Security Analysis of RSA-type Cryptosystems

Marc Joye

Submitted for the degree of Doctor of Philosophy in Applied Sciences

Jury: P. Sobieski, UCL (Chairman)
F. Borceux, UCL
J.-M. Couveignes, U. Bordeaux
A. Magnus, UCL
D. Naccache, Gemplus
J.-J. Quisquater, UCL (Thesis supervisor)
J. Stern, ENS

October 1997
ABSTRACT. In 1978, Rivest, Shamir and Adleman introduced the public-key cryptosystem RSA. Thereafter, it was extended to Lucas sequences and elliptic curves. This thesis analyses the security of these cryptosystems in given contexts. In particular, all known major attacks against RSA-type systems (including some attacks due to the author) are reviewed. We also see how these attacks can be avoided.

RéSUMÉ. En 1978, Rivest, Shamir et Adleman introduisaient le cryptosystème à clé publique RSA. Par après, ce dernier a été étendu aux suites de Lucas et aux courbes elliptiques. Cette thèse analyse la sécurité de ces cryptosystèmes dans des contextes donnés. En particulier, toutes les principales attaques connues contre les systèmes du type RSA (dont certaines sont dues à l'auteur) sont revues. Nous voyons également comment éviter ces attaques.

Author address:
UCL Cryptography Group, Université catholique de Louvain
Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium
E-mail address: joye@dice.ucl.ac.be
URL: http://www.dice.ucl.ac.be/crypto/joye/
To my two favorite persons,
Anne-Marie and Hung-Mei.

Toutes les guerres ayant leur caractère propre et présentant
dans leur évolution un grand nombre de caractères particuliers,
chacune d’elle peut être considérée comme une mer inconnue
du général en chef.

— Clausewitz (cited in the biography of de Gaulle by Lacouture)
Acknowledgements

I am most grateful to my advisor Jean-Jacques Quisquater for guiding me all along my research. I am also grateful to Francis Borceux, Jean-Marc Couveignes and Alphonse Magnus who served me on committee thesis. In particular, I want to thank Jean-Marc Couveignes for patiently answering many questions on the theory of elliptic curves. Thanks to David Naccache and Jacques Stern for accepting to be co-referees of this thesis. Thanks also to Feng Bao, Daniel Bleichenbacher, Robert Deng, Arjen Lenstra and Tsuyoshi Takagi for giving me the opportunity to write joint papers. Other researchers were of some utility for sending preprints of their work or for providing some fruitful comments; in alphabetical order, they are Dan Boneh, Seng-Kiat Chua, George Davida, Marc Girault, Burt Kaliski, David Kohel, Kenji Koyama, Masahiro Mambo, Alfred Menezes, Michael Merritt, Victor Miller, Peter Montgomery, François Morain, Willi More, Volker Müller, Richard Pinch, Mike Rose, Joe Silverman, Scott Vanstone and Moti Yung. Thanks also to all my colleagues and friends of the UCL Crypto Group* for many discussions on cryptography.

Most importantly, I want to thank my friend Hung-Mei for her patience during the writing of this thesis and my family for their encouragements.

Part of this work was supported by the European project ACCOPI (Prof. B. Macq and Prof. J.-J. Quisquater), the Action Concertée, the Communauté Française and the Fonds National de la Recherche Scientifique (F.N.R.S.). I have also benefited from teaching assistantships of the department of Mathematics of UCL.

*See <http://www.dice.ucl.ac.be/crypto/>.
Contents

Acknowledgements iii
List of Figures vii

Preface 1

Chapter 1. Mathematical Background 5
  1. Basic facts 5
  2. Eisenstein and Gaussian integers 7
     2.1. The ring $\mathbb{Z}[^1]$ 8
     2.2. The ring $\mathbb{Z}[i]$ 10
  3. Lucas sequences 12
     3.1. Definition and properties 12
     3.2. Dickson polynomials 15
  4. Elliptic curves 16
     4.1. Elliptic curves over a field 16
     4.2. Elliptic curves over a ring 20
     4.3. Division polynomials 22
  5. Lattice basis reduction 24

Chapter 2. RSA-type Cryptosystems 31
  1. RSA 31
  2. LUC 32
  3. Elliptic curve systems 33
     3.1. KMOV 33
     3.2. Demytko’s system 35

Chapter 3. Security Analysis 39
  1. Polynomial attacks 39
     1.1. Håstad’s attack 39
     1.2. GCD attack 46
     1.3. Garbage-man-in-the-middle attack 50
  2. Homomorphic attacks 56
     2.1. Chosen-message attack 57
2.2. Garbage-man-in-the-middle attack (II) 60
2.3. Common modulus attack 61
3. Other attacks 63
  3.1. Wiener’s attack 64
  3.2. Lenstra’s attack 66
  3.3. (Transient) faults based attack 69

Chapter 4. Summary and Conclusions 77

Bibliography 81

Appendix A. Index to Notations 101
Appendix B. Biography and Publications 105
Index 107
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Davida’s attack</td>
<td>2</td>
</tr>
<tr>
<td>0.2</td>
<td>Lenstra’s attack</td>
<td>3</td>
</tr>
<tr>
<td>1.1</td>
<td>Group law on $E(a, b)$</td>
<td>17</td>
</tr>
<tr>
<td>1.2</td>
<td>Gauss’ reduction algorithm</td>
<td>26</td>
</tr>
<tr>
<td>3.1</td>
<td>Basic Hástad’s attack</td>
<td>45</td>
</tr>
<tr>
<td>3.2</td>
<td>Coppersmith based variation</td>
<td>46</td>
</tr>
<tr>
<td>3.3</td>
<td>Maximum size of the public exponent $e$</td>
<td>50</td>
</tr>
<tr>
<td>3.4</td>
<td>Random tolerated padding $\Delta$</td>
<td>50</td>
</tr>
<tr>
<td>3.5</td>
<td>Garbage-man-in-the-middle attack</td>
<td>52</td>
</tr>
<tr>
<td>3.6</td>
<td>Order of a secure secret key $d$</td>
<td>66</td>
</tr>
<tr>
<td>3.7</td>
<td>Lenstra’s attack</td>
<td>67</td>
</tr>
<tr>
<td>4.1</td>
<td>Attacks against RSA-type systems</td>
<td>77</td>
</tr>
</tbody>
</table>
Cryptology, from the Greek kruptos (hidden) and logos (science), is the science of secure information. In particular, this science studies how two people, Alice and Bob, can exchange secret messages over an insecure channel. This can been achieved thanks to secret-key or public-key cryptography.

In secret-key cryptography, Alice and Bob have to share a common secret key $k$. Then, to send a message $m$ to Bob, Alice forms the ciphertext $c = E_k(M)$ where $E_k(\cdot)$ is the encryption function. To recover the plaintext $m$ from the ciphertext $c$, Bob computes $D_k(c) = D_k(E_k(m)) = m$, where $D_k(\cdot)$ is the decryption function. Example of such cryptosystems is DES [1], developed by IBM and later adopted by the US government as standard for unclassified data.

One drawback of secret-key systems is that they require the prior exchange of the secret key $k$. This assumes the existence of a secure channel. In 1976, Diffie and Hellman [97] overcome this limitation by introducing public-key cryptography. This one enables Alice and Bob to derive a secret key over an insecure channel. The idea behind public-key cryptography is concept of one-way function. A function $f$ is one-way if given $x$, one can easily compute $f(x)$; but given $f(x)$, it is infeasible to derive $x$. By infeasible, we mean computationally impossible.

In 1978, Rivest, Shamir and Adleman realized the first public-key cryptosystem, the so-called RSA [292]. This system is based on the difficulty of factoring large integers. It can briefly be described as follows. Suppose that Alice wants to send a message $m$ to Bob. To setup the system, Bob carefully selects two large primes $p$ and $q$, computes $n_B = pq$ and $\phi(n_B) = (p-1)(q-1)$, chooses an encryption key $e_B$ relatively prime to $\phi(n_B)$, and computes the decryption key $d_B = e_B^{-1} \mod \phi(n_B)$. The public key of Bob is the pair $(n_B, e_B)$ and his secret key is $d_B$. To send $m$ to Bob, Alice forms the ciphertext $c = m^{e_B} \mod n_B$ and sends it to Bob. Then, Bob recovers the plaintext by computing $m = c^{d_B} \mod n_B$ with his secret key $d_B$. 
However, the design of strong cryptoalgorithms is not sufficient to guarantee the security of information. We also have to deal with the soundness of the protocols using these algorithms. In some situations, a protocol may be completely subverted without compromising the security of the underpinning cryptoalgorithm. Such situations are called protocol failures [246]. Suppose that a pirate, say Carol, wants to recover the RSA-encrypted message $m$. She can proceed as follows. Carol chooses a random number $k$, intercepts the ciphertext $c$, replaces it by $c' = ck^{e_B} \mod n_B$, and sends $c'$ to Bob. Now, when Bob deciphers $c'$, he obtains $m' \equiv (c')^{d_B} \equiv mk \pmod{n_B}$. Since $m'$ is meaningless, Bob discards it. If Carol can get access to this discard, she finds $m = m'k^{-1} \mod n_B$. This failure was first pointed out by Davida [83].

![Figure 0.1: Davida’s attack](image)

Avoiding this attack is easy, users have to really destroy the discards, or in other words to protect their bins. The lesson of this failure is that the protocol designer has to pay attention to what is generally accepted but not explicitly stated.

The decryption process for RSA can be speeded up by the Chinese Remainder Theorem (CRT). From the secret factors $p$ and $q$, Bob computes $m_p = c^{d_B \mod (p-1)} \mod p$ and $m_q = c^{d_B \mod (q-1)} \mod q$, and finally finds $m = \text{CRT}(m_p, m_q)$ [283]. Suppose that Carol induces an external constraint on the deciphering device of Bob (e.g., ionizing or microwave radiation) so that the computation of $m_p$ is correctly performed but not computation of $m_q$. So, Bob gets $m' = \text{CRT}(m_p, m'_q)$ instead of $m$. If Bob discards $m'$ and if Carol can get access to $m'$, then she finds the secret factor $p$ by computing $\gcd((m')^{e_B} - c \mod n_B, n_B)$. Hence, $q = n_B/p$ and Carol can compute the secret decryption key $d_B$ [205, 156]. This
second attack is more dangerous because it completely breaks the system. This shows clearly the importance of checking cryptographic protocols for faults [40]. Note also that if Bob protects his bin, the attack does not remain applicable.

![Diagram](image)

**Figure 0.2:** Lenstra's attack

The two previous attacks show that it is extremely difficult for a protocol designer to determine whether his protocol is sound, even for very simple protocols. Some researchers proposed formal techniques for analyzing the soundness of protocols, such as the BAN logic [52] or the three systems presented in [173]. Another approach for the protocol designer is to try to find flaws in his protocol with all his experience of good and bad practice. In [2], Abadi and Needham give general rules helping protocol designers to avoid many of the pitfalls (see also [15]). In this thesis, we review for the first time all known attacks against RSA-type cryptosystems. We also present new failures. This may serve as guidelines to construct secure RSA-based protocols.

On the other hand, RSA was extended to other structures, including Lucas sequences and elliptic curves. A few authors have studied the underlying systems. These systems are sometimes claimed more resistant in some given contexts. My main contribution to this subject is to show that this is not always justified (see Table 4.1). Indeed, we will see that all major attacks against RSA can more or less successfully be extended to its analogues.

This thesis presumes almost no background in number theory or algebra. The relevant mathematics are introduced in Chapter 1. Chapter 2 reviews the RSA and its analogues based on Lucas sequences and elliptic curves. Chapter 3 is the main part of this thesis. We present attacks on RSA-type systems. These attacks are classified into three categories: polynomial attacks, homomorphic attacks and attacks resulting from a bad implementation. Finally, in Chapter 4, we compare the RSA-type
systems in terms of security. This can help to choose the most adequate system for a given application.

References


CHAPTER 1

Mathematical Background

There are numerous books devoted to the theory of numbers, good references are [137, 147, 182, 260, 295]. Computational aspects are treated in [67, 175].

For the Lucas sequences, we refer to [286, 287]. Also, the book of Lidl et al. [214] is an invaluable source on Dickson polynomials.

The classical reference for elliptic curves is [316] (see also [145, 174]). More accessible textbooks are [231, 317]. In [201], Lang gives a good introduction to division polynomials. Another introduction to division polynomials may be found in [59].

A lattice basis reduction algorithm was developed by Lenstra, Lenstra and Lovász [206]. The underlying mathematics are introduced in [55].

1. Basic facts

In this Section, we give some well-known results on number theory. All the proofs were omitted since they may be found in most textbooks on number theory (see [137] for example).

Theorem 1.1 (Lagrange’s Theorem). If \( a \) is an element of a (multiplicatively written) finite group \( G \) of order \( n \), then \( a^n = 1 \).

This beautiful theorem is a generalization of two theorems due to Fermat and Euler.

Theorem 1.2 (Fermat’s Little Theorem). If \( p \) is a prime and \( p \nmid a \), then

\[
(1.1) \quad a^{p-1} \equiv 1 \pmod{p}.
\]

Definition 1.3. The Euler’s totient function \( \phi(n) \) denotes the number of positive integers not greater than and relatively prime to \( n \).
Proposition 1.4. Let \( n = \prod_{i=1}^{r} p_i^{e_i} \) be the prime factorization of \( n \). Then,

\[
\phi(n) = p_1^{e_1}(1 - \frac{1}{p_1}) p_2^{e_2}(1 - \frac{1}{p_2}) \cdots p_r^{e_r}(1 - \frac{1}{p_r}) = n \prod_{p|n} (1 - \frac{1}{p}).
\]

Theorem 1.5 (Euler’s Theorem). If \( \gcd(a, n) = 1 \), then

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

Definition 1.6. Let \( a \) be an integer and let \( p \) be an odd prime. The Legendre symbol \( (a/p) \) is defined by

\[
(a/p) = \begin{cases} 
0 & \text{if } p \mid a, \\
1 & \text{if } a \text{ is a quadratic residue modulo } p, \\
-1 & \text{if } a \text{ is a quadratic non-residue modulo } p.
\end{cases}
\]

Proposition 1.7. For any odd prime \( p \),

\[
(a/p) \equiv a^{(p-1)/2} \pmod{p}.
\]

We usually do not use Eq. (1.4) to compute Legendre symbols. This can efficiently be achieved thanks to the next corollary and the Law of Quadratic Reciprocity, also known as the Gauss’ Theorem.

Corollary 1.8. The Legendre symbol satisfies the following properties:

(i) \( (a/p) = (a \mod p/p) \),

(ii) \( (ab/p) = (a/p)(b/p) \),

(iii) for \( b \) prime to \( a \), \( (ab^2/p) = (a/p) \),

(iv) \( (1/p) = 1 \) and \( (-1/p) = (-1)^{\frac{p-1}{2}} \).

Theorem 1.9 (Gauss’ Theorem). If \( p \) and \( q \) are two odd primes, then

\[
\left( \frac{2}{p} \right) = (-1)^{\frac{p^2-1}{8}},
\]

\[
\left( \frac{q}{p} \right) = (-1)^{\left( \frac{p-1}{2} \right) \left( \frac{q-1}{2} \right)} \left( \frac{p}{q} \right).
\]
One difficulty with this method of computing Legendre symbols is that it requires the factorization of the number on top. However, if we generalize Legendre symbols to Jacobi symbols, the Law of Quadratic Reciprocity remains valid for any two positive odd integers.

**Definition 1.10.** Let \( n \) be a positive odd integer. If \( n = \prod_{i=1}^{r} p_i^{e_i} \) is the prime factorization of \( n \), then the Jacobi symbol \( (a/n) \) is given by

\[
\left( \frac{a}{n} \right) = \prod_{i=1}^{r} \left( \frac{a}{p_i} \right)^{e_i}
\]

where \( (a/p_i) \) is the Legendre symbol of \( a \) modulo \( p_i \).

**Proposition 1.11.** If \( m \) and \( n \) are two positive odd integers, then

\[
\left( \frac{2}{n} \right) = (-1)^{\frac{n^2-1}{8}},
\]

\[
\left( \frac{n}{m} \right) = (-1)^{\left(\frac{m-1}{2}\right)\left(\frac{n-1}{2}\right)} \left( \frac{m}{n} \right).
\]
the Eisenstein integers, and the case \( d = 4 \) the Gaussian integers. It is well known that, for these choices of \( d \), these integers form unique factorization domains. So, we will analyze the units (i.e., invertible elements) and the primes for these new integers. We will also give the analogue of Fermat’s Little Theorem.

2.1. The ring \( \mathbb{Z}[^{\omega}] \).

**Definition 1.13.** Let \( \omega = e^{2\pi i/3} = \frac{1}{2}(-1 + \sqrt{-3}) \). The ring \( \mathbb{Z}[^{\omega}] \) of Eisenstein integers is the set of all numbers of the form \( \alpha = a + b\omega \) where \( a \) and \( b \) are rational integers.

Now, we study the units and the primes in \( \mathbb{Z}[^{\omega}] \).

**Lemma 1.14.** The units in \( \mathbb{Z}[^{\omega}] \) are \( \pm 1, \pm \omega \) and \( \pm \omega^2 \).

**Proof.** Let \( \alpha = a + b\omega = \frac{1}{2}((2a - b) + b\sqrt{-3}) \in \mathbb{Z}[^{\omega}] \), then \( N(\alpha) = \alpha \overline{\alpha} = \frac{1}{4}(2a - b)^2 + 3b^2 \geq 0 \). If \( \alpha \) is a unit, there exists \( \beta \in \mathbb{Z}[^{\omega}] \) such that \( \alpha \beta = 1 \). Hence, \( N(\alpha)N(\beta) = 1 \) and \( N(\alpha) = 1 \) since \( N(\alpha) \) and \( N(\beta) \) are positive integers. Therefore \( (2a - b)^2 + 3b^2 = 4 \). So, either \( 2a - b = \pm 2 \) and \( b = 0 \) or \( 2a - b = \pm 1 \) and \( b = \pm 1 \). This implies \( \alpha = \pm 1, \pm \omega \) or \( \pm (1 + \omega) = \mp \omega^2 \). \( \Box \)

It is important to note that the primes in \( \mathbb{Z} \) are not necessarily prime in \( \mathbb{Z}[^{\omega}] \). For example, \( 7 = (3 + \omega)(2 - \omega) \). The two following propositions tell more about the primes in \( \mathbb{Z}[^{\omega}] \).

**Proposition 1.15.** If \( p \equiv 2 \pmod{3} \) is a rational prime, then \( p \) is prime as an element of \( \mathbb{Z}[^{\omega}] \).

**Proof.** Suppose that \( p \) is not prime in \( \mathbb{Z}[^{\omega}] \), then \( p = \pi \gamma \) for some non-units \( \pi, \gamma \in \mathbb{Z}[^{\omega}] \). Therefore, \( N(\pi)N(\gamma) = p^2 \) and \( N(\pi) = p \). Writing \( \pi = a + b\omega \), we have \( p \equiv N(\pi) \equiv 4N(\pi) \equiv (2a - b)^2 \equiv 0, 1 \pmod{3} \), which is contrary to the hypothesis. \( \Box \)

**Proposition 1.16.** If \( p \equiv 1 \pmod{3} \) is a rational prime, then \( p = \pi \overline{\pi} \) where \( \pi \) is a prime in \( \mathbb{Z}[^{\omega}] \).

**Proof.** By Corollary 1.8 and Theorem 1.9,

\[
\left( \frac{-3}{p} \right) = \left( \frac{-1}{p} \right) \left( \frac{3}{p} \right) = (-1)^{(p-1)/2} \left( \frac{p}{3} \right) (-1)^{(p-1)/2} = \left( \frac{1}{3} \right) = 1.
\]

So, there exists \( x \in \mathbb{Z} \) such that \( x^2 \equiv -3 \pmod{p} \) \( \iff \) \( x^2 \equiv (2\omega + 1)^2 \pmod{p} \). Suppose \( p \) prime in \( \mathbb{Z}[^{\omega}] \), then \( p \mid (x + 1 + 2\omega) \) or \( p \mid (x - 1 - 2\omega) \). This implies \( bp = \pm 2 \) for some \( b \in \mathbb{Z} \), and thus \( p = 2 \). But \( p \equiv 1 \)
(mod 3), so \( p \) is not prime in \( \mathbb{Z}[\omega] \) and \( p = \pi \gamma \) where \( \pi \) and \( \gamma \) are non-units in \( \mathbb{Z}[\omega] \). Taking the norm, we obtain \( p^2 = N(\pi)N(\gamma) \) and thus \( p = N(\pi) = \pi \gamma \). It remains to prove that \( \pi \) is prime. If \( \pi \) was not prime, then \( \pi = \rho \theta \) for some non-units \( \rho \) and \( \theta \). So, \( p = N(\rho)N(\theta) \) which cannot be true since \( p \) is prime.

Therefore, if \( \pi \in \mathbb{Z}[\omega] \) is prime, then \( N(\pi) = p^2 \) or \( p \) for some rational prime \( p \). The first case corresponds to Proposition 1.15 and the second one to Proposition 1.16. The next proposition deals with the remaining case \( p \equiv 0 \pmod{3} \).

**Proposition 1.17.** \((1 - \omega)\) is prime in \( \mathbb{Z}[\omega] \).

**Proof.** Suppose that \( 1 - \omega = \pi \gamma \) for non-units \( \pi, \gamma \in \mathbb{Z}[\omega] \). Taking the norm, we get \( 3 = N(\pi)N(\gamma) \). This cannot be true since \( 3 \) is prime in \( \mathbb{Z} \).

The primes in \( \mathbb{Z}[\omega] \) are given up to multiplication by \( \pm 1, \pm \omega \) or \( \pm \omega^2 \). The notion of primary prime enables to distinguish among these primes.

**Definition 1.18.** Two integers \( \alpha, \beta \in \mathbb{Z}[\omega] \) are called associates if \( \alpha = \beta \nu \) for some unit \( \nu \in \mathbb{Z}[\omega] \).

**Definition 1.19.** Let \( \alpha \in \mathbb{Z}[\omega] \). If \( \alpha \equiv 2 \pmod{3} \), then \( \alpha \) is said primary.

**Proposition 1.20.** Let a prime \( \pi \in \mathbb{Z}[\omega] \). If \( N(\pi) \equiv 1 \pmod{3} \), then among the associates of \( \pi \), exactly one is primary.

**Proof.** Let \( \pi = a + b\omega \), then \( N(\pi) = a^2 - ab + b^2 \). The associates of \( \pi \) are

1. \( \pi = a + b\omega \);
2. \( -\pi = -a - b\omega \);
3. \( \omega \pi = -b + (a - b)\omega \);
4. \( -\omega \pi = b + (b - a)\omega \);
5. \( \omega^2 \pi = (b - a) - a\omega \);
6. \( -\omega^2 \pi = (a - b) + a\omega \).

Suppose first \( a \equiv 0 \pmod{3} \). Hence, \( a^2 - ab + b^2 \equiv b^2 \equiv 1 \pmod{3} \). So \( b \) mod 3 is either equal to 1 or to 2. If \( b \equiv 1 \pmod{3} \), then \( \pm(a - b) \equiv \pm 1 \pmod{3} \) and (6) is primary. If \( b \equiv 2 \pmod{3} \), then \( \pm(a - b) \equiv \pm 1 \pmod{3} \) and (5) is primary. The remaining cases \( (a \equiv 1, 2 \pmod{3}) \) are treated in a similar way.

Putting all together, we have:

**Theorem 1.21.** The primes in \( \mathbb{Z}[\omega] \) (up to multiplication by \( \pm 1, \pm \omega \) or \( \pm \omega^2 \)) are the rational primes congruent to 2 mod 3, \((1 - \omega)\), and the
integers of the form \(\pi = a + b\omega\) and \(\overline{\pi} = a + b\overline{\omega}^2\) such that \(a^2 - ab + b^2\) is a rational prime congruent to 1 mod 3. In the latter case, if \(N(\pi) \equiv 1\) (mod 3), then among the primes \(\pm \pi, \pm \omega \pi\) and \(\pm \omega^2 \pi\), exactly one is congruent to 2 mod 3. \(\square\)

The theorem of Fermat (Theorem 1.2) in \(\mathbb{Z}[\omega]\) becomes:

**Theorem 1.22.** If \(\pi\) is a prime in \(\mathbb{Z}[\omega]\) and if \(\pi \nmid \alpha\), then

\[
\alpha^{N(\pi)-1} \equiv 1 \pmod{\pi}.
\]

**Proof.** By Proposition 9.2.1 in [147, p. 111], \(K = \mathbb{Z}[\omega]/(\pi \mathbb{Z}[\omega])\) is a finite field with \(N(\pi)\) elements. Thus the multiplicative group of \(K\) has order \(N(\pi) - 1\). Using Theorem 1.1, this concludes the proof. \(\square\)

Finally, by analogy with the Legendre symbol, the cubic and sextic residue symbols are defined as follows.

