The Mathematical Cryptography of the RSA Cryptosystem

Abderrahmane Nitaj

Laboratoire de Mathématiques Nicolas Oresme
Université de Caen, France

abderrahmane.nitaj@unicaen.fr

http://www.math.unicaen.fr/~nitaj

Abstract

Invented in 1977 by Rivest, Shamir and Adleman, the RSA cryptosystem has played a very important role in the development of modern cryptography. Its various applications in industry, Internet, banking, online shopping, cell phones, smart cards, secure information transfers and electronic signatures have made RSA a standard at the heart of modern technologies. This chapter explores the mathematics behind the RSA cryptosystem including the encryption, decryption and signature schemes of RSA. We give a survey of the main methods used in attacks against the RSA cryptosystem. This includes the main properties of the continued fraction theory, lattices, the LLL algorithm of Lenstra, Lenstra and Lovász and the lattice reduction based technique of Coppersmith for solving modular polynomial equations.

1 Introduction

The concept of the public-key cryptosystem was proposed by Diffie and Hellman [5] in 1976. Since then, a number of public-key cryptosystems have been proposed to realize the notion of public-key cryptosystems. At the moment some of them are present in industrial standards. In 1977, Ronald Rivest, Adi Shamir and Leonard Adleman [10] proposed a scheme which became the most widely used asymmetric cryptographic scheme, RSA. For instance, the RSA public-key cryptosystem is used for securing web traffic, e-mail, remote login sessions, and electronic credit card payment systems. The underlying one-way function of RSA is the integer factorization problem: Multiplying two large primes is computationally easy, but factoring the resulting product is very hard. It is also well known that the security of RSA is based on the difficulty of solving the so-called RSA problem: Given an RSA public key (e, N) and a ciphertext $c \equiv m^e \pmod N$, compute the plaintext m. The RSA problem is not harder to solve than the integer factorization problem, because factoring the RSA modulus N leads to computing the private exponent d, and to solving the RSA problem. However, it is not clear, if the converse is true.

In the RSA cryptosystem, the public modulus $N = pq$ is a product of two primes of the same bit size. The public and private exponent e and d satisfy the congruence

$$ed \equiv 1 \pmod{\phi(N)},$$

where $\phi(N) = (p-1)(q-1)$ is the Euler totient function. Encryption, decryption, signature and signature-verification in RSA require the computation of heavy exponentiations. To reduce the encryption time or the signature-verification time, one can use a small public exponent e such as 3 or $2^{16}+1$. On the other hand, to reduce the decryption time or the signature-generation time, one can be tempted to use a small private exponent d. Many attacks show that using a very small private exponent is insecure. Indeed, Wiener [12] showed in 1990 how to break RSA when $d < N^{0.25}$ using Diophantine approximations. The bound was improved by Boneh and Durfee [2] in 1999 to $d < N^{0.292}$ using Coppersmith's lattice reduction based method [4].

In this chapter, we survey the state of research on RSA cryptography. We start from reviewing the basic concepts of RSA encryption, decryption, signature and signature-verification schemes, and subsequently review some algebraic attacks on RSA using elementary methods as well as tools from the theory of continued fractions and lattices. This includes the lattice reduction algorithm LLL of Lenstra, Lenstra and Lovász [8] and the technique of Coppersmith for solving univariate modular polynomial equations [4].

The rest of the paper is structured as follows. In section 2 we will introduce the basic mathematics behind the RSA cryptosystem including the encryption, decryption and signature schemes as well as some elementary attacks on the RSA cryptosystem. In Section 3, we review the theory of the continued fractions and present two applications in the cryptanalysis of RSA. In Section 4, we focus on lattices and their reduction using the LLL algorithm and review Coppersmith's method for finding small modular roots of univariate polynomial equations and some applications in the cryptanalysis of RSA. We conclude in section 5.

2 The Mathematics of the RSA Cryptosystem

2.1 The basic mathematics

The elementary arithmetic of the RSA cryptosystem is based on the rings $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{Z}_N = \mathbb{Z}/N\mathbb{Z}$.

Definition 2.1 (Division Algorithm for Integers)**.** Let $a, b \in \mathbb{Z}$ with $b > 1$. Then there exist unique $q, r \in \mathbb{Z}$ such that

$$a = bq + r, \qquad 0 \leq r < b.$$

If $r = 0$, we say that b divides a and denote this by $b|a$.

Definition 2.2 (Greatest common divisor)**.** Let $a, b \in \mathbb{Z}$. A positive integer d is the greatest common divisor of a and b if

1. $d|a$ and $d|b$,
2. if c is a positive integer satisfying $c|a$ and $c|b$, then $c|d$.

The greatest common divisor of a and b is denoted by $\gcd(a, b)$.

Primality and Coprimality play a central role in the arithmetic of the RSA cryptosystem.

Definition 2.3 (Prime Integer)**.** An integer $p \geq 2$ is said to be prime if its only positive divisors are 1 and p.

Definition 2.4 (Relatively Prime Integers)**.** Two integers a and b are said to be relatively prime or coprime if $\gcd(a, b) = 1$.

Definition 2.5 (RSA Modulus)**.** Let p and q be large prime numbers such that $p \neq q$. The product $N = pq$ is called an RSA modulus.

In the most standards of RSA, the modulus is a large integer of the shape $N = pq$ where p and q are large primes of the same bit-size. It is clear that the most direct method of breaking RSA is to factor the RSA modulus N. Consequently, the security of RSA is mainly based on the difficulty of factoring large integers.

Theorem 2.1 (The Fundamental Theorem of Arithmetic)**.** *Given a positive integer $n \geq 2$, the prime factorization of n is written*

$$n = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k} = \prod_{i=1}^{k} p_i^{a_i},$$

where $p_1, p_2, \cdots, p_k$ are the k distinct prime factors of n, each of order $a_i \geq 1$. Furthermore, the factorization is unique.

A very important number theoretical function in the RSA cryptosystem is the Euler totient function.

Definition 2.6 (The Euler Totient Function)**.** Given a positive integer $n \geq 2$, the Euler totient function $\phi(n)$ is defined by

$$\phi(n) = \#\mathbb{Z}_n^* = \#\left\{a, \quad 0 < a < n, \quad \gcd(a,n) = 1\right\}.$$

The set $\mathbb{Z}_n^*$ is called the group of units modulo n.

It is easy to see that $\phi(p) = p - 1$ whenever p is prime. The Euler totient function has many useful properties.

Theorem 2.2. *Let m and n two positive integers such that* $\gcd(m,n) = 1$. *Then*

$$\phi(mn) = \phi(m)\phi(n).$$

Proof. Suppose that $\gcd(m,n) = 1$. Consider the map

$$\begin{array}{lcccccc} \pi: & \quad & \mathbb{Z}_{mn} & \longrightarrow & \mathbb{Z}_m & \times & \mathbb{Z}_n \\ & & [x]_{mn} & \longmapsto & ([x]_m & , & [x]_n), \end{array}$$

where $[x]_a$ denotes x modulo a. We want to show that π is bijective. Let $x, y \in \mathbb{Z}_{mn}$ such that $\pi(x) = \pi(y)$. Then

$$[x]_m = [y]_m \Leftrightarrow [x-y]_m = 0 \Leftrightarrow x - y \equiv 0 \pmod{m}.$$

Similarly, we get $x - y \equiv 0 \pmod{n}$. Since $\gcd(m,n) = 1$, this implies that $x - y \equiv 0 \pmod{mn}$. On the other hand, $|x-y| < mn$. Hence $x - y = 0$ and $x = y$. This shows that π is injective. To show that π is surjective, let $(a,b) \in \mathbb{Z}_m \times \mathbb{Z}_n$. Define $M \in \mathbb{Z}_n$, $N \in \mathbb{Z}_m$ and $x \in \mathbb{Z}_{mn}$ by

$$M \equiv m^{-1} \pmod{n}, \quad N \equiv n^{-1} \pmod{m}, \quad x \equiv aNn + bMm \pmod{mn}.$$

Then

$$\begin{aligned} x &\equiv aNn + bMm \equiv aNn \equiv a \pmod{m}, \\ x &\equiv aNn + bMm \equiv bMm \equiv b \pmod{n}. \end{aligned}$$

It follows that the map π is surjective and finally bijective. Moreover, we have $\gcd(a, mn) = 1$ if only if $\gcd(a,m) = 1$ and $\gcd(a,n) = 1$. This implies that

$$\pi\left(\mathbb{Z}_{mn}^*\right) = \mathbb{Z}_{mn}^* \times \mathbb{Z}_{mn}^*.$$

Then $\phi(mn) = \phi(m)\phi(n)$. □

Theorem 2.3. *Let p be a prime number and $e \geq 1$. Then*

$$\phi\left(p^e\right) = p^{e-1}(p-1).$$

Proof. We have

$$\begin{aligned}
\phi\left(p^e\right) &= \#\left\{a, \quad 0 < a < p^e, \quad \gcd(a, p^e) = 1\right\} \\
&= \#\left\{a, \quad 0 < a < p^e, \quad \gcd(a, p) = 1\right\} \\
&= p^e - \#\left\{a, \quad 0 < a < p^e, \quad \gcd(a, p) > 1\right\}.
\end{aligned}$$

Notice that $\#\left\{a, \quad 0 < a < p^e, \quad \gcd(a, p) > 1\right\}$ is the number of positive integers not exceeding p^e that are not coprime to p. Such integers are p, $2p$,..., $p^{e-1}p$. Hence

$$\phi\left(p^e\right) = p^e - p^{e-1} = p^{e-1}(p-1),$$

which terminates the proof. □

If the factorization of n is given, then $\phi(n)$ can be expressed as in the following theorem.

