PELL EQUATION

Theory and applications to cryptography

CANDIDATE
Simone Dutto

SUPERVISORS
Danilo Bazzanella
Nadir Murru

Ph.D. in Pure and Applied Mathematics
DISMA "G. L. Lagrange" – Politecnico di Torino
Dip. di Mat. "G. Peano" – Università di Torino

Simone Dutto: *Pell equation,* Theory and applications to cryptography

ABSTRACT

The Pell equation $x^2 - Dy^2 = 1$ is a classical topic in number theory. There are well known methods for solving this equation, but there are still several important issues. One of the most interesting from the point of view of cryptographic applications is the study of its solutions over a generic field, in which case new interesting open problems arise. This work focuses on studying the theoretical and practical potential of the Pell equation in this context. Firstly, the required theoretical results from the state of the art are collected using a new unique and simple notation. This allows to obtain easily and elegantly new properties also for the generalization of the Pell equation in the cubic case. Then, all the theoretical results are adopted to formulate new public–key encryption and digital signature schemes with security based on the integer factorization problem or on the discrete logarithm problem, namely new RSA–like and ElGamal cryptosystems, and new Digital Signature Algorithms. The obtained cryptosystems are compared in terms of security, data–size and performance with the classical alternatives, and the results are very interesting especially in the case of the quadratic Pell equation. Finally, the properties of the Pell equation are exploited for defining new powerful probabilistic primality tests, related to the Lucas test included in the widely used Baillie–PSW test. In particular, the new primality tests are equipped with adaptations of the Selfridge method for choosing the parameters, resulting in very powerful tests.

PUBLICATIONS

Some ideas and results have appeared previously in the following publications:

Danilo Bazzanella, Antonio J. Di Scala, Simone Dutto, Nadir Murru,
"Primality tests, linear recurrent sequences and the Pell equation",
Ramanujan Journal 57 (2022),
https://doi.org/10.1007/s11139-020-00373-9.

Gessica Alecci, Simone Dutto, Nadir Murru,
"Pell hyperbolas in DLP–based cryptosystems",
Finite Fields and Their Applications 84 (2022),
https://doi.org/10.1016/j.ffa.2022.102112.

Simone Dutto,
"Developments on primality tests based on linear recurrent sequences of degree two",
Proceedings of the 5th Number Theory Meeting (2021).

Simone Dutto, Nadir Murru,
"On the cubic Pell equation over finite fields",
https://arxiv.org/abs/2203.05290 (2022).

Simone Dutto,
"DLP–based cryptosystems with Pell cubics",
Proceedings of NuTMiC 2021 (2022).

CONTENTS

PKE	Public-Key Encryption	44
RSA	Rivest, Shamir and Adleman PKE scheme	45
IFP	Integer Factorization Problem	46
DLP	Discrete Logarithm Problem	51
FFC	Finite Field Cryptography	53
ECC	Elliptic Curve Cryptography	53
DSA	Digital Signature Algorithm	53
ECDSA	DSA with Elliptic Curves	54
NIST	U.S. National Institute of Standard and Technology	54
spsp(a)	strong pseudoprime to base a	92
lpsp(P, Q)	Lucas pseudoprime with parameters P, Q	95
fpsp(f)	Frobenius pseudoprime with respect to $f(x)$	99
fpsp(P, Q)	Frobenius pseudoprime with respect to $x^2 - Px + Q$	100
slpsp(P, Q)	strong Lucas pseudoprime with parameters P, Q	96
xlpsp(P)	extra strong Lucas pseudoprime with parameter P	98
ppsp(D, X, Y)	Pell pseudoprime with parameters $D, (X, Y)$	101
sppsp(D, X, Y)	strong Pell pseudoprime with parameters $D, (X, Y)$	103
dlpsp(P, Q)	double Lucas pseudoprime with parameters P, Q	105
gppsp(D, X, Y)	generalized Pell pseudoprime with parameters $D, (X, Y)$	108
glpsp(P, Q, R)	generalized Lucas pseudoprime with parameters P, Q, R	111

Part I

WHAT IS IT KNOWN ABOUT THE PELL EQUATION?

This first part focuses on introducing the historical and new theoretical results concerning the Pell equation, as well as the formalism that will be adopted in all the following chapters. After an initial introduction in Chapter 1, the classical Pell equation is addressed in Chapter 2. The classical and new results are then adapted to the the generalization of the Pell equation of degree 3 in Chapter 3.

1

INTRODUCTION

The classical Pell equation is the Diophantine equation of the form

$$x^2 - Dy^2 = 1,$$

where D is a positive integer and solutions are sought for $(x, y) \in \mathbb{Z}^2$. The case with D square is trivial since the only solutions are $(\pm 1, 0)$. Thus, the interesting case is when D is a non–square, since with this choice for D the Pell equation has infinitely many solutions [35].

The Pell equation has ancient origins [22, 57], since it is related to the Archimedes cattle problem despite it is not known if Archimedes was able to solve it. Currently, there are still several important issues regarding the Pell equation. For instance, the study of the size of the fundamental solution is an interesting problem addressed in several papers, e.g., [15, 28, 60]. Recently, the solvability of simultaneous Pell equations and explicit formulas for their solutions have been also studied in [18, 29, 33]. However, the original problem of finding its solution is solved through methods that are mainly based on using continued fractions to find a fundamental solution, which is then used for generating all the other ones.

Historically, one of the first mathematicians that studied extensively the solutions of the Pell equation was surely Brahmagupta. In particular, he focused on the specific equation $92x^2 + 1 = y^2$ in his Brāhmasphuṭasiddhānta dating circa in 628. Then he discovered that, given two solutions (x_1, y_1) and (x_2, y_2) of the Pell equation,

$$(x_1x_2 + Dy_1y_2, x_1y_2 + y_1x_2), \quad (x_1x_2 - Dy_1y_2, x_1y_2 + y_1x_2),$$

are also solutions. This result, besides giving an easy method to find new solutions, can be related to the definition of an operation over the set of all the solutions of a Pell equation. This is the starting point for the applications of the Pell equation in cryptography.

In addition, since all the public–key cryptosystems require finite sets, it is interesting to study the Pell equation over finite fields, determining the number of solutions and their properties [19, 42, 54, 55]. In this context, the Pell equation has already been used for the formulation of many public–key cryptosystems. On one hand, cryptosystems with security relying on the difficulty of the Integer Factorization Problem are largely studied [11, 32, 36, 47, 48], resulting generally in a faster decryption procedure than the classical RSA. However, there are still some issues and theoretical results allow to obtain useful enhancements. On the other hand, the applications of the Pell equation in cryptosystems with security based on the Discrete Logarithm Problem are less studied [44], and new investigations are required.

Another interesting application of the Pell equation are the powers through the operation inspired by Brahmagupta. In particular, it is possible to find a relation with the Rédei polynomials and rational functions obtained from the development of $(z+\sqrt{D})^n$ [39, 53], which allows to speed up the exponentiations required in the previously mentioned cryptosystems. Moreover, Lucas sequences [40] are solutions, up to constants, of the Pell equation, and they can be seen as sequences of solutions obtained as powers of a given solution. Thus, following the definitions of primality tests based on the Lucas sequences, it is useful to investigate the uses of the solutions of the Pell equation over finite fields in new primality tests.