**Definition 1.23.** Let \(\pi\) be a prime in \(\mathbb{Z}[\omega]\) with \(N(\pi) \neq 3\). If \(\pi \nmid \alpha\), then the cubic residue symbol \((\alpha/\pi)_3\) is defined to be \(\omega^j\) where \(j\) is the unique integer (modulo 3) satisfying

\[
\alpha^{(N(\pi)-1)/3} \equiv \omega^j \pmod{\pi},
\]

and the sextic residue symbol \((\alpha/\pi)_6\) is defined to be \((-\omega)^j\) where \(j\) is the unique integer (modulo 6) satisfying

\[
\alpha^{(N(\pi)-1)/6} \equiv (-\omega)^j \pmod{\pi}.
\]

If \(\pi \mid \alpha\), then \((\alpha/\pi)_3 = (\alpha/\pi)_6 = 0\).

**Remark 1.24.** Note that \(\{\pm 1, \pm \omega, \pm \omega^2\}\) is a cyclic group of order 6. So by Theorem 1.1, \(6 \mid (N(\pi) - 1)\).

**2.2. The ring \(\mathbb{Z}[i]\).** The Gaussian integers are constructed by adding \(i = e^{2\pi i/4}\) to the rational integers. We can now mimic the presentation of Eisenstein integers.

**Definition 1.25.** Let \(i = \sqrt{-1}\). The ring \(\mathbb{Z}[i]\) of Gaussian integers is the set of all numbers of the form \(\alpha = a + bi\) where \(a\) and \(b\) are rational integers.

**Lemma 1.26.** The units in \(\mathbb{Z}[i]\) are \(\pm 1\) and \(\pm i\).

**Proof.** Suppose that \(\alpha = a + bi\) is a unit, then \(N(\alpha) = a^2 + b^2 = 1\). So, either \(a = \pm 1\) and \(b = 0\) or \(a = 0\) and \(b = \pm 1\). It follows \(\alpha = \pm 1\) or \(\alpha = \pm i\). \(\square\)
Proposition 1.27. If \( p \equiv 3 \pmod{4} \) is a rational prime, then \( p \) is prime as an element of \( \mathbb{Z}[i] \).

Proof. Suppose that \( p \) is not prime in \( \mathbb{Z}[i] \), then \( p = \pi \gamma \) with \( N(\pi) = N(\gamma) = p \). Writing \( \pi = a + bi \), we have \( p \equiv N(\pi) \equiv a^2 + b^2 \equiv 0, 1, 2 \pmod{4} \), which is contrary to the hypothesis.

Proposition 1.28. If \( p \equiv 1 \pmod{4} \) is a rational prime, then \( p = \pi \overline{\pi} \) where \( \pi \) is a prime in \( \mathbb{Z}[i] \).

Proof. Since \( p \equiv 1 \pmod{4} \), \( (-1/p) = 1 \). So, \( x^2 \equiv -1 \pmod{p} \) for some \( x \in \mathbb{Z} \). If \( p \) was prime in \( \mathbb{Z}[i] \), then \( p \mid (x + i) \) or \( p \mid (x - i) \), which implies \( bp = \pm 1 \) for some \( b \in \mathbb{Z} \). Therefore, \( p = \pi \gamma \) with \( N(\pi) = N(\gamma) = p \) and \( p = N(\pi) = \pi \overline{\pi} \). Furthermore, since \( N(\pi) \) is prime, it follows that \( \pi \) is prime in \( \mathbb{Z}[i] \).

Proposition 1.29. \( (1 + i) \) is prime in \( \mathbb{Z}[i] \).

Proof. Suppose that \( (1 + i) = \pi \gamma \) for some non-units \( \pi, \gamma \in \mathbb{Z}[i] \). This implies \( 2 = N(\pi)N(\gamma) \), which is impossible.

Definition 1.30. A non-unit \( \alpha \in \mathbb{Z}[i] \) is said primary if \( \alpha \equiv 1 \pmod{2 + 2i} \).

Lemma 1.31. A non-unit \( \alpha = a + bi \) is primary if and only if either \( a \equiv 1 \pmod{4} \) and \( b \equiv 0 \pmod{4} \) or \( a \equiv 3 \pmod{4} \) and \( b \equiv 2 \pmod{4} \).

Proof. Obvious, since \( a + bi \equiv 1 \pmod{2 + 2i} \) implies that \( a = 2k + 1 \) and \( b = 2k \) for some \( k \in \mathbb{Z} \).

Proposition 1.32. Let a prime \( \pi \in \mathbb{Z}[i] \). If \( N(\pi) \equiv 1 \pmod{4} \), then among the associates \( \{ \pm \pi, \pm i \pi \} \) of \( \pi \), exactly one is primary.

Proof. Let \( \pi = a + bi \) with \( a^2 + b^2 \equiv 1 \pmod{4} \). Its associates are

1. \( \pi = a + bi \);
2. \( -\pi = -a - bi \);
3. \( i\pi = -b + ai \);
4. \( -i\pi = b - ai \).

If \( a \equiv 0 \pmod{4} \), then either \( b \equiv 1 \pmod{4} \) and (4) is primary, or \( b \equiv 3 \pmod{4} \) and (3) is primary. The remaining cases are proceeded in a similar way.

More succinctly, we have shown:

Theorem 1.33. The primes in \( \mathbb{Z}[i] \) (up to multiplication by \( \pm 1 \) or \( \pm i \)) are the rational primes congruent to 3 mod 4, \( (1+i) \), and the integers of the form \( \pi = a + bi \) and \( \overline{\pi} = a - bi \) such that \( a^2 + b^2 \) is a rational prime.
congruent to 1 mod 4. In the latter case, if $N(\pi) \equiv 1 \pmod{4}$, then among the primes $\pm \pi$ and $\pm 2i$, exactly one is congruent to 1 mod $2 + 2i$.

\begin{theorem}
If $\pi$ is a prime in $\mathbb{Z}[i]$ and if $\pi \nmid \alpha$, then
\begin{equation}
\alpha^{N(\pi)-1} \equiv 1 \pmod{\pi}.
\end{equation}
\end{theorem}

\begin{proof}
The proof is identical to that of Theorem 1.22 because $\mathbb{Z}[i]/(\pi \mathbb{Z}[i])$ is a finite field with $N(\pi)$ elements from Proposition 9.8.1 in [147, p. 121].
\end{proof}

\begin{definition}
Let $\pi$ be a prime in $\mathbb{Z}[i]$ with $N(\pi) \neq 2$. If $\pi \nmid \alpha$, then the quartic residue symbol $(\alpha/\pi)_4$ is defined to $i^j$ where $j$ is the unique integer (modulo 4) satisfying
\begin{equation}
\alpha^{(N(\pi)-1)/4} \equiv i^j \pmod{\pi}.
\end{equation}
If $\pi \mid \alpha$, then $(\alpha/\pi)_4 = 0$.
\end{definition}

\section{3. Lucas sequences}
Lucas sequences have numerous applications in mathematics. For example, they can be used to construct compositeness and primality tests. They also play a role in the construction of irreducibles and optimal normal bases. Lucas sequences can be used to evaluate the determinants of certain circulant matrices. They allow to identify some properties of Kloosterman and Brewer character sums. The construction of sets of $q-1$ pairwise Latin squares of order $q$ can be achieved thanks to Lucas sequences. We refer to [214, Chapter 6] for a description of these applications. Note that these applications are described in terms of Dickson polynomials; but, as we will see, this formulation is equivalent. Moreover, using Weil’s Theorem, Lucas sequences enable to efficiently compute the number of points on an elliptic curve over $GF(2^m)$ [159]. Finally, Lucas sequences play a role in the design of cryptosystems [330, 331].

In this Section, we give the definition of Lucas sequences and derive some properties that will be useful later. We also show the connection between Lucas sequences and Dickson polynomials.

\subsection{3.1. Definition and properties.}

\begin{definition}
Let $P, Q$ be rational integers and let $\Delta = P^2 - 4Q$ be a non-square. If $\alpha = \frac{P + \sqrt{\Delta}}{2}$ and $\beta = \frac{P - \sqrt{\Delta}}{2}$ are the roots of
\[ x^2 - Px + Q = 0 \] in the quadratic field \( \mathbb{Q}(\sqrt{\Delta}) \), then the Lucas sequences \( \{U_k\}_{k \geq 0} \) and \( \{V_k\}_{k \geq 0} \) are the rational integers satisfying

\[
V_i + U_i \sqrt{\Delta} = 2\alpha^i.
\] (1.16)

**Notation.** The \( k \)th terms of the Lucas sequences \( \{U_i\}_{i \geq 0} \) and \( \{V_i\}_{i \geq 0} \) with parameters \( P \) and \( Q \) will respectively be denoted by \( U_k(P, Q) \) and \( V_{k}(P, Q) \).

Remark that since \( \alpha \in \mathbb{Q}(\sqrt{\Delta}) \) is a root of \( x^2 - Px + Q = 0 \), \( \alpha \) is an element of \( \mathcal{O}_{\sqrt{\Delta}} \), the ring of integers of the field \( \mathbb{Q}(\sqrt{\Delta}) \).

From Eq. (1.16) and since \( V_i - U_i \sqrt{\Delta} = 2\beta^i \), it follows that

\[
U_i = \frac{\alpha^i - \beta^i}{\alpha - \beta} \quad \text{and} \quad V_i = \alpha^i + \beta^i.
\] (1.17)

This relation is sometimes used as an alternative definition of Lucas sequences. This second definition enables to efficiently compute the Lucas sequences. We have

\[
U_{k+m} = \frac{\alpha^{k+m} - \beta^{k+m}}{\alpha - \beta}
\]

\[
= \frac{(\alpha^k - \beta^k)(\alpha^m + \beta^m)}{\alpha - \beta} - \frac{\alpha^m \beta^m(\alpha^{k-m} - \beta^{k-m})}{\alpha - \beta}
\]

\[
= U_k V_m - Q^m U_{k-m},
\] (1.18)

and

\[
V_{k+m} = \alpha^{k+m} + \beta^{k+m}
\]

\[
= (\alpha^k + \beta^k)(\alpha^m + \beta^m) - \alpha^m \beta^m(\alpha^{k-m} + \beta^{k-m})
\]

\[
= V_k V_m - Q^m V_{k-m}.
\] (1.19)

In particular, taking \( m = 1 \), we obtain

\[
U_{k+1} = PU_k - QU_{k-1} \quad \text{and} \quad V_{k+1} = PV_k - QV_{k-1}.
\]

So, using matrix notations, we finally get

\[
\begin{pmatrix}
U_{k+1} & V_{k+1} \\
U_k & V_k
\end{pmatrix} =
\begin{pmatrix}
P & -Q \\
1 & 0
\end{pmatrix}
\begin{pmatrix}
U_k & V_k \\
U_{k-1} & V_{k-1}
\end{pmatrix}
\]

\[
= \begin{pmatrix}
P & -Q \\
1 & 0
\end{pmatrix}^k
\begin{pmatrix}
U_1 & V_1 \\
U_0 & V_0
\end{pmatrix}.
\] (1.20)

**Lemma 1.37.** Any pair \( (U_i, V_i) \) with parameters \( P \) and \( Q \) satisfies

\[
V_i^2 - \Delta U_i^2 = 4Q^i.
\] (1.21)
PROOF. By definition, 
\[ V_i^2 - \Delta U_i^2 = (V_i + U_i \sqrt{\Delta})(V_i - U_i \sqrt{\Delta}) = 2\alpha^i 2\alpha^i = 4(\alpha \overline{\alpha})^i = 4Q^i. \]
\[ \Box \]

**Proposition 1.38.** Let \( U_i(P, Q) \) and \( V_i(P, Q) \) be the \( i \text{th} \) terms of the Lucas sequences with parameters \( P, Q \) and \( \Delta = P^2 - 4Q \). Then,

\[ U_{km}(P, Q) = U_k(P, Q)U_m(V_k(P, Q), Q^k), \tag{1.22} \]
\[ V_{km}(P, Q) = V_m(V_k(P, Q), Q^k), \tag{1.23} \]
\[ 2U_{k+m}(P, Q) = U_k(P, Q)V_m(P, Q) + U_m(P, Q)V_k(P, Q), \tag{1.24} \]
\[ 2V_{k+m}(P, Q) = V_k(P, Q)V_m(P, Q) + \Delta U_k(P, Q)U_m(P, Q), \tag{1.25} \]
\[ V_{-k}(P, Q) = Q^{-k}V_k(P, Q), \tag{1.26} \]
\[ U_{-k}(P, Q) = -Q^{-k}U_k(P, Q). \tag{1.27} \]

**Proof.** By Lemma 1.37 and letting \( P' = V_k(P, Q) \) and \( Q' = Q^k \), we have \( \Delta' := P'^2 - 4Q' = \Delta U_k^2(P, Q) \). Hence,

\[ 2\alpha^{km} = 2^{1-m}(2\alpha^k)^m = 2^{1-m}(V_k(P, Q) + U_k(P, Q)\sqrt{\Delta})^m \]
\[ = 2^{1-m}(P' + \sqrt{\Delta'})^m = 2^{1-m}Q^k = V_m(P', Q') + U_m(P', Q')\sqrt{\Delta'} \]
\[ = V_m(V_k(P, Q), Q^k) + U_m(V_k(P, Q), Q^k)U_k(P, Q)\sqrt{\Delta}, \]

which proves Eqs (1.22) and (1.23) by comparison with the coefficients of 
\[ 2\alpha^{km} = V_{km}(P, Q) + U_{km}(P, Q)\sqrt{\Delta}. \] 

The two next equations are proved in a similar way,

\[ 4\alpha^{k+m} = 2\alpha^k 2\alpha^m \]
\[ = (V_k(P, Q) + U_k(P, Q)\sqrt{\Delta})(V_m(P, Q) + U_m(P, Q)\sqrt{\Delta}) \]
\[ = V_k(P, Q)V_m(P, Q) + U_k(P, Q)U_m(P, Q)\Delta \]
\[ \qquad + (U_k(P, Q)V_m(P, Q) + U_m(P, Q)V_k(P, Q))\sqrt{\Delta}. \]

Finally, from \( \alpha \overline{\alpha} = Q \), we have

\[ V_{-k}(P, Q) + U_{-k}(P, Q)\sqrt{\Delta} = 2\alpha^{-k} = 2\alpha^k/Q^k \]
\[ = \frac{V_k(P, Q)}{Q^k} - \frac{U_k(P, Q)\sqrt{\Delta}}{Q^k}. \]
\[ \Box \]
THEOREM 1.39. For any \( \alpha \in \mathbb{O}\sqrt{\Delta} \),

\[
\alpha^p \mod p = \begin{cases} 
\alpha & \text{if } (\Delta/p) = 1, \\
\overline{\alpha} & \text{if } (\Delta/p) = -1,
\end{cases}
\]

where \((\Delta/p)\) denotes the Legendre symbol.

PROOF. From Proposition 1.7, \( \frac{\Delta^{p-1}}{2} \equiv (\Delta/p) \pmod{p} \). Therefore,

\[
\alpha^p \equiv \frac{(P + \sqrt{\Delta})^p}{2^p} \equiv \frac{1}{2} \sum_{i=0}^{p} \binom{p}{i} P^i (\sqrt{\Delta})^{p-i} 
\equiv \frac{P^p + (\sqrt{\Delta})^p}{2} \equiv \frac{P + (\Delta/p)\sqrt{\Delta}}{2} \pmod{p}.
\]

\[\Box\]

COROLLARY 1.40. If \( p \) is an odd prime that does divide \( \Delta \), then

\[
\begin{cases} 
U_{p-(\Delta/p)}(P,1) \equiv 0 \pmod{p}, \\
V_{p-(\Delta/p)}(P,1) \equiv 2 \pmod{p}.
\end{cases}
\]

PROOF. Let \( \alpha \) be a non-zero integer of \( \mathbb{O}\sqrt{\Delta} \). From Theorem 1.39 follows \( \alpha^{p-1} \equiv 1 \pmod{p} \) if \( (\Delta/p) = 1 \) and \( \alpha^{p+1} \equiv \alpha Q \equiv 1 \) if \( (\Delta/p) = -1 \). Hence, by Eq. (1.16), \( V_{p-(\Delta/p)}(P,1) + U_{p-(\Delta/p)}(P,1)\sqrt{\Delta} \equiv 2 \pmod{p} \). \[\Box\]

3.2. Dickson polynomials. These polynomials were introduced by Dickson in [96]. They enable to consider Lucas sequences as polynomials.

DEFINITION 1.41. The Dickson polynomials of the first kind, denoted by \( D_n(x,\alpha) \), are given by

\[
D_n(x,\alpha) = \sum_{i=0}^{[n/2]} \binom{n}{i} \binom{n-i}{i} (-\alpha)^i x^{n-2i}.
\]

The Dickson polynomials of the second kind \( E_n(x,\alpha) \) are given by

\[
E_n(x,\alpha) = \sum_{i=0}^{[n/2]} \binom{n}{i} (-\alpha)^i x^{n-2i}.
\]

The next proposition shows that Lucas sequences and Dickson polynomials are essentially the same functions.
Proposition 1.42. The Dickson polynomials and the Lucas sequences are connected by the following relations
\begin{equation}
D_n(x, a) = V_n(x, a) \quad \text{and} \quad E_n(x, a) = U_{n+1}(x, a).
\end{equation}

Proof. We shall only prove the first equality. Using the same notations as in Definition 1.36, we have $V_n(P, Q) = \alpha^n + \beta^n$. By Waring’s formula \[216, p. 30], we obtain
\[
\alpha^n + \beta^n = \sum_{i_1 + 2i_2 = n} (-1)^{i_2} \frac{(i_1 + i_2 - 1)! n}{i_1! i_2!} (\alpha + \beta)^{i_1} (\alpha \beta)^{i_2}
\]
\[
= \sum_{i_1 + 2i_2 = n} \frac{(i_1 + i_2 - 1)! n}{i_1! i_2!} P_{i_1}(-Q)^{i_2}
\]
\[
= \sum_{i_2=0}^{[n/2]} \frac{n}{n - i_2} \binom{n - i_2}{i_2} P_{n-2i_2}(-Q)^{i_2} = D_n(P, Q).
\]

4. Elliptic curves

Elliptic curves is one of the oldest and most fascinating branch in mathematics. They recently gained more interest thanks to cryptography. All began when Lenstra discovered a factorization algorithm over these structures \[209\]. Thereafter, Koblitz \[177\] and Miller \[240\] independently proposed to adapt existing cryptographic protocols on elliptic curves.

The theory of elliptic curves is quite difficult; we will only introduce the rudiments that will be useful for our purposes. The interested reader may for example consult \[316\] for further study. In this Section, we recall the definition of an elliptic curve over a field and over a ring. We also give some basic properties. Finally, we introduce the division polynomials.

4.1. Elliptic curves over a field.

Definition 1.43. Let \( K \) be a field of characteristic \( \neq 2, 3 \), and let \( x^3 + ax + b \) (where \( a, b \in K \)) be a cubic with no multiple roots. An elliptic curve \( E(a, b) \) over \( K \) is the set of points \( (x, y) \in K \times K \) satisfying the Weierstraß equation
\begin{equation}
y^2 = x^3 + ax + b
\end{equation}

\( y^2 = x^3 + ax + b \)
together with a single element denoted \( O_K \) and called the point at infinity.
Remark 1.44. If $K$ is the prime field $\mathbb{F}_p$, then the equation $x^3 + ax + b$ has no multiple roots if and only if $\gcd(4a^3 + 27b^2, p) = 1$.

Let $P, Q \in E(a, b)$, let $\ell$ be the line connecting $P$ and $Q$ (tangent line if $P = Q$), and let $T$ be the third point of intersection of $\ell$ with $E(a, b)$. If $\ell'$ is the line connecting $T$ and $O_K$, then $P + Q$ is the point such that $\ell'$ intersects $E(a, b)$ at $T, O_K$ and $P + Q$.

Proposition 1.45. The previous composition law makes $E(a, b)$ into an Abelian group with identity element $O_K$.

Proof. Elementary proofs of this proposition are quite long (see [152] for example). However this proposition can also be seen as a consequence of Abel’s Theorem [221, 9].

Algebraically, we have

(i) $O_K$ is the identity element, i.e. $\forall P \in E(a, b), P + O_K = P$.

(ii) The inverse of $P = (x_1, y_1)$ is $-P = (x_1, -y_1)$.

(iii) Let $P = (x_1, y_1)$ and $Q = (x_2, y_2) \in E(a, b)$ with $P \neq -Q$.

Then $P + Q = (x_3, y_3)$ where

\begin{align}
  x_3 &= \lambda^2 - x_1 - x_2 \\
  y_3 &= \lambda(x_1 - x_3) - y_1
\end{align}

and $\lambda = \begin{cases} 
  \frac{3x_1^2 + a}{2y_1} & \text{if } x_1 = x_2, \\
  \frac{y_1 - y_2}{x_1 - x_2} & \text{otherwise}.
\end{cases}$
Note that if $P = (x_1, 0) \in E(a, b)$, then $[2]P = O_K$.

In the sequel, we will see some properties of the order and the structure of elliptic curves over finite prime fields.

**Theorem 1.46 (Hasse).** If $K$ is the finite prime field $\mathbb{F}_p$, then the order of the group $E_p(a, b)$ (i.e. the elliptic curve $E(a, b)$ over $\mathbb{F}_p$) is given by

$$
\#E_p(a, b) = p + 1 - a_p
$$

where $|a_p| \leq 2\sqrt{p}$.

**Proof.** See [316, p. 131], Theorem 1.1.

**Theorem 1.47.** The group $E_p(a, b)$ is either cyclic or isomorphic to a product of two cyclic groups. Furthermore, we can write $E_p(a, b) \cong \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$ with $n_2 | n_1$ and $n_2 | p - 1$.

**Proof.** The proof involves higher mathematics. The required material can be found in [56].

**Corollary 1.48.** In $E_p(a, b)$, the number of points of order dividing $j$ is equal to

$$
gcd(j, n_1) \cdot gcd(j, n_2).
$$

**Proof.** From Theorem 1.47, we can write

$$
E_p(a, b) = \{P = [u]R + [v]S \mid 0 \leq u < n_1 \text{ and } 0 \leq n_2 < v\}.
$$

Hence $[j]P = [ju]R + [jv]S = O_p$ if and only if $ju \equiv 0 \pmod{n_1}$ and $jv \equiv 0 \pmod{n_2}$. There are thus $gcd(j, n_1) \cdot gcd(j, n_2)$ points of order dividing $j$.

**Definition 1.49.** Let $E_p(a, b)$ be an elliptic curve over the prime field $\mathbb{F}_p$. Let $D_p$ be a quadratic non-residue modulo $p$. The *complementary group* of $E_p(a, b)$, denoted by $\overline{E_p(a, b)}$, is the elliptic curve given by the (extended) Weierstraß equation

$$
D_p y^2 = x^3 + ax + b
$$

together with the point at infinity $O_p$. The sum of two points (that are not inverse of each other) $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$ can be computed by

$$
x_3 = \lambda^2 D_p - x_1 - x_2
$$

$$
y_3 = \lambda(x_1 - x_3) - y_1
$$
and \( \lambda = \begin{cases} 3x_1^2 + a & \text{if } x_1 = x_2, \\ \frac{2Dpy_1}{y_1 - y_2} & \text{otherwise}. \end{cases} \)

**Corollary 1.50.** If \( \#E_p(a, b) = p + 1 - a_p \), then \( \#\overline{E_p(a, b)} = p + 1 + a_p \).

**Proof.** Letting \( a_p = -\sum_{x \in \mathbb{F}_p} \left( 1 + \left( x^3 + ax + b/p \right) \right) \) and counting the point at infinity, we have
\[
\#E_p(a, b) = 1 + \sum_{x \in \mathbb{F}_p} \left( 1 + \left( \frac{x^3 + ax + b}{p} \right) \right) = 1 + p - a_p.
\]
Hence,
\[
\#\overline{E_p(a, b)} = 1 + \sum_{x \in \mathbb{F}_p} \left( 1 - \left( \frac{x^3 + ax + b}{p} \right) \right) = 1 + p + a_p.
\]

For some special cases, the order and the structure of an elliptic curve can easily be determined.