Theorem 2.4. *Let*

$$n = \prod_{i=1}^{k} p_i^{a_i},$$

be the factorization of $n \geq 2$. Then

$$\phi(n) = \prod_{i=1}^{k} p_i^{a_i - 1}(p_i - 1).$$

Proof. Using Theorem 2.2 and Theorem 2.3, we get

$$\phi(n) = \phi\left(\prod_{i=1}^{k} p_i^{a_i}\right) = \prod_{i=1}^{k} \phi\left(p_i^{a_i}\right) = \prod_{i=1}^{k} p_i^{a_i - 1}(p_i - 1).$$

□

As we will see later, the decryption process of RSA is based on the following result.

Theorem 2.5 (Euler)**.** *Let n be a positive integer. If a is an integer such that* $\gcd(a, n) = 1$*, then*

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

Proof. Recall that $\phi(n) = \#\mathbb{Z}_n^*$ where

$$\mathbb{Z}_n^* = \left\{a_1 = 1 < a_2 < \cdots < a_{\phi(n)}\right\}.$$

Suppose that $\gcd(a, n) = 1$ and consider the set

$$\left\{aa_1 \pmod{n}, aa_2 \pmod{n}, \cdots, aa_{\phi(n)} \pmod{n}\right\}.$$

If $aa_i \equiv aa_j \pmod{n}$ for some i, j, then $a(a_i - a_j) \equiv 0 \pmod{n}$. Since $\gcd(a, n) = 1$, then $a_i - a_j \equiv 0 \pmod{n}$ and since $|a_i - a_j| < n$, then $a_i = a_j$. Hence

$$\left\{a_1, a_2, \cdots, a_{\phi(N)}\right\} = \left\{aa_1 \pmod{n}, aa_2 \pmod{n}, \cdots, aa_{\phi(N)} \pmod{n}\right\}.$$

Next, consider the product of the integers in both sides. We get

$$\prod_{i=1}^{\phi(n)} a_i = \prod_{i=1}^{\phi(n)} (aa_1 \pmod{n}) \equiv a^{\phi(n)} \prod_{i=1}^{\phi(n)} a_i \pmod{n}.$$

Since each a_i satisfies $\gcd(a_i, n) = 1$, then $\gcd\left(n, \prod_{i=1}^{\phi(n)} a_i\right) = 1$. Simplifying by $\prod_{i=1}^{\phi(n)} a_i$, we get

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

□

Let a, e and n be positive integers. A practical concern in implementing RSA and many cryptographic protocols is the computation of $a^e \pmod{n}$. Suppose that the binary representation of e is

$$e = \sum_{i=0}^{k} 2^i e_i, \quad e_i \in \{0, 1\}.$$

Then

$$a^e = \left(\cdots\left(\left(\left(a^{e_k}\right)^2 a^{e_{k-1}}\right)^2 a^{e_{k-2}}\right)^2 a^{e_{k-3}}\right)^2 \cdots\right)^2 a^{e_0}.$$

We summarize the modular exponentiation in Algorithm 1.

2.2 The basic RSA cryptosystem

The RSA cryptosystem was created in 1977 by Ronald Rivest, Adi Shamir and Leonard Adleman [10]. It has become fundamental to e-commerce and is widely used to secure communication in the Internet and ensure confidentiality and authenticity

Algorithm 1 Square-and-multiply algorithm for exponentiation in $\mathbb{Z}_n$

```
INPUT: $a \in \mathbb{Z}_n$ and an integer $0 < e < n$ whose binary representation is $e = \sum_{i=0}^{k} 2^i e_i$.
OUTPUT: $b \equiv a^e \pmod{n}$.
Set $b = 1$.
for $i$ from $k$ down to 0 do
    Compute $b \equiv b^2 \pmod{n}$.
    if $e_i = 1$ then
        Compute $b \equiv ba \pmod{n}$.
    end if
end for
Print $b$ and stop.
```

of e-mail. The RSA cryptosystem is based on the generation of two random primes, p and q, of equal bit-size and the generation of random exponents, d and e satisfying $ed \equiv 1 \pmod{\phi(N)}$ where $\phi(N) = (p-1)(q-1)$ is Euler's totient function. The RSA modulus N is the product $N = pq$. The pair n and e are made public and p, q, d are secret. The integer e is sometimes called the public exponent and d the private exponent. The generation process is illustrated in Algorithm 2. The pair (N, e) is often called the public key and (N, d) the private key.

Algorithm 2 : Standard RSA key generation

```
INPUT: A number $k$ of bits of the primes.
OUTPUT: A public key $(N, e)$ and a private key $(N, d)$.
Pick random primes $p$ and $q$ of bit-size $k$.
Set $N = pq$ and $\phi(N) = (p-1)(q-1)$.
repeat
    Pick a random integer $e < \phi(N)$,
until $\gcd(e, \phi(N)) = 1$.
Compute $d \equiv e^{-1} \pmod{\phi(N)}$.
Return $(N, e)$ and $(N, d)$.
```

Now, we describe the encryption, decryption and the signature schemes of the RSA cryptosystem.

- **RSA Encryption**
  INPUT: The public key (N, e) and the plaintext message m.
  OUTPUT: The cyphertext c.

  1. Represent the message as an integer $m < N$ such that $\gcd(m, N) = 1$.

2. Compute $c \equiv m^e \pmod{N}$.
3. Return c.

- **RSA Decryption**
  INPUT: The private key (N, d) and the cyphertext c.
  OUTPUT: The plaintext message m.

  1. Compute $m \equiv c^e \pmod{N}$.
  2. Return m.

- **RSA Signature**
  INPUT: The public key (N_A, e_A), the private key (N_B, d_B), and the plaintext message m.
  OUTPUT: The cyphertext c and the signature S.

  1. Compute $c \equiv m^{e_A} \pmod{N_A}$.
  2. Compute $S \equiv c^{d_B} \pmod{N_B}$.
  3. Return c and S.

- **RSA Signature Verification**
  INPUT: The private key (N_A, d_A), the public key (N_B, e_B), cyphertext c and the signature S.
  OUTPUT: The cyphertext c and the signature S.

  1. Compute $S' \equiv S^{e_B} \pmod{N_B}$.
  2. Return c and S'. The signature is verified if $S' = c$.

To show that encryption and decryption are inverse operations, recall that $ed \equiv 1 \pmod{\phi(N)}$. Therefore

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

for some positive integer k. Hence

$$c^d \equiv m^{ed} \equiv m^{1+k\phi(N)} = m \times \left(m^{\phi}(N)\right)^k \equiv m \pmod{n},$$

where we used Euler's Theorem 2.5.

2.3 Elementary attacks on RSA

It is well known that most successful attacks on RSA, are not based on factoring the modulus N. Rather, they exploit the mathematical weakness of the RSA algorithm or the improper use of the RSA system, such as lower exponents, common modulus, and knowledge of parts of the private exponent. We shall study here two elementary attacks on the RSA system.

Let $N = pq$ be an RSA modulus with $q < p < 2q$. Suppose that an adversary knows the Euler totient function $\phi(N)$ in addition to N. Then he can easily break the RSA system.

Proposition 2.6. *Let $N = pq$ be an RSA modulus. Suppose that $\phi(N)$ is known. Then one can factor N.*

Proof. Suppose that $N = pq$ and $\phi(N)$ are known. Consider the equations in p, q

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

Then, eliminating q, we get

$$p^2 - (N+1-\phi(N))p + N = 0.$$

This leads to the solutions

$$\begin{aligned} p &= \frac{N+1-\phi(N)+\sqrt{(N+1-\phi(N))^2-4N}}{2}, \\ q &= \frac{N+1-\phi(N)-\sqrt{(N+1-\phi(N))^2-4N}}{2}. \end{aligned}$$

□

Another well known attack on RSA makes use of the Fermat method for factoring. suppose that p and q are too close, namely $|p-q| < cN^{0.25}$ for some small constant c. de Weger [11] showed in 2002 that Fermat's factoring method could find the primes p and q.

