a univariate modular equation. In *Proceedings of Eurocrypt'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 155–165, 1996.

[16] D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, volume 10(4): pages 233–260, 1997.

[17] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter. Low exponent RSA with related messages. In *Proceedings of Eurocrypt'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 1–9, 1996.

[18] J.-S. Coron. Finding small roots of bivariate integer polynomial equations revisited. In *Proceedings of Eurocrypt'04*, volume 3027 of *Lecture Notes in Computer Science*, pages 492–505, 2004.

[19] J.-S. Coron. Finding small roots of bivariate integer polynomial equations: a direct approach. In *Proceedings of Crypto'07*, volume 4622 of *Lecture Notes in Computer Science*, 2007.

[20] J.-S. Coron and A. May. Deterministic polynomial-time equivalence of computing the RSA secret key and factoring. *Journal of Cryptology*, 20: pages 39–50, 2007.

[21] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms. An introduction to computational algebraic geometry and commutative algebra.* Undergraduate texts in mathematics. Springer, 1992.

[22] J.-F. Dhem, F. Koeune, P.-A. Leroux, P. Mestre, J.-J. Quisquater, and J.-L. Willems. A practical implementation of the timing attack. In *Proceedings of CARDIS'98*, volume 1820 of *Lecture Notes in Computer Science*, pages 167–182, 1998.

[23] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, volume 22(6): pages 644–654, 1976.

[24] ECRYPT-AZTEC. Hardness of the main computational problems used in cryptography. IST-2002-507932, *http://www.ecrypt.eu.org/documents/D.AZTEC.4-1.1.pdf.*

[25] M. Ernst, E. Jochemsz, A. May, and B.M.M. de Weger. Partial key exposure attacks on RSA up to full size exponents. In *Proceedings of Eurocrypt'05*, volume 3494 of *Lecture Notes in Computer Science*, pages 371–386, 2005.

[26] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases. *Journal of Pure and Applied Algebra*, 139: pages 61–88, 1999.

[27] S.D. Galbraith, C. Heneghan, and J.F. McKee. Tunable balancing of RSA, full version of [28]. http://www.isg.rhul.ac.uk/∼sdg/full-tunable-rsa.pdf.

[28] S.D. Galbraith, C. Heneghan, and J.F. McKee. Tunable balancing of RSA. In *Proceedings of ACISP'05*, volume 3574 of *Lecture Notes in Computer Science*, pages 280–292, 2005.

[29] K.O. Geddes, S.R. Czapor, and G. Labahn. *Algorithms for computer algebra*. Kluwer Academic Publishers, 1992.

[30] G.H. Hardy and E.M. Wright. *An introduction to the theory of numbers*. Oxford University Press, 1979.

[31] J. Håstad. Solving simultaneous modular equations of low degree. *SIAM Journal on Computing*, volume 17(2): pages 336–341, 1988.

[32] M.J. Hinek. Another look at small RSA exponents. In *Proceedings of CT-RSA'06*, volume 3860 of *Lecture Notes in Computer Science*, pages 82–98, 2006.

[33] M.J. Hinek. On the security of multi-prime RSA. Technical report, CACR, 2006.

[34] M.J. Hinek and Douglas R. Stinson. An inequality about factors of multivariate polynomials. Technical report, CACR, 2006.

[35] N. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Proceedings of IMA Int. Conf.*, volume 1355 of *Lecture Notes in Computer Science*, pages 131–142, 1997.

[36] E. Jochemsz and B.M.M. de Weger. A partial key exposure attack on RSA using a 2-dimensional lattice. In *Proceedings of ISC'06*, volume 4176 of *Lecture Notes in Computer Science*, pages 203–216, 2006.

[37] E. Jochemsz and A. May. A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. In *Proceedings of Asiacrypt'06*, volume 4284 of *Lecture Notes in Computer Science*, pages 267–282, 2006.

[38] E. Jochemsz and A. May. A polynomial time attack on RSA with private CRT-exponents smaller than $N^{0.073}$. In *Proceedings of Crypto'07*, volume 4622 of *Lecture Notes in Computer Science*, 2007.

[39] A. Joux and F. Olivier. Side-channel analysis. In *Encyclopedia of Cryptography and Security*, pages 571–576. Springer, 2005.

[40] C.S. Jutla. On finding small solutions of modular multivariate polynomial equations. In *Proceedings of Eurocrypt'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 158–170, 1998.