These are the ideas that led the first theoretical part of this work (Chapter 2) to obtain new results about the Pell equation, all formulated under a new simple notation, so that they can be exploited to explore new enhancements and formulations for public–key cryptosystems and primality tests. For sake of completeness, the focus moves also to the case with D square and with constant term not 1, introducing also a generalized operation, so that a full classification is obtained.

As a direct generalization of the classical Pell equation, it is possible to consider its cubic version. As observed in [6], while a natural choice could be simply

$$x^3 - Dy^3 = 1,$$

a more interesting and theoretically correct cubic generalization of the Pell equation is given by

$$x^3 + Ry^3 + R^2z^3 - 3Rxyz = 1,$$

where R is a non–cube integer. Differently from the quadratic case, the first studies on the cubic Pell equation are more modern, since they can be found at the end of the XIX century in [41] and at the beginning of the XX century in [59]. Some years later, a method for solving the cubic Pell equation by means of a generalization of continued fractions due to Jacobi [34] was proposed [20]. However, it is not working always because of the periodicity of the Jacobi algorithm that is still an open problem. This issue was also addressed in the second half of the last century, e.g., in [13] and [14]. The solutions of the cubic Pell equation were also studied in [5] from the point of view of recurrent sequences, generalizing how Lucas sequences are solutions of the quadratic Pell equation. Moreover, in [6], the author exhibited an algorithm for finding the fundamental solutions of the cubic Pell equation that works only in some cases. Thus, in general, there is a wide state of the art on the cubic Pell equation but the problem of finding its solutions is still very hard to solve for any cube–free R. A good starting point could be considering a cubic generalization of the operation inspired by Brahmagupta and studying the cubic Pell equation over a generic field as for the quadratic version.

The theoretical properties of the cubic Pell equation can be exploited for cryptographic applications as in the quadratic case. In this context, the focus is on the solutions over a finite field and the strategy adopted for the classical Pell equation can be generalized in the cubic case. The idea is to explore and enhance formulations with the cubic Pell equation of cryptosystems with security based on the Integer Factorization Problem [46] as well as cryptographic schemes based on the Discrete Logarithm Problem.

The second theoretical part (Chapter 3) adopts a generalization of the notation introduced for the quadratic case in order to study the properties of the cubic Pell equation, so that the new results can be exploited for the mentioned cryptographic applications.

2

PELL CONICS

The classical Pell equation introduced in Chapter 1 can be used for defining a conic over a field $\mathbb{F}$. Moreover, the set of solutions can be equipped with a product such that the resulting structure is a group.

In this chapter, this construction and a particular parametrization are introduced in Section 2.1. The latter is useful for studying the Pell equation over finite fields in Section 2.2 and it is also handy for a generalization in the cubic case studied in Chapter 3. Moreover, this parametrization allows to reduce the data-size of the points of the conic, so that it could be useful in concrete applications, like cryptography. In Section 2.3, a group structure on the generalized Pell conic defined by the equation $x^2 - Dy^2 = Q$ is introduced, together with explicit group isomorphisms with the classical Pell conic and among different generalized Pell conics. Finally, Section 2.4 focuses on an algorithm for the exponentiation over the Pell conic that requires less operations than the classical square–multiply algorithm.

2.1 SOLUTIONS OF THE PELL EQUATION OVER A FIELD

The best way to consider the solutions over a field $\mathbb{F}$ of a Pell equation with parameter $D \in \mathbb{F}^\times$ is by taking the polynomial ring

$$\mathcal{R}_D = \mathbb{F}[t]/\langle t^2 - D\rangle,$$

which inherits from the polynomial product the operation

$$(x_1 + y_1 t) \cdot (x_2 + y_2 t) = (x_1 x_2 + D y_1 y_2) + (x_1 y_2 + y_1 x_2)t.$$

DEFINITION 2.1 Given the parameter $D \in \mathbb{F}^\times$, considering as conjugate of $x + yt \in \mathcal{R}_D$ the element $x - yt$, the product of an element with its conjugate defines the *norm over* $\mathcal{R}_D$ given by

$$\mathcal{N}_D(x + yt) = (x + yt) \cdot (x - yt) = x^2 - Dy^2 \in \mathbb{F}.$$

The unitary elements of $\mathcal{R}_D$ with respect to the norm $\mathcal{N}_D$ are

$$\mathcal{U}(\mathcal{R}_D) = \{x + yt \in \mathcal{R}_D \,|\, \mathcal{N}_D(x + yt) = 1\},$$

and form a commutative group that is isomorphic to the curve defined by the Pell equation, i.e., the *Pell conic with parameter* D:

$$\mathcal{C}_D = \{(x, y) \in \mathbb{F}^2 \,|\, x^2 - Dy^2 = 1\}.$$

If equipped with the *Brahmagupta product with parameter* D:

$$(x_1, y_1) \otimes_D (x_2, y_2) = (x_1 x_2 + D y_1 y_2, x_1 y_2 + y_1 x_2), \tag{2.1}$$

$\mathcal{C}_D$ is a commutative group with identity $(1, 0)$ and $(x, y)^{-1} = (x, -y)$.

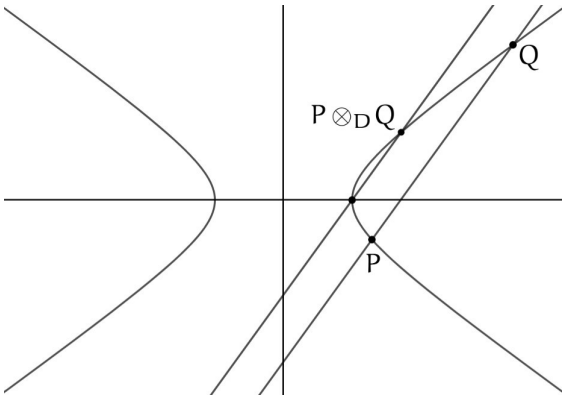

Figure 1: Geometric interpretation of $P \otimes_D Q \in \mathcal{C}_D$.

Due to this group isomorphism, in the following $\otimes_D$ is used also for the product over $\mathcal{R}_D$, so that its dependence on D is highlighted.

From a n geometrical point of view, the operation $\otimes_D$ over $\mathcal{C}_D$ can be introduced in a very similar way to the one of the elliptic curves, as observed, e.g., in [12]. Indeed, given two points P and Q of an elliptic curve, their sum $P \oplus Q$ is obtained by considering the point R, intersection between the elliptic curve and the line through P and Q, so that $P \oplus Q$ is the intersection between the elliptic curve and the line through R and the identity point, that is the point at infinity. This construction works also considering two points P and Q of $\mathcal{C}_D$, with the difference that the line through P and Q intersects the conic at the point R that is, in this case, the point at infinity. Consequently, the product $P \otimes_D Q$ is the intersection between the conic and the line through R (point at infinity) and the identity, that is the point $(1, 0)$, i.e., the line parallel to the line through P and Q, as shown in Figure 1.