Theorem 2.7. *Let $N = pq$ be an RSA modulus with $|p-q| < cN^{1/4}$ where c is a positive constant. Then one can factor N in time polynomial in c.*

Proof. Fermat's method consists in finding two integers x, y such that

$$4N = x^2 - y^2 = (x+y)(x-y).$$

If $x - y \neq 2$, then the factorization of N is given by

$$p = \frac{x+y}{2}, \quad q = \frac{x-y}{2}.$$

To find x, y, we consider the sequence of candidates for x defined by

$$x_i = \left[2\sqrt{N}\right] + i, \quad y_i = \sqrt{x_i^2 - 4N}, \quad i = 0, 1, \cdots, k,$$

where $[x]$ is the integral part of x. We stop the process when $x_k^2 - 4N$ is a perfect square. Since $p = \frac{x_k + y_k}{2}$ and $q = \frac{x_k - y_k}{2}$, then $x_k = p + q$. Now, suppose that $|p - q| < cN^{1/4}$. Then

$$\begin{aligned}
k &= x_k - \left[2\sqrt{N}\right] \\
&= p + q - \left[2\sqrt{N}\right] \\
&< p + q - 2\sqrt{N} + 1 \\
&= \frac{(p+q)^2 - 4N}{p + q + 2\sqrt{N}} + 1 \\
&= \frac{(p-q)^2}{p + q + 2\sqrt{N}} + 1 \\
&< \frac{c^2\sqrt{N}}{2\sqrt{N}} + 1 \\
&< \frac{c^2}{2} + 1.
\end{aligned}$$

It follows that Fermat's method can factor N in less than $\frac{c^2}{2} + 1$ steps, which is efficient for small values of c. □

If RSA is not used properly, it may be possible to break the RSA encryption by recovering the secret message m without use of any knowledge of the private key (N, d). One such improper use is the use of two public keys (N, e_1) and (N, e_2) with common modulus N and message m.

Theorem 2.8. *Let $N = pq$ be an RSA modulus and m a secret message. Let e_1 and e_2 be two public exponents such that $\gcd(e_1, e_2) = 1$. If $c_1 \equiv m^{e_1} \pmod{N}$ and $c_2 \equiv m^{e_2} \pmod{N}$ are public, then one can recover m.*

Proof. Suppose that the encrypted messages c_1, c_2 defined by

$$\begin{aligned}
c_1 &\equiv m^{e_1} \pmod{N}, \\
c_2 &\equiv m^{e_2} \pmod{N},
\end{aligned}$$

are public. If e_1 and e_2 are public with $\gcd(e_1, e_2) = 1$, then there exist integers x_1 and x_2 such that $e_1x_1 - e_2x_2 = 1$. Hence

$$c_1^{x_1}c_2^{-x_2} \equiv m^{e_1x_1}m^{-e_2x_2} \equiv m^{e_1x_1-e_2x_2} \equiv m \pmod{N}.$$

This terminates the proof. □

The RSA cryptosystem standards recommends to generate the primes p, q with the same bit size, that is $q < p < 2q$. This leads to the following result.

Proposition 2.9. *Let $N = pq$ be an RSA modulus with $q < p < 2q$. Then*

$$2^{-\frac{1}{2}}N^{\frac{1}{2}} < q < N^{\frac{1}{2}} < p < 2^{\frac{1}{2}}N^{\frac{1}{2}},$$

and

$$2N^{\frac{1}{2}} < p + q < \frac{3\sqrt{2}}{2}N^{\frac{1}{2}}.$$

Proof. Assume that $q < p < 2q$. Then multiplying by q, we get $q^2 < N < 2q^2$ and $2^{-\frac{1}{2}}N^{\frac{1}{2}} < q < N^{\frac{1}{2}}$. Since $p = \frac{N}{q}$, this gives easily $N^{\frac{1}{2}} < p < 2^{\frac{1}{2}}N^{\frac{1}{2}}$. Next, consider $f(p) = p + q = p + \frac{N}{p}$. The derivative satisfies $f'(p) = 1 - \frac{N}{p^2} > 0$, hence $f\left(N^{\frac{1}{2}}\right) \leq f(p) \leq f\left(2^{\frac{1}{2}}N^{\frac{1}{2}}\right)$ and

$$2N^{\frac{1}{2}} < p + q < \frac{3\sqrt{2}}{2}N^{\frac{1}{2}}.$$

This terminates the proof. □

3 Diophantine Approximations

In this section we introduce the basics of continued fractions and see how they arise out from attacking the RSA cryptosystem in some cases. For a general background we refer to [6] and [3].

3.1 Continued fractions

Let $x \in \mathbb{R}$ such that $\lfloor x \rfloor \neq x$ where $\lfloor x \rfloor$ is the integral part of x. Write $x_0 = x$ and

$$x_0 = a_0 + \frac{1}{x_1},$$

where $a_0 = \lfloor x_0 \rfloor$ and $x_1 > 1$. If $x_1 \neq 0$, then write

$$x_1 = a_1 + \frac{1}{x_2},$$

where $a_1 = \lfloor x_1 \rfloor$ and $x_2 > 1$. Next, if $x_2 \neq 0$, then write

$$x_2 = a_2 + \frac{1}{x_3},$$

where $a_2 = \lfloor x_2 \rfloor$ and $x_3 > 1$. Observe that

$$x = a_0 + \frac{1}{x_1} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{x_2}} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{x_3}}}.$$

Alternatively, one may write $x = [a_0, a_1, a_2, x_3]$.

Definition 3.1 (Continued Fraction Expansion)**.** The continued fraction representation of a real number x will be denoted by $x = [a_0, a_1, \cdots, a_m]$ where

$$[a_0, a_1, \cdots, a_m] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\cdots + \cfrac{1}{a_m}}},$$

and m may be infinite. All a_i, called partial quotients, are positive integers, except for a_0 which may be any integer.

Definition 3.2 (Convergent)**.** Let $x \in \mathbb{R}$ with $x = [a_0, a_1, \cdots, a_m]$. For $0 \leq n \leq m$, the nth convergent of the continued fraction expansion of x is $[a_0, a_1, \cdots, a_n]$.

Proposition 3.1. *For each $n \geq 0$, define integers p_n and q_n as follows:*

$$\begin{aligned} p_{-2} &= 0, \quad p_{-1} = 1, \quad p_n = a_n p_{n-1} + p_{n-2}, \\ q_{-2} &= 1, \quad q_{-1} = 0, \quad q_n = a_n q_{n-1} + q_{n-2}. \end{aligned}$$

Then, for $0 \leq n \leq m$, the nth convergent of the continued fraction expansion of x is $[a_0, a_1, \cdots, a_n] = \frac{p_n}{q_n}$.

Proof. We use induction. We have $p_0 = a_0 p_{-1} + p_{-2} = a_0$ and $q_0 = a_0 q_{-1} + q_{-2} = 1$ so that

$$[a_0] = \frac{p_0}{q_0}.$$

Suppose the proposition is true for $n-1$, that is

$$[a_1, a_2, a_3, \cdots, a_{n-1}] = \frac{a_{n-1}p_{n-2} + p_{n-3}}{a_{n-1}q_{n-2} + q_{n-3}}.$$

Then

$$\begin{aligned}
[a_0, a_1, a_2, \cdots, a_{n-1}, a_n] &= [a_0, a_1, a_2, \cdots, a_{n-1} + \frac{1}{a_n}] \\
&= \frac{\left(a_{n-1} + \frac{1}{a_n}\right) p_{n-2} + p_{n-3}}{\left(a_{n-1} + \frac{1}{a_n}\right) q_{n-2} + q_{n-3}} \\
&= \frac{(a_{n-1}a_n + 1)\, p_{n-2} + a_n p_{n-3}}{(a_{n-1}a_n + 1)\, q_{n-2} + a_n q_{n-3}} \\
&= \frac{a_n\,(a_{n-1}p_{n-2} + p_{n-3}) + p_{n-2}}{a_n\,(a_{n-1}q_{n-1} + q_{n-3}) + q_{n-2}} \\
&= \frac{a_n p_{n-1} + p_{n-2}}{a_n q_{n-1} + q_{n-2}} \\
&= \frac{p_n}{q_n},
\end{aligned}$$

Hence the proposition is true for n. □

Proposition 3.2. *For $-2 \le n \le m-1$, we have*

$$p_n q_{n+1} - q_n p_{n+1} = (-1)^{n+1}$$

Proof. We use induction. For $n = -2$, we have $p_{-2}q_{-1} - q_{-2}p_{-1} = -1 = (-1)^{-2+1}$. Assume that $p_{n-1}q_n - q_{n-1}p_n = (-1)^n$. Using Proposition 3.1 for $n+1$, we get

$$\begin{aligned}
p_n q_{n+1} - q_n p_{n+1} &= p_n(a_{n+1}q_n + q_{n-1}) - q_n(a_{n+1}p_n + p_{n-1}) \\
&= p_n q_{n-1} - q_n p_{n-1} \\
&= -(-1)^n \\
&= (-1)^{n+1},
\end{aligned}$$

which terminates the proof. □

As a consequence, we easily get the following result.

Proposition 3.3. *For $0 \le n \le m$, the fraction $\frac{p_n}{q_n}$ is in lowest terms, that is* $\gcd(p_n, q_n) = 1$.

Proof. By Proposition 3.2, for $0 \le n \le m-1$, we have $p_n q_{n+1} - q_n p_{n+1} = (-1)^{n+1}$, then $\gcd(p_n, q_n) = 1$ and $\gcd(p_{n+1}, q_{n+1}) = 1$. □

The following result is a direct consequence of Proposition 3.1 and Proposition 3.2.

Corollary 3.4. *For $n \geq 0$, let $\frac{p_n}{q_n}$ be a convergent of the continued fraction expansion of x. Then*

(a) $(q_n x - p_n)(q_{n+1}x - p_{n+1}) < 0$.

(b) $|q_{n+1}x - p_{n+1}| < |q_n x - p_n|$.