**Lemma 1.51.** Let \( p \) be an odd prime congruent to 2 mod 3. Then the elliptic curve \( E_p(0, b) \) is a cyclic group of order \( p + 1 \).

**Proof.** Since \( p \equiv 2 \pmod{3} \), cube roots exist and are unique. Consequently, there are \( p + 1 \) points on \( E_p(0, b) \), namely \( p \) points of the form \( \left( \sqrt[3]{y^2 - b}, y \right) \) with \( y \in \mathbb{F}_p \) and the point at infinity.

By Theorem 1.47, \( E_p(0, b) \cong \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \) with \( n_1 \mid n_2 \) and \( n_1, n_2 \mid p - 1 \). Since \( n_1n_2 = p + 1 \), \( n_2 \) is equal to 1 or to 2. But \( n_2 \) cannot be 2 because \( E_p(0, b) \) has only 2 points of order dividing 2, namely \( (\sqrt[3]{-b}, 0) \) and \( O_p \).

Therefore, by Corollary 1.48, \( n_2 = 1 \) and \( E_p(0, b) \) is cyclic.

**Lemma 1.52.** Let \( p \) be a prime congruent to 3 mod 4. If \( a \) is a quadratic residue modulo \( p \), then \( E_p(a, 0) \) is a cyclic group of order \( p + 1 \). If \( a \) is a quadratic non-residue modulo \( p \), then \( E_p(a, 0) \) is a group isomorphic to \( \mathbb{Z}_{(p+1)/2} \times \mathbb{Z}_2 \) of order \( p + 1 \).

**Proof.** Since \( p \equiv 3 \pmod{4} \), \( -1/p = -1 \) and
\[
\left( \frac{-x^3 + ax}{p} \right) = \left( \frac{-1}{p} \right) \left( \frac{x^3 + ax}{p} \right) = -\left( \frac{x^3 + ax}{p} \right).
\]
So, \( \sum_{x \in \mathbb{F}_p} \left( x^3 + ax/p \right) = 0 \) because the term for \( x \) and the term for \( -x \) cancel in the sum. Therefore, \( a_p = 0 \) and \( \#E(a, 0) = p + 1 \).
As in the proof of Lemma 1.51, \( E_p(a,0) \cong \mathbb{Z}/n_1 \times \mathbb{Z}/n_2 \) with \( n_2 = 1 \) or 2. If \( (a/p) = 1 \), then the equation \( x^3 + ax \equiv 0 \pmod{p} \) has only the solution \( x = 0 \) because \( p \equiv 3 \pmod{4} \) implies \( (-a/p) = -1 \). Therefore, \( E_p(a,0) \) has only two points of order dividing 2, namely \((0,0)\) and \( \mathcal{O}_p \), and \( E_p(a,0) \) is cyclic. Otherwise, if \( (a/p) = -1 \), the equation \( x^3 + ax \equiv 0 \pmod{p} \) has 3 solutions, namely \( x = 0 \) and \( x = \pm \sqrt{a} \). Hence, \( E_p(a,0) \) has 4 points of order dividing 2, namely \((0,0)\), \((\pm \sqrt{a},0)\) and \( \mathcal{O}_p \), and \( n_2 = 2 \) by Corollary 1.48.

**Lemma 1.53.** Let \( p \) be a prime congruent to 1 modulo 3. If \( p = \pi_p \pi_p' \) with \( \pi_p \in \mathbb{Z}[w] \) and \( \pi_p \equiv 2 \pmod{3} \), then

\[
1.39 \quad #E_p(0,b) = p + 1 + \left( \frac{4b}{\pi_p} \right)_6 \pi_p + \left( \frac{4b}{\pi_p} \right)_6 \pi_p'.
\]

**Proof.** See [147, p. 305], Theorem 4.

**Lemma 1.54.** Let \( p \) be prime congruent to 1 modulo 4. If \( p = \pi_p \pi_p' \) with \( \pi_p \in \mathbb{Z}[i] \) and \( \pi_p \equiv 1 \pmod{2 + 2i} \), then

\[
1.40 \quad #E_p(a,0) = p + 1 - \left( \frac{-a}{\pi_p} \right)_4 \pi_p - \left( \frac{-a}{\pi_p} \right)_4 \pi_p'.
\]

**Proof.** See [147, p. 307], Theorem 5.

### 4.2. Elliptic curves over a ring

Elliptic curves over rings are defined similarly as over the field \( \mathbb{F}_p \). The main difference is that they do not define an Abelian group.

**Definition 1.55.** Let \( n \) be the product of two primes \( p \) and \( q \), and let \( a, b \) such that \( \gcd(4a^3 + 27b^2, n) = 1 \). An elliptic curve \( E_n(a,b) \) over the ring \( \mathbb{Z}/n \) is the set of the points \((x,y) \in \mathbb{Z}/n \times \mathbb{Z}/n \) satisfying the equation

\[
1.41 \quad y^2 = x^3 + ax + b
\]

together with the point \( \mathcal{O}_n \).

Consider the group \( \bar{E}_n(a,b) \) given by the direct product

\[
1.42 \quad \bar{E}_n(a,b) = E_p(a,b) \times E_q(a,b).
\]

By Theorem 1.12, there exists a unique point \( P = (x_1,y_1) \in E_n(a,b) \) for every pair of points \( P_p = (x_{1p},y_{1p}) \in E_p(a,b) \setminus \{ \mathcal{O}_p \} \) and \( P_q = (x_{1q},y_{1q}) \in E_q(a,b) \setminus \{ \mathcal{O}_q \} \) such that \( x_1 \pmod{p} = x_{1p} \), \( x_1 \pmod{q} = x_{1q} \), \( y_1 \pmod{p} = y_{1p} \), and \( y_1 \pmod{q} = y_{1q} \). This equivalence will be denoted by \( P = [P_p,P_q] \). Since \( \mathcal{O}_n = [\mathcal{O}_p,\mathcal{O}_q] \), the group \( \bar{E}_n(a,b) \) consists of
all the points of $E_n(a, b)$ together with a number of points of the form $[P_p, O_q]$ or $[O_p, P_q]$.

**Lemma 1.56.** The tangent-and-chord addition on $E_n(a, b)$, whenever it is defined, coincides with the group operation on $E_n(a, b)$.

**Proof.** Let $P$ and $Q \in E_n(a, b)$. Assume $P + Q$ is well-defined by the tangent-and-chord rule. Therefore $P + Q = [(P + Q)_p, (P + Q)_q] = [P_p + Q_p, P_q + Q_q]$. □

If $n$ is the product of two large primes, it is extremely unlikely that the “addition” is not defined on $E_n(a, b)$. Consequently, computations in $E_n(a, b)$ can be performed without knowing the two prime factors of $n$.

Although $E_n(a, b)$ is not a group, a proposition similar to Lagrange’s Theorem holds.

**Proposition 1.57.** Let $n = pq$ and let $E_n(a, b)$ be an elliptic curve over $\mathbb{Z}_n$. If $N_n = \text{lcm}(\#E_p(a, b), \#E_q(a, b))$, then

$$\forall P \in E_n(a, b), \forall k \in \mathbb{Z} : [kN_n + 1]P = P,$$

with the overwhelming probability for large $p$ and $q$.

**Proof.** Since $[kN_n + 1]P_p = P_p$ and $[kN_n + 1]P_q = P_q$, $[kN_n + 1]P = P$ by Lemma 1.56.

Multiple of points can be computed in polynomial time (see for example [249]). Therefore, there exists a polynomial chain of intermediate points $P_j := [j]P$ from which we can compute $[kN_n + 1]P$. The computation of $[kN_n + 1]P$ cannot be achieved if one of the points $P_j$ is of the form $[O_p, P_q]$ or $[P_j, O_q]$. Let us analyze the probability that some $P_j$ is equal to $O_p$. From Theorem 1.47, we can write

$$E_p(a, b) = \{P_p = [u]R + [v]S \mid 0 \leq u < n_1 \text{ and } 0 \leq v < n_2\}.$$

Hence, from Corollary 1.48 and assuming that $P_p \neq O_p$, the probability that $P_j = O_p$ is equal to

$$\frac{\gcd(n_1, j) \times \gcd(n_2, j) - 1}{n_1n_2 - 1}.$$

So, since the chain of intermediate points is polynomial while the number of “bad” points $P_j$ in this chain is exponentially small, the probability that $[kN_n + 1]P$ cannot be computed is negligible. □
4.3. Division polynomials. Division polynomials play the same role as Dickson polynomials for Lucas sequences. They enable to express multiples of points in terms of polynomials.

**Definition 1.58.** Let $E_n(a, b)$ be an elliptic curve over the ring $\mathbb{Z}_n$. Consider the polynomials $P_i \in \mathbb{Z}_n[a, b, x]$ defined by

(i) $P_0 = 0$,
(ii) $P_1 = 1$,
(iii) $P_2 = 1$,
(iv) $P_3 = 3x^4 + 6ax^2 + 12bx - a^2$,
(v) $P_4 = 2(x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - 8b^2 - a^3)$,
(vi) for $i > 2$,

$$P_{2i} = P_i(P_{i+2}P_{i-1}^2 - P_{i-2}P_{i+1}^2),$$

and for $i \geq 2$,

$$P_{2i+1} = \begin{cases} P_{i+2}^3 - P_{i+1}P_{i-1}^2 & \text{if } i \text{ is odd,} \\ 16(x^3 + ax + b)^2P_{i+2}P_i^3 - P_{i+1}P_{i-1}^2 & \text{if } i \text{ is even.} \end{cases}$$

Letting $\Psi_i(x, y) = \left\{ \begin{array}{ll} P_i(x) & \text{if } i \text{ is odd} \\ 2yP_i(x) & \text{if } i \text{ is even} \end{array} \right.$, define the polynomials $\Phi_i$ and $\omega_i \in \mathbb{Z}_n[a, b, x, y]$ by

$$\Phi_i = x\Psi_i^2 - \Psi_{i+1}\Psi_{i-1},$$
$$4y\omega_i = \Psi_{i+2}\Psi_{i-1}^2 - \Psi_{i-2}\Psi_{i+1}^2.$$

One of the most important results about division polynomials is given in the next proposition.

**Proposition 1.59.** Let $E_n(a, b)$ be an elliptic curve over $\mathbb{Z}_n$.

(a) $\Psi_i, \Phi_i, y^{-1}\omega_i$ (for $i$ odd) and $y^{-1}\Psi_i, \Phi_i, \omega_i$ (for $i$ even) are polynomials in $\mathbb{Z}_n[a, b, x, y^2]$. Hence, by replacing $y^2$ by $x^3 + ax + b$, they will be considered as polynomials in $\mathbb{Z}_n[a, b, x]$.

(b) As polynomials in $x$,

$$\Phi_k(x) = x^{k^2} + \text{lower order terms},$$
$$\Psi_k(x)^2 = k^2x^{k^2-1} + \text{lower order terms}.$$

(c) If $P \in E_n(a, b)$, then

$$Q = [k]P = \left( \frac{\Phi_k(P)}{\Psi_k(P)^2}, \frac{\omega_k(P)}{\Psi_k(P)^2}, \frac{\Psi_k(P)}{\Psi_k(P)^3} \right) \pmod n.$$

**Proof.** See [201, p. 38], Theorem 2.1. \qed
Corollary 1.60. Let $F_i, G_i$ and $H_i$ be the rational functions defined by

(i) $F_1(x) = 1$ and $G_1(x) = x \mod n$,
(ii) for $i \geq 2$,

$$F_i(x) = \begin{cases} 
\frac{P_{i+2}(x)P_{i-1}(x)^2 - P_{i-2}(x)P_{i+1}(x)^2}{P_i(x)^3} \mod n & \text{for } i \text{ odd} \\
\frac{16(x^3 + ax + b)^2 P_i(x)^3}{P_{i+2}(x)P_{i-1}(x)^2 - P_{i-2}(x)P_{i+1}(x)^2} \mod n & \text{for } i \text{ even}
\end{cases}$$

$$G_i(x) = \begin{cases} 
x - 4(x^3 + ax + b) \frac{P_{i+1}(x)P_{i-1}(x)}{P_i(x)^2} \mod n & \text{for } i \text{ odd} \\
x - \frac{1}{4(x^3 + ax + b)} \frac{P_{i+1}(x)P_{i-1}(x)}{P_i(x)^2} \mod n & \text{for } i \text{ even}
\end{cases}$$

$H_i(x, y) = y F_i(x) \mod n$.

If $P = (p_1, p_2)$ and $Q = [k]P = (q_1, q_2)$, then

\begin{equation}
q_1 = G_k(p_1) \quad \text{and} \quad q_2 = p_2 F_k(p_1) = H_k(p_1, p_2).
\end{equation}

Proof. If $k \geq 3$ is odd, then

$$\Phi_k(p_1, p_2) = p_1 \Psi_k(p_1, p_2)^2 - \Psi_{k+1}(p_1, p_2) \Psi_{k-1}(p_1, p_2)$$

$$= p_1 P_k(p_1)^2 - 4(p_1^3 + ap_1 + b) P_{k+1}(p_1) P_{k-1}(p_1),$$

and $\Psi_k(p_1, p_2)^2 = P_k(p_1)^2$. Hence, $q_1 = \Phi_k(p_1, p_2)/\Psi_k(p_1, p_2)^2 = G_k(p_1)$. Note that if $k = 1$, then $q_1 = p_1 = G_1(p_1)$. Moreover,

$$\omega_k(p_1, p_2) = \frac{1}{4p_2} \left( \Psi_{k+2}(p_1, p_2) \Psi_{k-1}(p_1, p_2)^2 - \Psi_{k-2}(p_1, p_2) \Psi_{k+1}(p_1, p_2)^2 \right)$$

$$= p_2 \left( P_{k+2}(p_1) P_{k-1}(p_1)^2 - P_{k-2}(p_1) P_{k+1}(p_1)^2 \right),$$

and $\Psi_k(p_1, p_2)^3 = P_k(p_1)^3$. Therefore, $q_2 = \omega_k(p_1, p_2)/\Psi_k(p_1, p_2)^3 = p_2 F_k(p_1) = H_k(p_1, p_2)$. Note also that if $k = 1$, then $q_2 = p_2 = H_1(p_1, p_2)$.

The case $k$ even is treated in a similar way.

\[ \square \]

Proposition 1.61. Functions $G_i$ and $F_i$ satisfy

\begin{align}
(1.48) \quad & G_{km}(x) = G_m(G_k(x)) , \\
(1.49) \quad & F_{km}(x) = F_k(x) F_m(G_k(x)) ,
\end{align}
if \( k \neq m \), then
\[
G_{k+m}(x) = -G_k(x) - G_m(x) + (x^3 + ax + b) \left( \frac{F_k(x) - F_m(x)}{G_k(x) - G_m(x)} \right)^2 \tag{1.50}
\]

and
\[
F_{k+m}(x) = -F_k(x) + \frac{F_k(x) - F_m(x)}{G_k(x) - G_m(x)}(G_k(x) - G_{k+m}(x)) \tag{1.51}
\]

\[
G_{-k}(x) = G_k(x), \tag{1.52}
\]

\[
F_{-k}(x) = -F_k(x). \tag{1.53}
\]

**Proof.** Let \( \mathbf{P} = (x, y) \) be a generic point on \( E_n(a, b) \). Then,
\[
G_{km}(x) = x([km][\mathbf{P}]) = x([m][[k][\mathbf{P}]] = G_m(x([k][\mathbf{P}])) = G_m(G_k(x))
\]

and
\[
F_{km}(x) = \frac{y([km][\mathbf{P}])}{y(\mathbf{P})} = \frac{y([m][([k][\mathbf{P}])]}{y([k][\mathbf{P}])} \frac{y([k][\mathbf{P}])}{y(\mathbf{P})} = F_m(G_k(x))F_k(x),
\]

which proves Eqs (1.48) and (1.49). Eqs (1.50) and (1.51) are an application of addition formulae on elliptic curves. Finally, since \([-k][\mathbf{P}] = [k][-\mathbf{P}] = [k](x, -y)\), we have Eqs (1.52) and (1.53).

**Remark 1.62.** Note the great similarity of this proposition with Proposition 1.38 on Lucas sequences. Indeed, using the alternative definition in terms of Dickson polynomials (see Proposition 1.42), we have
\[
D_{km}(x, 1) = D_m(D_k(x, 1), 1),
\]
\[
E_{km}(x, 1) = E_k(x, 1)E_m(D_k(x, 1), 1),
\]
\[
D_{-k}(x, 1) = D_k(x, 1),
\]
\[
E_{-k}(x, 1) = (-1)^k E_k(x, 1).
\]

This has to be compared to Eqs (1.48), (1.49), (1.52) and (1.53), respectively.

5. **Lattice basis reduction**

The theory of lattice reduction was introduced by Lagrange in order to define a reduced form associated to a quadratic form. This notion was later reconsidered by several mathematicians, including Hermite, Korkine, Zolotarev, Gauss and Minkowski. The theory of lattice reduction re-emerged in 1982 when Lenstra, Lenstra and Lovász [206]
proposed a polynomial time algorithm, the so-called LLL algorithm, that transforms a basis of a lattice into a reduced basis in the sense of Lovász. Although the resulting basis is not always optimal, this technique revealed itself to be a powerful tool to mount successful attacks on knapsack based cryptosystems [150] or to find roots of polynomial equations [70, 68, 69].

**Definition 1.63.** A subset $L$ of the vector space $\mathbb{R}^n$ is called a lattice if there exists $n$ linearly independent vectors $\vec{b}_i \in \mathbb{R}^n$ such that

$$L = \left\{ \sum_{i=1}^{n} r_i \vec{b}_i \mid r_i \in \mathbb{Z} \right\}.$$ 

The set $\{\vec{b}_1, \ldots, \vec{b}_n\}$ is called a basis of $L$ and $n$ is its rank. The determinant of $L$ is the quantity $\Delta(L) = |\det(\vec{b}_1, \ldots, \vec{b}_n)|$, where the $\vec{b}_i$ are written as column vectors.

**Proposition 1.64.** $\Delta(L)$ is a positive quantity that does not depend on the choice of the basis.

**Proof.** See [55, p. 10], Section I.2. \qed

**Definition 1.65.** The $k$th minimum of a lattice $L$ is the smallest positive real number $\Lambda_k(L)$ such that there exists $k$ linearly independent vectors $\vec{v}_1, \ldots, \vec{v}_k \in L$ satisfying $\| \vec{v}_i \|^2 \leq \Lambda_k(L)$, for $i = 1, \ldots, k$.

In dimension 2, a basis is called reduced if it realizes the minima $\Lambda_1(L)$ and $\Lambda_2(L)$. This can be achieved by the Gaussian algorithm.

**Algorithm 1.66 (Gauss).**

- **Input:** $\{\vec{b}_1, \vec{b}_2\}$ basis with $\| \vec{b}_1 \| \geq \| \vec{b}_2 \|$ 
  - do 
    - $\lambda := \frac{\langle \vec{b}_1, \vec{b}_2 \rangle}{\| \vec{b}_2 \|^2}$ 
    - $\vec{b}_1 := \vec{b}_1 - \lambda \vec{b}_2$ 
    - exchange $\vec{b}_1$ and $\vec{b}_2$ 
  - until $\| \vec{b}_1 \| \leq \| \vec{b}_2 \|$ 
- **Output:** $\{\vec{b}_1, \vec{b}_2\}$ reduced basis

In higher dimension, a reduced basis cannot be defined as the set of vectors realizing the minima $\Lambda_i(L)$ because this set of vectors does not necessarily exist.

Until 1982, there was no general definition for lattice reduction. The LLL reduction [206] relies on the Gram-Schmidt orthogonalization process.
Algorithm 1.67 (Gram-Schmidt).

Input: \( \{\vec{b}_1, \ldots, \vec{b}_n\} \) a basis of \( \mathbb{R}^n \)

for \( i := 1 \) to \( n \) do

\[ \vec{b}_i^2 := \vec{b}_i \]

for \( j := 1 \) to \( i - 1 \) do

\[ \mu_{i,j} := \frac{\langle \vec{b}_i, \vec{b}_j^2 \rangle}{\langle \vec{b}_j, \vec{b}_j \rangle} \]

\[ \vec{b}_i^2 := \vec{b}_i^2 - \mu_{i,j} \vec{b}_j^2 \]

od

od

Output: \( \{\vec{b}_1^2, \ldots, \vec{b}_n^2\} \) an orthogonal basis

Remark 1.68. At step \( i \), the vector \( \vec{b}_i^2 \) is not only orthogonal to all the \( \vec{b}^2_\ell \) \((\ell < i)\), but also to all the \( \vec{b}_i \) since \( \sum_j \mathbb{R} \vec{b}_j = \sum_j \mathbb{R} \vec{b}_j^2 \).

Definition 1.69. The basis \( \{\vec{b}_1, \ldots, \vec{b}_n\} \) of a lattice \( L \) is called **LLL-reduced** if

\begin{align}
|\mu_{i,j}| &\leq 1/2 \quad \text{for } 1 \leq j < i \leq n \quad \text{[size reduction]}, \\
\text{and for } 1/4 \leq t \leq 1 \\
\|\vec{b}_i^2 + \mu_{i,i-1} \vec{b}_{i-1} \|^2 &\geq t \|\vec{b}_{i-1} \|^2 \quad \text{for } 1 < i \leq n \quad \text{[Lovász condition]}. 
\end{align}

The LLL algorithm is a combination of Gauss and Gram-Schmidt. We shall describe it in its simplest form. Suppose that the vectors \( \{\vec{b}_1, \ldots, \vec{b}_{k-1}\} \) are already reduced. The next vector to be reduced is \( \vec{b}_k \). We first have to satisfy the size reduction, i.e. \( |\mu_{k,j}| \leq 1/2 \) for \( 1 \leq j < k \).
Algorithm 1.70 (Size Reduction).

**Input:** \( \{ \vec{b}_1, \ldots, \vec{b}_{n-1} \} \) a set of LLL-reduced vectors

**for** \( j := k - 1 \) **downto** 1 **do**

\[ q := \lfloor \mu_{k,j} \rfloor \left( = \left\lfloor \frac{\langle \vec{b}_j, \vec{b}_k \rangle}{\langle \vec{b}_j, \vec{b}_j \rangle} \right\rfloor \right) \]

**if** \( q > 1/2 \) **then**

\[ \vec{b}_k := \vec{b}_k - q \vec{b}_j \]

update \( \mu_{k,\ell} \) for \( \ell = 1, \ldots, j \)

**fi**

**od**

**Output:** \( \{ \vec{b}_1, \ldots, \vec{b}_{n-1}, \vec{b}_k \} \) a set of size-reduced vectors

Suppose that \( q > 1/2 \) at step \( j \), then \( |\mu_{k,j}^{\text{updated}}| = |\mu_{k,j} - q| \leq 1/2 \).

Furthermore, the values of \( \mu_{k,\ell} \) are not affected by the updating for \( \ell > j \):

\[ \mu_{k,\ell}^{\text{updated}} = \mu_{k,\ell} - q \frac{\langle \vec{b}_j, \vec{b}_k \rangle}{\langle \vec{b}_j, \vec{b}_j \rangle} = \mu_{k,\ell} \]

because \( \vec{b}_j \) is orthogonal to \( \vec{b}_j \) if \( j < \ell \) (see Remark 1.68).

Now, we have to fulfill the Lovász condition. If

\[ \| \vec{b}_k + \mu_{k,k-1} \vec{b}_{k-1} \|^2 \geq \ell \| \vec{b}_{k-1} \|^2, \]

we are done and \( \{ \vec{b}_1, \ldots, \vec{b}_k \} \) are LLL-reduced. Otherwise, we exchange the vectors \( \vec{b}_{k-1} \) and \( \vec{b}_k \). We then reiterate the same process with input \( \{ \vec{b}_1, \ldots, \vec{b}_{k-2} \} \) are LLL-reduced vectors.

Lenstra, Lenstra and Lovász proved that their algorithm terminates. It runs in polynomial time at most \( O(n^6 \log^3 B) \) where \( B = \max_i \| \vec{b}_i \| \). However, in practice, this bound is quite pessimistic. Efficient implementations of this algorithm are presented in [305].

Once a basis is LLL-reduced, it fulfills the following useful properties.