It is quite easy to check that, from this geometrical construction of $\otimes_D$, the algebraic expression described in Equation 2.1 can be obtained. Indeed, given $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ on $\mathcal{C}_D$, it is sufficient to check that the slope of the line through P and Q is equal to that of the line through $(x_1x_2 + Dy_1y_2, x_1y_2 + y_1x_2)$ and $(1, 0)$. Thanks to this geometrical approach, it is also possible to observe that the identity point can be an arbitrary point of the Pell conic, and this choice leads to the generalized Brahmagupta product that will be introduced in Section 2.3.

In order to introduce a parametrization for $\mathcal{C}_D$, by denoting with $\mathcal{R}_D^{\otimes_D}$ the invertible elements of $\mathcal{R}_D$ with respect to $\otimes_D$, there are two possible cases:

1. if $D \in \mathbb{F}^\times$ is a non–square, then
$$\mathcal{R}_D^{\otimes_D} = \mathcal{R}_D \setminus \{0\};$$
2. if $D \in \mathbb{F}^\times$ is a square and $s \in \mathbb{F}^\times$ is a square root of D, then
$$\mathcal{R}_D^{\otimes_D} = \mathcal{R}_D \setminus \{0, \pm sy + yt \,|\, y \in \mathbb{F}\}.$$

DEFINITION 2.2 The *projectivization of* $\mathcal{R}_D$ is defined as

$$\mathbb{P}_D = \mathcal{R}_D^{\otimes_D}/\mathbb{F}^\times.$$

In particular, its elements are of the form

$$[m : n] = \{\lambda(m + nt) \,|\, \lambda \in \mathbb{F}^\times\}, \quad \text{for } m + nt \in \mathcal{R}_D^{\otimes_D}.$$

Since if $n \neq 0$ then $m + nt$ is equivalent to $mn^{-1} + t$, it is useful to define the *canonical representatives* as the elements of the form

$$\begin{aligned} \mathbb{P}_D &= \begin{cases} \{[m : 1], [1 : 0] \,|\, m \in \mathbb{F}\}, & \text{if } D \text{ is a non–square,} \\ \{[m : 1], [1 : 0] \,|\, m \in \mathbb{F} \setminus \{\pm s\}\}, & \text{otherwise} \end{cases} \\ &\sim \begin{cases} \mathbb{F} \cup \{\alpha\}, & \text{if } D \text{ is a non–square,} \\ \mathbb{F} \setminus \{\pm s\} \cup \{\alpha\}, & \text{otherwise,} \end{cases} \end{aligned} \tag{2.2}$$

where α denotes the point at infinity and, in case of D square, $s \in \mathbb{F}$ is a fixed square root of D. Since $\otimes_D$ consists of homogeneous polynomials, it is well defined also on $\mathbb{P}_D$ and determines a commutative group with identity $[1 : 0]$ and inverse of $[m : 1]$ given by $[-m : 1]$.

The operation $\otimes_D$ over $\mathbb{P}_D$ between canonical representatives is

$$m_1 \otimes_D m_2 = \begin{cases} m_1, & \text{if } m_2 = \alpha, \\ m_2, & \text{if } m_1 = \alpha, \\ \frac{m_1 m_2 + D}{m_1 + m_2}, & \text{if } m_1 + m_2 \neq 0, \\ \alpha, & \text{otherwise.} \end{cases} \tag{2.3}$$

This projectivization is actually a parametrization of the Pell conic, which is useful for studying some of its properties over finite fields and will be naturally generalized also for the cubic case. The following theorem provides an explicit group isomorphism between $\mathbb{P}_D$ and $\mathcal{C}_D$. The result was introduced in [7], but can be obtained with the different formulation described below, which will be adapted to the cubic case in Chapter 3.

THEOREM 2.1 Given $D \in \mathbb{F}^\times$, the following map is a group isomorphism between $\mathbb{P}_D$ and $\mathcal{C}_D$:

$$\begin{aligned} \phi_D : \left(\mathbb{P}_D, \otimes_D\right) &\xrightarrow{\sim} \left(\mathcal{C}_D, \otimes_D\right), \\ [m : n] &\longmapsto \frac{(m, n)^{\otimes_D 2}}{\mathcal{N}_D(m, n)} = \left(\frac{m^2 + Dn^2}{m^2 - Dn^2}, \frac{2mn}{m^2 - Dn^2}\right). \end{aligned}$$

Proof. In order for ϕ_D to be a group isomorphism, it must be

- well defined: for any $[m : n] \in \mathbb{P}_D$, $\lambda \in \mathbb{F}^\times$,

$$\phi_D([\lambda m : \lambda n]) = \frac{\lambda^2 (m, n)^{\otimes_D 2}}{\lambda^2 \mathcal{N}_D(m, n)} = \phi_D([m : n]),$$

and $\mathcal{N}_D\big(\phi_D([m : n])\big) = \frac{\mathcal{N}_D(m,n)^2}{\mathcal{N}_D(m,n)^2} = 1$, so that $\phi_D(\mathbb{P}_D) \subseteq \mathcal{C}_D$;

- a group homomorphism: for any $[m_1 : n_1], [m_2 : n_2] \in \mathbb{P}_D$,

$$\begin{aligned}\phi_D([m_1:n_1]\otimes_D[m_2:n_2]) &= \frac{(m_1m_2+Dn_1n_2, m_1n_2+n_1m_2)^{\otimes_D 2}}{\mathcal{N}_D(m_1m_2+Dn_1n_2, m_1n_2+n_1m_2)}\\ &= \frac{(m_1,n_1)^{\otimes_D 2}\otimes_D (m_2,n_2)^{\otimes_D 2}}{\mathcal{N}_D(m_1,n_1)\mathcal{N}_D(m_2,n_2)}\\ &= \phi_D([m_1:n_1])\otimes_D \phi_D([m_2:n_2]);\end{aligned}$$

- injective: $\ker(\phi_D) = \{[1:0]\}$ since, for any $[m:n] \in \mathbb{P}_D$,

$$\begin{aligned}\phi_D([m:n]) = (1,0) &\Leftrightarrow \begin{cases} m^2 - Dn^2 = m^2 + Dn^2, \\ 0 = 2mn \end{cases}\\ &\Leftrightarrow n = 0;\end{aligned}$$

- surjective: if $(x,0) \in \mathcal{C}_D$, then $x^2 = 1$, which has solutions $x = \pm 1$. Specifically, these correspond to

$$\begin{aligned}\phi_D([1:0]) &= (1,0),\\ \phi_D([0:1]) &= (-1,0).\end{aligned}$$

  If $(x,y) \in \mathcal{C}_D$ with $y \neq 0$, so that $x \neq \pm 1$ and $D = \frac{x^2-1}{y^2} \neq 0$. The preimage of such (x,y) is an element $[m:n] \in \phi_D$ such that

$$\begin{cases} x = \frac{m^2y^2+(x^2-1)n^2}{m^2y^2-(x^2-1)n^2}, \\ y = \frac{2mny^2}{m^2y^2-(x^2-1)n^2} \end{cases} \Leftrightarrow \begin{cases} m^2y^2-2mnxy+n^2(x^2-1) = 0, \\ m^2y^2 - n^2(x^2-1) = 2mny. \end{cases}$$

  When subtracting the second equation to the first one, the resulting equation is given by

$$2n^2(x-1)^2 = 2mn(x-1).$$

  Since $x \neq 1$ and $n \neq 0$, this gives $m = n\frac{x+1}{y}$ and the surjectivity is confirmed because $\phi_D([x+1:y]) = (x,y)$.