[41] D. Kahn. *The codebreakers: the story of secret writing.* MacMillan, 1967.

[42] P.C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Proceedings of Crypto'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113, 1996.

[43] P.C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Proceedings of Crypto'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397, 1999.

[44] J.L. Lagrange. *Recherches d'arithmétique.* Nouveaux Mémoires de l'Académie Royale des Sciences et des Belles Lettres de Berlin, 1773.

[45] A.-M. Legendre. *Essai sur la théorie des nombres.* Duprat, Paris, An VI, 1798.

[46] A.K. Lenstra. Integer factoring. In *Encyclopedia of Cryptography and Security*, pages 290–297. Springer, 2005.

[47] A.K. Lenstra and H.W. Lenstra Jr. (eds.). The development of the number field sieve. *Lecture Notes in Mathematics*, volume 1554, 1993.

[48] A.K. Lenstra, H.W. Lenstra Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, volume 261(4): pages 515–534, 1982.

[49] H.W. Lenstra Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, volume 126(2): pages 649–673, 1987.

[50] C.H. Lim and P.J. Lee. Security and performance of server-aided RSA computation protocols. In *Proceedings of Crypto'95*, volume 963 of *Lecture Notes in Computer Science*, pages 70–83, 1995.

[51] A. May. RSA & meet-in-the-middle Angriffe. Chapter from the course "Public Key Kryptanalyse", available via http://www.informatik.tu-darmstadt.de/KP/lehre/ws0506/vl/pkk.html.

[52] A. May. Cryptanalysis of unbalanced RSA with small CRT-exponent. In *Proceedings of Crypto'02*, volume 2442 of *Lecture Notes in Computer Science*, pages 242–256, 2002.

[53] A. May. *New RSA Vulnerabilities Using Lattice Reduction Methods.* PhD thesis, University of Paderborn, 2003.

[54] A. May. Computing the RSA secret key is deterministic polynomial time equivalent to factoring. In *Proceedings of Crypto'04*, volume 3152 of *Lecture Notes in Computer Science*, pages 213–219, 2004.

[55] A. May. Secret exponent attacks on RSA-type schemes with moduli $p^r q$. In *Proceedings of PKC'04*, volume 2947 of *Lecture Notes in Computer Science*, pages 218–230, 2004.

[56] J.F. McKee and R. Pinch. Further attacks on server-aided RSA cryptosystems. http://citeseer.ist.psu.edu/388295.html, 1998.

[57] H. Minkowski. *Geometrie der Zahlen*. Teubner Verlag, 1896.

[58] P.Q. Nguyen and D. Stehlé. Floating-point LLL revisited. In *Proceedings of Eurocrypt'05*, volume 3494 of *Lecture Notes in Computer Science*, pages 215–233, 2005.

[59] P.Q. Nguyen and J. Stern. The two faces of lattices in cryptology. In *Proceedings of CaLC'01*, volume 2146 of *Lecture Notes in Computer Science*, pages 146–180, 2001.

[60] G. Qiao and K.-Y. Lam. RSA signature algorithm for microcontroller implementation. In *Proceedings of CARDIS'98*, volume 1820 of *Lecture Notes in Computer Science*, pages 353–356, 1998.

[61] J.-J. Quisquater and C. Couvreur. Fast decipherment algorithm for RSA public-key cryptosystem. *Electronic Letters*, volume 18: pages 905–907, 1982.

[62] J.-J. Quisquater and F. Koeune. Side channel attacks, state of the art. Available via http://www.crypto.rub.de/en_sclounge.html, 2002.

[63] D. Redmond. *Number theory: an introduction*, volume no. 201 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker, 1996.

[64] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, volume 21(2): pages 120–126, 1978.

[65] W. Scharlau and H. Opolka. *From Fermat to Minkowski. Lectures on the theory of numbers and its historical development*. Undergraduate texts in mathematics. Springer-Verlag, 1985.

[66] V. Shoup. NTL: A library for doing number theory. http://www.shoup.net/ntl.

[67] S. Singh. *The code book: the science of secrecy from ancient Eqypt to quantum cryptography*. Anchor books, 2000.

[68] R. Steinfeld and Y. Zheng. An advantage of low-exponent RSA with modulus primes sharing least significant bits. In *Proceedings of CT-RSA'01*, volume 2020 of *Lecture Notes in Computer Science*, pages 52–62, 2001.