**Theorem 1.71.** Let \( \{ \vec{b}_1, \ldots, \vec{b}_n \} \) be a LLL-reduced basis of a lattice \( L \). Then,

\[
\Delta(L) \leq \prod_{i=1}^{n} \| \vec{b}_i \| \leq 2^{n(n-1)/4} \Delta(L),
\]

\[
\| \vec{b}_j \| \leq 2^{(i-1)/2} \| \vec{b}_i \| \quad \text{for} \quad 1 \leq j \leq i \leq n,
\]

\[
\| \vec{b}_i \| \leq 2^{n-1/4} \Delta(L)^{1/n},
\]

\[
\| \vec{b}_1 \| \leq 2^{(n-1)/2} \| \vec{x} \| \quad \forall \vec{x}(\neq \vec{0}) \in L.
\]
Furthermore, for any linearly independent vectors $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_t \in L$, 
\begin{equation}
\| \hat{y}_j \| \leq 2^{(n-1)/2} \max(\|\vec{x}_1\|, \|\vec{x}_2\|, \ldots, \|\vec{x}_t\|) \quad \text{for} \ 1 \leq j \leq t.
\end{equation}

Proof. See [67, p. 84], Theorem 2.6.2. \qed

References


REFERENCES

[221] A. Magnus, Personal communication.
1. MATHEMATICAL BACKGROUND
RSA-type Cryptosystems

In 1978, Rivest, Shamir and Adleman [292] introduced the so-called RSA cryptosystem. Its security mainly relies on the intractability of factoring large numbers, or more precisely on the RSA hypothesis.

**Definition 2.1.** Given a positive integer $n$ that is the product of two odd primes $p$ and $q$, a positive integer $e$ relatively prime to $(p - 1)(q - 1)$ and an integer $c$, the *RSA problem* is: “Find an integer $m$ such that $m^e \equiv c \pmod{n}$”.

**Conjecture 2.2 (RSA hypothesis).** The RSA problem and the integer factorization problem are computationally equivalent.

In Section 1, we will show that if the factors of $n$ are known, then solving the RSA problem is obvious. However, it has never been proved (although widely believed) that the knowledge of the factors of $n$ is the only way to solve the RSA problem.

Later, other structures were envisaged to implement analogues of RSA. So, in 1981, a cryptosystem based on Dickson polynomials was proposed by Müller and Nöbauer [251] and analyzed in [252]. This system re-emerged later in terms of Lucas sequences to produce LUC [328, 330].

In 1985, Koblitz [177] and Miller [240] independently suggested the use of elliptic curves in cryptography. Afterwards, Koyama, Maurer, Okamoto and Vanstone [187] and later Demytko [89] exhibited new one-way trapdoor functions on elliptic curves over the ring $\mathbb{Z}_n$ (see also [169]). The resulting cryptosystems are respectively called KMOV (from the last names of their inventors) and Demytko’s system.

**1. RSA**

Each user chooses two large primes $p$ and $q$, and publishes the product $n = pq$. Next, he chooses a public encryption key $e$ that is relatively prime to $(p - 1)$ and $(q - 1)$. Finally, he computes the secret decryption key $d$ according to

\[
ed \equiv 1 \pmod{\text{lcm}(p - 1, q - 1)}.
\]
To send a message $m \in \mathbb{Z}_n$ to Bob, Alice looks to Bob’s public key $e$ and forms the ciphertext $c = m^e \mod n$. Next, to recover the plaintext $m$, Bob uses his secret decryption key $d$ to obtain

\[(2.2) \quad m = c^d \mod n.\]

**Proof.** Since $ed \equiv 1 \pmod{\text{lcm}(p - 1, q - 1)}$, $ed \equiv 1 \pmod{(p - 1)}$ and $ed \equiv 1 \pmod{(q - 1)}$. By Theorem 1.2, if $p \nmid m$, then $c^d \equiv m^{ed} \equiv m \pmod{p}$. Otherwise, without loss of generality, we may assume $\gcd(m, p) = p$. Then $c \equiv 0 \pmod{p}$ and we still have $c^d \equiv m \pmod{p}$. Similarly, $c^d \equiv m \pmod{q}$. Hence, by Theorem 1.12, $c^d \equiv m \pmod{n}$.

This encryption scheme can be converted into a signature scheme. If Bob wants to sign a message $m$, he uses his secret key $d$ to compute the signature $s = m^d \mod n$. Next, he sends $m$ and $s$ to Alice. Then Alice can verify that $s$ is the Bob’s signature of message $m$ by checking whether $s^e \equiv m \pmod{n}$ where $e$ is the public key of Bob.

2. LUC

To setup the system, each user proceeds in a similar way as for RSA. The public parameters are the RSA-modulus $n = pq$ that is the product of two large primes, and the encryption key $e$ that is relatively prime to $(p - 1), (p + 1), (q - 1)$ and $(q + 1)$. The secret decryption key $d$ is computed according to

\[(2.3) \quad ed \equiv 1 \pmod{\Psi(n)},\]

where $\Psi(n) = \text{lcm}(p - (\Delta/p), q - (\Delta/q))$.

To send a message $m$ to Bob, Alice uses Bob’s public key $e$ to compute the ciphertext $c = V_e(m, 1) \mod n$. Then, Bob recovers the plaintext $m$ with his secret key $d$ by computing

\[(2.4) \quad m = V_d(c, 1) \mod n.\]

**Proof.** From Eq. (1.23) and Corollary 1.40, it follows

$V_d(c, 1) \equiv V_d(V_e(m, 1), 1) \equiv V_{de}(m, 1) \equiv m \pmod{n}.$

Apparently, the drawback in this method is that $d$ depends on message $m$ because $\Delta = m^2 - 4$. In fact, according to the values of $(\Delta/p)$ and $(\Delta/q)$, there are four possibilities for $d$ that satisfy Eq. (2.3). However,
the decryption key corresponding to a given message can be determined a priori since

$$\Psi(n) = \text{lcm}(p - (\Delta/p), q - (\Delta/q))$$

$$= \text{lcm} \left( p - (\Delta U^2(m,1)/p), q - (\Delta U^2(m,1)/q) \right)$$

$$= \text{lcm} \left( p - \left( \frac{c^2 - 4}{p} \right), q - \left( \frac{c^2 - 4}{q} \right) \right) \text{ by Eq. (1.21).}$$

Remark 2.3. It is also possible to construct a message-independent cryptosystem, taking $d$ such that $ed \equiv 1 \pmod{\text{lcm}(p-1, p+1, q-1, q+1)}$. This method avoids the computation of the two Legendre symbols in the expression of $\Psi(n)$, but it doubles the length of the deciphering key, on average.

3. Elliptic curve systems

Proposition 1.57 seems to establish a RSA-type system. However, some problems occur. Suppose that Alice wants to send a message $m$ to Bob. Bob fixes an elliptic curve $E_n(a, b)$ over $\mathbb{Z}_n$ and computes $N_n = \text{lcm}(\#E_p(a, b), \#E_q(a, b))$. He chooses a public key $e$ that is relatively prime to $N_n$, and computes $d$ such that $ed \equiv 1 \pmod{N_n}$. The values of $e$ and $n$, and the elliptic curve $E_n(a, b)$ are public. To encode the message $m$, Alice represents it, in a publicly known way, as a point $M$ of the elliptic curve $E_n(a, b)$. Then she computes $C = [e]M$ and sends $C$ to Bob. To recover the message $m$, Bob uses his secret key $d$ to compute $[d]C = [de]M = M$.

This scheme is not correct because algorithms known for imbedding a message as a point of a given elliptic curve $E_n(a, b)$ require the knowledge of the factors $p$ and $q$. For example, Koblitz proposes the following probabilistic method [182, pp. 179–180]. Given a security parameter $\kappa$, a message $m < [n/\kappa]$ is represented by $x_j = m \kappa + j$, where $1 \leq j \leq \kappa$ is chosen so that we can find some $y_j$ satisfying $y_j^2 = x_j^3 + ax_j + b$. In this case, we take $M = (x_j, y_j) \in E_n(a, b)$. From $M$, message $m$ is recovered as $m = \lfloor (x_j - 1)/\kappa \rfloor$. However, the evaluation of $y_j$ cannot be achieved without knowing $p$ and $q$. Therefore, this method can only be used to construct signature schemes. To construct encryption schemes, other solutions were proposed.

3.1. KMOV.

3.1.1. Basic scheme. KMOV system relies on Lemma 1.51. Each user chooses two primes $p$ and $q$ both congruent to 2 modulo 3, and publishes their product $n = pq$. Next, he selects a public key $e$ relatively
prime to \( N_n = \text{lcm}(p + 1, q + 1) \) and computes the secret key \( d \) so that
\[
ed \equiv 1 \pmod{N_n}.
\]

To send a message \( M = (m_1, m_2) \) to Bob, Alice chooses the parameter \( b \) according to
\[
b = m_2^2 - m_1^2 \pmod{n}.
\]

Next, using Bob’s public key \( e \), she encrypts \( M \in E_n(0, b) \) as
\[
C = [e]M = (c_1, c_2),
\]
and sends it to Bob. From \( C \), Bob computes the parameter \( b \) as \( b = c_2^2 - c_1^2 \pmod{n} \). Then, he recovers the original message with his secret key \( d \) by computing
\[
M = [d]C
\]
on the curve \( E_n(0, b) \).

**Proof.** Obvious by Proposition 1.57. \( \square \)

Note that, from the addition formulae on elliptic curves (see Eqs (1.34) and (1.35)), the knowledge of parameter \( b \) is not really required in the decryption process.

**Remark 2.4.** The minimum value of \( e \) is 5 because \( 6 \mid N_n \).

**Remark 2.5.** As mentioned in [187], from Lemma 1.52, it is also possible to work on a curve of the form \( E_n(a, 0) \) by choosing \( a \) as
\[
a = \frac{m_2^2 - m_1^3}{m_1} \pmod{n}.
\]
In this case, the minimum value of \( e \) is 3 because \( 4 \mid N_n \).

3.1.2. **Kuwakado/Koyama’s extension.** KMOV system was extended to form-free primes by Kuwakado and Koyama [191]. Their extension is based on Lemmas 1.53 and 1.54.

Let the elliptic curve
\[
E_n(0, b) : y^2 \equiv x^3 + b \pmod{n}.
\]

To setup the system, each user chooses two large primes \( p \) and \( q \) and publishes \( n = pq \). Suppose \( p \equiv 1 \pmod{3} \). By Lemma 1.53, \( \#E_p(0, b) = p + 1 + (4b/\pi_p)\pi_p + (4b/\pi_p)\pi_p \) where \( \pi_p = r + s\omega \) is a prime in \( \mathbb{Z}[\omega] \).
such that $\pi_p \pi_p = p$ and $\pi_p \equiv 2 \pmod{3}$. So, depending on the values of the sixtic symbols, we have

\[
\#E_p(0, b) = \begin{cases} 
N_{p,1} = p + 1 + 2r - s & \text{if } (4b/\pi_p)_6 = 1, \\
N_{p,2} = p + 1 - 2r + s & \text{if } (4b/\pi_p)_6 = -1, \\
N_{p,3} = p + 1 - r + 2s & \text{if } (4b/\pi_p)_6 = \omega, \\
N_{p,4} = p + 1 + r - 2s & \text{if } (4b/\pi_p)_6 = -\omega, \\
N_{p,5} = p + 1 - r - s & \text{if } (4b/\pi_p)_6 = \omega^2, \\
N_{p,6} = p + 1 + r + s & \text{if } (4b/\pi_p)_6 = -\omega^2.
\end{cases}
\]

If $q \equiv 1 \pmod{3}$, then the user similarly computes $\#E_q(0, b) = N_{q,i}$ ($1 \leq i \leq 6$). Next, he computes $N_{n,i,j} = \text{lcm}(N_{p,i}, N_{q,j})$ for $1 \leq i, j \leq 6$ and chooses a public encryption key $e$ such that $\gcd(e, N_{n,i,j}) = 1$ for $1 \leq i, j \leq 6$. The decryption keys $d_{i,j}$ are computed so that

\[(2.11) \quad ed_{i,j} \equiv 1 \pmod{N_{n,i,j}} \quad (i, j = 1, \ldots, 6).\]

If $q \equiv 2 \pmod{3}$, then $\#E_q(0, b) = q + 1$. So, the user computes $N_{n,i} = \text{lcm}(N_{p,i}, q + 1)$ for $1 \leq i \leq 6$ and chooses $e$ relatively prime to $N_{n,i}$ for $1 \leq i \leq 6$. In this case, the decryption keys $d_i$ are computed according to

\[(2.12) \quad ed_i \equiv 1 \pmod{N_{n,i}} \quad (i = 1, \ldots, 6).\]

For both cases, the encryption process is the same as for KMOV. To decrypt a ciphertext, Bob has to use the corresponding decryption key $d_{i,j}$ according to the values of $(4b/\pi_p)_6$ and of $(4b/\pi_q)_6$ if $q \equiv 1 \pmod{3}$. If $q \equiv 2 \pmod{3}$, then Bob chooses the decryption key $d_i$ according to the value of $(4b/\pi_p)_6$.

**Remark 2.6.** A similar system can be constructed from Lemma 1.54 instead of Lemma 1.53.

Another cryptosystem based on elliptic curves was later proposed by Demytko.

**3.2. Demytko’s system.** Each user chooses two large primes $p$ and $q$ and makes $n = pq$ public. He also chooses once for all the parameters $a, b$. Next, he computes

\[
\begin{align*}
N_{n,1} &= \text{lcm}(p + 1 - a_p, q + 1 - a_q), \\
N_{n,2} &= \text{lcm}(p + 1 - a_p, q + 1 + a_q), \\
N_{n,3} &= \text{lcm}(p + 1 + a_p, q + 1 - a_q), \\
N_{n,4} &= \text{lcm}(p + 1 + a_p, q + 1 + a_q),
\end{align*}
\]
where \( a_p = p + 1 - \#E_p(a, b) \) and \( a_q = q + 1 - \#E_q(a, b) \). He chooses a public encryption key \( e \) that is relatively prime to \( N_{n,i} \) (\( 1 \leq i \leq 4 \)) and computes the secret decryption keys \( d_i \) so that

\[
ed_i \equiv 1 \pmod{N_{n,i}} \quad (i = 1, \ldots, 4).
\]

Let \( m \) be the message being encoded. Demytko’s system is based on the fact that if \( m \) (modulo \( p \)) is not the \( x \)-coordinate of a point on \( E_p(a, b) \), it will be the \( x \)-coordinate of a point on the twisted curve \( E_p(a, b) \).

It is useful to introduce some notation. Since the computation of the \( y \)-coordinate can be avoided (see Corollary 1.60 and the resulting algorithm described in [48, p. 214], for example), \([k]_x p_1\) will denote the \( x \)-coordinate of \( k \) times the point \( P = (p_1, p_2) \), i.e. \([k]_x p_1 = G_k(p_1) = x([k]P)\). To encrypt \( m \), Alice computes

\[
c = [e]_x m.
\]

To decrypt the ciphertext \( c \), Bob computes

\[
[d_i]_x c = [d_i e]_x m = m,
\]

where the decryption key is

\[
\begin{align*}
d_1 & \text{ if } (w/p) = 1 \text{ and } (w/q) = 1, \\
d_2 & \text{ if } (w/p) = 1 \text{ and } (w/q) \neq 1, \\
d_3 & \text{ if } (w/p) \neq 1 \text{ and } (w/q) = 1, \\
d_4 & \text{ if } (w/p) \neq 1 \text{ and } (w/q) \neq 1,
\end{align*}
\]

with \( w = e^3 + ac + b \pmod{n} \).

**Proof.** Suppose, for example, \( (w/p) = 1 \) and \( (w/q) = -1 \). Then \((c, \cdot) \in E_p(a, b)\) and \((c, \cdot) \in E_q(a, b)\). So, \([d_2]_x c = [e d_2]_x m = m\) by Proposition 1.57, which concludes the proof.

**Remark 2.7.** It is also possible to construct a message-independent cryptosystem by choosing \( d \) according to

\[
ed \equiv 1 \pmod{\operatorname{lcm}(N_{n,1}, N_{n,2}, N_{n,3}, N_{n,4})}.
\]

**Remark 2.8.** In all RSA-type systems, the decryption process can be speeded up by using the Chinese Remainder Theorem (Theorem 1.12). This was first pointed out by Quisquater and Couvreur [283].
References


2. RSA-TYPE CRYPTOSYSTEMS
CHAPTER 3

Security Analysis

Due to its popularity, the original RSA was subject to an extensive cryptanalysis. The attacks can basically be classified into three categories independently of the protocol in use for encryption or signature:

1. attacks exploiting the polynomial structure of RSA;
2. attacks based on its homomorphic nature;
3. attacks resulting from a bad choice of parameters.

Most of these attacks can more or less successfully be extended to their Lucas-based and elliptic curve analogues.

The first category of attacks relies on the polynomial structure of RSA. Since Lucas sequences can be expressed in terms of Dickson polynomials, all these attacks can almost straightforwardly be adapted on LUC. Using division polynomials, the same conclusion holds for elliptic curve cryptosystems.

The second type of attacks does not extend so easily to LUC or Demytko’s system, because of their non homomorphic nature. Therefore, they apparently seem to be resistant. However, multiplicative attacks can sometimes be rewritten in order to be applicable on these latter systems.

The last category of attacks does not really result from a weakness of RSA but rather from a bad implementation. Parameters have to be carefully chosen. Unfortunately, there is no general recipe to extend this kind of attacks.

1. Polynomial attacks

1.1. Håstad’s attack.

1.1.1. Basic attack. Extending a simple attack of Blum and Davida, Håstad showed that sending linearly related messages over a large network using RSA with low public exponent is insecure [139, 140]. To mount his attack, Håstad developed a technique to solve a system of univariate modular equations. This technique was later generalized to the multivariate case by Takagi and Naito [340] and improved in [155].
On the other hand, Coppersmith [70] recently proposed a new method for finding a (small) root of a modular equation, which turned out to be a better way to mount a successful attack [314, 34].

We will first review the original attack of Håstad and then we will discuss the Coppersmith based variation.

**Theorem 3.1.** Consider a system of $k$ modular polynomial equations of degree $\leq \delta$ with $l$ variables given by

$$
\sum_{j_1,j_2,\ldots,j_l=0}^{j_1+j_2+\ldots+j_l \leq \delta} a_{i,j_1,j_2,\ldots,j_l} x_1^{j_1} x_2^{j_2} \cdots x_l^{j_l} \equiv 0 \pmod{n_i}
$$

for $i = 1, \ldots, k$, and where $x_1, \ldots, x_l < n$ and $n = \min_{1 \leq i \leq k} n_i$.

Let

$$
N = \prod_{i=1}^{k} n_i, \quad f = \sum_{m=1}^{\delta} m^{(m+l-1)} \quad \text{and} \quad g = \sum_{m=0}^{\delta} \left(\frac{l+m-1}{m}\right).
$$

If the moduli $n_i$ are coprime, if $\gcd((a_{i,j_1,j_2,\ldots,j_l} x_1^{j_1} x_2^{j_2} \cdots x_l^{j_l} \equiv 0 \pmod{n_i}) = 1$

for $i = 1, \ldots, k$ and if $N > 2^{\frac{g(g+1)}{4}} g^g n^f$, then we can get in polynomial time a real-valued equation which is equivalent to Eq. (3.1).

**Proof.** Consider a lattice $L$ whose basis is given by

$$
\vec{b}_1 = (\mu_0, 0, 0, 0, \ldots, 0, 0, \ldots, 0, 0),
$$

$$
\vec{b}_2 = (N, 0, 0, 0, \ldots, 0, 0, \ldots, 0, 0),
$$

$$
\vec{b}_3 = (0, nN, 0, 0, \ldots, 0, 0, \ldots, 0, 0),
$$

$$
\vec{b}_4 = (0, 0, nN, 0, \ldots, 0, 0, \ldots, 0, 0),
$$

$$
\vdots
$$

$$
\vec{b}_{g+1} = (0, 0, 0, 0, \ldots, 0, 0, \ldots, 0, n^\delta N),
$$

A vector of this lattice is of the form $\vec{V} = S\vec{b}_1 + \sum_{i=1}^{g} s_i \vec{b}_{i+1}$. Its $i^{th}$ coordinate (apart from the last one) is given by

$$
V_i = n^{j_1+\cdots+j_l} (S_{j_1,\ldots,j_l} + s_i N).
$$
Suppose that we can find a vector $\vec{V}(\neq \vec{0}) \in L$ such $\|\vec{V}\| < N/g$, then $|V_i| < N/g$. So,

\[
(*) \quad \left| \frac{V_i}{n^j_{1} + \ldots + j_l} \right| = \frac{V_i}{n^j_{1} + \ldots + j_l} \mod \frac{N}{g^{n^j_{1} + \ldots + j_l}} = |S_{j_1, \ldots, j_l} \mod \frac{N}{g^{n^j_{1} + \ldots + j_l}}| < \frac{N}{g^{n^j_{1} + \ldots + j_l}}.
\]

for all $j_1, \ldots, j_l$.

Let $u_j \equiv \delta_{ij} \mod n_i$ where $\delta_{ij}$ is Kronecker’s delta. Using Theorem 1.12, we get

\[
0 \equiv \sum_{j_1, j_2, \ldots, j_l=0}^{j_1+j_2+\ldots+j_l=\delta} \left( \prod_{i=1}^{k} \mu_{j_1, j_2, \ldots, j_l} a_{i,j_1,j_2,\ldots,j_l} u_i \right) x_1^{j_1} x_2^{j_2} \ldots x_l^{j_l},
\]

\[
(**) \quad \equiv \sum_{j_1, j_2, \ldots, j_l=0}^{j_1+j_2+\ldots+j_l=\delta} S_{j_1, j_2, \ldots, j_l} x_1^{j_1} x_2^{j_2} \ldots x_l^{j_l} \pmod{N},
\]

From Eq. (*), for any $x_1, x_2, \ldots, x_l < n$, we have

\[
\left| \sum_{j_1, j_2, \ldots, j_l=0}^{j_1+j_2+\ldots+j_l=\delta} (S_{j_1, j_2, \ldots, j_l} \mod \frac{N}{g^{n^j_{1} + \ldots + j_l}}) x_1^{j_1} x_2^{j_2} \ldots x_l^{j_l} \right| < \sum_{j_1, j_2, \ldots, j_l=0}^{j_1+j_2+\ldots+j_l=\delta} |S_{j_1, j_2, \ldots, j_l} \mod \frac{N}{g^{n^j_{1} + \ldots + j_l}}| n_1^{j_1} n_2^{j_2} \ldots n_l^{j_l}.
\]

We can thus consider Eq. (***) as a real-valued equation. If non-trivial (i.e. there is at least one nonzero coefficient), this equation is equivalent to Eq. (3.1). Note that the last coefficient of $\vec{V}$ is equal to $S/g$, and therefore $|S| < N$ since $\|\vec{V}\| < N/g$. Note also that $S \neq 0$ because all nonzero vectors $\vec{V}$ with $S = 0$ are of length at least $N$. So, $0 < |S| < N$ whence $S \neq 0$ (mod $N$), and thus $S \neq 0$ (mod $n_i$) for some $n_i$. Furthermore, since $\mu_{j_1, j_2, \ldots, j_l} \equiv \mu_{j_1, j_2, \ldots, j_l} \mod n_i$, and $\gcd((a_{i,j_1,j_2,\ldots,j_l})_{j_1+j_2+\ldots+j_l=\delta, j_1, j_2, \ldots, j_l=0}, n_i) = 1$, there exists at least one $\mu_{j_1, j_2, \ldots, j_l} \neq 0$ (mod $n_i$). Consequently, Eq. (***) is nontrivial.
It remains to prove how to find a vector $\vec{V}$ such that $\|\vec{V}\| < N/g$. From Theorem 1.71, the LLL algorithm can find a vector $\vec{V}$ such that

$$\|\vec{V}\| \leq 2^{\delta/4} \Delta(L)^{1/(g+1)}$$

within polynomial time. Therefore, this algorithm will provide the required vector $\vec{V}$ if

$$2^{\delta/4} \left( N^{g/n} \right)^{1/(g+1)} < N/g \iff 2^{(g+1)/4} g^{n/2} < N.$$  

**Corollary 3.2 (Håstad’s Theorem).** Let $N = \prod_{i=1}^{k} n_i$ and let $n = \min n_i$. Given a set of $k$ equations $\sum_{j=0}^{\delta} a_{ij}x^j \equiv 0 \pmod{n_i}$ where the moduli $n_i$ are pairwise relatively prime and $\gcd((a_{ij})_{j=0}^{\delta}, n_i) = 1$ for all $i$. Then it is possible to find $x < n$ in polynomial time if $N > 2^{(\delta+1)(\delta+2)/4}(\delta + 1)^{\delta+1} n^{\delta(\delta+1)/2}$.