In conclusion, ϕ_D is a group isomorphism. □

This proof gives as the inverse of ϕ_D the group homomorphism

$$\begin{aligned}\phi_D^{-1} : \left(\mathcal{C}_D, \otimes_D\right) &\xrightarrow{\sim} \left(\mathbb{P}_D, \otimes_D\right),\\ (-1,0) &\longmapsto [0:1],\\ (x,y) &\longmapsto [x+1:y].\end{aligned}$$

From a geometrical point of view, when taking the canonical representative of the image, the first entry results in the slope of the line through $(-1,0)$ and (x,y) evaluated with x depending on y, except for the image of $(1,0)$ which is the point at infinity $\alpha = [1:0]$.

The group isomorphism ϕ_D is also a direct method to generate all the solutions of the Pell equation $x^2 - Dy^2 = 1 \in \mathbb{F}$ from the elements of $\mathbb{P}_D$, which requires half the size to be stored, since when using the canonical representatives in $\mathbb{P}_D$, it becomes

$$\phi_D(m) = \begin{cases} \left(\frac{2m}{m^2-D}m-1, \frac{2m}{m^2-D}\right) = \left(\frac{m^2+D}{m^2-D}, \frac{2m}{m^2-D}\right), & \text{if } m \neq \alpha, \\ (1,0), & \text{otherwise.} \end{cases} \tag{2.4}$$

2.2 THE PELL CONIC OVER FINITE FIELDS

When $\mathbb{F} = \mathbb{F}_q$ with $q = p^k$ and p odd prime, the group structure of the Pell conic depends on the quadratic character in $\mathbb{F}_q$ of the parameter D, i.e.,

$$\chi_q(D) = \begin{cases} -1, & \text{if } D \text{ is a non–square in } \mathbb{F}_q, \\ 0, & \text{if } D = 0, \\ 1, & \text{if } D \text{ is a square in } \mathbb{F}_q. \end{cases}$$

In the non–zero cases, it is easy to obtain also the isomorphism between two Pell conics with different parameters.

THEOREM 2.2 For $\chi_q(D) = \chi_q(D')$, there is $\delta \in \mathbb{F}_q^\times$ such that $D = \delta^2 D'$ and the following map is a group isomorphism between $\mathcal{C}_D$ and $\mathcal{C}_{D'}$:

$$\begin{aligned} \delta_{D,D'} : \left(\mathcal{C}_D, \otimes_D\right) &\xrightarrow{\sim} \left(\mathcal{C}_{D'}, \otimes_{D'}\right), \\ (x,y) &\longmapsto (x, \delta y). \end{aligned}$$

Proof. In order for $\delta_{D,D'}$ to be a group isomorphism, it must be

- well defined: $\delta_{D,D'}(\mathcal{C}_D) \subseteq \mathcal{C}_{D'}$ since, for any $(x,y) \in \mathcal{C}_D$,

$$\mathcal{N}_{D'}(x, \delta y) = x^2 - D'(\delta y)^2 = x^2 - D'\delta^2 y^2 = x^2 - Dy^2 = 1;$$

- a group homomorphism: for any $(x_1,y_1), (x_2,y_2) \in \mathcal{C}_D$,

$$\begin{aligned} \delta_{D,D'}\left((x_1,y_1) \otimes_D (x_2,y_2)\right) &= \delta_{D,D'}(x_1x_2 + Dy_1y_2, x_1y_2 + y_1x_2) \\ &= \left(x_1x_2 + Dy_1y_2, \delta(x_1y_2 + y_1x_2)\right) \\ &= (x_1x_2 + D'\delta^2 y_1y_2, x_1\delta y_2 + \delta y_1x_2) \\ &= \delta_{D,D'}(x_1,y_1) \otimes_{D'} \delta_{D,D'}(x_2,y_2). \end{aligned}$$

- injective: for any $(x,y) \in \mathcal{C}_D$,

$$\delta_{D,D'}(x,y) = (1,0) \Leftrightarrow (x, \delta y) = (1,0) \Leftrightarrow (x,y) = (1,0);$$

- surjective: the preimage of $(x,y) \in \mathcal{C}_{D'}$ is simply $(x, y/\delta) \in \mathcal{C}_D$.

In conclusion, $\delta_{D,D'}$ is a group isomorphism. □

The map $\delta_{D,D'}$ can be also a group isomorphism between projectivizations that, if $D = \delta^2 D'$, is given by

$$\begin{aligned}\delta_{D,D'} : \left(\mathbb{P}_D, \otimes_D\right) &\xrightarrow{\sim} \left(\mathbb{P}_{D'}, \otimes_{D'}\right),\\ m &\longmapsto m/\delta.\end{aligned}$$

The cases with $\chi_q(D) \neq 0$ are fully described by Menezes and Vanstone [42], which give also the order of the Pell conic in both situations, i.e., the number of solutions of the Pell equation over finite fields. The case with $\chi_q(D) = -1$ is tackled in Section 2.2.1, while Section 2.2.2 deals with the case with $\chi_q(D) = 1$. These subsections include the proofs from [42] as well as new alternative proofs connected to the parametrization ϕ_D from Theorem 2.1. This introduces the ideas that will be exploited to study the cubic Pell equation over finite fields in Section 3.2.

2.2.1 D *non–square*

When $\chi_q(D) = -1$, the polynomial $t^2 - D \in \mathbb{F}_q[t]$ is irreducible over $\mathbb{F}_q$, so that

$$\mathcal{R}_D = \mathbb{F}_q[t]/\langle t^2 - D\rangle \cong \mathbb{F}_{q^2},$$

and the following result holds.

THEOREM 2.3 [42] If $\chi_q(D) = -1$, then $\left(\mathcal{C}_D, \otimes_D\right)$ is a cyclic group of order $q+1$.