[69] H.-M. Sun, M.J. Hinek, and M.-E. Wu. On the design of rebalanced RSA-CRT, revised version of [70]. Technical report, CACR, 2005.

[70] H.-M. Sun and M.-E. Wu. An approach towards RSA-CRT with short public exponent. http://eprint.iacr.org/2005/053, 2005.

[71] T. Takagi. Fast RSA-type cryptosystem modulo $p^k q$. In *Proceedings of Crypto'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 318–326, 1998.

[72] E.R. Verheul and H.C.A. van Tilborg. Cryptanalysis of 'less short' RSA secret exponents. *Applicable Algebra in Engineering Communication and Computing*, volume 8(5): pages 425–435, 1997.

[73] P.S. Wang and L.P. Rothschild. Factoring multivariate polynomials over the integers. *ACM SIGSAM Bulletin*, issue 28: pages 21–29, 1973.

[74] B.M.M. de Weger. Cryptanalysis of RSA with small prime difference. *Applicable Algebra in Engineering, Communication and Computing*, volume 13(1): pages 17–28, 2002.

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

# Index

# Nederlandse samenvatting

Cryptologie is een onderzoeksgebied dat o.a. het vercijferen en ontcijferen van berichten omvat. 'Public key' cryptografie is een tak van cryptografie waarin we cryptosystemen onderzoeken waarbij elke gebruiker een publieke en een geheime sleutel heeft, waarbij de geheime sleutel niet af te leiden is uit de publieke sleutel. Met zijn geheime sleutel kan een gebruiker berichten ontcijferen of een digitale handtekening zetten, terwijl anderen met de publieke sleutel berichten kunnen vercijferen of handtekeningen kunnen verifiëren. Dit heeft veel toepassingen, met name op het gebied van internetcommunicatie. Het systeem RSA was in 1978 het eerste voorstel voor een public key cryptosysteem, en tot op de dag van vandaag is het nog altijd het populairste public key cryptosysteem.

RSA heeft de volgende parameters. De RSA-modulus $N$ (publiek) is het product van twee grote priemgetallen $p$ en $q$ (geheim). Verder is er een publieke sleutel $e$ en een geheime sleutel $d$. Deze $d$ is gemakkelijk uit $N$ en $e$ af te leiden *mits* de factorisatie van $N$ bekend is. Dit komt omdat $e$ en $d$ voldoen aan de relatie

$$ed \equiv 1 \mod (p-1)(q-1).$$

Als men RSA wil breken (oftewel, de geheime $(d, p, q)$ achterhalen als $(N, e)$ bekend is), kan men natuurlijk proberen om $N$ te factoriseren. Maar aangezien er geen factorisatiemethoden bekend zijn met een looptijd die polynomiaal is in de bitlengte van $N$, lukt dit niet als de priemgetallen $p$ en $q$ groot genoeg gekozen zijn.

Desondanks blijkt het mogelijk om $N$ te factoriseren in polynomiale tijd als we wat extra informatie krijgen over de geheime parameters. Het is bijvoorbeeld mogelijk dat een aanvaller een deel van de bits van $d$ heeft achterhaald door middel van een zogenaamde 'side channel attack'. Een ander voorbeeld waarin een aanvaller extra informatie heeft is als hij weet dat de parameters van het RSA-systeem dat hij aanvalt op een bepaalde manier gekozen zijn. In sommige RSA-voorstellen worden bijvoorbeeld priemgetallen gebruikt die aan een speciale relatie voldoen, of sleutels $e$ en $d$ gekozen die relatief 'klein' zijn.

In dit proefschrift bespreken we bekende en nieuwe aanvallen op verscheidene RSA-varianten. De aanvallen komen voort uit het feit dat de speciale relaties tussen de parameters $N$, $p$, $q$, $e$ en $d$ vertaald kunnen worden in een veelterm met een 'klein' nulpunt. Als dit kleine nulpunt gevonden kan worden dan vinden we direct de geheime informatie $(d, p, q)$, en hebben we het betreffende RSA-systeem gebroken.