Proof. (a) Write $x = [a_1, a_2, a_3, \cdots, a_n, x_{n+1}] = [a_0, a_1, a_2, \cdots, a_{n+1}, x_{n+2}]$ where $x_{n+1} = [a_{n+1}, \cdots]$ and $x_{n+2} = [a_{n+2}, \cdots]$. For $n \geq 0$, we have

$$x - \frac{p_n}{q_n} = \frac{x_{n+1}p_n + p_{n-1}}{x_{n+1}q_n + q_{n-1}} - \frac{p_n}{q_n} = \frac{p_{n-1}q_n - p_n q_{n-1}}{q_n(x_{n+1}q_n + q_{n-1})} = \frac{(-1)^n}{q_n(x_{n+1}q_n + q_{n-1})}.$$

Hence

$$q_n x - p_n = \frac{(-1)^n}{x_{n+1}q_n + q_{n-1}}, \quad q_{n+1}x - p_{n+1} = \frac{(-1)^{n+1}}{x_{n+2}q_{n+1} + q_n}.$$

It follows that $(q_n x - p_n)(q_{n+1}x - p_{n+1}) < 0$.

(b) To show $|q_n x - p_n| > |q_{n+1}x - p_{n+1}|$, write $x_{n+1} = a_{n+1} + \frac{1}{x_{n+2}}$ with $x_{n+2} > 1$. Then

$$a_{n+1} < x_{n+1} < a_{n+1} + 1.$$

Hence

$$x_{n+1}q_n + q_{n-1} < (a_{n+1} + 1)q_n + q_{n-1} = q_{n+1} + q_n < x_{n+2}q_{n+1} + q_n,$$

which leads to

$$\frac{1}{x_{n+1}q_n + q_{n-1}} > \frac{1}{x_{n+2}q_{n+1} + q_n}.$$

We get finally

$$|q_n x - p_n| > |q_{n+1}x - p_{n+1}|,$$

which terminates the proof. □

Theorem 3.5. *For $n \geq 0$, let $\frac{p_n}{q_n}$ be a convergent of the continued fraction expansion of x. Let $\frac{p}{q}$ be a rational number with $\gcd(p, q) = 1$.*

(a) If $q < q_{n+1}$, then $|q_n x - p_n| \leq |qx - p|$.

(b) If $q \leq q_n$, then $\left|x - \frac{p_n}{q_n}\right| \leq \left|x - \frac{p}{q}\right|$.

Proof. (a) Assume $0 < q < q_{n+1}$. To show $|q_n x - p_n| \leq |qx - p|$, write p and q as

$$p = ap_n + bp_{n+1},$$
$$q = aq_n + bq_{n+1},$$

where

$$a = (-1)^{n+1}(pq_{n+1} - qp_{n+1}), \qquad b = (-1)^{n+1}(qp_n - pq_n).$$

Since $q < q_{n+1}$, then the expression of q implies that $ab < 0$. On the other hand, we have

$$\begin{aligned} qx - p &= (-1)^{n+1}(aq_n + bq_{n+1})x - (-1)^{n+1}(ap_n + bp_{n+1}) \\ &= (-1)^{n+1}a(q_n x - p_n) + (-1)^{n+1}b(q_{n+1}x - p_{n+1}). \end{aligned}$$

Observe that, using Corollary 3.4, the product of the terms gives

$$ab(q_n x - p_n)(q_{n+1}x - p_{n+1}) > 0.$$

Then

$$|qx - p| = |a(q_n x - p_n)| + |b(q_{n+1}x - p_{n+1})| \geq |q_n x - p_n|,$$

and the first assertion follows.

To prove (b), assume that $q \leq q_n$. Then

$$\left| x - \frac{p_n}{q_n} \right| = \frac{|q_n x - p_n|}{q_n} \leq \frac{|qx - p|}{q} = \left| x - \frac{p}{q} \right|,$$

which proves the second assertion. □

In 1798 Legendre proved the following result. This is the main result from the theory of continued fractions that we use to attack RSA.

Theorem 3.6. *Let $x \in \mathbb{R}$ and $\frac{p}{q}$ be a rational fraction such that $\gcd(p, q) = 1$ and $q < b$ if $x = \frac{a}{b}$ with $\gcd(a, b) = 1$. If*

$$\left| x - \frac{p}{q} \right| < \frac{1}{2q^2},$$

then $\frac{p}{q}$ is a convergent of the continued fraction expansion of x.

Proof. Let $\frac{p}{q}$ be a rational number with $\gcd(a, b) = 1$. Let $\frac{p_n}{q_n}$ be a convergente of x such that $q_n \leq q < q_{n+1}$. Suppose that $\left| x - \frac{p}{q} \right| < \frac{1}{2q^2}$. Using Theorem 3.5, we get

$$\left| \frac{p}{q} - \frac{p_n}{q_n} \right| = \left| \frac{p}{q} - x + x - \frac{p_n}{q_n} \right| \leq \left| \frac{p}{q} - x \right| + \left| x - \frac{p_n}{q_n} \right| \leq 2 \left| x - \frac{p}{q} \right| < \frac{1}{q^2}.$$

Hence

$$|pq_n - p_n q| < \frac{q_n}{q} \leq 1,$$

which leads to $pq_n - p_n q = 0$ and $\frac{p}{q} = \frac{p_n}{q_n}$. □

3.2 Attacks on RSA using continued fractions

Let $N = pq$ be an RSA modulus and e, d be the public and private exponents of the RSA cryptosystem satifying $ed \equiv 1 \pmod{\phi(N)}$ where $\phi(N) = (p-1)(q-1)$. A well known attack on RSA, described by Wiener [12], uses continued fractions, and applies when the private exponent d is small.

Theorem 3.7 (Wiener)**.** *Let $N = pq$ be an RSA modulus with $q < p < 2q$. If $d < \frac{1}{3}N^{\frac{1}{4}}$, then one can factor N in polynomial time.*

Proof. Suppose that $e < \phi(N)$ and $q < p < 2q$. Then $N = pq > q^2$, and $q < \sqrt{N}$. Expanding $\phi(N) = (p-1)(q-1)$, we get

$$N - \phi(N) = p + q - 1 < 2q + q - 1 < 3q < 3\sqrt{N}.$$

On the other hand, since $ed \equiv 1 \pmod{\phi(N)}$, then

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

for some positive integer k and, since $e < \phi(N)$, it satisfies

$$k = \frac{ed-1}{\phi(N)} < \frac{ed}{\phi(N)} < d.$$

Using $N - \phi(N) < 3\sqrt{N}$, we have

$$\begin{aligned}
\left|\frac{e}{N} - \frac{k}{d}\right| &= \frac{|ed - kN|}{Nd} \\
&= \frac{|ed - k\phi(N) - kN + k\phi(N)|}{Nd} \\
&= \frac{|1 - k(N - \phi(N))|}{Nd} \\
&< \frac{k(N - \phi(N))}{Nd} \\
&< \frac{3k\sqrt{N}}{Nd} \\
&= \frac{3k}{d\sqrt{N}}.
\end{aligned}$$

Using $k < d < \frac{1}{3}N^{\frac{1}{4}}$, we get

$$\frac{3k}{d\sqrt{N}} < \frac{N^{\frac{1}{4}}}{d\sqrt{N}} = \frac{1}{dN^{\frac{1}{4}}} < \frac{1}{3d^2} < \frac{1}{2d^2}.$$

Hence $\left|\frac{e}{N}-\frac{k}{d}\right|<\frac{1}{2d^2}$ and therefore, from Theorem 3.6, it follows that $\frac{k}{d}$ is one of the convergents in the continued fraction expansion of $\frac{e}{N}$. Notice that the continued fraction algorithm gives the convergents in polynomial time. Using this convergent, we get

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

which, by Proposition 2.6, leads to the factorization of N. □

The bounds on the private exponent can be increased considerably when there are three instances of RSA, having the same modulus, with small private exponents. As described in [7], an unpublished attack by Guo can be used to factor the modulus when the private exponents are each smaller than $N^{\frac{1}{3}}$.

Theorem 3.8 (Guo)**.** *Let $N = pq$ be an RSA modulus. Consider three instances of RSA with a common modulus N and public exponents e_1, e_2, e_3 satisfying*

$$e_1d_1\equiv 1 \pmod{\phi(N)},\quad e_2d_2\equiv 1 \pmod{\phi(N)},\quad e_3d_3\equiv 1 \pmod{\phi(N)}.$$

If all the k_i and d_i are pairwise relatively prime and $d_i < N^{\frac{1}{3}-\varepsilon}$ for $i = 1, 2, 3$, with $\varepsilon > 0$, then factor N can be factored in polynomial time.