Proof. Since $\mathcal{R}_D^{\otimes_D} \cong \mathbb{F}_{q^2}^{\times}$ has $q^2 - 1$ elements, there is a multiplicative subgroup $G \subset \mathbb{F}_{q^2}^{\times}$ of order $q+1$. In particular, $x + yt \in G$ if and only if $(x+yt)^{q+1} = 1$, where

$$\begin{aligned}(x+yt)^{q+1} &= (x+yt)^q(x+yt) = (x+yt^q)(x+yt)\\ &= \left(x + y(t^2)^{(q-1)/2}t\right)(x+yt) = \left(x + yD^{(q-1)/2}t\right)(x+yt)\\ &= (x-yt)(x+yt) = x^2 - Dy^2,\end{aligned}$$

so that $x+yt \in G$ if and only if $(x,y) \in \mathcal{C}_D$. This association is a group isomorphism between G and $\left(\mathcal{C}_D, \otimes_D\right)$, hence the Pell conic is a cyclic group of order $q+1$. □

Looking at $\mathbb{P}_D$, since there are no square roots of D in $\mathbb{F}_q$, then $\#\mathbb{P}_D = q+1$ from Equation 2.2. This is confirmed also by considering

$$\left(\mathbb{P}_D, \otimes_D\right) = \mathcal{R}_D^{\otimes_D}/\mathbb{F}_q^{\times} \cong \mathbb{F}_{q^2}^{\times}/\mathbb{F}_q^{\times},$$

which proves also that $\mathbb{P}_D$ is cyclic because quotient of cyclic groups.

Thus, using the group isomorphism ϕ_D obtained in Theorem 2.1 for a general field gives an alternative proof that $\left(\mathcal{C}_D, \otimes_D\right)$ is cyclic of order $q+1$. In addition, ϕ_D allows to describe each point of the conic with half the size with respect to the group isomorphism obtained in Theorem 2.3.

2.2.2 D *square*

When $\chi_q(D) = 1$, $\mathcal{R}_D$ is a ring and, as in the previous case, the Pell conic is a cyclic group.

THEOREM 2.4 [42] If $\chi_q(D) = 1$, then $\left(\mathcal{C}_D, \otimes_D\right)$ is a cyclic group of order $q-1$.

Proof. Fixed a square root $s \in \mathbb{F}_q^\times$ of D, the norm $\mathcal{N}_D$ of $(x,y) \in \mathcal{C}_D$ can be factorized as

$$1 = x^2 - Dy^2 = (x - sy)(x + sy) = uv,$$

so that

$$x = \frac{v+u}{2}, \quad y = \frac{v-u}{2s},$$

which results in a bijective correspondence between $(x,y) \in \mathcal{C}_D$ and $(u,v) \in \mathbb{F}_q^2$ such that $uv = 1$. This equation has exactly $q-1$ solutions in $\mathbb{F}_q^2$ and, in particular, a unique solution for each $u \in \mathbb{F}_q^\times$. Thus,

$$\begin{aligned} \left(\mathcal{C}_D, \otimes_D\right) &\xleftrightarrow{\sim} \mathbb{F}_q^\times, \\ (x,y) &\longmapsto x - sy, \\ \left(\frac{1+u^2}{2u}, \frac{1-u^2}{2su}\right) &\longmapsto u, \end{aligned}$$

is a group isomorphism, i.e., $\left(\mathcal{C}_D, \otimes_D\right)$ is cyclic of order $q-1$. □

When considering $\mathbb{P}_D$, from Equation 2.2, $\#\mathbb{P}_D = q-1$. This is confirmed by the following result.

THEOREM 2.5 If $\chi_q(D) = 1$, then $\left(\mathbb{P}_D, \otimes_D\right)$ is a cyclic group of order $q-1$.

Proof. Fixed a square root $s \in \mathbb{F}_q^\times$ of D, $t^2 - D$ is reducible over $\mathbb{F}_q$ as

$$t^2 - D = (t-s)(t+s),$$

so that, using the Chinese remainder theorem, there is the ring isomorphism

$$\begin{aligned} \mathcal{R}_D = \mathbb{F}_q[t]/\langle t^2 - D\rangle &\xrightarrow{\sim} \mathbb{F}_q[t]/\langle t-s\rangle \times \mathbb{F}_q[t]/\langle t+s\rangle, \\ x + yt &\longmapsto (x+sy, x-sy). \end{aligned}$$

In addition, $\mathbb{F}_q[t]/\langle t-s\rangle \cong \mathbb{F}_q[t]/\langle t+s\rangle \cong \mathbb{F}_q$, so that when passing to the quotients there is the group isomorphism

$$\begin{aligned} \left(\mathbb{P}_D, \otimes_D\right) = \mathcal{R}_D^{\otimes_D}/\mathbb{F}_q^\times &\xleftrightarrow{\sim} (\mathbb{F}_q^\times \times \mathbb{F}_q^\times)/\mathbb{F}_q^\times \cong \mathbb{F}_q^\times, \\ [m:n] &\longmapsto \frac{m-sn}{m+sn}, \\ [s(1+u) : 1-u] &\longmapsto u, \end{aligned}$$

which confirms that $\left(\mathbb{P}_D, \otimes_D\right)$ is a cyclic group of order $q-1$. □

In particular, the composition of the group isomorphisms obtained in Theorem 2.4 and Theorem 2.5 is

$$\begin{aligned}
&\left(\mathbb{P}_D, \otimes_D\right) \xrightarrow{\sim} \mathbb{F}_q^\times \qquad \xrightarrow{\sim} \left(\mathcal{C}_D, \otimes_D\right), \\
&[m:n] \quad \longmapsto \frac{m-sn}{m+sn} \longmapsto \left(\frac{1+\left(\frac{m-sn}{m+sn}\right)^2}{2\frac{m-sn}{m+sn}}, \frac{1-\left(\frac{m-sn}{m+sn}\right)^2}{2s\frac{m-sn}{m+sn}}\right) = (x,y),
\end{aligned}$$

where

$$\begin{aligned}
(x,y) &= \left(\frac{(m+sn)^2+(m-sn)^2}{2(m-sn)(m+sn)}, \frac{(m+sn)^2-(m-sn)^2}{2s(m-sn)(m+sn)}\right) \\
&= \left(\frac{2m^2+2Dn^2}{2(m^2-Dn^2)}, \frac{4smn}{2s(m^2-Dn^2)}\right) = \left(\frac{m^2+Dn^2}{m^2-Dn^2}, \frac{2mn}{m^2-Dn^2}\right).
\end{aligned}$$

The inverse is given by

$$\begin{aligned}
&\left(\mathcal{C}_D, \otimes_D\right) \xrightarrow{\sim} \mathbb{F}_q^\times \qquad \xrightarrow{\sim} \left(\mathbb{P}_D, \otimes_D\right), \\
&(x,y) \quad \longmapsto x-sy \longmapsto [s(1+x-sy):1-x+sy] = [m:n],
\end{aligned}$$

where, if $1+x+sy \neq 0$, then

$$\begin{aligned}
[m:n] &= [s(1+x-sy)(1+x+sy) : (1-x+sy)(1+x+sy)] \\
&= [s(1+2x+x^2-Dy^2) : 1+2sy+Dy^2-x^2] \\
&= [2s(1+x) : 2sy] = [1+x:y],
\end{aligned}$$

while

$$\begin{cases} 1+x+sy=0, \\ x^2-Dy^2=1 \end{cases} \Rightarrow \begin{cases} x=-1-sy, \\ 1+2sy+Dy^2-Dy^2=1 \end{cases} \Rightarrow \begin{cases} x=-1, \\ y=0, \end{cases}$$

so that

$$[m:n] = \begin{cases} [0:1], & \text{if } (x,y)=(-1,0), \\ [1+x:y], & \text{otherwise.} \end{cases}$$

These are exactly ϕ_D and ${\phi_D}^{-1}$ obtained for a general field in Theorem 2.1, which allow to describe each point of the conic with half the size with respect to the group isomorphism obtained in Theorem 2.4.