**Proof.** In the univariate case, we have $f = \sum_{m=1}^{\delta} m^{m(m)} = \delta(\delta + 1)/2$ and $g = \sum_{m=0}^{\delta} (m^{m}) = \delta + 1$. 

If the public encryption exponents are small, the encryption process is fast. However, such a choice can be dangerous. We will illustrate the failure with an exponent $e = 3$. Suppose that the same message $m$ has to be sent to three different users. The corresponding ciphertexts are $c_1 = m^3 \pmod{n_1}$, $c_2 = m^3 \pmod{n_2}$ and $c_3 = m^3 \pmod{n_3}$. If $n_1$, $n_2$ and $n_3$ are not relatively prime, then we find the secret factors of some $n_i$ and thus recover the message $m$. Otherwise, by Theorem 1.12,

$$m^3 \equiv c_1 u_1 + c_2 u_2 + c_3 u_3 \pmod{n_1 n_2 n_3},$$

where $u_j \equiv \delta_{ij} \pmod{n_i}$. But, since $m^3 < n_1 n_2 n_3$, it can be recovered.

To foil this attack, we can use larger exponents or send not exactly the same message by adding a time-stamp, for example. This latter solution does not always prevent the recovery of messages. Suppose that the messages are linearly related

$$m_i = \alpha_i m + \beta_i \pmod{n_i} \quad (1 \leq i \leq k),$$

where $\alpha_i$ and $\beta_i$ are known constants. The corresponding ciphertexts are $c_i = m_i^{e_i} \pmod{n_i}$. In this case, we have:

**Corollary 3.3.** In the RSA cryptosystem, a set of $k$ linearly related messages encrypted with public encryption keys $e_i$ and RSA-moduli $n_i$ can be recovered if

$$k > e(e+1)/2 \quad \text{and} \quad n_i > 2^{(e+1)(e+2)/4}(e+1)^{e+1},$$

where $e = \max e_i$. 

Indeed, from Proposition 1.42, we can consider of $k$ and $n_i$ are pairwise coprime and also that the coefficients of polynomial $P_i$ are relatively prime to $n_i$; otherwise we recover the message by factoring $n_i$. Since $k > e(e + 1)/2$ and $n_i > 2^{(e+1)(e+2)/4}(e + 1)^{e+1}$, it follows

$$N = \prod_{i=1}^{k} n_i \geq n_1 \prod_{i=2}^{k} n_i > 2^{(e+1)(e+2)/4}(e + 1)^{e+1} n_i^{(e+1)/2},$$

where $n = \min n_i$. 

Describing Lucas sequences in terms of quadratic rings [254], Pinch showed the previous corollary remains valid for the LUC cryptosystem [276]. This can also be proved thanks to the Dickson polynomials. Indeed, from Proposition 1.42, we can consider $c_i = V_{e_i}(\alpha_im + \beta_i, 1) \mod e_i$ as polynomials in $m$ of degree $e_i$. Kuwakado and Koyama [192] obtained a similar result for elliptic curve cryptosystems by using division polynomials (see also [190]).

**Corollary 3.4.** In the KMOV or Demykko’s cryptosystems, a set of $k$ linearly related messages encrypted with public encryption keys $e_i$ and RSA-moduli $n_i$ can be recovered if

$$k > e^2(e^2 + 1)/2 \quad \text{and} \quad n_i > 2^{(e^2+1)(e^2+2)/4}(e^2 + 1)^{e^2+1},$$

where $e = \max e_i$.

**Proof.** We only focus on the $x$-coordinate. Let $k$ plaintexts $m_i = \alpha_i m + \beta_i \mod n_i$ and the corresponding ciphertexts $c_i = [e_i]_x m_i$. From Proposition 1.59, we obtain $k$ modular equations of degree at most $e^2$

$$c_i \Psi_{e_i}(\alpha_i m + \beta_i)^2 - \Phi_{e_i}(\alpha_i m + \beta_i) \equiv 0 \pmod{n_i} \quad (1 \leq i \leq k).$$

By Corollary 3.2, this set of equations can be solved if $k > e^2(e^2 + 1)/2$ and $n_i > 2^{(e^2+1)(e^2+2)/4}(e^2 + 1)^{e^2+1}$.

1.1.2. **Coppersmith based variation.** The previous approach is not optimal [314, 34]. The amount of messages can significantly be reduced if we use a powerful technique due to Coppersmith.

**Theorem 3.5 (Coppersmith’s Theorem).** Let a monic integer polynomial $\mathcal{P}(x)$ of degree $\delta$ and a positive integer $N$ of unknown factorization. In time polynomial in $\log N$ and $\delta$, we can find all integer solutions $x_0$ to $\mathcal{P}(x_0) \equiv 0 \pmod{N}$ with $|x_0| < N^{1/\delta}$.

**Proof.** See [69, p. 159], Corollary 2.
Corollary 3.6. Let a positive integer \( e \). Sending more than \( e \) linearly related messages that are encrypted via RSA or LUC with public exponents \( e_i \leq e \) and RSA-moduli \( n_i \) is dangerous.

Proof. Let \( m_i = \alpha_i m + \beta_i \mod n_i \) (\( 1 \leq i \leq k \)) be \( k \) linearly related plaintexts and let \( N = \prod_{i=1}^{k} n_i \). We can assume that the RSA-moduli \( n_i \) are pairwise coprime; otherwise the messages are recovered by factoring \( n_i \). Let \( e = \max e_i \). From the ciphertexts \( c_i \), we can derive \( k \) monic polynomial equations of degree \( e \) given by

\[
P_i(m) \equiv (\alpha_i m + \beta_i)^{e_i} - c_i \equiv (m + \alpha_i^{-1} \beta_i)^{e_i} - \alpha_i^{-e_i} c_i
\]

\[
\equiv m^{e-e_i} [(m + \alpha_i^{-1} \beta_i)^{e_i} - \alpha_i^{-e_i} c_i] \equiv 0 \pmod{n_i},
\]

if the RSA cryptosystem is used. Combining these equations by Chinese remaindering (Theorem 1.12), we get a monic polynomial (modulo \( N \)) in \( m \) of degree \( e \). By Theorem 3.5, we can recover \( m \) since \( m < \min n_i \leq N^{1/k} \leq N^{1/e} \) if \( k \geq e \).

If the LUC cryptosystem is used, we express the ciphertexts in terms of Dickson polynomials and compute

\[
P_i(m) \equiv m^{e-e_i} \alpha_i^{-e_i} [D_{e_i}(\alpha_i m + \beta_i, 1) - c_i] \equiv 0 \pmod{n_i}
\]

which are monic polynomial equations of degree \( e \). We combine them by Chinese remaindering. The resulting polynomial can then be solved thanks to Theorem 3.5 if \( k \geq e \).

\( \square \)

Corollary 3.7. Let a positive integer \( e \). Sending more than \( e^2 \) linearly related messages that are encrypted via Demytko’s cryptosystem with public exponents \( e_i \leq e \) and RSA-moduli \( n_i \) is dangerous.

Proof. As before, from the ciphertexts \( c_i = [e_i]_x (\alpha_i m + \beta_i) \), we compute

\[
P_i(m) \equiv m^{e^2-e_i} \alpha_i^{-e_i} [\Phi_{e_i}(\alpha_i m + \beta_i) - c_i \Psi_{e_i}(\alpha_i m + \beta_i)^2] \equiv 0 \pmod{n_i},
\]

which are monic and of degree \( e^2 \) by Proposition 1.59. We conclude the proof in a similar way as for Corollary 3.6.

\( \square \)

Bleichenbacher [36] also improves Håstad’s attack against KMOV. Its improvement relies on another result of Coppersmith.

Proposition 3.8. Let \( P(x_1, \ldots, x_m) \) (mod \( N \)) be a polynomial of total degree \( \delta \). If there exists a solution \( x_i = y_i \) with \( |y_i| < N^{\alpha_i} \), then we can find this solution as long as \( \sum_{i=1}^{m} \alpha_i < (1/\delta) - \epsilon \) for some \( \epsilon > 0 \).

Proof. See [69, p. 160], Section 3.

\( \square \)

Let \( k \) linearly related messages \( M_i = (\alpha_1 m_1 + \beta_1, \alpha_2 m_2 + \beta_2) \). Since each message respectively belongs to the elliptic curve \( E_m(0, b_1) \):

\[
\square \]
$y^2 = x^3 + b_i$, we have

$$(\alpha_2 m_2 + \beta_2)^2 \equiv (\alpha_1 m_1 + \beta_1)^3 + b_i \pmod{n_i},$$

for $i = 1, \ldots, k$. If we combine these $k$ equations by Chinese remaindering, we obtain a bivariate modular polynomial

$$
P(m_1, m_2) \equiv \left(\sum_{i=1}^{k} \alpha_{1i}^3 u_i\right) m_1^3 + 3 \left(\sum_{i=1}^{k} \alpha_{1i}^2 \beta_{1i} u_i\right) m_1^2 + 3 \left(\sum_{i=1}^{k} \alpha_{2i}^2 u_i\right) m_2^3 - 2 \left(\sum_{i=1}^{k} \alpha_{2i} \beta_{2i}\right) m_2^2 + \sum_{i=1}^{k} (\beta_{1i}^3 + b_i - \beta_{2i}^2) u_i \equiv 0 \pmod{N},$$

where $u_j \equiv \delta_{ij} \pmod{n_i}$ and $N = \prod_{i=1}^{k} n_i$. Let $n = \min n_i$. We know that $m_1$ and $m_2$ lie in the interval $[0, n)$. Therefore,

$$|m_1 \mod n| \text{ and } |m_2 \mod n| \leq \left\lfloor n/2 \right\rfloor < \frac{N^{1/k}}{2} = N^{\frac{1}{k} - \frac{1}{\log_2 N}}.$$

By Proposition 3.8, we can solve the polynomial $P(m_1, m_2)$ if

$$\frac{2}{k} - \frac{2}{\log_2 N} < \frac{1}{3} - \varepsilon.$$

This inequality can already be satisfied for $k = 6$. This means that sending more than 6 linearly related messages with KMOV cryptosystem is dangerous. Note that the amount of messages does not depend on the public encryption exponents.

1.1.3. Summary. The following tables summarizes the number of messages required to mount a successful Håstad’s attack.

<table>
<thead>
<tr>
<th>$e$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>33</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA</td>
<td>–</td>
<td>7</td>
<td>–</td>
<td>16</td>
<td>29</td>
<td>562</td>
</tr>
<tr>
<td>LUC</td>
<td>–</td>
<td>7</td>
<td>–</td>
<td>16</td>
<td>29</td>
<td>562</td>
</tr>
<tr>
<td>KMOV</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>326</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Demytko</td>
<td>11</td>
<td>46</td>
<td>137</td>
<td>326</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Table 3.1: Basic Håstad’s attack
Table 3.2: Coppersmith based variation

Table 3.1 was constructed for 512-bit RSA-moduli $n_i$. The ‘−’ means that the system is not defined for this value of $e$ and the ‘∗’ means that the attack is not applicable. We can see that the attack fails for elliptic curve systems if public encryption keys $e$ are greater than 7; while for RSA and LUC, if only 29 linearly related messages are sent then they can be recovered.

The Coppersmith based variation is more powerful, it works whatever the size of the RSA moduli $n_i$. In particular, only 6 linearly related KMOV-encrypted messages can be sufficient to mount a successful attack. We also see that Demytko’s system is more resistant.

Bleichenbacher [36] implemented this attack against KMOV on an Ultra Sparc. He was able to recover the plaintexts from the ciphertexts of 9 linearly related messages in a few minutes, and from the ciphertexts of 8 linearly related messages in about 2 weeks. The theoretical lower bound of 6 messages seemed to be computationally unreachable.

1.2. GCD attack. At the rump session of Crypto’95, Franklin and Reiter identified a new attack against RSA with public exponent 3 [108]. Later, it was extended for exponents up to $\simeq 32$ bits by Patarin [273] and generalized to other RSA-type cryptosystems [164]. If two messages differ only from a known fixed value $\Delta$ and are RSA-encrypted under the same RSA-modulus $n$, then it is possible to recover both of them. This situation occurs quite often, as for example,

- texts differing only from their date of compilation;
- letters sent to different addressees;
- retransmission of a message with a new ID number due to an error …

1.2.1. Review of the attack. Let $m_1$ and $m_2 = m_1 + \Delta$ be the two messages, and let $c_1 = m_1^e \mod n$ and $c_2 = m_2^e \mod n$ be the corresponding ciphertexts. Then, form the polynomials $P$ and $Q \in \mathbb{Z}_n[x]$, defined

<table>
<thead>
<tr>
<th>System</th>
<th># of messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA</td>
<td>$e$</td>
</tr>
<tr>
<td>LUC</td>
<td>$e$</td>
</tr>
<tr>
<td>KMOV</td>
<td>6</td>
</tr>
<tr>
<td>Demytko</td>
<td>$e^2$</td>
</tr>
</tbody>
</table>

| Table 3.2: Coppersmith based variation |
by
\[
(P(x) = x^e - c_1 \mod n \quad \text{and} \quad Q(x) = (x + \Delta)^e - c_2 \mod n.
\]

Since the message \(m_1\) is a root of \(P\) and \(Q\), \(m_1\) will be a root of
\[
\mathcal{R} = \gcd(P, Q)
\]
which is, with a high probability, a polynomial of degree 1. Solving polynomial \(\mathcal{R}\) in \(x\) gives the value of \(m_1\), and \(m_2 = m_1 + \Delta\).

**Example 3.9.** With exponent \(e = 3\), the plaintext \(m_1\) is given by
\[
m_1 = \frac{\Delta(2c_1 + c_2 - \Delta^3)}{-c_1 + c_2 + 2\Delta^3} \mod n,
\]
and \(m_2 = m_1 + \Delta\).

This attack was later generalized to any known polynomial relation between the messages and to any number of messages [71]. Suppose that two messages satisfy a known polynomial relation of the form \(m_1 \equiv P(m_2) \pmod{n}\). So, from the corresponding ciphertexts \(c_1\) and \(c_2\), we construct
\[
(P(x) = x^e - c_1 \mod n \quad \text{and} \quad Q(x) = P(x)^e - c_2 \mod n,
\]
for which \(m_1\) is a root. Therefore, we can obtain \(m_1\) by computing \(\mathcal{R} = \gcd(P, Q)\).

Suppose now that \(m_1\) and \(m_2\) satisfy an implicit polynomial relation given by \(P(m_1, m_2) \equiv 0 \pmod{n}\). In this case, from \(P(x, y) = y^e - c_2\) considered as univariate polynomials in \(\mathbb{Z}_n[y]\), we compute their resultant \(\varrho \in \mathbb{Z}_n[x]\) (see [42], pp. 415-419)
\[
\varrho(x) = \text{Resultant}_y(P(x, y), Q(y))
\]
for which \(m_1\) is a root. Then, from \(P(x) = x^e - c_1 \mod n\), we compute \(\mathcal{R} = \gcd(P, \varrho)\) that will give the value of \(m_1\).

If there are several polynomial related messages, the same technique enables to recover all of them. Let \(k\) messages \(m_i\) satisfying \(P(m_1, \ldots, m_k) \equiv 0 \pmod{n}\) and let \(c_i = m_i^e \mod n\) the corresponding ciphertexts. We construct the \(k\) polynomials \(P_i(x_i) = x_i^e - c_i \mod n\) and we let \(\varrho_0(x_1, \ldots, x_k) = P(x_1, \ldots, x_k) \mod n\). Next, we iteratively compute
\[
\varrho_i(x_1, \ldots, x_{k-i}) = \text{Resultant}_{x_{k-i+1}, P_{k-i+1}, \varrho_{i-1}}\]
until we obtain \(\varrho_{k-1}(x_1)\) for which \(m_1\) is a root. Consequently, we find \(m_1\) by computing \(\gcd(P_1, \varrho_{k-1})\). To recover the other messages \(m_i\) \((i = 2, \ldots, k)\), we first construct
\[
Q_i(m_1, \ldots, m_{i-1}, x_i) = \varrho_{k-i}(m_1, \ldots, m_{i-1}, x_i),
\]
and then we solve

\[(3.9) \quad R_i = \gcd(P_i, Q_i).\]

### 1.2.2. Extension to LUC

Let \(m_1\) and \(m_2 = m_1 + \Delta\) be two plaintexts encrypted by LUC to produce the corresponding ciphertexts \(c_1 = V_e(m_1, 1) \mod n\) and \(c_2 = V_e(m_2, 1) \mod n\).

Unlike RSA, the polynomial relation between the plaintext and the ciphertext is not explicitly given. However, by Proposition 1.42, Lucas sequences can be rephrased in terms of Dickson polynomials:

\[
c_1 \equiv V_e(m_1, 1) \equiv D_e(m_1, 1)
\]

\[
\equiv \sum_{i=0}^{\lfloor e/2 \rfloor} \frac{e}{e-i} \binom{e}{i} (-1)^i \cdot 1^e \equiv 0 \quad (\text{mod } n).
\]

Consequently, the previous attack applies with \(P(x) = D_e(x, 1) - c_1 \mod n\) and \(Q(x) = D_e(x + \Delta, 1) - c_2 \mod n\) [164].

**Example 3.10.** With a public encryption exponent \(e = 3\), the plaintext \(m_1\) can be recovered from \(c_1, c_2\) and \(\Delta\) by

\[
m_1 = \frac{\Delta(2c_1 + \Delta - \Delta^3 + 3\Delta)}{-c_1 + c_2 + 2\Delta^3 - 6\Delta} \mod n,
\]

and \(m_2 = m_1 + \Delta\).

### 1.2.3. Extension to Demytko’s system

Demytko’s system makes only use of the first coordinate of points of elliptic curves. Let \(m_1\) and \(m_2 = m_1 + \Delta\) be two plaintexts, and let \(c_1 = [e]_x m_1\) and \(c_2 = [e]_x m_2\) be the corresponding ciphertexts. We will exhibit the polynomial dependence thanks to Proposition 1.59 [164]:

\[
c_1 \Psi_e(m_1)^2 \equiv \Phi_e(m_1) \quad (\text{mod } n),
\]

where \(\Psi\) and \(\Phi\) are considered as univariate polynomials. So, we construct polynomials \(P\) and \(Q \in \mathbb{Z}_n[x]\) as

\[
(3.10) \quad P(x) = c_1 \Psi_e(x)^2 - \Phi_e(x) \mod n \quad \text{and} \quad Q(x) = c_2 \Psi_e(x + \Delta)^2 - \Phi_e(x + \Delta) \mod n.
\]

Therefore, \(R = \gcd(P, Q)\) will give the desired value of \(m_1\) and \(m_2 = m_1 + \Delta\).
1.2.4. Extension to KMOV. Let two plaintexts \( M_1 = (m_{1,1}, m_{1,2}) \) and \( M_2 = (m_{2,1}, m_{2,2}) \) related by
\[
m_{2,1} = m_{1,1} + \Delta_1 \quad \text{and} \quad m_{2,2} = m_{1,2} + \Delta_2.
\]
These messages are encrypted as \( C_1 = [e]M_1 = (c_{1,1}, c_{1,2}) \) and \( C_2 = [e]M_2 = (c_{2,1}, c_{2,2}) \) on the elliptic curves \( E_n(0, b_1) \) and \( E_n(0, b_2) \), respectively.

Bleichenbacher [36] showed that we can recover the messages independently of the size of the public encryption key \( e \) as follows. First, we construct polynomials \( \mathcal{P} \) and \( Q \in \mathbb{Z}_n[x, y] \) as
\[
\mathcal{P}(x, y) = x^3 - y^2 + b_1 \mod n \quad \text{and} \quad Q(x, y) = (x + \Delta_1)^3 - (y + \Delta_2)^2 + b_2 \mod n,
\]
where \( b_i \) \( (i = 1, 2) \) is derived from the ciphertexts, i.e. \( b_i = c_{i,2}^2 - c_{i,1} \mod n \). Note that \( \mathcal{P}(m_{1,1}, m_{1,2}) = Q(m_{1,1}, m_{1,2}) = 0 \). Next, we compute
\[
\varrho(x) = \text{Resultant}_y(\mathcal{P}, Q)
\]
\[
= 9\Delta_1^2 x^4 + (18\Delta_1^3 - 4\Delta_2^2) x^3
\]
\[
+ \left(15\Delta_1^4 - 6\Delta_1(\Delta_2^2 + b_1 - b_2)\right) x^2
\]
\[
+ 6\Delta_1^2(\Delta_1^2 - \Delta_2 - b_1 + b_2) x
\]
\[
+ \Delta_1^3 \left(\Delta_1^2 - 2(\Delta_2^2 + b_1 - b_2)\right)
\]
\[
+ (\Delta_2^2 - b_1 - b_2)^2 - 4b_1 b_2,
\]
for which \( m_{1,1} \) is a root. Finally, we compute
\[
(3.12) \quad \mathcal{C}(x) = c_{1,1} \Psi_e(x)^2 - \Phi_e(x) \mod (n, \varrho(x))
\]
over \( \mathbb{Z}_n[x]/(\varrho(x)) \). Since \( c_{1,1} = [e] m_{1,1} \), \( m_{1,1} \) is a root of \( \mathcal{C} \). Consequently, \( m_{1,1} \) can be recovered by computing \( \mathcal{R} = \gcd(\varrho, \mathcal{C}) \). The second part of \( M_1 \), i.e. \( m_{1,2} \), is recovered thanks to Corollary 1.60, \( m_{1,2} = c_{1,2}/\mathcal{F}_e(x) \mod n \). Then, \( m_{2,1} = m_{1,1} + \Delta_1 \) and \( m_{2,2} = m_{1,2} + \Delta_2 \).

1.2.5. Analysis. In [273], Patarin estimated that, for RSA, this attack applies with a public encryption exponent \( e \) up to typically 32 bits. From Proposition 1.42, the same conclusion holds for LUC. However, this is not true for Demytko’s system, since the polynomial relation \( \mathcal{P}(x) \) is of order \( e^2 \) instead of \( e \). It means that for the same modulus \( n \), a public exponent \( e \) of length \( \ell \) for Demytko’s system will be as secure as a public exponent \( e \) of length \( 2\ell \) for RSA or LUC. Putting all this information together, we get the following table.
Furthermore, from Theorem 3.5, Coppersmith [69] showed that even if \( \Delta \) (the difference between the two messages) is unknown, then \( m_1 \) and \( m_2 \) can sometimes be recovered. In particular, this means that adding a random padding to the messages being encrypted does not always prevent the recovery of messages.

Let \( \vartheta(\Delta) \) be the resultant in \( x \) of polynomials \( P \) and \( Q \) (previously defined), which is an univariate polynomial in \( \Delta \). It is possible to solve this polynomial \( \vartheta \) if the solution \( \Delta \) is smaller than \( n^{1/k} \), where \( n \) is the public modulus and \( k \) is the degree of \( \vartheta \). For RSA and LUC, we have \( k = e^2 \); for the elliptic curve systems, we have \( k = e^4 \). So, we obtain:

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
\text{e} & 2 & 3 & 5 & 7 \\
\text{RSA} & 512 & 1024 & 512 & 1024 & 512 & 1024 \\
\text{LUC} & - & - & 56 & 113 & 20 & 40 \\
\text{KMOV} & - & - & - & - & * & 1 \\
\text{Demytko} & 31 & 63 & 6 & 12 & * & 1 \\
\hline
\end{array}
\]

Table 3.4: Random padding \( \Delta \) (in bits) tolerated for a 512/1024-bit modulus \( n \)

Note that, with a 512-bit modulus \( n \), the attack only applies on RSA and LUC with a public exponent \( e \) equal to 3, because in this case the maximum size of the random padding \( \Delta \) is of \( \frac{512}{3} \approx 56 \) bits. In the other cases, the tolerated random padding becomes too small and that would be better treated by exhaustion.

1.3. Garbage-man-in-the-middle attack. The basic idea of this attack relies on the possibility to get access to the “bin” of the recipient. In fact, if the cryptanalyst intercepts, transforms and resends a ciphertext, then the corresponding plaintext will be meaningless when the authorized receiver (say, Bob) will decrypt it. So, Bob will discard
poly

it. If the cryptanalyst can get access to this discard, he will be able to recover the original plaintext if the transformation is done in a clever way. Such an attack was already been mounted against RSA by Davida [83]. In many situations, we can get access to the discards, as for example,

- bad implementation of softwares or bad architectures;
- negligent secretaries;
- recovering of a previously deleted message, by a tool like the <un-delete> command with MS-DOS ...

Another scenario is to ask the victim to sign the forged messages. The working hypothesis are thus not unrealistic.