Proof. Transforming the three congruences $e_id_i \equiv 1 \pmod{\phi(N)}$, $i = 1, 2, 3$ to equations, we get

$$e_1d_1=1+k_1\phi(N),\quad e_2d_2=1+k_2\phi(N),\quad e_3d_3=1+k_3\phi(N),$$

where k_1, k_2, k_3 are positive integers. Removing $\phi(N)$, we get the system

$$\begin{aligned}
e_1d_1k_2-e_2d_2k_1&=k_2-k_1,\\
e_1d_1k_3-e_3d_3k_1&=k_3-k_1,\\
e_2d_2k_3-e_3d_3k_2&=k_3-k_2.
\end{aligned}$$

Dividing the first equation by $d_1k_2e_2$, we get

$$\left|\frac{e_1}{e_2}-\frac{d_2k_1}{d_1k_2}\right|=\frac{|k_2-k_1|}{d_1k_2e_2}.$$

Under the conditions $\gcd(d_2k_1, d_1k_2) = 1$ and $\frac{|k_2-k_1|}{d_1k_2e_2}<\frac{1}{2(d_1k_2)^2}$, Theorem 3.6 implies that $\frac{d_2k_1}{d_1k_2}$ is a convergent of the continued expansion of $\frac{e_1}{e_2}$. The last condition leads to

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

Similarly, $\frac{d_3k_1}{d_1k_3}$ is a convergent of the continued expansion of $\frac{e_1}{e_3}$ if

$$d_1 < \frac{e_3}{2k_3|k_3 - k_1|},$$

and $\frac{d_3k_2}{d_2k_3}$ is a convergent of the continued expansion of $\frac{e_2}{e_3}$ if

$$d_2 < \frac{e_3}{2k_3|k_3 - k_2|}.$$

Assuming that all the k_i and d_i are pairwise relatively prime, we get

$$d_1 = \gcd(d_1k_2, d_1k_3), \qquad k_1 = \gcd(d_2k_1, d_3k_1),$$

which leads to $\phi(N) = \frac{ed_1-1}{k_1}$ and finally to the factorization of N. If we suppose that $k_i < d_i < N^\delta$ for a positive constant δ, and $e_1 < N$, then the condition $\frac{|k_2-k_1|}{d_1k_2e_2} < \frac{1}{2(d_1k_2)^2}$ can be rewritten as

$$N^{3\delta} < \frac{1}{2}N = N^{1-3\varepsilon},$$

or equivalently $\delta < \frac{1}{3} - \varepsilon$, where $\varepsilon > 0$ is a small constant depending only on N. □

4 Lattices

In this section we give some basic backgrounds about lattices and the LLL algorithm [8]. This includes definitions about lattices, some very useful lattice properties and some necessary theorems that will allow us to try some attacks on RSA. For more information about the algorithmic theory of lattices, see [3].

4.1 Lattices preliminaries and the LLL algorithm

A lattice $\mathcal{L}$ is a discrete additive subgroup of R^m.

Definition 4.1 (Lattice)**.** Let $b_1, \cdots, b_n \in \mathbb{R}^m$ be $n \leq m$ linearly independent vectors. The lattice generated by $\{b_1, \cdots, b_n\}$ is the set

$$\mathcal{L} = \sum_{i=1}^{n} \mathbb{Z}b_i = \left\{ \sum_{i=1}^{n} x_ib_i \mid x_i \in \mathbb{Z} \right\}.$$

The set $B = \langle b_1 \ldots, b_n \rangle$ is called a lattice basis for $\mathcal{L}$. The lattice dimension is $\dim(\mathcal{L}) = n$. If $n = m$ then $\mathcal{L}$ is said to be a full rank lattice.

A lattice $\mathcal{L}$ can be represented by a basis matrix. Given a basis B, a basis matrix M for the lattice generated by B is the $n \times m$ matrix defined by the rows of the set $\{b_1 \dots, b_n\}$

$$M = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

It is often useful to represent the matrix M by B. A very important notion for the lattice $\mathcal{L}$ is the determinant.

Definition 4.2 (Determinant)**.** Let $\mathcal{L}$ be a lattice generated by the basis $B = \langle b_1 \dots, b_n \rangle$. The determinant of $\mathcal{L}$ is defined as

$$\det(\mathcal{L}) = \sqrt{\det\left(BB^T\right)}.$$

If $n = m$, we have

$$\det(\mathcal{L}) = \sqrt{\det\left(BB^T\right)} = |\det(B)|.$$

In the following we show that the determinant of a lattice is an invariant, that is does not depend on the particular choice of the basis.

Proposition 4.1. *Any two bases for a lattice $\mathcal{L}$ are related by a matrix U having integer coefficients and determinant* $\det(U) = \pm 1$.

Proof. Suppose that the lattice $\mathcal{L}$ is generated by the bases $B = \langle b_1 \dots, b_n \rangle$ and $B' = \langle b'_1 \dots, b'_n \rangle$. Since every b_i can be expressed in the basis B' using integer coefficients, there exists a $n \times n$ matrix U' with integer coefficients such that $B = U'B'$. Similarly, there exists a $n \times n$ matrix U with integer coefficients such that $B' = UB$. Hence $B' = UB = UU'B'$ which leads to $UU' = I$ and $\det(U)\det(U') = 1$. Since $\det(U), \det(U') \in \mathbb{Z}$, then $\det(U) = \det(U') = \pm 1$. □

Corollary 4.2. *The determinant of a lattice does not depend on the selection of the basis.*

Proof. Suppose that the lattice $\mathcal{L}$ is generated by the bases $B = \langle b_1 \dots, b_n \rangle$ and $B' = \langle b'_1 \dots, b'_n \rangle$. Then there exists a matrix U' with integer coefficients such that $B = U'B'$. We have

$$\begin{aligned}
\det(\mathcal{L}) &= \sqrt{\det\left(BB^T\right)} \\
&= \sqrt{\det\left(U'B'(U'B')^T\right)} \\
&= \sqrt{\det\left((U'B')B'^T U'^T\right)} \\
&= \sqrt{\det\left(U'U'^T B'B'^T\right)} \\
&= \sqrt{\det\left(U'U'^T\right)}\sqrt{\det\left(B'B'^T\right)} \\
&= \sqrt{\det\left(B'B'^T\right)},
\end{aligned}$$

which terminates the proof. □

Definition 4.3 (Inner Product)**.** Let $v = \sum_{i=1}^{n} v_i b_i$ and $v' = \sum_{i=1}^{n} v'_i b_i$ be two vectors. The inner product of v and v' is defined as

$$\langle u, v \rangle = \sum_{i=1}^{n} v_i v'_i.$$

A short lattice vector is a vector v in $\mathcal{L}$ such that its Euclidean norm $\|v\|$ is relatively small.

Definition 4.4 (Euclidean Norm)**.** The Euclidean norm of a vector $v = \sum_{i=1}^{n} v_i b_i$ is defined as

$$\|v\| = \left(\sum_{i=1}^{n} v_i^2 \right)^{\frac{1}{2}}.$$

For a given lattice $\mathcal{L}$ with dimension $n \geq 2$ some bases are better than others. It is often useful to represent a lattice in a basis of short vectors. The LLL algorithm, designed by Lenstra, Lenstra and Lovász [8] in 1982, can be used to find a basis of lattice vectors which are relatively small in norm. The LLL algorithm makes use of the the Gram-Schmidt procedure of computing an orthogonal basis of the same determinant. Given a set of independent vectors $\{b_1 \dots, b_n\}$, the Gram-Schmidt procedure constructs a set of vectors $\{b_1^* \dots, b_n^*\}$ such that

$$\begin{aligned} & b_1^* = b_1, \\ & \text{for } i \geq 2, \; b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^*, \text{ where } \mu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle} \text{ for } \quad 1 \leq j < i. \end{aligned}$$

The above conditions can be rewritten as $B = MB^*$, where basis vectors are rows of B and B^*, and

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ \mu_{2,1} & 1 & 0 & 0 & \cdots & 0 \\ \mu_{3,1} & \mu_{3,2} & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mu_{n_1,1} & \mu_{n-1,2} & \mu_{n-1,3} & \cdots & 1 & 0 \\ \mu_{n,1} & \mu_{n,2} & \mu_{n,3} & \cdots & \mu_{n,n-1} & 1 \end{bmatrix},$$

is the Gram-Schmidt matrix. Obviously $\det(M) = 1$.

Theorem 4.3 (Gram-Schmidt)**.** *Let $\{b_1, \dots, b_n\}$ be a set of independant vectors and let $\{b_1^*, \dots, b_n^*\}$ be the associated Gram-Schmidt set of vectors. Then $\{b_1^* \dots, b_n^*\}$ is orthogonal.*

Proof. We use induction. Since $b_1^* = b_1$ and $b_2^* = b_2 - \mu_{2,1} b_1^* = b_2 - \mu_{2,1} b_1$, then

$$\langle b_1^*, b_2^* \rangle = \langle b_1, b_2 - \mu_{2,1} b_1 \rangle = \langle b_1, b_2 \rangle - \mu_{2,1} \langle b_1, b_1 \rangle = \langle b_1, b_2 \rangle - \frac{\langle b_2, b_1 \rangle}{\langle b_1, b_1 \rangle} \langle b_1, b_1 \rangle = 0.$$

Hence $\{b_1^*, b_2^*\}$ is orthogonal. Next, suppose that $\{b_1^* \cdots, b_{i-1}^*\}$ is orthogonal for $i \geq 3$. Then, for $1 \leq k \leq i-1$, we have

$$\begin{aligned}
\langle b_k^*, b_i^* \rangle &= \left\langle b_k^*, b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^* \right\rangle \\
&= \langle b_k^*, b_i \rangle - \sum_{j=1}^{i-1} \mu_{i,j} \langle b_k^*, b_j^* \rangle \\
&= \langle b_k^*, b_i \rangle - \mu_{i,k} \langle b_k^*, b_k^* \rangle \\
&= \langle b_k^*, b_i \rangle - \frac{\langle b_i, b_k^* \rangle}{\langle b_k^*, b_k^* \rangle} \langle b_k^*, b_k^* \rangle \\
&= 0.
\end{aligned}$$