2.3 GENERALIZED PELL CONICS

In this section, a generalization of the Pell equation and the resulting generalized Pell conic are introduced. In addition, an explicit group isomorphism between the standard Pell conic and a generalized Pell conic is obtained.

DEFINITION 2.3 Given the parameters $D, Q \in \mathbb{F}^\times$, the solutions of a generalized Pell equation $x^2 - Dy^2 = Q$ correspond to the elements of $\mathcal{R}_D$ with norm equal to Q, which are the points of the *generalized Pell conic with parameter* D *and norm* Q:

$$\mathcal{C}_{D,Q} = \{(x, y) \in \mathbb{F}^2 \mid x^2 - Dy^2 = Q\}.$$

This is the canonical form of all hyperbolas and ellipses.

The classical Brahmagupta product with parameter D does not give a group structure on $\mathcal{C}_{D,Q}$, but it can be used to define the *generalized Brahmagupta product with identity* $(a, b) \in \mathcal{C}_{D,Q}$ as

$$(x_1, y_1) \otimes_{D,Q,a,b} (x_2, y_2) = \frac{1}{Q}(a, -b) \otimes_D (x_1, y_1) \otimes_D (x_2, y_2).$$

Clearly, the identity point for $\otimes_{D,Q,a,b}$ is the chosen point $(a, b) \in \mathcal{C}_{D,Q}$, the inverse of a point $(x, y) \in \mathcal{C}_{D,Q}$ is the point

$$\frac{1}{Q}(a, b) \otimes_D (a, b) \otimes_D (x, -y),$$

and $\left(\mathcal{C}_{D,Q}, \otimes_{D,Q,a,b}\right)$ is a commutative group.

When $Q = 1$ and the chosen identity point is $(a, b) = (1, 0)$, the product $\otimes_{D,Q,a,b}$ coincides with the classical $\otimes_D$.

Despite the introduction of the new parameter $Q \in \mathbb{F}^\times$, it is possible to obtain the explicit group isomorphism between two generalized Pell conics with same D by exploiting the definition of $\otimes_{D,Q,a,b}$.

THEOREM 2.6 Given $D, Q \in \mathbb{F}^\times$ and a point $(a, b) \in \mathcal{C}_{D,Q}$, the following map is a group isomorphism between $\mathcal{C}_D$ and $\mathcal{C}_{D,Q}$:

$$\begin{aligned} \tau_{D,Q}^{a,b} : \left(\mathcal{C}_D, \otimes_D\right) &\xrightarrow{\sim} \left(\mathcal{C}_{D,Q}, \otimes_{D,Q,a,b}\right), \\ (x, y) &\longmapsto (a, b) \otimes_D (x, y). \end{aligned}$$

Proof. In order for $\tau_{D,Q}^{a,b}$ to be a group isomorphism, it must be

- well defined: $\tau_{D,Q}^{a,b}(\mathcal{C}_D) \subseteq \mathcal{C}_{D,Q}$ since, for any $(x, y) \in \mathcal{C}_D$,

$$\mathcal{N}_D\big((a, b) \otimes_D (x, y)\big) = \mathcal{N}_D(a, b)\,\mathcal{N}_D(x, y) = Q;$$

- a group homomorphism: for any $(x_1, y_1), (x_2, y_2) \in \mathcal{C}_D$,

$$\begin{aligned} \tau_{D,Q}^{a,b}\big((x_1, y_1) \otimes_D (x_2, y_2)\big) &= (a, b) \otimes_D (x_1, y_1) \otimes_D (x_2, y_2) \\ &= \frac{(a, -b) \otimes_D (a, b)}{Q} \otimes_D (a, b) \otimes_D (x_1, y_1) \otimes_D (x_2, y_2) \\ &= \tau_{D,Q}^{a,b}(x_1, y_1) \otimes_{D,Q,a,b} \tau_{D,Q}^{a,b}(x_2, y_2); \end{aligned}$$

- injective: for any $(x, y) \in \mathcal{C}_D$,

$$\tau_{D,Q}^{a,b}(x, y) = (a, b) \Leftrightarrow (x, y) = (1, 0);$$

- surjective: for any $(x, y) \in \mathcal{C}_{D,Q}$,

$$\begin{aligned}(x, y) &= \frac{(a, b) \otimes_D (a, -b)}{Q} \otimes_D (x, y) \\ &= (a, b) \otimes_D (1, 0) \otimes_{D,Q,a,b} (x, y) \\ &= \tau_{D,Q}^{a,b}\big((1, 0) \otimes_{D,Q,a,b} (x, y)\big),\end{aligned}$$

  where $(1, 0) \otimes_{D,Q,a,b} (x, y) \in \mathcal{C}_D$ since it has unitary norm $\mathcal{N}_D$.

Thus, $\tau_{D,Q}^{a,b}$ is a group isomorphism. □

This constructive proof gives the inverse group homomorphism

$$\begin{aligned}(\tau_{D,Q}^{a,b})^{-1} : \big(\mathcal{C}_{D,Q}, \otimes_{D,Q,a,b}\big) &\xrightarrow{\sim} \big(\mathcal{C}_D, \otimes_D\big), \\ (x, y) &\longmapsto (1, 0) \otimes_{D,Q,a,b} (x, y).\end{aligned}$$

The composition of $(\tau_{D,Q}^{a,b})^{-1}$ with $\tau_{D,Q'}^{a',b'}$ is an explicit group isomorphism between generalized Pell conics with same D:

$$\begin{aligned}\big(\mathcal{C}_{D,Q}, \otimes_{D,Q,a,b}\big) &\xrightarrow{\sim} \big(\mathcal{C}_{D,Q'}, \otimes_{D,Q',a',b'}\big), \\ (x, y) &\longmapsto (a', b') \otimes_{D,Q,a,b} (x, y).\end{aligned} \tag{2.5}$$