1.3.1. Davida's attack. Suppose Alice wants to send a message $m$ to Bob. Using Bob's public encryption key $e$, she computes $c = m^e \mod n$, and sends it to Bob. Then, because only Bob knows the secret decryption key $d$, he can recover the message $m = c^d \mod n$.

However, a cryptanalyst (Carol) can also recover the message as follows. She intercepts the ciphertext $c$, and replaces it by $c' = c k^e \mod n$ where $k$ is a random number. Then, when Bob will decrypt $c'$, he will compute $m' = c'^d \mod n$. Since the message $m'$ is meaningless, he will discard it. Consequently, if Carol can get access to $m'$, she recovers the original message $m$ by computing

$$m' k^{-1} \equiv c'^d k^{-1} \equiv c^d \equiv m \pmod{n}.$$  

This attack relies on the homomorphic nature of RSA. Thus, it can easily be extended to other systems which have the same property [90]. For example, KMOV is also susceptible to this attack. Let $M$ the message being encrypted. Carol intercepts the corresponding ciphertext $C = [c]M$ and transforms it into $C' = C + [ek]C$ where $k$ is randomly chosen. Then, Bob deciphers $C'$ and obtains $M' = M + [k]C$. Therefore, Carol recovers $M = M' - [k]C$.

1.3.2. Generalization. Non-homomorphic cryptosystems seem to be resistant to the Davida's attack. However, thanks to Theorem 1.1, a similar result can be found [163].

Let $E_e(\cdot)$, $D_d(\cdot)$ and $T_k(\cdot)$ be respectively the public encryption function, the corresponding secret decryption function and the cryptanalytic transformation function.

Now, imagine Alice wants to send a message $m$ to Bob. She first computes the ciphertext $c = E_e(m)$, and sends it to Bob. Suppose that the cryptanalyst, Carol, intercepts the ciphertext $c$, and modifies it into $c' = T_k(c)$ according to Lagrange's Theorem. Next, Bob decrypts $c'$, and
Figure 3.5: Garbage-man-in-the-middle attack

gets \( m' = D_d(c') \). Since the message \( m' \) has no meaning, Bob discards it. Finally, if Carol can get access to \( m' \), then she recovers the message \( m \) from \( c, k, c' \) and \( m' \) from a non-trivial relation.

1.3.3. Illustrations.

1) Attacking LUC. For LUC cryptosystem, the enciphering function and the deciphering function are respectively defined by

(3.14) \[ c = E_e(m) := V_e(m, 1) \mod n \]

and

(3.15) \[ m' = D_d(c') := V_d(c', 1) \mod n. \]

In order to modify the ciphertext \( c \) into \( c' \), the cryptanalyst uses the transformation function

(3.16) \[ c' = T_k(c) := V_k(c, 1) \mod n. \]

The non-trivial relation enabling the attack comes from Proposition 1.42. It is possible to express \( V_k(x, 1) \) as a polynomial of degree \( k \) in the indeterminate \( x \). Consequently, to recover the message \( m \), the cryptanalyst, Carol, does the following.

[Lagrange’s modification]: Carol intercepts \( c = V_e(m, 1) \mod n \) and replaces it by \( c' = V_k(c, 1) \mod n \), where \( k \) is relatively prime to \( e \).
[Recovery of \(m'\):] Next, she gets from Bob the value of \(m'\), the plaintext corresponding to \(c'\):
\[
m' \equiv V_d(c', 1) \equiv V_{dk}(c, 1) \equiv V_{de}(m, 1) \equiv V_{dk}(m, 1) \quad \text{(mod } n).\]

[Non-trivial relation:] She constructs the polynomials \(P, Q \in \mathbb{Z}_n[x]\) given by
\[
(3.17) \quad P(x) = D_e(x, 1) - c \text{ mod } n
\]
and
\[
(3.18) \quad Q(x) = D_k(x, 1) - m' \text{ mod } n,
\]
for which \(m\) is a root. Then, she computes
\[
R = \gcd(P, Q),
\]
that is with a very high probability a polynomial of degree 1. So, Carol obtains the value of \(m\) by solving \(R\) in \(x\).

2) **Attacking Demytko’s system.** To illustrate the attack against Demytko’s system, we shall use division polynomials. Let \(m\) be the message to be encrypted. Then, the cryptanalyst proceeds in a similar way as for LUC to recover the message.

[Lagrange’s modification:] Carol intercepts \(c = [e]_x m\) and replaces it by \(c' = [k]_x c\), where \(k\) is relatively prime to \(e\).

[Recovery of \(m'\):] Next, she gets from Bob the value of \(m'\), the plaintext corresponding to \(c'\):
\[
m' = [d]_x c' = [dk]_x m = [k]_x m.
\]

[Non-trivial relation:] She constructs the polynomials \(P, Q \in \mathbb{Z}_n[x]\)
\[
(3.19) \quad P(x) = \Phi_e(x) - c\Psi_e(x)^2 \text{ mod } n
\]
and
\[
(3.20) \quad Q(x) = \Phi_k(x) - m'\Psi_k(x)^2 \text{ mod } n,
\]
for which \(m\) is a root. Finally, she recovers \(m\) by computing \(R = \gcd(P, Q)\).

1.3.4. **Analysis.** The computation of a GCD may be done as long as the degree of the polynomials is less than 32-bit long [273]. So, unlike the basic attack of Davida against RSA, from Eqs (3.17) and (3.18), this attack applies on LUC only if the public encryption exponent \(e\) has length less than 32 bits. Moreover, when Alice encrypts a message with Demytko’s system with a public exponent \(e\), the polynomial
relations (3.19) and (3.20) are of order $e^2$ instead of $e$ as in LUC. Therefore, this attack is useless against Demytko’s system if the encryption exponent $e$ has length greater than 16 bits.

To overcome this drawback, the cryptanalyst (Carol) has to apply the attack twice. We shall illustrate the technique on Demytko’s system. Let $m$ be the message to be encrypted. Then, the attack goes as follows. Carol intercepts $c = [e]_x m$, and replaces it by $c' = [k_1]_x c$. Next, Bob computes

$$m'_1 = [d]_x c'_1 = [dk_1]_x m = [k_1]_x m.$$  

Carol chooses $k_2$ (relatively prime to $k_1$), and sends $c'_2 = [k_2]_x c$ to Bob. Then, Bob computes

$$m'_2 = [d]_x c'_2 = [dk_2]_x m = [k_2]_x m.$$  

Therefore, from relations (3.21) and (3.22), Carol forms the polynomials $\mathcal{P}$ and $\mathcal{Q} \in \mathbb{Z}_n[x]$ given by

$$\mathcal{P}(x) = \Phi_{k_1}(x) - m'_1 \Psi_{k_1}(x)^2$$

and

$$\mathcal{Q}(x) = \Phi_{k_2}(x) - m'_2 \Psi_{k_2}(x)^2,$$

for which $m$ is a root. So, if $k_1$ and $k_2$ are “small” (typically less than 16-bit long), then by solving the polynomial $R = \gcd(\mathcal{P}, \mathcal{Q})$, Carol obtains the message $m$.

1.3.5. Further results.

1) Substituting the authority. Imagine that we deal with a key distribution scheme. If the cryptanalyst intercepts the encrypted key $c$ sent by the authority to Bob, and modifies it into $c'$, then Bob will discover that the key is corrupted when he will decrypt it. Therefore, he will ask to the authority to re-send the key. If now, the cryptanalyst plays the role of the authority, i.e. if he sends the encrypted key $c$, then the authority will never know that a pirate knows the secret key of Bob. The cryptanalyst behaves thus transparently for the authority.

2) Concealing the cryptanalyst. The aim of the cryptanalyst is to modify the ciphertext $c$ in such a way that Bob is not able to make the difference between his modification and noise (error of transmission) on the public channel. Consequently, he has to modify $c$ in an apparently random way. So, he has to use “large” exponents for the transformation or to use the basic attack of Davida (when applicable).

We shall illustrate this topic on LUC. Imagine that the cryptanalyst, Carol, chooses a small exponent $k$ in order to speed up the computation of the GCD. Then, Bob can “prove” that somebody (namely Carol)
modified the ciphertext $c$ into $c'$ by recovering $k$. He has just to compare (modulo $n$) $V_j(m, 1)$ with $m'$, for $j = 2, 3, \ldots, k$. To prevent this, Carol has to choose a relatively large exponent $k$. However, in order to recover the plaintext, she has to get access two or three times to the bin.

Let $m$ be the message that Alice wants to send to Bob, and let $c = V_e(m, 1) \mod n$ be the corresponding ciphertext, where $e$ is the public key of Bob. Then, the attack is the following.

Carol intercepts $c$, and replaces it by $c' = V_{k_1}(c, 1) \mod n$. Bob receives $c'$, and decrypts it as $m'_1 = V_d(c', 1) \mod n$ with his secret key $d$. Bob asks Alice to re-send $c'$, and Carol sends $c'' = V_{k_2}(c, 1) \mod n$. When Bob computes $m''_2 = V_d(c'', 1) \mod n$, he finds a meaningless message. So, he asks to Alice to send a third time the ciphertext $c''$. Next, Carol sends $c'_3 = V_{k_3}(c, 1) \mod n$. Bob decrypts $c'_3$ to $m'_3 = V_d(c'_3, 1) \mod n$, ...

Now, from the discards $m'_i$ ($i = 1, 2, 3$), Carol can recover the original message $m$ if $k_1, k_2$ and $k_3$ and correctly chosen. This can for instance be done by selecting

$$k_1 = rs, k_2 = ru$$

where $r$ and $s$ are small, and $\gcd(st, u) = \gcd(rt, v) = 1$. Note that $t, u$ and $v$ must be sufficiently large to disable Bob to distinguish noise with piracy on the public channel.

From our choices on the transformation keys $k_i$, Carol obtains

$$m'_1 \equiv V_{st}(m, 1) \equiv V_{st}(V_r(m, 1), 1) \equiv V_{rt}(V_s(m, 1), 1) \quad (\mod n),$$

$$m'_2 \equiv V_{ru}(m, 1) \equiv V_{ru}(V_r(m, 1), 1) \quad (\mod n),$$

$$m'_3 \equiv V_{sv}(m, 1) \equiv V_{sv}(V_s(m, 1), 1) \quad (\mod n).$$

Next, she forms the polynomials $Q_1, Q_2 \in \mathbb{Z}_n[y]$ given by

$$Q_1(y) = V_{st}(y, 1) - m'_1 \quad \text{and} \quad Q_2(y) = V_{ru}(y, 1) - m'_2,$$

for which $V_r(m, 1)$ is a root. Hence, by computing $R = \gcd(Q_1, Q_2)$, she gets (with a high probability) a polynomial of degree 1, for which $m_r = V_r(m, 1) \mod n$ is the root.

Now, if $e$ is small, Carol recovers the original message $m$ by constructing $P, R \in \mathbb{Z}_n[x]$ given by

$$P(x) = V_e(x, 1) - c \quad \text{and} \quad R(x) = V_r(x, 1) - m_r,$$

and by computing $\gcd(P, R)$, she recovers the original message $m$ as explained before.
Otherwise, she constructs the polynomials $Q_1, Q_3 \in \mathbb{Z}_n[z]$ as

$$Q_1(z) = V_r(z, 1) - m'_1 \quad \text{and} \quad Q_3(z) = V_v(z, 1) - m'_3,$$

for which $V_s(m, 1)$ is a root.

Hence, by computing $S = \gcd(Q_1, Q_3)$, she gets (with a high probability) a polynomial of degree 1, for which $m_s = V_s(m, 1) \mod n$ is the root.

Next, from polynomials $R, S \in \mathbb{Z}_n[x]$ given by $R(x) = V_r(x, 1) - m_r$ and $S(x) = V_s(x, 1) - m_s$, she computes $\gcd(R, S)$ and recovers the message $m$.

**Remark 3.11.** It is possible to speed up the computation by using the ideas developed in [37] (see [171] for Demytko’s system).

3) *Combinations/Broadcast encryption.* There are basically three ways for a cryptanalyst to recover a message

1. to force the retransmission;
2. to have a look in the bin;
3. to ask a signature.

Therefore, by combining these methods, it is possible to recover the message.

Furthermore, if the same message is broadcasted to several people, the security is compromised if only two or three persons are negligent (i.e. they do not protect their bins).

**Remark 3.12.** The previous attack was later improved (see Subsection 2.2) against LUC and Demytko’s system. However, we have presented it because of its very general nature.

### 2. Homomorphic attacks

The inherent homomorphic structure of RSA enables to mount some attacks. One example is the Davida’s attack (see § 1.3.1). Naturally, all known homomorphic attacks also apply to KMOV since $[k](P + Q) = [k]P + [k]Q$. This seems not to be the case for LUC and for Demytko’s system. However, the existence of a chosen-message forgery against LUC that needs two messages has been described in [37]. Kaliski found a similar attack on Demytko's system [171].

In this Section, we describe a new chosen-message attack which needs only one message. This new attack shows that RSA-type cryptosystems are even closer related to RSA, i.e. several attacks based on the multiplicative nature of the original RSA can straightforwardly be adapted to any RSA-type cryptosystem. We shall illustrate this with the common modulus failure. We also revisit the garbage-man-in-the-middle attack.
2.1. Chosen-message attack.

2.1.1. Sketch of the attack. Suppose that a cryptanalyst (say Carol) wants to make Alice to sign message \( m \) without her consent. Carol can proceed as follows. She chooses a random number \( k \) and asks Alice to sign (or to decrypt) \( m' = mk^e \mod n \). Carol gets then \( s' \equiv m'^d (k^e)^d \equiv m^d k \) (mod \( n \)) and therefore the signature \( s \) of message \( m \) as \( s = s' k^{-1} \mod n \).

Consequently, chosen-message attacks against RSA seem quite naturally to be a consequence of its multiplicative structure. By reformulating this attack with the extended Euclidean algorithm, it appears that non-homomorphic cryptosystems are also susceptible to a chosen-message attack [38]. Applying on RSA, the attack goes as follows.

- **Input**: A message \( m \) and the public key \( n, e \) of Alice.
- **Step 1**: Carol chooses an integer \( k \) relatively prime to \( e \). Then by the extended Euclidean algorithm, she finds \( u, v \in \mathbb{Z} \) such that \( ku + ev = 1 \).
- **Step 2**: Carol computes \( m' = m^k \mod n \).
- **Step 3**: Next, she asks Alice to sign \( m' \) and gets therefore
  \[
  s' = m'^d \mod n.
  \]
- **Step 4**: Consequently, Carol can compute the signature \( s \) of \( m \) by
  \[
  s = s'^uv \mod n.
  \]

**Proof.** Since \( ku + ev = 1 \), \( d = d(ku + ev) \equiv dku + v (\mod \lcm(p-1, q-1)) \). Hence, \( s \equiv m^d \equiv m'^d k^u v^v \equiv (m^d)^u m^v \equiv s'^u m^v \) (mod \( n \)).

2.1.2. Attacking LUC. The cryptanalyst Carol can try to get a signature \( s \) on a message \( m \) in the following way [35].

- **Input**: A message \( m \) and the public key \( n, e \) of Alice.
- **Step 1**: Carol chooses an integer \( k \) relatively prime to \( e \). Then she uses extended Euclidean algorithm to find \( u, v \in \mathbb{Z} \) such that \( ku + ev = 1 \).
- **Step 2**: Next she computes \( m' = V_k(m, 1) \mod n \).
- **Step 3**: Now she asks Alice to sign \( m' \). If Alice does so then Carol knows \( s' \) given by
  \[
  s' = V_d(m', 1) \mod n.
  \]
3. SECURITY ANALYSIS

[Step 4]: Finally Carol computes

\[ V_{ukd}(m, 1) = V_u(s', 1) \mod n, \]

\[ U_{ukd}(m, 1) = \frac{U_k(m, 1)U_u(s', 1)}{U_e(s', 1)} \mod n, \]

and finds the signature \( s \) of \( m \)

\[ s \equiv V_s(m, 1) \]

\[ \equiv \frac{V_{ukd}(m, 1)V_e(m, 1)}{2} + \frac{\Delta U_{ukd}(m, 1)U_e(m, 1)}{2} \pmod{n} \]

where \( \Delta = m^2 - 4 \).

[Output]: The signature \( s \) of message \( m \).

**Proof.** Eq. (3.24) follows from Eq. (1.23) since

\[ V_u(s', 1) = V_u(V_{kd}(m, 1), 1) \equiv V_{ukd}(m, 1) \pmod{n}. \]

Eq. (3.25) is a consequence of Eq. (1.22) and

\[ U_{ukd}(m, 1)U_e(s', 1) \equiv U_u(V_{kd}(m, 1), 1)U_{kd}(m, 1)U_e(V_{kd}(m, 1), 1) \]

\[ \equiv U_u(V_{kd}(m, 1), 1)U_{ke}(m, 1) \equiv U_u(V_{kd}(m, 1), 1)U_k(m, 1) \pmod{n}. \]

Moreover, \( ku + ev = 1 \) implies \( V_u(m, 1) = V_{ukd+dev}(m, 1) \equiv V_{ukd+e}(m, 1) \pmod{n} \). Hence Eq. (3.26) is an application of Eq. (1.25).

2.1.3. **Attacking Demytko’s system.** Demytko’s system is also vulnerable thanks to Corollary 1.60. The cryptanalyst proceeds similarly as for LUC.

**Input**: A message \( m \) and the public key \( n, e \) of Alice.

**Step 1**: The cryptanalyst Carol chooses a random number \( k \) relatively prime to \( e \). Then she uses extended Euclidean algorithm to find \( u, v \in \mathbb{Z} \) such that \( ku + ev = 1 \).

**Step 2**: Carol computes \( m' = [k]_x m \). Next, she asks Alice to sign \( m' \). So, Carol obtains the signature

\[ s' = [d]_x m'. \]

**Step 3**: Finally, Carol finds the signature \( s = [d]_x m \) of message \( m \) as follows.

**Step 3a**: If \([u]_x s' \neq [v]_x m \) then, by Corollary 1.60, Carol can compute

\[ F_k(m), F_u(s'), F_e(s') \text{ and } F_v(m), \]
and

\[ s = (m^3 + am + b) \left( \frac{F_k(c_m)F_{s'}(s')/F_{s''}(s'') - F_u(m)}{[u]_x s' - [v]_x m} \right)^2 - [u]_x s' - [v]_x m \mod n. \]

[Step 3b]: Otherwise, the signature is given by

\[ s = \frac{(3([u]_x s')^2 + a)^2}{4([u]_x s')^2 + a[u]_x s' + b} - 2[u]_x s' \mod n. \]

[Output]: The signature \( s \) of message \( m \).

**Proof.** \( ku + ev = 1 \) implies that \( d = kud + evd \equiv kud + v \mod N_n \). Consequently,

\[ s = [d]_x m = [kud + v]_x m = [u]_x s' + [v]_x m. \]

Let \( M \) and \( S' \) respectively be the points corresponding to \( m \) and \( s' \), i.e. \( M = (m, \cdot) \) and \( S' = (s', \cdot) \).

a) If \([u]_x s' \neq [v]_x m\), then

\[
x([u]S' + [v]M) \equiv \left( \frac{y([u]S') - y([v]M)}{x([u]S') - x([v]M)} \right)^2 - x([u]S') - x([v]M)
\]

\[
\equiv y(M)^2 \left[ \frac{y([k]M) y([u]S') y(S') - y([v]M) y(M)}{x([u]S') - x([v]M)} \right]^2
\]

\[
- x([u]S') - x([v]M) \mod n
\]

since

\[
\frac{y([u]S')}{y(M)} = \frac{y([u]S')}{y(S')} \cdot \frac{y(S')}{y([k]M)} \cdot \frac{y([k]M)}{y(M)}
\]

and \( y([k]M) = y([edk]M) = y([v]S'). \)

b) Otherwise, since \( \gcd(d, N_n) = 1 \), it follows that \([u]S' \neq -[v]M\) and therefore

\[
x([u]S' + [v]M) = \left( \frac{3x([u]S')^2 + a}{2y([u]S')} \right)^2 - x([u]S') - x([v]M) \mod n.
\]

\( \Box \)
2.2. Garbage-man-in-the-middle attack (II). A one-chosen-message attack straightforwardly extends to the garbage-man-in-the-middle attack. In Subsection 1.3, one, two or three accesses to the bin were required for LUC or for Demytko’s system. By the previous chosen-message attack, only one access to the bin is necessary to recover the message \( m \).

Using the same notations as in Subsection 1.3, the garbage-man-in-the-middle attack on LUC and Demytko’s system becomes:

1) Attacking LUC.

[Lagrange’s modification]: Carol intercepts \( c = V_e(m, 1) \mod n \) and replaces it by \( c' = V_k(c, 1) \mod n \), where \( k \) is relatively prime to \( e \).

[Recovery of \( m' \)]: Next, she gets from Bob the value of \( m' \), the plaintext corresponding to \( c' \):

\[
m' \equiv V_d(c', 1) \equiv V_d(c) \pmod{n}.
\]

[Non-trivial relation]: By the extended Euclidean algorithm, Carol computes \( u, v \in \mathbb{Z} \) such that \( ku + ev = 1 \). Then, she computes

\[
V_{ukd}(c, 1) = V_u(m', 1) \mod n, \\
U_{ukd}(c, 1) = \frac{U_k(c)U_u(m', 1)}{U_e(m', 1)} \mod n,
\]

and finally recovers \( m \) as

\[
m = \frac{V_{ukd}(c, 1)V_v(c, 1)}{2} + \frac{\Delta U_{ukd}(c, 1)U_v(c, 1)}{2} \mod n
\]

where \( \Delta = c^2 - 4 \).

2) Attacking Demytko’s system.

[Lagrange’s modification]: Carol intercepts \( c = [e]_x m \) and replaces it by \( c' = [k]_x c \), where \( k \) is relatively prime to \( e \).

[Recovery of \( m' \)]: Next, she gets from Bob the value of \( m' \), the plaintext corresponding to \( c' \):

\[
m' = [d]_x c' = [dk]_x c.
\]

[Non-trivial relation]: By the extended Euclidean algorithm, Carol computes \( u, v \in \mathbb{Z} \) such that \( ku + ev = 1 \). If \([u]_x m' \neq [v]_x c\), then Carol computes \( \mathcal{F}_k(c), \mathcal{F}_u(m'), \mathcal{F}_e(m') \) and \( \mathcal{F}_v(c) \), and finally
reverses the message $m$ as

$$m = (c^3 + ac + b) \left( \frac{\mathcal{F}_k(c)\mathcal{F}_u(m')/\mathcal{F}_v(m') - \mathcal{F}_v(c)}{[u]_x m' - [v]_x c} \right)^2 - [u]_x m' - [v]_x c \mod n;$$

otherwise, the message $m$ is given by

$$m = \frac{(3[u]_x m')^2 + a)^2}{4((u]_x m')^3 + a[u]_x m' + b)} - 2[u]_x m' \mod n.$$

2.3. Common modulus attack. Simmons pointed out in [320] that the use of a common RSA modulus is dangerous. Indeed, if a message $m$ is sent to two users that have coprime public encryption keys, then the message can be recovered. Suppose that the ciphertexts corresponding to message $m$ are given by $c_1 = m^{e_1} \mod n$ and $c_2 = m^{e_2} \mod n$, with $\gcd(e_1, e_2) = 1$. Then the cryptanalyst uses the extended Euclidean algorithm to find integers $u, v \in \mathbb{Z}$ such that $ue_1 + ve_2 = 1$. Therefore, the message $m$ is recovered as

$$m = m^{ue_1 + ve_2} \equiv c_1^u c_2^v \pmod{n}.$$  

The same attack applies on KMOV. From the ciphertexts $C_1 = [e_1]M$ and $C_2 = [e_2]M$, the cryptanalyst computes $u, v \in \mathbb{Z}$ such that $ue_1 + ve_2 = 1$ and finds

$$M = [u]C_1 + [v]C_2,$$

if $\gcd(e_1, e_2) = 1$.

Because the previous chosen-message attack (Subsection 2.1) requires only one message, Lucas-based systems and Demytko’s elliptic curve system are vulnerable to the common modulus attack [153, 38].

2.3.1. Extension to LUC and Demytko’s system. Let $(e_1, d_1)$ and $(e_2, d_2)$ be two pairs of encryption/decryption keys and let $m$ be the message being encrypted. Assuming $e_1$ and $e_2$ are relatively prime, the cryptanalyst Carol use the extended Euclidean algorithm to find integers $u$ and $v$ such that $ue_1 + ve_2 = 1$.