It follows that b_i^* is orthogonal to each vector b_k^* with $1 \leq k \leq i-1$. Hence $(b_1^* \cdots, b_i^*)$ is orthogonal, which terminates the proof. □

Corollary 4.4 (Hadamard)**.** *Let $B = \{b_1, \ldots, b_n\}$ be a basis of a lattice $\mathcal{L}$ and let $B^* = \{b_1^*, \ldots, b_n^*\}$ be the associated Gram-Schmidt basis. Then*

$$\det(\mathcal{L}) = \prod_{i=1}^{n} \|b_i^*\| \leq \prod_{i=1}^{n} \|b_i\|.$$

Proof. Since $B = MB^*$ where M is the Gram-Schmidt matrix with coefficients $\mu_{i,j}$ and determinant 1, we have

$$\det(\mathcal{L})^2 = \det\left(BB^T\right)^2 = \det\left(B^*(B^*)^T\right)^2.$$

Using the orthogonal basis $B^* = \{b_1^*, \ldots, b_n^*\}$, we get

$$\det(\mathcal{L})^2 = \det\left(B^*(B^*)^T\right) = \det\left[\langle b_i^*, b_j^* \rangle\right]_{1 \leq i,j \leq n} = \prod_{i=1}^{n} \|b_i^*\|^2.$$

On the other hand, we have $\|b_1\| = \|b_1^*\|$ and for $2 \leq i \leq n$,

$$b_i = b_i^* + \sum_{j=1}^{i-1} \mu_{i,j} b_j^*.$$

Observe that $\langle b_r^*, b_s^* \rangle = 0$ whenever $r \neq s$. Hence

$$\|b_i\|^2 = \|b_i^*\|^2 + \sum_{j=1}^{i-1} \mu_{i,j}^2 \|b_j^*\|^2,$$

and we deduce

$$\det(\mathcal{L})^2 = \prod_{i=1}^{n} \|b_i^*\|^2 \leq \prod_{i=1}^{n} \|b_i\|^2.$$

This terminates the proof. □

In 1982, Lenstra, Lenstra and Lovász [8] proposed an algorithm for lattice reduction that runs in polynomial time and produces a basis B with many remarkable properties. When the LLL reduction algorithm is performed on a lattice $\mathcal{L}$ generated by a basis $V = \{v_1, \ldots, v_n\}$, it outputs a basis $B = \{b_1, \ldots, b_n\}$ which is LLL reduced.

Definition 4.5 (LLL Reduction)**.** Let $B = \{b_1, \ldots, b_n\}$ be a basis for a lattice $\mathcal{L}$ and let $B^* = \{b_1^*, \ldots, b_n^*\}$ be the associated Gram-Schmidt orthogonal basis. The basis B is said to be LLL reduced if it satisfies the following two conditions:

$$|\mu_{i,j}| \leq \frac{1}{2}, \quad \text{for} \quad 1 \leq j < i \leq n, \tag{1}$$

$$\frac{3}{4}\|b_{i-1}^*\|^2 \leq \|b_i^* + \mu_{i,i-1} b_{i-1}^*\|^2 \quad \text{for} \quad 1 < i \leq n. \tag{2}$$

The fundamental result of Lenstra, Lenstra, and Lovász says that an LLL reduced basis is a good basis for shortness of the vectors and that it is possible to compute an LLL reduced basis in polynomial time. Some useful properties of a LLL reduced basis are stated in the following theorem.

Theorem 4.5. *Let $B = \{b_1, \ldots, b_n\}$ be a LLL reduced basis and let $B^* = \{b_1^*, \ldots, b_n^*\}$ be the associated Gram-Schmidt orthogonal basis. Then*

- *(a)* $\|b_j^*\|^2 \leq 2^{i-j}\|b_i^*\|^2$ *for* $1 \leq j \leq i \leq n$.
- *(b)* $\prod_{i=1}^{n} \|b_i\| \leq 2^{\frac{n(n-1)}{4}} \det(\mathcal{L})$.
- *(c)* $\|b_j\| \leq 2^{\frac{i-1}{2}} \|b_i^*\|$ *for* $1 \leq j \leq i \leq n$.
- *(d)* $\|b_1\| \leq 2^{\frac{n-1}{4}} (\det(\mathcal{L}))^{\frac{1}{n}}$.

Proof.
(a) Suppose that the basis $B = \{b_1, \ldots, b_n\}$ is LLL reduced. Expanding (2) and using (1), we get

$$\frac{3}{4}\|b_{i-1}^*\|^2 \leq \|b_i^*\|^2 + \mu_{i,i-1}^2\|b_{i-1}^*\|^2 \leq \|b_i^*\|^2 + \frac{1}{4}\|b_{i-1}^*\|^2,$$

which leads to

$$\|b_{i-1}^*\|^2 \leq 2\|b_i^*\|^2. \tag{3}$$

Hence, for $j \leq i$, we get

$$\|b_j^*\|^2 \leq 2^{i-j}\|b_i^*\|^2.$$

(b) Recall that $b_i = b_i^* + \sum_{j=1}^{i-1} \mu_{i,j} b_j^*$. Using (1), we get

$$\|b_i\|^2 = \|b_i^*\|^2 + \sum_{j=1}^{i-1} \mu_{i,j}^2 \|b_j^*\|^2 \leq \|b_i^*\|^2 + \frac{1}{4}\sum_{j=1}^{i-1} \|b_j^*\|^2.$$

Since $\|b_j^*\|^2 \leq 2^{i-j}\|b_i^*\|^2$, we get

$$\|b_i\|^2 \leq \|b_i^*\|^2 + \frac{1}{4}\sum_{j=1}^{i-1} 2^{i-j}\|b_i^*\|^2 = \left(1 + 2^{i-2} - 2^{-1}\right)\|b_i^*\|^2 \leq 2^{i-1}\|b_i^*\|^2. \tag{4}$$

Using Corollary 4.4, we have

$$\prod_{i=1}^{n} \|b_i\|^2 \leq \prod_{i=1}^{n} 2^{i-1}\|b_i^*\|^2 = 2^{\frac{n(n-1)}{2}} \prod_{i=1}^{n} \|b_i^*\|^2 = 2^{\frac{n(n-1)}{2}} \det(L)^2,$$

which leads to

$$\prod_{i=1}^{n} \|b_i\| \leq 2^{\frac{n(n-1)}{4}} \det(L).$$

(c) Considering (4) with $i = j$, we get $\|b_j\|^2 \leq 2^{j-1}\|b_j^*\|^2$. Combining with (3), we get

$$\|b_j\|^2 \leq 2^{j-1}2^{i-j}\|b_i^*\|^2 = 2^{i-1}\|b_i^*\|^2,$$

which leads to

$$\|b_j\| \leq 2^{\frac{i-1}{2}}\|b_i^*\|. \tag{5}$$

(d) Taking $j = 1$ in (5) and squaring, we get $\|b_1\|^2 \leq 2^{i-1}\|b_i^*\|^2$ for $1 \leq i \leq n$. Hence

$$\|b_1\|^{2n} \leq \prod_{i=1}^{n} 2^{i-1}\|b_i^*\|^2 = 2^{\frac{n(n-1)}{2}} \prod_{i=1}^{n} \|b_i^*\|^2 = 2^{\frac{n(n-1)}{2}} (\det(L))^2.$$

From this, we deduce $\|b_1\| \leq 2^{\frac{n-1}{4}}(\det(L))^{\frac{1}{n}}$. □

4.2 Attacks on RSA using lattice reduction

An important application of lattice reduction found by Coppersmith [4] in 1996 is finding small roots of low-degree polynomial equations. This includes modular univariate polynomial equations, and bivariate integer equations.

Let M be some large integer of unknown factorization and

$$f(x) = \sum_{i=1}^{d} a_i x^i.$$

be a polynomial of degree d with integer coefficients. Consider the equation $f(x) \equiv 0 \pmod{M}$. In general there is no known efficient algorithm that find integer roots of the above equation. However, Coppersmith [4] introduced an efficient method for finding small integer solutions using the LLL algorithm. Suppose we know that there exists an integer x_0 such that $f(x_0) \equiv 0 \pmod{M}$ and that $|x_0| < N^{\frac{1}{d}}$. The problem is to find x_0. The main idea is that if the coefficients of f are small enough so that $|f(x_0)| = \sum_{i=1}^{d} |a_i x^i| < M$, then one might have $f(x_0)$ over the integers. Coppersmith's idea is to build from $f(x)$ a polynomial $h(x)$ which has small coefficients and the same solution x_0. Soon after Coppersmith proposed his method in 1996, Howgrave-Graham [7] proposed in 1997 a a new method for finding all small integer roots. Suppose we know an upper bound X such that $|x_0| < X$. The following theorem by Howgrave-Graham reformulates Coppersmith's idea of finding modular roots. We define the Euclidean norm of a polynomial $f(x)$ as

$$\|f(x)\| = \left(\sum_{i=0}^{d} a_i^2 \right)^{\frac{1}{2}}.$$

Theorem 4.6 (Howgrave-Graham). *Let $h(x) \in \mathbb{Z}[x]$ be a polynomial of at most ω monomials satisfying*

(1) $|x_0| < X$, for some positive integer X.