Since the group $\big(\mathbb{P}_D, \otimes_D\big)$ introduced in Definition 2.2 is independent of the choice of the parameter $Q \in \mathbb{F}^\times$ and of the identity point $(a, b) \in \mathcal{C}_{D,Q}$, all the results in Section 2.1 can be adapted to generalized Pell conics. In particular, the composition of ϕ_D with $\tau_{D,Q}^{a,b}$ results in the group isomorphism between $\mathbb{P}_D$ and $\mathcal{C}_{D,Q}$, explicitly given by

$$\begin{aligned}\phi_{D,Q}^{a,b} : \big(\mathbb{P}_D, \otimes_D\big) &\xrightarrow{\sim} \big(\mathcal{C}_{D,Q}, \otimes_{D,Q,a,b}\big), \\ [m : n] &\longmapsto (a, b) \otimes_D \phi_D([m : n]) = \frac{(a, b) \otimes_D (m, n)^{\otimes_D 2}}{\mathcal{N}_D(m, n)},\end{aligned}$$

that, using the canonical representatives in $\mathbb{P}_D$, becomes

$$\phi_{D,Q}^{a,b}(m) = \begin{cases} \left(2\frac{am+Db}{m^2-D}m - a, 2\frac{am+Db}{m^2-D} + b\right), & \text{if } m \neq \alpha, \\ (a, b), & \text{otherwise,} \end{cases}$$

which is a generalization of Equation 2.4, with inverse given by

$$\begin{aligned}(\phi_{D,Q}^{a,b})^{-1} : \big(\mathcal{C}_{D,Q}, \otimes_{D,Q,a,b}\big) &\xrightarrow{\sim} \big(\mathbb{P}_D, \otimes_D\big), \\ (-a, b) &\longmapsto [Db, -a], \\ (x, y) &\longmapsto [x + a : y - b].\end{aligned}$$

Using the canonical representatives in $\mathbb{P}_D$, the inverse becomes

$$(\phi_{D,Q}^{a,b})^{-1}(x, y) = \begin{cases} \frac{x+a}{y-b}, & \text{if } y \neq b, \\ -D\frac{b}{a}, & \text{if } (x, y) = (-a, b), \\ \alpha, & \text{if } (x, y) = (a, b). \end{cases}$$

This parametrization and its inverse can be used as an alternative way to obtain the group isomorphism in Equation 2.5.

From a geometrical point of view, the parameter m of a point (x, y) is the slope of the line through (x, y) and $(-a, b)$ written considering x variable with y, analogously to the geometric interpretation of $\otimes_D$.

All the results in Section 2.2 can be adapted for generalized Pell conics in the case of finite fields. The main result is that, for all the parameters $D, Q \in \mathbb{F}_q$ and any identity point $(a, b) \in \mathcal{C}_{D,Q}$, the group $\left(\mathcal{C}_{D,Q}, \otimes_{D,Q,a,b}\right)$ is cyclic of order $q - \chi_q(D)$.

In addition, if $\left(\mathcal{C}_{D,Q}, \otimes_{D,Q,a,b}\right)$ and $\left(\mathcal{C}_{D',Q'}, \otimes_{D',Q',a',b'}\right)$ have $D \neq D'$ but $\chi_q(D) = \chi_q(D')$, then the composition of $(\tau_{D,Q}^{a,b})^{-1}$, $\delta_{D,D'}$ and $\tau_{D',Q'}^{a',b'}$ results in a group isomorphism between the two generalized Pell conics given explicitly by

$$\tau_{D',Q'}^{a',b'} \circ \delta_{D,D'} \circ (\tau_{D,Q}^{a,b})^{-1}(x, y) = \frac{1}{Q}\left(a'(ax - Dby) + D'b'\delta(ay - bx),\right.$$
$$\left. a'\delta(ay - bx) + b'(ax - Dby)\right).$$

2.4 EXPONENTIATION AND RÉDEI POLYNOMIALS

In this section, an alternative algorithm for the exponentiation with respect to $\otimes_D$ is described.

Improving the exponentiation algorithm is very useful because it is the usual computational bottleneck in practical implementations. Generally, exponentiation is implemented with a square–multiply algorithm, eventually enhanced with a precomputation phase, so that the total time is mainly determined by the speed of the single product, which is required in both square and multiply steps. Thus, the new algorithm is compared with the exponentiation with $\otimes_D$ through the classical square–multiply algorithm.

As can be easily observed in Equation 2.1, the operation $\otimes_D$ on $\mathcal{C}_D$ requires 5 products and 2 additions in $\mathbb{F}$, while $\otimes_D$ between canonical representatives in $\mathbb{P}_D$ from Equation 2.3 requires 1 inversion, 2 products and 2 additions in $\mathbb{F}$. However, the inversion is largely more expensive than the additional 3 products required in the first case.

Therefore, in a comparison of square–multiply implementations, the first one is the most efficient. Figure 2 describes the algorithm `BrahSquaMult` for the square–multiply exponentiation with $\otimes_D$ on $\mathcal{C}_D$.

An interesting improvement exploits that the exponentiation of a canonical representative in $\mathbb{P}_D$ can be evaluated with Rédei polynomials or Rédei rational functions. They are classical number theory objects firstly introduced in [53], while a more general overview can be found in [39]. Here only the definitions and the useful properties are recalled, while alternative Rédei polynomials are introduced in order to have an improvement also for the exponentiation over $\mathcal{C}_D$.

```
BrahSquaMult(D, x, y, k):
1.  a, b = 1, 0
2.  kbin = binary(k)
3.  for bit in kbin:
4.      a, b = a^2 + D*b^2, 2*a*b
5.      if bit == 1:
6.          a, b = a*x + D*b*y, a*y + b*x
7.  return a, b
```

Figure 2: Square–multiply algorithm with $\otimes_D$ on $\mathcal{C}_D$.

DEFINITION 2.4 Given a parameter $D \in \mathbb{F}^\times$ with $t^2 = D$ not necessarily in $\mathbb{F}$, the classical *Rédei polynomials* $(A_k)_{k\geqslant 0}$ and $(B_k)_{k\geqslant 0}$ result from

$$(m+t)^k = A_k(D,m) + B_k(D,m)t, \quad \text{for } k \geqslant 0.$$

These two sequences clearly correspond to the coordinates resulting from $[m:1]^{\otimes_D k} \in \mathbb{P}_D$.

The ratios of Rédei polynomials are the *Rédei rational functions*

$$Q_k(D,m) = \frac{A_k(D,m)}{B_k(D,m)}, \quad \text{for } k > 0,$$

which are the canonical representatives of $m^{\otimes_D k} \in \mathbb{P}_D$.