If $m$ is encrypted as $c_1 = Vc_1(m, 1) \mod n$ and $c_2 = Vc_2(m, 1) \mod n$ by LUC, then Carol recovers it by computing

$$m = \frac{V_u(c_1, 1)V_v(c_2, 1)}{2} + \frac{(e_1^2 - 4)U_u(c_1, 1)U_v(c_2, 1)}{2U_v(c_2, 1)} \mod n.$$
Proof. By Proposition 1.38 and since \(ue_1 + ve_2 = 1\), we have

\[
2m = 2V_{d_2}(e_2, 1) = 2V_{d_2[ue_1+ve_2]}(e_2, 1) = 2V_{d_2ue_1+v}(e_2, 1)
\]

\[
\equiv V_{d_2ue_1}(e_2, 1)V_v(e_2, 1) + (e_2^2 - 4)U_{d_2ue_1}(e_2, 1)U_v(e_2, 1)
\]

\[
\equiv V_{ue_1}(m, 1)V_v(e_2, 1) + (e_2^2 - 4)U_{d_2e_1}(e_2, 1)U_v(V_{d_2e_1}(e_2, 1), 1)U_v(e_2, 1)
\]

\[
\equiv V_u(c_1, 1)V_v(e_2, 1) + (e_2^2 - 4)U_{d_2e_1}(e_2, 1)U_u(c_1, 1)U_v(e_2, 1) \pmod n.
\]

Hence, since

\[
U_{e_1}(e_2, 1) \equiv U_{e_1e_2d_2}(e_2, 1) \equiv U_{d_2e_1}(e_2, 1)U_{e_2}(V_{d_2e_1}(e_2, 1), 1)
\]

\[
\equiv U_{d_2e_1}(e_2, 1)U_{e_2}(c_1, 1) \pmod n,
\]

the proof is complete.

Suppose now that Demytko’s system is used for encryption. Then, Carol can recover \(m = [ue_1 + ve_2]_x m\) from the ciphertexts \(c_1 = [e_1]_x m\) and \(c_2 = [e_2]_x m\) as follows. If \([u]_x c_1 \neq [v]_x c_2\), then

\[
m = (c_2^3 + ac_2 + b) \left[ \frac{F_{e_1}(e_2)F_u(e_1)/F_{e_2}(c_1) - F_v(e_2)}{[u]_x c_1 - [v]_x c_2} \right]^2
\]

otherwise

\[
m = \frac{(3[u]_x c_1^2 + a)^2}{4([u]_x c_1^2 + a[u]_x c_1 + b)} - 2[u]_x c_1 \pmod n.
\]

Proof. Let \(C_1\) and \(C_2\) be the points respectively corresponding to ciphertexts \(c_1\) and \(c_2\), i.e., \(C_1 = (c_1, \cdot)\) and \(C_2 = (c_2, \cdot)\). Since \(ue_1 + ve_2 = 1\), it follows that

\[
m = [d_1]_x c_1 = [d_1(ue_1 + ve_2)]_x c_1 = x([u]C_1 + [v]C_2).
\]

Noting that \(y([e_2]C_1) = y([e_1]C_2)\), Eqs (3.36) and (3.37) are simply an application of addition formulae on elliptic curves.

2.3.2. Faulty encryption. We will suppose that an error occurs during the computation of the ciphertext. More precisely, if \(e = \sum_{i=0}^{t-1} e_i 2^i\) (with \(e_{t-1} = 1\)) denotes the binary decomposition of the public exponent \(e\), we will suppose that the \(j\)th bit of \(e\) flips to its complementary value.
For example, if RSA is used, the ciphertext corresponding to message $m$ will be $\hat{c} = m^{\hat{e}} \mod n$ instead of $c = m^e \mod n$, where

$$\hat{e} = \begin{cases} e + 2^j & \text{if } e_j = 0, \\ e - 2^j & \text{if } e_j = 1. \end{cases}$$

Let $\delta = \gcd(\hat{e}, e)$. Since $\delta = \gcd(e \pm 2^j, e)$, $\delta$ divides $2^j$. This implies $\delta = 1$ because $e$ is odd for RSA.

Therefore, a cryptanalyst can recover $m$ from $c$ and $\hat{c}$ by the common modulus attack described in Subsection 2.3 [162]. This attack also applies to LUC and KMOV since the encryption exponent must be odd for these systems. This is not always the case for Demytko’s system.

When several bits of the encryption exponent flip, the attack may still apply or not depending on the value of $\gcd(\hat{e}, e)$. Note also that using prime public encryption exponents is dangerous because, in this case, the attack is always applicable.

3. Other attacks

This last section will address attacks resulting from bad implementations or from the presence of faults.

If the secret prime factors $p$ and $q$ of the RSA-moduli are improperly chosen, then factoring attacks [278, 352] can recover the secret keys. Note also that $p$ and $q$ must carefully be generated [282]. Factoring attacks were recently reconsidered by Silverman and Rivest [318, 293]. Furthermore, if a portion of the bits of $p$ (or $q$) is known, then the RSA-moduli can be factorized [291, 68]. Although dangerous, we will not discuss this kind of attacks because there apply identically to any RSA-type system. There are also attacks depending on specific software implementations, such as the timing attacks [183] or the cycling attacks against defective hardware [115]. Other attacks were mounted against some standards using the RSA cryptosystem (see for example the attacks of Girault and Misarsky [118, 242]). On the other hand, the RSA cryptosystem was sometimes modified for efficiency or security purposes. A good example is the Shamir’s unbalanced RSA [313] which use extremely large moduli. However, unless carefully implemented and used, this version can be completely broken [113]. The range of attacks resulting from bad implementations is quite large. So we will only explain how to choose the secret (decryption or signature) exponent $d$.

The second kind of attacks that we will analyze relies on the presence of faults during the computations. If no precaution is taken, we will see that this can give some information on the secret parameters.
Faults based attacks are quite old [255]. In his book "Seizing the Enigma", Kahn reported that, before sending a ciphertext, the operators of Enigma (used by the Germans during the World War II) encrypted a given message twice and compared the resulting ciphertexts. They already knew that Enigma could be attacked if an error occurred during the computation of a ciphertext. We will here study the consequences of faults in the RSA-type cryptosystems (see also § 2.3.2).

### 3.1. Wiener’s attack

Wiener [350] showed that if the secret key \( d \) is chosen too small, then it can be recovered. In the RSA cryptosystem, the public key \( e \) and the secret key \( d \) are related by

\[
ed \equiv 1 \pmod{\text{lcm}(p-1, q-1)}.
\]

So, \( \exists k, h \in \mathbb{Z} \) with \( \gcd(k, h) = 1 \) such that

\[
ed = 1 + \frac{k}{h}(p-1)(q-1) = 1 + \frac{k}{h}(n-p-q+1).
\]

Therefore, dividing both sides by \( dn \) yields

\[
\frac{k}{hd} - \frac{e}{n} = \frac{k}{hd} \left( \frac{1}{p} + \frac{1}{q} - \frac{1}{n} \right) - \frac{1}{dn}. 
\]

Following the presentation of Pinc[277], the attack of Wiener can be explained by the following theorem.

**Theorem 3.13.** If \( \left| \frac{a}{b} - x \right| < \frac{1}{2b^2} \), then \( \frac{a}{b} \) is a continued fraction approximant for \( x \).

**Proof.** See [137, p. 153], Theorem 184. \( \square \)

So, if the condition of the previous theorem is fulfilled, then \( k/(hd) \) is a continued approximant for \( e/n \). Since \( e/n \) is public and since continued fractions can easily be computed, it is possible to find the secret exponent \( d \) under certain assumptions. More precisely, Wiener proved the following corollary.

**Corollary 3.14.** Assume that \( p \sim q \sim \sqrt{n} \), that \( h < d \) and that \( e \sim n \). Then, \( k \sim hd \) and the continued fraction attack will succeed for secret exponents of order up to \( n^{1/4} \).

The assumption \( h < d \) comes from the fact that the RSA-modulus \( n \) is generally chosen as a **rigid** integer, i.e. \( n \) is the product of two primes of the form \( p = 2p' + 1 \) and \( q = 2q' + 1 \) where \( p' \) and \( q' \) are prime (in this case, \( h = 2 \)). Indeed, if \( p - 1 \) or \( q - 1 \) does not contain large prime factors then the Pollard’s \( p - 1 \) method [278] enables to factorize \( n \).
Proof (of Corollary 3.14). If \( e \sim n \), then \( k \sim hd \) since the right side of Eq. (3.39) tends to zero. We now have to verify the condition of Theorem 3.13:

\[
\left| \frac{k}{hd} - \frac{e}{n} \right| \leq \frac{k + h}{hdn} + \frac{k}{hd} \left( \frac{1}{p} + \frac{1}{q} \right) \sim \frac{1}{n} + \frac{1}{\sqrt{n}} \sim \frac{1}{\sqrt{n}}.
\]

So \( \left| \frac{k}{hd} - \frac{e}{n} \right| < \frac{1}{2\sqrt{dn}} \sim \frac{1}{d} \) if \( d \) is of order at most \( n^{1/4} \).

For LUC, the public and the secret keys are chosen according to \( ed \equiv 1 \pmod{\text{lcm}(p, q)} \). Hence, fixing the sign of the two Legendre symbols, we can write \[277\] (3.40)

\[
\frac{k}{hd} - \frac{e}{n} = \frac{k}{hd} \left( \frac{1}{p} + \frac{1}{q} \right) - \frac{1}{dn},
\]

for some \( k \) and \( h \) such that \( \gcd(k, h) = 1 \). From this equation, we can prove that Corollary 3.14 remains valid.

**Proof.** Straightforward since

\[
\left| \frac{k}{hd} - \frac{e}{n} \right| \leq \frac{k + h}{hdn} + \frac{k}{hd} \left( \frac{1}{p} + \frac{1}{q} \right) \sim \frac{1}{\sqrt{n}}.
\]

So the condition of Theorem 3.13 is fulfilled if \( d \) is of order at most \( n^{1/4} \).

This attack also applies to KMOV under the same assumptions. In KMOV, we have \( ed \equiv 1 \pmod{\text{lcm}(p, q)} \). So this can be seen as a special case of LUC where \( (\Delta/p) = (\Delta/q) = -1 \).

Pinch \[277\] also extended the previous attack to Demytko’s cryptosystem by using the Hasse’s Theorem.

**Corollary 3.15.** Assume that \( p \sim q \sim \sqrt{n} \), that \( h < d \) and that \( e \sim n \). Then, \( k \sim hd \) and the continued fraction attack will succeed for secret exponents of order up to \( n^{1/8} \).

**Proof.** For Demytko’s system, the secret and public keys are related by \( ed \equiv 1 \pmod{\text{lcm}(p, q)} \). So if \( h \) is the highest common factor of \( p + 1 \pm a_p \) and \( q + 1 \pm a_q \), then

\[
ed = 1 + \frac{k}{h}(p + 1 \pm a_p)(q + 1 \pm a_q),
\]
for some integer $k$ so that
\[
\frac{k}{hd} - \frac{e}{n} = \frac{k}{hd} \left( \frac{\pm a_p - 1}{p} + \frac{\pm a_q - 1}{q} + \frac{\pm a_p \pm a_q \pm a_pa_q - 1}{n} \right) - \frac{1}{dn}
\]

\[(*) \quad \sim \frac{k}{hd} \left( \frac{\pm a_p}{p} \pm \frac{a_q}{q} \pm \frac{a_pa_q}{n} \right) - \frac{1}{dn}.
\]

By Theorem 1.46, $|a_p| \leq 2\sqrt{p}$ and $|a_q| \leq 2\sqrt{q}$. Hence, $k \sim hd$ since $e \sim n$ and (*) tends to zero. Furthermore, this implies that
\[
\left| \frac{k}{hd} - \frac{e}{n} \right| \lesssim \frac{k}{hd} \left( \frac{2}{\sqrt{p}} + \frac{2}{\sqrt{q}} + \frac{4}{\sqrt{n}} \right) + \frac{1}{dn} \sim \frac{1}{n^{1/4}}.
\]

Therefore, from Theorem 3.13, $k/(hd)$ will be a continued approximant for $e/n$ as long as $d$ has order at most $n^{1/8}$.

The choice of short secret exponents may be attractive in order to quickly perform decryptions or to efficiently sign messages. However, some precautions must be taken. Table 3.6 summarizes the minimum key length for the secret exponent to avoid the Wiener’s attack.

<table>
<thead>
<tr>
<th>$n$</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA</td>
<td>$\sim 128$</td>
<td>$\sim 256$</td>
</tr>
<tr>
<td>LUC</td>
<td>$\sim 128$</td>
<td>$\sim 256$</td>
</tr>
<tr>
<td>KMOV</td>
<td>$\sim 128$</td>
<td>$\sim 256$</td>
</tr>
<tr>
<td>Demytko</td>
<td>$\sim 64$</td>
<td>$\sim 128$</td>
</tr>
</tbody>
</table>

**Table 3.6**: Order (in bits) of a secure secret key $d$ for a 512/1024-bit modulus $n$

3.2. Lenstra’s attack. In September 1996, Boneh, Demillo and Lipton from Bellcore identified a new attack against RSA when performed with Chinese remaindering. This attack was reported in a Bellcore press release, but no technical details were provided. Thereafter and independently, Lenstra wrote a short memo [205] that after the publication of the Bellcore researchers [40] appeared as a more realistic attack. In case of computation error, the Bellcore researchers showed how to recover the secret factors $p$ and $q$ of the public modulus $n$ from two signatures of the same message: a correct one and a faulty one; Lenstra showed that actually only the faulty signature is required.

We will show that Lenstra’s attack is of very general nature and applies on all Chinese remaindering based cryptosystems [156].
3. OTHER ATTACKS

3.2.1. Review of Lenstra’s attack. Let $p$ and $q$ be two primes and let $n = pq$. Imagine that a message $m$ is signed with the secret exponent $d$ using RSA: $s = m^d \mod n$. Using Theorem 1.12, the value of $s$ can be computed more efficiently [283] from

\begin{equation}
sp = m_{p}^{d_p} \mod p \quad \text{and} \quad sq = m_{q}^{d_q} \mod q,
\end{equation}

with $m_p = m \mod p$, $m_q = m \mod q$, $d_p = d \mod (p - 1)$ and $d_q = d \mod (q - 1)$.

Suppose that an error occurs during the computation of $s_p$ (we note $\hat{s}_p$ the faulty value), but not during the computation of $s_q$. Applying Chinese remaindering on $\hat{s}_p$ ($\neq s_p$) and $s_q$ will give the faulty signature $\hat{s}$ for message $m$. Then, the computation of

\begin{equation}
\gcd(\hat{s}^e - m \mod n, n)
\end{equation}

will give the secret factor $q$, where $e$ is the public exponent.

**Proof.** This is a special case of Proposition 3.17.

\[ \square \]

**Figure 3.7:** Lenstra’s attack

\[ \text{Figure 3.7: Lenstra’s attack} \]

**Remark 3.16.** The same attack applies to decryption process, if the attacker has access to the faulty decryption (see pp. 2–3).

3.2.2. Generalization. In this Paragraph, $n = pq$ denotes the RSA-modulus, and $e$ and $d$ are respectively the public and the secret exponents. The message to be signed is $m$, and the corresponding signature is $s$. Let

\[ S : \mathbb{Z}_n \to \mathbb{Z}_n, m \mapsto s = S(m) \]
be an RSA-type signature function. If the signature \( s \) of message \( m \) is performed with the Chinese Remainder Theorem, then the previous attack still applies. More explicitly, we have:

**Proposition 3.17.** Let two primes \( p \) and \( q \) whose product is \( n \). Suppose that \( s = S(m) \) is the signature of a message \( m \) and that \( \hat{s} \) is a faulty signature. If \( \hat{s} \not\equiv s \pmod{p} \) but \( \hat{s} \equiv s \pmod{q} \), then

\[
\gcd(S^{-1}(\hat{s}) - m \pmod{n})
\]

will give the secret factor \( q \).

**Proof.** Since \( \hat{s} \equiv s \pmod{q} \) and \( \hat{s} \not\equiv s \pmod{p} \),

\[
S^{-1}(\hat{s}) \equiv S^{-1}(s) \equiv m \pmod{q}
\]

and \( S^{-1}(\hat{s}) \not\equiv m \pmod{p} \). Hence, \( S^{-1}(\hat{s}) - m \pmod{n} \) is divisible by \( q \) and not by \( p \).

Consequently, the attack of Lenstra works with all RSA-type cryptosystems. With LUC, the signature function is defined as \( S(m) := V_d(m, 1) \pmod{n} \), and the verification function is defined as \( S^{-1}(s) := V_e(s, 1) \pmod{n} \). If \( \hat{s} \not\equiv s \pmod{p} \) but \( \hat{s} \equiv s \pmod{q} \), then

\[
\gcd(V_e(\hat{s}, 1) - m \pmod{n})
\]

will give \( q \).

Demytko’s cryptosystem uses the \( x \)-coordinate of points on elliptic curves over the ring \( \mathbb{Z}_n \). The \( x \)-coordinate of the multiple of a point can be computed thanks to division polynomials (see Proposition 1.59) considered as univariate polynomials. The signature function is defined as \( S(m) := \Phi_d(m)/\Psi_d(m)^2 \pmod{n} \), and the verification function as \( S^{-1}(s) := \Phi_e(s)/\Psi_e(s)^2 \pmod{n} \). If \( \hat{s} \not\equiv s \pmod{p} \) but \( \hat{s} \equiv s \pmod{q} \), then

\[
\gcd\left(\Phi_e(\hat{s})/\Psi_e(\hat{s})^2 - m \pmod{n}\right)
\]

will give \( q \). For KMOV, the two coordinates of the points are used. Let \( S = (s_1, s_2) = [d]M \) be the KMOV-signature of message \( M = (m_1, m_2) \). Suppose that the computation of the \( x \)-coordinate of \( S \) is faulty. More precisely, if \( \hat{s}_1 \not\equiv s_1 \pmod{p} \) and \( \hat{s}_1 \equiv s_1 \pmod{q} \), then, as for Demytko’s system, \( q \) can be recovered by computing \( \gcd(\Phi_e(\hat{s}_1)/\Psi_e(\hat{s}_1)^2 - m_1 \pmod{n}) \). If only the computation of the \( y \)-coordinate of \( S \) is faulty,
3. OTHER ATTACKS

i.e. \( \hat{s}_2 \not\equiv s_2 \pmod{p} \) and \( \hat{s}_2 \equiv s_2 \pmod{q} \), then, thanks to Proposition 1.59, \( q \) can be found by computing

\[
\text{gcd} \left( \frac{\omega_{e}(s_1, \hat{s}_2)}{\Psi_{e}(s_1, \hat{s}_2)^3} - m_2 \pmod{n, n} \right).
\]

3.2.3. Countermeasure. Shamir [309] presented a simple solution to prevent the previous attack. We shall illustrate his technique for RSA, but it remains valid for the other RSA-type systems. The signer first chooses a (small) random number \( r \) relatively prime to \( n \). Then he computes \( s_{rp} = m^{d_{\text{mod}(rp)}} \pmod{rp} \) and \( s_{rq} = m^{d_{\text{mod}(rq)}} \pmod{rq} \). If \( s_{rp} \equiv s_{rq} \pmod{r} \), then the computations are assumed correct, and \( s \) is computed by applying Chinese remaindering on \( (s_{rp} \pmod{p}) \) and \( (s_{rq} \pmod{q}) \).

3.2.4. Further results. Imagine that someone finds a lost smart card. The card performs RSA signatures (using Chinese remaindering), but does not mention any information about its owner. The attacker is thus in a situation where he does not possess any information about the parameters, even not the public ones. Even in such a challenging situation, it is possible to recover much sensitive information if the public exponent is known. Since the public exponent is often a standard value (for example, \( e = 3, 5 \) or \( 2^{16} + 1 \)), identical for each card of a given organization, this hypothesis is not so unlikely [154].

As before, suppose that the computation modulo \( p \) is faulty. From two faulty signatures \( \hat{s}_1 \) and \( \hat{s}_2 \) of messages \( m_1 \) and \( m_2 \) respectively, we compute \( g = \text{gcd}(\hat{s}_1^e - m_1, \hat{s}_2^e - m_2) \) over the rational integers. Removing the small factors of \( g \) gives the secret value of \( q \). Moreover, once \( q \) is known, we can compute \( d_q = e^{-1} \pmod{q - 1} \).

Furthermore, from the knowledge of two (valid) signatures \( s_3 \) and \( s_4 \) of messages \( m_3 \) and \( m_4 \), we can find \( p \) by computing \( \text{gcd}((s_3^e - m_3)/q, (s_4^e - m_4)/q) \). Of course, these results straightforwardly extend to any Chinese remaindering based cryptosystem, including LUC, KMOV and Demytko’s system.

3.3. (Transient) faults based attack. At the last Workshop on Security Protocols, some researchers from the University of Singapore exhibited new attacks against several cryptosystems [17]. Their attacks exploit the presence of transient faults. By exposing a device to external constraints, one can induce some faults with a non-negligible probability [14]. We will show that their attacks are of very general nature [165]. Moreover, we will focus on signatures generation, reducing the number of required signatures for a successful attack to one.
In the sequel, \( n = pq \) will denote the RSA-modulus, and \( e \) and \( d \) will respectively denote the public verification key and the secret signature key. Let \( d = \sum_{i=0}^{t-1} d_i 2^i \) be the binary expansion of \( d \). We assume that one bit of \( d \) flips to its complementary value when the signature is performed. This corrupted signature key will be denoted \( \hat{d} \). So, if bit \( j \) of \( d \) flips, then

\[
\hat{d} = \begin{cases} 
    d + 2^j & \text{if } d_j = 0, \\
    d - 2^j & \text{if } d_j = 1.
\end{cases}
\]

3.3.1. Attacking RSA. Let \( s = m^d \mod n \) and \( \hat{s} = m^{\hat{d}} \mod n \) be the correct and the faulty signatures corresponding to message \( m \), respectively. Mimicking the attack in [17], we find the flipped bit of \( d \) by computing

\[
\hat{s} \equiv m^{d-d} \equiv \begin{cases} 
    m^{2^j} & \text{if } d_j = 0, \\
    \frac{1}{m^{2^j}} & \text{if } d_j = 1.
\end{cases} (\mod n)
\]

However, this formulation is not optimal in signature context. If we put Eq. (3.48) to the \( e \), we obtain

\[
\hat{s}^e \equiv m^{\frac{d}{e}} \equiv \begin{cases} 
    (m^e)^{\frac{2^j}{e}} & \text{if } d_j = 0, \\
    \frac{1}{(m^e)^{\frac{2^j}{e}}} & \text{if } d_j = 1.
\end{cases} (\mod n)
\]

So only the faulty signature \( \hat{s} \) is required to recover the flipped bit. The attack can thus be summarized as follows. The attacker randomly chooses a message \( m \). He computes \( F = m^e \mod n \) and \( \alpha_j = F^{2^j} \mod n \). Then, inducing a physical effort, he asks the device to sign message \( m \). If one bit of \( d \) has flipped, he easily recovers it by comparing \( \hat{s}^e / m \) to \( \alpha_j \) and \( 1/\alpha_j \).

3.3.2. Attacking LUC. The attack on LUC works similarly. The attacker chooses a random message \( m \). He computes \( F = V_e(m, 1) \mod n \) and \( G = (m^2 - 4) U_e(m, 1) \mod n \). He also computes \( \alpha_j = V_{2j}(F, 1) \mod n \) and \( \beta_j = U_{2j}(F, 1) \mod n \). Now, inducing an external effort to the signing device, he gets the faulty signature of message \( m \),

\[
\hat{s} = V_d(m, 1) \mod n.
\]

Finally, he finds the flipped bit of \( d \) as

\[
2V_e(\hat{s}, 1) \equiv \begin{cases} 
    \alpha_j m + \beta_j G & (\mod n) \text{ if } d_j = 0, \\
    \alpha_j m - \beta_j G & (\mod n) \text{ if } d_j = 1.
\end{cases}
\]
3. OTHER ATTACKS

Proof. From the properties of Lucas sequences (see Proposition 1.38), it follows that
\[
2V_e(s, 1) \equiv 2V_e(V_d(m, 1), 1) \equiv V_{e(d-d)+1}(m, 1) \equiv 2V_{e(d-d)+1}(m, 1)
\]
\[
\equiv V_{e(d-d)}(m, 1)V_{1}(m, 1) + \Delta U_{e(d-d)}(m, 1)U_{1}(m, 1)
\]
\[
\equiv V_{d-d}(V_e(m, 1), 1)m + (m^2 - 4)U_e(m, 1)U_{d-d}(V_e(m, 1), 1)
\]
\[
\equiv V_{d-d}(F, 1)m + U_{d-d}(F, 1)G \quad \text{(mod } n\text{).}
\]

Moreover, since \(V_{-k}(P, 1) = V_{k}(P, 1)\) and \(U_{-k}(P, 1) = -U_{k}(P, 1)\), we have the required result. \(\square\)

Note that the computation of \(\alpha_j\) and \(\beta_j\) is not expensive. Taking \(m = k\) in Eqs. (1.19) and (1.18) yields \(V_{2k}(P, 1) = V_{k}^2(P, 1) - 2\) and \(U_{2k}(P, 1) = U_{k}(P, 1)V_{k}(P, 1)\). Therefore, \(\alpha_j\) and \(\beta_j\) can recursively be evaluated by
\[
(3.51) \quad \begin{cases} \alpha_0 = F \\ \alpha_j = \alpha_{j-1}^2 - 2 \mod n \end{cases} \quad \text{and} \quad \begin{cases} \beta_0 = 1 \\ \beta_j = \alpha_{j-1}\beta_{j-1} \mod n \end{cases}
\]

3.3.3. Attacking KMOV. In KMOV, messages are represented as points of elliptic curves. Let \(M = (m_1, m_2)\) be the message being signed, and let \(S = [d]M\) and \(\hat{S} = [d]M\) respectively be the correct and the faulty signatures. We still assume that bit \(j\) of \(d\) has flipped during the computation of the signature, i.e., \(d' = d + 2^j\).