(2) $h(x_0) \equiv 0 \pmod{M}$, for some positive integer M.

(3) $\|h(xX)\| < \frac{M}{\sqrt{\omega}}$.

Then $h(x_0) = 0$ over $\mathbb{Z}$.

Proof. Let $h(x) = \sum_i^d a_i x^i$ with ω monomials. Suppose $|x_0| < X$. Then

$$|h(x_0)| = \left| \sum_i a_i x_0^i \right| \leq \sum_i \left| a_i x_0^i \right| < \sum_i \left| a_i X^i \right|. \tag{6}$$

Recall that Cauchy-Schwarz inequality asserts that for $\alpha, \beta \in \mathbb{R}$, we have

$$\left(\sum_i \alpha_i \beta_i\right)^2 \leq \left(\sum_i \alpha_i^2\right)\left(\sum_i \beta_i^2\right).$$

Using this, we get

$$\left(\sum_i \left|a_i X^i\right|\right)^2 \leq \left(\sum_i 1^2\right)\left(\sum_i \left(a_i X^i\right)^2\right) = \omega \sum_i \left(a_i X^i\right)^2.$$

If $\|h(xX)\| < \frac{M}{\sqrt{\omega}}$, then, using (6), we get

$$|h(x_0)| < \sum_i \left|a_i X^i\right| < \sqrt{\omega}\sqrt{\sum_i \left(a_i X^i\right)^2} = \sqrt{\omega}\|h(xX)\| < M.$$

Hence $|h(x_0)| < M$. Finally, if $h(x_0) \equiv 0 \pmod{M}$, then $h(x_0) = 0$ over $\mathbb{Z}$ which terminates the proof. □

To solve $f(x_0) \equiv 0 \pmod{M}$, Theorem 4.6 suggests we should look for a polynomial $h(x)$ of small norm satisfying $h(x_0) \in Z$. To do this we will build a lattice of polynomials related to f and use LLL to find short vectors in the lattice. The following result, as given below, is from May [9].

Theorem 4.7. *For every $\varepsilon > 0$ there exists an N_0 such that the following holds: Let $N > N_0$ be an integer with unknown factorization which has a divisor $b > N^\beta$. Let $f_b(x)$ be a monic univariate polynomial of degree δ. All solutions x_0 of the congruence $f_b(x) \equiv 0 \pmod{b}$, such that*

$$|x_0| < 2^{-\frac{1}{2}} N^{\frac{\beta^2}{\delta}-\varepsilon},$$

can be found in time polynomial in $\log(N)$.

Proof. Fix two positive integers m and t and consider the polynomials

$$\begin{aligned} g_{i,j}(x) &= x^j N^i (f_b(x))^{m-i}, \quad j = 0, \cdots, \delta - 1, \quad i = m, \cdots, 1, \\ h_i(x) &= x^i (f_b(x))^m, \qquad i = 0, \cdots, t-1, \end{aligned}$$

where $\delta = \deg(f_b)$. Observe that all the polynomials share the root x_0 modulo N^m. Rewriting the polynomials explicitly, we get

$$\begin{array}{c|ccccc} & j=0 & j=1 & j=2 & \cdots & j=\delta-1 \\ \hline i=m & N^m, & N^m x, & N^m x^2, & \cdots & N^m x^{\delta-1}, \\ i=m-1 & N^{m-1} f_b(x) & N^{m-1} x f_b(x) & N^{m-1} x^2 f_b(x) & \cdots & N^{m-1} x^{\delta-1} f_b(x) \\ i=m-2 & N^{m-2} f_b(x)^2 & N^{m-2} x f_b(x)^2 & N^{m-2} x^2 f_b(x)^2 & \cdots & N^{m-2} x^{\delta-1} f_b(x)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ i=2 & N^2 f_b(x)^{m-2} & N^2 x f_b(x)^{m-2} & N^2 x^2 f_b(x)^{m-2} & \cdots & N^2 x^{\delta-1} f_b(x)^{m-2} \\ i=1 & N f_b(x)^{m-1} & N x f_b(x)^{m-1} & N x^2 f_b(x)^{m-1} & \cdots & N x^{\delta-1} f_b(x)^{m-1} \end{array} \tag{7}$$

Observe that the maximal degree is $\delta - 1 + (m-1)\delta = m\delta - 1$. The details of the polynomials $h_i(x)$ are as follows

$$i = 0, \cdots, t-1 \Rightarrow f_b^m(x), xf_b^m(x), x^2 f_b^m(x), \cdots, x^{t-1} f_b^m(x). \tag{8}$$

Observe here that the maximal degree is $t - 1 + m\delta > m\delta - 1$. Replacing x by Xx in the rows $i = m, m-1, \ldots, 1$ of the table (7) and in the sequence (8) and expressing in the basis $\left(1, x, x^2, \cdots, x^{m\delta+t-1}\right)$, we get a sequence of matrices of the shape

$$\begin{array}{rcl}
M_m & = & \left[\begin{array}{cccc} N^m & & & \\ & N^m X & & \\ & & \ddots & \\ & & & N^m X^{\delta-1} \end{array}\right], \\
M_{m-1} & = & \left[\begin{array}{cccccccc} - & - & - & - & N^{m-1}X^{\delta} & & & \\ - & - & - & - & - & N^{m-1}X^{\delta+1} & & \\ - & - & - & - & - & - & \ddots & \\ - & - & - & - & - & - & - & N^{m-1}X^{2\delta-1} \end{array}\right], \\
\vdots & = & \vdots \\
M_1 & = & \left[\begin{array}{cccccccc} - & - & \cdots & - & NX^{(m-1)\delta} & & & \\ - & - & \cdots & - & - & NX^{(m-1)\delta+1} & & \\ - & - & \cdots & - & - & - & \ddots & \\ - & - & \cdots & - & - & - & - & NX^{(m-1)\delta+\delta-1} \end{array}\right], \\
M_0 & = & \left[\begin{array}{cccccccc} - & - & \cdots & - & X^{m\delta} & & & \\ - & - & \cdots & - & - & X^{m\delta+1} & & \\ - & - & \cdots & - & - & - & \ddots & \\ - & - & \cdots & - & - & - & - & X^{m\delta+t-1} \end{array}\right].
\end{array}$$

Gathering the matrices, we get a triangular matrix of the form

$$M = \left[\begin{array}{c} M_m \\ M_{m-1} \\ \vdots \\ M_1 \\ M_0 \end{array}\right], \tag{9}$$

which generates a lattice $\mathcal{L}$. Obviously, we have

$$\det(\mathcal{L}) = N^{m\delta} \cdot N^{(m-1)\delta} \cdot \cdots \cdot N^{\delta} X^{1+2+\cdots+n-1} = N^{\frac{1}{2}m(m+1)\delta} X^{\frac{1}{2}n(n-1)},$$

where $n = m\delta + t$. Using the LLL-algorithm, we can find a small element in $\mathcal{L}$ that corresponds to a polynomial $h(x)$ satisfying (d) of Theorem 4.5, namely

$$\|h(xX)\| \leq 2^{\frac{n-1}{4}} \det(L)^{\frac{1}{n}} = 2^{\frac{n-1}{4}} N^{\frac{m(m+1)\delta}{2n}} X^{\frac{1}{2}(n-1)}.$$

In order to apply Theorem 4.6 on $h(x)$, it is sufficient that $\|h(xX)\| \leq \frac{b^m}{\sqrt{n}}$, holds. This is satisfied if

$$2^{\frac{n-1}{4}} N^{\frac{m(m+1)\delta}{2n}} X^{\frac{1}{2}(n-1)} < \frac{b^m}{\sqrt{n}}.$$

Plugging $b > N^\beta$, we find

$$2^{\frac{n-1}{4}} N^{\frac{m(m+1)\delta}{2n}} X^{\frac{1}{2}(n-1)} < \frac{N^{m\beta}}{\sqrt{n}}.$$

Solving for X, we get

$$X < 2^{-\frac{1}{2}} n^{\frac{-1}{n-1}} N^{\frac{2mn\beta - m(m+1)\delta}{n(n-1)}}.$$

Consider the exponent of N as a polynomial in m. The exponent is maximal for

$$m = \frac{2n\beta - \delta}{2\delta},$$

which leads to the bound

$$X < 2^{-\frac{1}{2}} n^{\frac{-1}{n-1}} N^{\frac{\beta^2}{\delta} + \frac{\beta^2}{(n-1)\delta} + \frac{\delta}{4n(n-1)} - \frac{\beta}{n-1}}.$$

This can be rewritten as

$$X < 2^{-\frac{1}{2}} N^{\frac{\beta^2}{\delta} - \varepsilon},$$

where

$$\varepsilon = \frac{\log n}{(n-1)\log N} + \frac{\beta}{n-1} - \frac{\beta^2}{(n-1)\delta} - \frac{\delta}{4n(n-1)}.$$

Observe that ε depends on n and satisfies $\lim_{n \to +\infty} \varepsilon = 0$. □

Theorem 4.7 has various applications in cryptography. We will now present an attack on RSA - also due to Coppersmith - that finds the factorization of $N = pq$, provided that one knows half of the bits of one of the factors.