Dit brengt ons bij de theorie van het vinden van nulpunten. In 1996 introduceerde Don Coppersmith methodes om 'kleine' nulpunten $x_0$ te vinden van een veelterm $f(x)$ modulo $N$, en om 'kleine' geheeltallige nulpunten $(x_0, y_0)$ te vinden van een veelterm $f(x, y)$. Voor de veeltermen in onze aanvallen hebben we soms uitbreidingen van deze methodes naar meer variabelen nodig. Bij deze uitbreidingen worden technieken als resultanten, Gröbner bases en reductie van roosterbases gebruikt. Ook op het gebied van het vinden van kleine nulpunten is vooruitgang geboekt: in dit proefschrift is een algemene strategie opgenomen die gevolgd kan worden voor een willekeurige veelterm $f$. Ook is uit deze strategie makkelijk af te leiden hoe klein een bepaald nulpunt van een gegeven veelterm precies moet zijn, om het te kunnen vinden in polynomiale tijd.

# Summary: Cryptanalysis of RSA variants using small roots of polynomials

Cryptology is a research area that includes (among other things) methods to encipher and decipher messages. In the area of public key cryptography, we examine cryptosystems in which every user has a public key and a private key, where the private key cannot be derived from the public key. With the private key, the user can decrypt messages or produce a digital signature, whereas the public key can be used by everyone to encrypt messages, or verify that a signature is correct. These public key schemes have many applications, especially in the area of safe internet communication. The RSA cryptosystem, published in 1978, was the first proposal for a multipurpose public key cryptosystem, and has remained the most popular public key cryptosystem ever since.

The RSA cryptosystem has the following parameters. The RSA modulus $N$ (public) is the product of two large prime numbers $p$ and $q$ (secret). Moreover, there is a public key $e$ and a secret key $d$. This $d$ can easily be derived from $e$ and $N$ *provided that* the factorization of $N$ is known, because $e$ and $d$ satisfy the relation

$$ed \equiv 1 \mod (p-1)(q-1).$$

If one wants to break RSA (that is, retrieve the secret information $(d, p, q)$ from a given $(N, e)$), one could try to factor $N$. However, since we do not know factorization methods with a running time that is polynomial in the bitsize of $N$, these factorization methods will not succeed if the prime numbers $p$ and $q$ are chosen large enough.

Nevertheless, it turns out to be possible to factor $N$ in polynomial time if we get some extra information on the secret parameters $d$, $p$, and $q$. It is for instance possible that an attacker has obtained a part of the bits of $d$ by performing a so-called 'side-channel attack'. Another example in which an attacker has extra information is when it is known that the parameters of the system are chosen in a special way. In some RSA proposals, primes are used that satisfy special relations. In other RSA variants, $e$ or $d$ is chosen to be relatively 'small'.

In this thesis we discuss many known and new attacks on several RSA variants. The attacks arise from the fact that the special relation between the parameters $N$, $p$, $q$, $e$, and $d$ can be translated into a polynomial with a 'small' root. If the root can be found, then one immediately finds the secret information $(d, p, q)$, thereby breaking this RSA variant.

This brings us to the theory of finding small roots. In 1996 Don Coppersmith introduced methods to find 'small' roots $x_0$ of a polynomial $f(x)$ modulo $N$, and to find 'small' integer roots $(x_0, y_0)$ of a polynomial $f(x, y)$. For the polynomials in our attacks we sometimes need extensions of these methods to more variables. In these extensions, techniques like resultants, Gröbner bases, and lattice basis reduction are used. In the area of finding small roots we have also made progress: in this thesis one can find a general strategy that can be followed for an arbitrary polynomial $f$. From the strategy it is easy to derive a bound that describes how small a root of the given polynomial should be, such that it can be found in polynomial time.

# Acknowledgments

This thesis could not have been made without the help and support of many people.

First of all, I owe a great deal to my supervisors, Henk van Tilborg and Benne de Weger. Henk initiated the NWO project"Key Issues in Cryptology" for which I was hired, and has followed my research and coached my development as a researcher from the start. Besides that he is mostly responsible for the great working environment in the Coding theory & Cryptology group of the TU Eindhoven. Benne has been a great, enthusiastic 'daily supervisor'. I am very thankful that he chose RSA cryptanalysis as a suitable topic for my research, a topic that captured me and provided me with many possibilities to find new results. I always found his door open to discuss the many research issues that bothered me, and left his room again with new ideas to process.

Besides working with Benne, I obtained most of my research results in cooperation with Alexander May. I have been very lucky to work together with Alex, and was able to profit from his expertise (I hardly traveled anywhere without his thesis in my bag). Besides his knowledge of the subject, I also learnt from him important skills like writing good introductions and programming the experiments.