To recover this flipped bit, the attacker has just to compare \([e]\hat{S} - M\) to \([d - d']\) \(([e]M)\). More precisely,
\[
(3.52) \quad [e]\hat{S} - M = \begin{cases} [2^j](F_1, F_2) & \text{if } s_j = 0 \\ [2^j](F_1, -F_2) & \text{if } s_j = 1 \end{cases},
\]

where \((F_1, F_2) = [e]M\).

Proof. Obvious since
\[
[e]\hat{S} = [ed]M = [e(d - d + d)]M = [e(d - d)]M + M = [d - d']([e]M) + M.
\]

Moreover, if \(P = (P_1, P_2)\) is a point on an elliptic curve, then its inverse is given by \(-P = (P_1, -P_2)\). \(\square\)
3.3.4. Attacking Demytko’s system. Demytko’s system only uses the $x$-coordinate of points on elliptic curves. Let $m$ be the message to be signed. This message is represented as a point $M = (m, \cdot)$, or equivalently $m = x(M)$. As before, we suppose that one bit of $d$ has flipped during the computation of the signature. So, the resulting signature is $\hat{s}$.

The attack relies on Corollary 1.60. To recover the flipped bit of $d$, the attacker computes $F = x([e]M) = \mathcal{G}_e(m)$, $G = y([e]M)/y(M) = \mathcal{F}_e(m)$, $H = m^3 + am + b \mod n$, $\alpha_j = \mathcal{G}_2(F)$ and $\beta_j = \mathcal{F}_2(F)$. Then,

\begin{equation}
[e]_x \hat{s} + m \equiv \begin{cases} 
\frac{[\beta_j G - 1]^2 H}{\alpha_j - M} - \alpha_j \pmod{n} & \text{if } s_j = 0, \\
\frac{[\beta_j G + 1]^2 H}{\alpha_j - M} - \alpha_j \pmod{n} & \text{if } s_j = 1.
\end{cases}
\end{equation}

The chord-and-tangent addition on elliptic curves yields

$[e]_x \hat{s} \equiv \mathcal{G}_e(\hat{s}) \equiv x([\hat{d}e]M) \equiv x([\hat{d} - d]e]M + M)$

$\equiv \left[ y([\hat{d} - d]e]M) - y(M) \right]^2 x([\hat{d} - d]e]M) - x(M)$

$\equiv \frac{m^3 + am + b}{(x([\hat{d} - d]([e]M) - m)^2 - x([\hat{d} - d]([e]M) - m}$

$\equiv H \left[ \frac{y([\hat{d} - d]([e]M) - y([e]M)}{y([e]M)} - 1 \right] - \mathcal{G}_{\hat{d} - d}(F) - m$

$\equiv H \left( \mathcal{F}_{\hat{d} - d}(F) G - 1 \right) (\mathcal{G}_{\hat{d} - d}(F) - m)^2 - \mathcal{G}_{\hat{d} - d}(F) - m \pmod{n}$.

Hence, since $\mathcal{G}_{-k}(x) = \mathcal{G}_k(x)$ and $\mathcal{F}_{-k}(x) = -\mathcal{F}_k(x)$, the proof is complete. \qed

3.3.5. Further results.

Generalizing the number of faulty bits. Assume that two bits of $d$ are faulty when performing the signature of a message $m$ with the RSA. Then, the signature is $\hat{s} = m^d \pmod{n}$, where

\begin{equation}
\hat{d} = d \pm 2^l \pm 2^k.
\end{equation}
As before, the attacker computes $F = m^e \mod n$, $\alpha_j = F^{2^j} \mod n$, and compares $\alpha_j / \alpha_k$ to $(\alpha_j)^{\pm 1}(\alpha_k)^{\pm 1}$ until a match is found. In this case, he finds two bits of the secret key $d$. This method naturally extends to multiple faulty bits and to Lucas-based and elliptic curve systems.

Davida's attack. This attack also applies for encryption process. This can been considered as a special case of the Davida's attack (see § 1.3.1). We shall illustrate this attack on RSA, but it remains valid for Lucas-based and elliptic curve systems.

Let $e$ and $d$ be the pair of public encryption key and secret decryption key of Bob. The attacker chooses a random message $m$, computes the ciphertext $c = m^e \mod n$ with the public key $e$ of Bob. Then, inducing a external constraint, he asks Bob to decipher $c$. If Bob does so, he obtains $\tilde{m} = d^j \mod n$. Since $\tilde{m}$ is meaningless, Bob will discard it. Suppose that the attacker can get access to this discard and that only one bit of $d$ is faulty, i.e. $d = d \pm 2^i$. Then,

$$
\frac{\tilde{m}^e}{c} = \begin{cases} 
(c^2)^{2^i} \pmod{n} & \text{if } d_j = 0, \\
\frac{1}{(c^2)^{2^i}} \pmod{n} & \text{if } d_j = 1.
\end{cases}
$$

References


[34] D. Bleichenbacher, Personal communication.


3. SECURITY ANALYSIS


REFERENCES


[255] D. Naccache, Personal communication.


[300] A. Shamir, How to check modular exponentiation, Presented at the rump session of Eurocrypt '97.


3. SECURITY ANALYSIS
CHAPTER 4

Summary and Conclusions

The following table summarizes the vulnerability of the RSA-type cryptosystems against a collection of attacks.

<table>
<thead>
<tr>
<th>attacks</th>
<th>RSA 512</th>
<th>1024</th>
<th>LUC 512</th>
<th>1024</th>
<th>KMOV 512</th>
<th>1024</th>
<th>Demytko 512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hassad (1.1)</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>6</td>
<td>6</td>
<td>e²</td>
<td>e²</td>
</tr>
<tr>
<td>GCD (1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max. size of e (bits)</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>∞</td>
<td>∞</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>random padding (bits)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>for e = 3</td>
<td>56</td>
<td>113</td>
<td>56</td>
<td>113</td>
<td>–</td>
<td>–</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>for e = 5</td>
<td>20</td>
<td>40</td>
<td>20</td>
<td>40</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Garbage-man (1.3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># access to the bin</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Chosen-message (2.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of messages</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Common modulus (2.3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>applicable</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Faulty encryption (2.3.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>susceptible</td>
<td>often</td>
<td>often</td>
<td>often</td>
<td>often</td>
<td>often</td>
<td>often</td>
<td>sometimes</td>
<td>some-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>times</td>
<td>times</td>
</tr>
<tr>
<td>Wiener (3.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max. size of d (bits)</td>
<td>128</td>
<td>256</td>
<td>128</td>
<td>256</td>
<td>128</td>
<td>256</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>Lenstra (3.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>applicable</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Transient faults (3.3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>susceptible</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 4.1: Attacks against 512- and 1024-bit RSA-type systems

The security of all these systems is based on the difficulty of factoring the public modulus $n$. We can see that LUC presents no advantage comparatively to RSA. Moreover, since the computation of Lucas sequences is more expensive, there is practically no reason to use LUC instead of RSA for security purposes. For elliptic curve RSA, addition
of points on elliptic curves consumes also more time than exponentiation. Table 4.1 shows that the security of KMOV is comparable to that of RSA, except against polynomial attacks for which KMOV is less resistant. It also shows that Demytko's system generally is more resistant. So, in my own opinion, original RSA system provides the best ratio between security and efficiency. Since the size of the secret prime factors for systems based on the factorization problem is independent of the underlying structure, devising a competitive RSA-type cryptosystem seems quite difficult. This does not mean that elliptic curve-based implementations must be discarded. Elliptic curves still present some advantages, especially for systems based on the discrete logarithm problem [177, 240]. It is believed (except in few cases [230, 326]) that computing of a discrete logarithm is harder over an elliptic curve than in a finite field of the same size.

Can we still trust RSA-type cryptosystems? The answer is yes. As we will see, there is always a practical and efficient countermeasure. A first weakness of these systems is their polynomial structure. This can enable an attacker to find some secret information. However, two polynomial relations are generally required to mount a successful attack. So there is already a lesson we can learn, sending related messages is dangerous. This also clearly shows a potential weakness of KMOV over the other systems; because with KMOV, the $x$- and $y$-coordinates of messages are already related as points on elliptic curves. This weakness has for example been exploited by Bleichenbacher to reduce the number of messages to 6 for a successful Hästad’s attack (see p. 44).

The second type of attacks against RSA relies on its multiplicative nature. As shown before, non-homomorphic systems can also be subject to this kind of attacks. The remedy is simple, the attacker should never be able to obtain the raw decryption (or signature) of an arbitrary value. For example, users may be very careful about meaningless messages and have to really destroy them (see the Garbage-man-in-the-middle-attack, Subsection 1.3). Another lesson is that the use of the same cryptoalgorithm for encryption and signature is definitely not a good practice. It is better to have two pairs of public/secret keys, one for encryption and the other one for signature.

The last type of attacks is rather the domain of the mathematician or the cryptographer than the one of the protocol designer. So Wiener pointed out that, even if attractive, the use of short secret exponents is insecure (see Subsection 3.1). Therefore, the protocol designer has to pay much attention for the choice of good and bad parameters for the underpinning cryptoalgorithm that he makes use. He has also to be
careful about the implementation (hardware and software) of protocols. In particular, good protocols must be faults resistant, or at least must provide for how to correctly react in the case of faults. The correctness verification cannot always be achieved by doing calculations twice. This must be done by another (proved) secure way (see for example Shamir’s countermeasure against Lenstra’s attack, § 3.2.3).

The previous discussion shows that no implementation of RSA-type based protocols can be considered fully secure unless all security issues are considered. However, we can be quite confident in these systems if they are carefully and properly used.

References


34. D. Bleichenbacher, Personal communication.


42. F. Borceux, Initiation à la géométrie, Ciaco, Louvain-la-Neuve, 1986.

43. F. Borceux and J.-J. Quisquater, Number theory and cryptography, To appear.


46. H. Brandstrom, A public-key cryptosystem based upon equations over a finite field, Cryptologia 7 (1983), no. 4, 347–358.


63. B. Chor and O. Goldreich, RSA/Rabin least significant bits are \( \frac{1}{2} + \frac{1}{\log n} \) secure, Advances in Cryptology – Crypto ’84 (G. R. Blakley and D. Chaum, eds.), Lecture Notes in Computer Science, vol. 196, Springer-Verlag, 1986, pp. 303–313.


BIBLIOGRAPHY


104. ____, Which new RSA-signatures can be computed from certain given RSA-signatures, Journal of Cryptology 5 (1992), no. 1, 41–52.


BIBLIOGRAPHY

141. T. Herlestan, Critical remarks on some public-key cryptosystems, BIT 17 (1978), 493–496.
152. ———, Introduction à la théorie des courbes elliptiques, Mémoire de DEA en mathématiques, Université catholique de Louvain, June 1995.
158. ———, Reducing the elliptic curve cryptosystem of Meyer-Müller to the cryptosystem of Rabin-Williams, Presented to the rump session of Eurocrypt ’96, Zaragoza, Spain, 12-16th May 1996. To appear in Designs, Codes and Cryptography.


208. _______, *Factoring with two primes*, Math. of Comp. 63 (1994), no. 208, 785–798.
221. A. Magnus, Personal communication.
BIBLIOGRAPHY


244. P. L. Montgomery, *Evaluating recurrences of form X_{m+n} = f(X_m, X_n, X_{m-n}) via Lucas chains*, To appear in Fibonacci Quaterly.


255. D. Naccache, Personal communication.


273. **Patarin**, *Some serious protocol failures for RSA with exponent \( e \) of less than \( \approx 32 \) bits*, Presented at the conference of cryptography, CIRM Luminy, France, September 1995.

309. A. Shamir, How to check modular exponentiation, Presented at the rump session of Eurocrypt ’97.
311. , How to share a secret, Communications of the ACM 22 (1979), no. 11, 120–126.


### APPENDIX A

**Index to Notations**

1. **General**

<table>
<thead>
<tr>
<th>Formal symbolism</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>group</td>
</tr>
<tr>
<td>$#G$</td>
<td>order of the (finite) group $G$</td>
</tr>
<tr>
<td>$R$</td>
<td>ring</td>
</tr>
<tr>
<td>$R^\times$</td>
<td>group of units of the ring $R$</td>
</tr>
<tr>
<td>$K$</td>
<td>field</td>
</tr>
<tr>
<td>$\mathcal{O}_{\sqrt{\Delta}}$</td>
<td>ring of integers of the quadratic field $\mathbb{Q}(\sqrt{\Delta})$</td>
</tr>
<tr>
<td>$\mathbb{F}_p$</td>
<td>finite prime field</td>
</tr>
<tr>
<td>$\mathbb{Z}_n$</td>
<td>ring of integers modulo $n$</td>
</tr>
<tr>
<td>$\mathbb{Z}[i]$</td>
<td>ring of Gaussian integers</td>
</tr>
<tr>
<td>$\mathbb{Z}[\omega]$</td>
<td>ring of Eisenstein integers</td>
</tr>
<tr>
<td>$\phi(n)$</td>
<td>Euler’s totient function</td>
</tr>
<tr>
<td>Resultant$_x(P, Q)$</td>
<td>resultant in $x$ of $P$ and $Q \in R[x]$</td>
</tr>
<tr>
<td>$[a]$</td>
<td>largest integer $\leq a$</td>
</tr>
<tr>
<td>$\lfloor a \rfloor$</td>
<td>least integer $\geq a$</td>
</tr>
<tr>
<td>$\lceil a \rceil$</td>
<td>nearest integer to $a$</td>
</tr>
<tr>
<td>$\bar{\alpha}$</td>
<td>conjugate of the quadratic integer $\alpha$</td>
</tr>
<tr>
<td>$N(\alpha)$</td>
<td>norm of $\alpha$, i.e., $N(\alpha) = \alpha \bar{\alpha}$</td>
</tr>
<tr>
<td>$(a/p)$</td>
<td>Legendre symbol of $a$ modulo $p$</td>
</tr>
<tr>
<td>$(\alpha/\pi)_3$</td>
<td>cubic residue symbol of $\alpha$ modulo $\pi$</td>
</tr>
<tr>
<td>$(\alpha/\pi)_4$</td>
<td>quartic residue symbol of $\alpha$ modulo $\pi$</td>
</tr>
<tr>
<td>$(\alpha/\pi)_6$</td>
<td>sextic residue symbol of $\alpha$ modulo $\pi$</td>
</tr>
<tr>
<td>$(a/n)$</td>
<td>Jacobi symbol of $a$ modulo $n$</td>
</tr>
<tr>
<td>$n!$</td>
<td>$n$ factorial</td>
</tr>
<tr>
<td>$\binom{n}{k}$</td>
<td>binomial coefficient</td>
</tr>
<tr>
<td>$\gcd(a, b)$</td>
<td>greatest common divisor of $a$ and $b$</td>
</tr>
<tr>
<td>$\gcd(\langle a_i \rangle_{i=1}^k, b)$</td>
<td>$\gcd(a_1, a_2, \ldots, a_k, b)$</td>
</tr>
</tbody>
</table>
### 2. Lucas sequences

<table>
<thead>
<tr>
<th>Formal symbolism</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_k(P, Q)$</td>
<td>‘sister’ Lucas sequence with parameters $P$ and $Q$</td>
</tr>
<tr>
<td>$V_k(P, Q)$</td>
<td>Lucas sequence with parameters $P$ and $Q</td>
</tr>
<tr>
<td>$D_n(x, a)$</td>
<td>Dickson polynomials of the first kind</td>
</tr>
<tr>
<td>$E_n(x, a)$</td>
<td>Dickson polynomials of the second kind</td>
</tr>
</tbody>
</table>

### 3. Elliptic curves

<table>
<thead>
<tr>
<th>Formal symbolism</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E(a, b)$</td>
<td>elliptic curve</td>
</tr>
<tr>
<td>$E_p(a, b)$</td>
<td>elliptic curve over the prime field $\mathbb{F}_p$</td>
</tr>
<tr>
<td>$#E_p(a, b)$</td>
<td>order of the elliptic curve $E_p(a, b)$</td>
</tr>
<tr>
<td>$a_p$</td>
<td>$p + 1 - $#E_p(a, b)$</td>
</tr>
<tr>
<td>$E_p(a, b)$</td>
<td>complementary group of $E_p(a, b)$</td>
</tr>
<tr>
<td>$E_n(a, b)$</td>
<td>elliptic curve over the ring $\mathbb{Z}_n$ ($n = pq$)</td>
</tr>
<tr>
<td>$\mathcal{O}$</td>
<td>point at infinity of an elliptic curve</td>
</tr>
<tr>
<td>$x(P)$</td>
<td>$x$-coordinate of point $P \in E(a, b)$</td>
</tr>
<tr>
<td>$y(P)$</td>
<td>$y$-coordinate of point $P \in E(a, b)$</td>
</tr>
<tr>
<td>$[k]P$</td>
<td>$P + \cdots + P$ ($k$ times) on $E(a, b)$</td>
</tr>
<tr>
<td>$[k]_p$</td>
<td>$x$-coordinatewise multiplication</td>
</tr>
</tbody>
</table>
### 4. Lattices

<table>
<thead>
<tr>
<th>Formal symbolism</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$</td>
<td>lattice</td>
</tr>
<tr>
<td>$\Delta(L)$</td>
<td>determinant of the lattice $L$</td>
</tr>
<tr>
<td>$\Lambda_k(L)$</td>
<td>$k^{th}$ minimum of lattice $L$</td>
</tr>
<tr>
<td>$|\vec{v}|$</td>
<td>Euclidean length of the vector $\vec{v}$</td>
</tr>
<tr>
<td>$\langle \vec{u}, \vec{v} \rangle$</td>
<td>usual inner product of the vectors $\vec{u}$ and $\vec{v}$</td>
</tr>
</tbody>
</table>
APPENDIX B

Biography and Publications

1. Biographical sketch

Marc Joye was born in Mouscron (Belgium) on December 27th, 1969. He graduated as engineer in applied mathematics at the University of Louvain (January 1994) and received a degree in mathematics at the same university (June 1995). He participated to the European project ACCOPI (Access Control and Copyright Protection for Images): study of access control in a multimedia environment in April-August 1994. He was teaching assistant in mathematics from Sept. 1994 till Sept. 1997. He has been member of the International Association of Cryptologic Research (IACR) since 1995. He is also editor of the technical reports of the UCL Crypto Group.

2. Publications and communications


6. Jean-Jacques Quisquater, Benoît Macq, Marc Joye, Nicolas Degand, and Alexis Bernard. Practical solution to authentication of images with a secure camera. In


Index

$D_n(x, a)$, 15
$E_n(a, b)$, 20
$E_n(x, a)$, 15
$E_p(a, b)$, 18
$U_k(P, Q)$, 13
$V_k(P, Q)$, 13
$[k]$, 36
$#E_p(a, b)$, see elliptic curves, order
$\Delta(L)$, 25
$\Lambda_k(L)$, 25
$O_k$, 16
$\Phi_k(x, y)$, 22
$\Psi_i(x, y)$, 22
$\mathbb{Z}[\omega]$, see Eisenstein integers
$\mathbb{Z}[i]$, see Gaussian integers
$\delta_{ij}$, 7
$(\alpha/\pi)_h$, 10
$(\alpha/\pi)_k$, 12
$(a/\pi)_h$, 10
$(a/n)$, 7
$(a/p)$, 6
$\mathcal{F}_i(x)$, 22–24
$\mathcal{G}_i(x)$, 22–24
$\mathcal{P}_i(x)$, 22
$O_{\sqrt{m}}$, 13, 15
$\phi(n)$, 5
$\omega(x, y)$, 22
$E_p(a, b)$, 18
$E_n(a, b)$, 20
$a_p$, see elliptic curves, order
algorithm
Gaussian reduction, 25
Gram-Schmidt, 25–26
LLL, 25–27, 42
complexity, 27
size reduction, 27
attack
chosen-message attack, 56–59
common modulus attack, 61–63
continued fraction attack, see attack, Wiener’s attack
CRT attack, see attack, Lenstra’s attack
David’s attack, 2, 51, 73
faults based attack, 62–63, 66–73
transient faults, 69–73
garbage-man-in-the-middle attack, 50–56, 59–61
GCD attack, 46–50
Håstad’s attack, 39–46
Coppersmith based variation, 43–46
Lenstra’s attack, 2–3, 66–69
low exponent attack, see attack, Håstad’s attack
Simmons’ attack, see attack, common modulus attack
Wiener’s attack, 63–66
Chinese remaindering, see theorem, Chinese Remainder Theorem
cryptanalysis, see security analysis
cryptology, 1
cryptosystem
Demytko’s system, 35–36
KMOV, 33–35
Kuwakado/Koyama’s extension, 34–35
LUC, 32–33
RSA, 31–32
cubic residue symbol, 10
Demytko, see cryptosystem, Demytko’s system
Dickson polynomials, 15–16, 24
division polynomials, 21–24
Eisenstein integers, 8–10
associates, 9
definition, 8
primary, 9
primes, 9
units, 8
elliptic curves, 16–24
addition formulae, see elliptic curves, group law
complementary group, 18
division polynomials, see division polynomials
elliptic curve based cryptosystems, 33–36
group law, 17–18
order, see theorem, Hasse’s Theorem
order and structure, 18–20
over a field, 16–20
over a ring, 20–21
twist, see elliptic curves, complementary group
Weierstraß equation, 16, 18
Euler’s Theorem, 6
Euler’s totient function, 5
Gaussian integers, 10–12
associates, 11
definition, 10
primary, 11
primes, 11
units, 10
homomorphic attacks, 56–63
Jacobi symbol, 7
KMOV, see cryptosystem, KMOV
Kronecker’s delta, 7
lattice
kth minimum, 25
basis, 25
definition, 25
determinant, 25
LLL-reduced basis, 26
rank, 25
lattice basis reduction, 24–28
Gaussian reduction algorithm, see algorithm, Gaussian reduction
Gram-Schmidt orthogonalization process, see algorithm, Gram-Schmidt in dimension 2, 25
LLL algorithm, see algorithm, LLL
Lovász condition, 26
Legendre symbol, 6
LUC, see cryptosystem, LUC
Lucas sequences, 12–16
definition, 12
Dickson polynomials, see Dickson polynomials
Lucas based cryptosystems, see cryptosystem, LUC
matrix notation, 13
properties, 13–16
one-way function, 1
polynomial attacks, 39–56
protocol failures, 2
public-key cryptography, 1
quartic residue symbol, 12
random padding, 49–50
RSA, see cryptosystem, RSA
hypothesis, 31
problem, 31
RSA-type cryptosystems, 31–36
secret-key cryptography, 1
security analysis, 39–73
sixth residue symbol, 10
soundness, 2, 3
theorem
Chinese Remainder Theorem, 7, 36
Coppersmith’s Theorem, 43, 50
Fermat’s Little Theorem, 5
in \( \mathbb{Z}[\omega] \), 10
in \( \mathbb{Z}[i] \), 12
in \( \mathbb{Q}(i) \), 15
Gauss’ Theorem, see theorem, Law of Quadratic reciprocity
Hasse’s Theorem, 18
Håstad’s Theorem, 42
Lagrange’s Theorem, 5
for elliptic curves, 21
Law of Quadratic Reciprocity, 6
Waring’s formula, 16