Theorem 4.8. *Let $N = pq$ be an RSA modulus with $p > q$. If $\tilde{p}$ is an approximation of p with*

$$|\tilde{p} - p| < N^{\frac{1}{4}},$$

then N can be factored in polynomial time in $\log N$.

Proof. Suppose we know an approximation $\tilde{p}$ of p with $|\tilde{p} - p| < N^{\frac{1}{4}}$. Consider the polynomial $f_p(x) = x + \tilde{p}$. Then $f_p(p - \tilde{p}) = p \equiv 0 \mod p$. Hence, $x_0 = p - \tilde{p}$ satisfies

$$f_p(x_0) \equiv 0 \mod p, \quad |x_0| < N^{\frac{1}{4}}.$$

Since $p > N^{\frac{1}{2}}$, one can then apply Theorem 4.7 with $b = p$, $f_p(x) = x + \tilde{p}$, $\delta = 1$ and $\beta = \frac{1}{2}$. This gives explicitly x_0 which leads to $p = x_0 + \tilde{p}$. □

In 2004, Blömer and May [1] improved upon Wiener's result by showing that every public exponent e satisfying an equation $ex - k\phi(N) = y$ with suitable bounds for x and y yields the factorization of N. The Blömer-May attack makes use of Coppersmith's method, namely Theorem 4.8.

Theorem 4.9. *Let $c \leq 1$ and let (N, e) be an RSA public key tuple with $N = pq$ and $p - q \geq cN^{\frac{1}{2}}$. Suppose that e satisfies an equation $ex - k\phi(N) = y$ with*

$$0 < x \leq \frac{1}{3}N^{\frac{1}{4}}, \quad \text{and} \quad |y| \leq cN^{-\frac{3}{4}}ex.$$

Then N can be factored in polynomial time.

Proof. Rewrite the equation $ex - k\phi(N) = y$ as $ex - kN = y - k(p+q-1)$. Dividing by Nx, we get

$$\left|\frac{e}{N} - \frac{k}{x}\right| = \frac{|y - k(p+q-1)|}{Nx}. \tag{10}$$

Next, suppose $|y| \leq cN^{-\frac{3}{4}}ex$ and $e < \phi(N)$. Then

$$k = \frac{ex - y}{\phi(N)} < \frac{ex + |y|}{\phi(N)} < \frac{ex + \frac{1}{4}ex}{\phi(N)} < \frac{5}{4}x.$$

Combining with Proposition 2.9 and using $e < N$, this implies an upper bound for $|y - k(p+q-1)|$ as follows

$$\begin{aligned}
|y - k(p+q-1)| &\leq |y| + k(p+q-1) \\
&\leq |y| + k(p+q) \\
&\leq cN^{-\frac{3}{4}}ex + \frac{5}{4}x \times 3N^{\frac{1}{2}} \\
&= cN^{-\frac{3}{4}}ex + \frac{15}{4}N^{\frac{1}{2}}x \\
&< cN^{\frac{1}{4}}x + \frac{15}{4}N^{\frac{1}{2}}x \\
&< 4N^{\frac{1}{2}}x,
\end{aligned}$$

for sufficiently large N. Plugging in (10), we get

$$\left|\frac{e}{N} - \frac{k}{x}\right| < \frac{4N^{\frac{1}{2}}x}{Nx} = \frac{4}{N^{\frac{1}{2}}}.$$

If x satisfies $0 < x \leq \frac{1}{3}N^{\frac{1}{4}}$, then $\frac{4}{N^{\frac{1}{2}}} < \frac{1}{2x^2}$. Hence, by Theorem 3.6, $\frac{k}{x}$ is a convergent of the continued fraction expansion of $\frac{e}{N}$. Using k and x we deduce

$$p + q = N - \frac{ex}{k} + 1 + \frac{y}{k}. \tag{11}$$

On the other hand, we have

$$k = \frac{ex - y}{\phi(N)} > \frac{ex - |y|}{\phi(N)} > \frac{ex - \frac{1}{4}ex}{\phi(N)} = \frac{3ex}{4\phi(N)}.$$

This implies the following upper bound for $\frac{|y|}{k}$

$$\frac{|y|}{k} < \frac{cN^{-\frac{3}{4}}ex}{\frac{3}{4}ex}\phi(N) = \frac{4}{3}cN^{-\frac{3}{4}}\phi(N) < \frac{4}{3}cN^{\frac{1}{4}},$$

where we used $\phi(N) < N$. Hence, using (11), we see that $N - \frac{ex}{k} + 1$ is is an approximation of $p + q$ up to an error term $\frac{|y|}{k} < \frac{4}{3}cN^{\frac{1}{4}}$ which can be transformed into an approximation of $p - q$. Indeed, setting $s = N - \frac{ex}{k} + 1$ and $t = \sqrt{|s^2 - 4N|}$, we have

$$\begin{aligned}
|p-q-t| &= \frac{|(p-q)^2 - t^2|}{p-q+t} \\
&= \frac{|(p-q)^2 - |s^2 - 4n||}{p-q+t} \\
&\leq \frac{|(p-q)^2 - (s^2 - 4n)|}{p-q+t} \\
&= \frac{|(p-q)^2 + 4n - s^2|}{p-q+t} \\
&= \frac{|(p+q)^2 - s^2|}{p-q+t} \\
&= \frac{|p+q-s|\,(p+q+s)}{p-q+t}.
\end{aligned}$$

Observe that $|p + q - s| < \frac{4}{3}cN^{\frac{1}{4}}$, $p + q + s < 3(p + q)$ and $p - q + t > p - q$. Then

$$|p-q-t| < \frac{4cN^{\frac{1}{4}}(p+q)}{p-q}$$

Assuming $p - q \geq cN^{\frac{1}{2}}$ and using Proposition 2.9, we get

$$|p-q-t| < \frac{12cN^{\frac{1}{4}}N^{\frac{1}{2}}}{cN^{\frac{1}{2}}} = 12N^{\frac{1}{4}}.$$

We get finally

$$\begin{aligned}\left|p-\frac{s+t}{2}\right| &= \frac{1}{2}\left|p+q-s+p-q-t\right| \\ &< \frac{1}{2}\left|p+q-s\right|+\frac{1}{2}\left|p-q-t\right| \\ &< \frac{2}{3}cN^{\frac{1}{4}}+6N^{\frac{1}{4}} \\ &< 7N^{\frac{1}{4}}.\end{aligned}$$

This implies that one of the values $\frac{s+t}{2}+jN^{\frac{1}{4}}$, $-6 \leq j \leq 6$, is an approximation of p up to an error of at most $N^{\frac{1}{4}}$. Hence, using Coppersmith's Theorem 4.8, N can be factored in polynomial time. □

5 Conclusion

In this chapter, we review the mathematical foundations of the RSA cryptosystem. We described the elementary arithmetic of the RSA encryption, decryption and signature. We introduced the tools to launch some cryptanalytic attacks on the RSA cryptosystem, namely the diophantine approximation based attacks and the lattice reduction based attacks. This includes the theory of the continued fractions, the Lenstra, Lenstra, and Lovász famous LLL algorithm as well as the work of Coppersmith for finding small roots of univariate modular polynomial equations.

References

[1] J. Blömer, A. May: A generalized Wiener attack on RSA. In Public Key Cryptography - PKC 2004, volume 2947 of Lecture Notes in Computer Science, pp. 1—13. Springer-Verlag, 2004.

[2] D. Boneh, G. Durfee: Cryptanalysis of RSA with private key d less than $N^{0.292}$, Advances in Cryptology – Eurocrypt'99, Lecture Notes in Computer Science Vol. 1592, Springer-Verlag, 1—11, 1999.

[3] H. Cohen: A Course in Computational Number Theory, Graduate Texts in Mathematics, Springer, 1993.

[4] D. Coppersmith: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. Journal of Cryptology, 10(4), 233—260 (1997)

[5] W. Diffie, E. Hellman: New directions in cryptography, IEEE Trans- actions on Information Theory, 22, 5 (1976), pp. 644–654.

[6] G.H. Hardy, E.M. Wright: An Introduction to the Theory of Numbers. Oxford University Press, London, 1965.

[7] N.A. Howgrave-Graham: Finding small roots of univariate modular equations revisited. In Cryptography and Coding, LNCS 1355, pp. 131—142, Springer-Verlag, 1997.

[8] A.K. Lenstra, H.W. Lenstra and L. Lovász: Factoring polynomials with rational coefficients, Mathematische Annalen, Vol. 261, 513—534, 1982.

[9] A. May: New RSA Vulnerabilities Using Lattice Reduction Methods, Ph.D. thesis, Paderborn, 2003,
`http://www.cits.rub.de/imperia/md/content/may/paper/bp.ps`

[10] R. Rivest, A. Shamir, L. Adleman: A Method for Obtaining digital signatures and public-key cryptosystems, Communications of the ACM, Vol. 21 (2), 120—126 (1978)

[11] B. de Weger: Cryptanalysis of RSA with small prime difference, Applicable Algebra in Engineering, Communication and Computing,Vol. 13(1), 17—28, 2002.

[12] M. Wiener: Cryptanalysis of short RSA secret exponents, IEEE Transactions on Information Theory, Vol. 36, 553—558, 1990.