The remaining members of my Ph.D. committee are professors Ronald Cramer, Tanja Lange, Arjen Lenstra, Eric Verheul, and the chairman, prof. Kees van Hee. Their work in reading my thesis and commenting on it is greatly appreciated. Arjen already influenced my work in a positive way in an earlier stage, by suggesting changes to the papers I submitted, and by commenting on the direction in which my research should go.

Other people who have read my thesis and have given me valuable comments are Reinier Bröker, Mehmet Kiraz, and Joris De Kaey.

Throughout my short scientific career I have had the opportunity to discuss with many other researchers about my topic. I would specifically like to thank Steven Galbraith, Jason Hinek, Phong Nguyen, Jean-Sébastien Coron, Claus Diem, Aurelie Bauer, and the many people in the ECRYPT-AZTEC Cryptanalysis group for the helpful discussions. The existence of the ECRYPT-labs, summerschools, etc. have made it very easy to learn from other researchers, ranging from professors to other Ph.D. students, and this has been very valuable to me. The same statement holds for the activities, meetings, and minicourses organized by the Dutch clusters EIDMA and DIAMANT. I'm grateful for the stipends that I received from the IACR in order to be able to attend and give presentations at Eurocrypt'05, Asiacrypt'06, and Crypto'07.

I have already mentioned the pleasant working atmosphere inside the Coding theory & Cryptology group in Eindhoven. Discussion sessions with Henk, Benne, Berry, Ruud, Tanja, and Dan, ('the supervisors'), and Andrey, Mehmet, Reza, José, Peter, and Christiane ('the students') were a nice way to think about the open research problems and the possibilities to pursue, and I liked learning about the other students' research topics in the process. Mehmet has been a great 'roommate' at the university, and I enjoyed his company throughout the last $3\frac{1}{2}$ years. Bram, Wil, Anita and Henny completed this great group, with tea&cake-meetings on Tuesdays, outings, and the yearly dinner at Henk's.

As a nice bonus, we share our floor in the building with the members of Arjeh Cohen's Discrete Algebra and Geometry group, and I spent many nice lunches with Tim, Jos, Erik, Rikko, Mark, Jan-Willem, Tyrell, Dan, and Jan (in order of pick-up). The contact with these people and their knowledge of very hard mathematics gave me the opportunity to ask silly questions sometimes and get decent answers in return.

From the DAG-students, I would specifically like to thank Tim for his friendship (and for introducing me to a fellow student), and Rikko for sharing my NS-disasters.

During the organization of the Studygroup Mathematics with Industry, a yearly event that I would recommend to any mathematician, it was a good experience to work together with Mark, Jaap, Remco, Erik, Georg, and Tim.

From the people that were really involved in the work related to my thesis, to the people that contributed to a good working environment, we now come to the people that have nothing to do with my university life but are of great importance to me all the same.

Most importantly, I've had an enormous support from my family, that is, my parents and my sister Mieke, in the last 4 years. I'm extremely happy to have such a great home base to rely on, and I owe them for everything that I have achieved in my life.

I'm thankful for my friends for the many enjoyable moments, and for understanding my full schedule in the last year.

And last but not least, I've had the luck to meet Joris, who has given me his love and support, a new and welcoming group of friends and family in Belgium, a funny accent, and so much more...

# Curriculum Vitae

Ellen Jochemsz was born on June 23, 1980 in Rijnsburg, The Netherlands.

In 1997, she graduated from the secondary school Northgo in Noordwijk, and started her studies in mathematics at the Vrije Universiteit Amsterdam, where she finished her masters in 2002 with specialization 'discrete mathematics, coding theory and cryptology'. Her masters thesis was titled 'Goppa codes and the McEliece cryptosystem', and was supervised by dr. Evert Wattel. During her studies, she worked as a teaching assistant, giving instructions for several mathematics courses at the university.

From 2002 to 2003, she worked as a teacher in mathematics at the Fons Vitae Lyceum in Amsterdam. In this period, she completed the universitary teachers program, and obtained her diploma for teaching mathematics at any level of the secondary school.

From 2003 to 2007, she was a Ph.D. student in the Coding theory & Cryptology group of the Technische Universiteit Eindhoven, under the supervision of prof. Henk van Tilborg and dr. Benne de Weger. The present thesis is the result of her work in this period.

Her research interests are in the area of applied mathematics, in particular: discrete mathematics, cryptology, coding theory, and discrete optimization.