On the other hand, considering the coordinates of $(x,y)^{\otimes_D k} \in \mathcal{C}_D$, it is possible to define the *generalized Rédei polynomials* as

$$(x+yt)^k = a_k(D,x,y) + b_k(D,x,y)t, \quad \text{for } k \geqslant 0.$$

In particular, the classical Rédei polynomials for $k \geqslant 0$ are

$$A_k(D,m) = \sum_{i=0}^{\lfloor k/2\rfloor} \binom{k}{2i} D^i m^{k-2i} = \sum_{\substack{i+j=k,\\ j \text{ even}}} \binom{k}{i,j} D^{j/2} m^i,$$

$$B_k(D,m) = \sum_{i=0}^{\lfloor k/2\rfloor} \binom{k}{2i+1} D^i m^{k-2i-1} = \sum_{\substack{i+j=k,\\ j \text{ odd}}} \binom{k}{i,j} D^{(j-1)/2} m^i.$$

On the other hand, the generalized Rédei polynomials for $k \geqslant 0$ can be written as

$$a_k(D,x,y) = \sum_{i=0}^{\lfloor k/2\rfloor} \binom{k}{2i} D^i x^{k-2i} y^{2i} = \sum_{\substack{i+j=k,\\ j \text{ even}}} \binom{k}{i,j} D^{j/2} x^i y^j,$$

$$b_k(D,x,y) = \sum_{i=0}^{\lfloor k/2\rfloor} \binom{k}{2i+1} D^i x^{k-2i-1} y^{2i+1} = \sum_{\substack{i+j=k,\\ j \text{ odd}}} \binom{k}{i,j} D^{(j-1)/2} x^i y^j,$$

where the multinomial coefficients are $\binom{k}{i,j} = \frac{k!}{i!\,j!}$.

```
ModMore(D, m, k):
1.  A, B = 1, 0
2.  kbin = binary(k)
3.  for bit in kbin:
4.      A, B = A^2 + DB^2, 2AB
5.      if bit == 1:
6.          A, B = Am + DB, A + Bm
7.  return A/B
```

Figure 3: Modified More algorithm for the exponentiation over $\mathbb{P}_D$ using Rédei rational functions.

It also possible to use a matrix notation, so that

$$\begin{pmatrix} A_k(D,m) \\ B_k(D,m) \end{pmatrix} = \begin{pmatrix} m & D \\ 1 & m \end{pmatrix}^k \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} a_k(D,x,y) \\ b_k(D,x,y) \end{pmatrix} = \begin{pmatrix} x & Dy \\ y & x \end{pmatrix}^k \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

where the second matrix is the same exploited in Chapter 8 for a primality test based on the Pell conic.

In [45], the More algorithm was proposed for evaluating the Rédei rational function Q_k, i.e., the result of the exponentiation in $\mathbb{P}_D$, with complexity $O(\log k)$ considering additions and multiplications over $\mathbb{F}$. In [12], an improvement of the performance of this algorithm was obtained and the result was called modified More algorithm. This improved algorithm is detailed in Figure 3 under the denomination `ModMore`. In a comparison with Figure 2, the two algorithms have the same number of operations at each step except for step 6, where `BrahSquaMult` requires an additional product, and step 7, where `ModMore` requires a final inversion.

Thus, from the point of view of performance, the two algorithms are comparable. The main advantage in evaluating the exponentiation of canonical representatives in $\mathbb{P}_D$ through the modified More algorithm is that the size of the data is halved because they are elements of the field $\mathbb{F}$ and not of the Pell conic $\mathcal{C}_D$.

3

PELL CUBICS

The idea of this chapter is to generalize the results obtained for the Pell conic in Chapter 2 to the cubic Pell equation, which can be defined analogously to the Pell equation of degree 2.

After considering its solutions over a field in Section 3.1, together with a generalization of the Brahmagupta product that gives a group structure, the cases over finite fields are characterized in Section 3.2. A further generalization of the Pell cubic is introduced and studied in Section 3.3, where its relations with the standard Pell cubic are obtained. Finally, in Section 3.4, a generalization of Rédei polynomials and rational functions with dimension three is exploited to obtain a well-performing exponentiation algorithm.

3.1 SOLUTIONS OF THE CUBIC PELL EQUATION OVER A FIELD

As for the classical quadratic Pell equation, it is possible to define the cubic Pell equation over a field $\mathbb{F}$ by taking a parameter $r \in \mathbb{F}^\times$ and the polynomial ring

$$\mathcal{R}_r = \mathbb{F}[t]/\langle t^3 - r \rangle,$$

which inherits from the polynomial product the operation

$$\begin{aligned}&(x_1 + y_1 t + z_1 t^2) \cdot (x_2 + y_2 t + z_2 t^2)\\ &= x_1x_2 + r(y_1z_2 + z_1y_2) + (x_1y_2 + y_1x_2 + rz_1z_2)t + (x_1z_2 + y_1y_2 + z_1x_2)t^2.\end{aligned}$$

DEFINITION 3.1 Given the parameter $r \in \mathbb{F}^\times$, considering the primitive cubic roots of unity ω, ω^2, the conjugates of $x + yt + zt^2 \in \mathcal{R}_r$ are the polynomials

$$x + y\omega t + z\omega^2 t^2, \quad x + y\omega^2 t + z\omega t^2,$$

and, analogously to the quadratic case, a *norm over* $\mathcal{R}_r$ is defined as

$$\begin{aligned}&\mathcal{N}_r(x + yt + zt^2)\\ &= (x + yt + zt^2) \cdot (x + y\omega t + z\omega^2 t^2) \cdot (x + y\omega^2 t + z\omega t^2)\\ &= x^3 + ry^3 + r^2z^3 - 3rxyz.\end{aligned}$$

The unitary elements of $\mathcal{R}_r$ with respect to the norm $\mathcal{N}_r$ are

$$\mathcal{U}(\mathcal{R}_r) = \{x + yt + zt^2 \in \mathcal{R}_r \,|\, \mathcal{N}_r(x + yt + zt^2) = 1\},$$

and form a commutative group.

Thus, it is natural to define the *cubic Pell equation* as

$$x^3 - 3Rxyz + Ry^3 + R^2z^3 = 1,$$

and the *Pell cubic with parameter* R as

$$\mathcal{C}_R = \{(x, y, z) \in \mathbb{F}^3 \,|\, x^3 + Ry^3 + R^2z^3 - 3Rxyz = 1\}.$$

Considering the *cubic Brahmagupta product with parameter* R given by

$$\begin{aligned}&(x_1, y_1, z_1) \odot_R (x_2, y_2, z_2) \qquad (3.1)\\ &= \big(x_1x_2 + R(y_1z_2 + z_1y_2), x_1y_2 + y_1x_2 + Rz_1z_2, x_1z_2 + y_1y_2 + z_1x_2\big),\end{aligned}$$

$\mathcal{C}_R$ is a commutative group with identity $(1, 0, 0)$ and inverse of an element (x, y, z) given the product of its conjugates

$$\begin{aligned}(x, y, z)^{-1} &= (x, y\omega, z\omega^2) \odot_R (x, y\omega^2, z\omega)\\ &= (x^2 - Ryz, Rz^2 - xy, y^2 - xz),\end{aligned}$$

where ω, ω^2 are primitive cubic roots of unity.

Clearly, the group over the Pell cubic is isomorphic to that of the unitary elements of $\mathcal{R}_R$ with respect to the norm $\mathcal{N}_R$, so that in the following $\odot_R$ is used also for denoting the product over $\mathcal{R}_R$, in order to highlight its dependence on R.

In order to find a parametrization for $\mathcal{C}_R$, inspired by Definition 2.2 for the quadratic case, the invertible elements of $\mathcal{R}_R$ with respect to $\odot_R$ are considered. They are all but the non–zero divisors or, equivalently, all those with non–zero norm, i.e.,

$$\mathcal{R}_R^{\odot_R} = \{x + yt + zt^2 \in \mathcal{R}_R \,|\, \mathcal{N}_R(x + yt + zt^2) \neq 0\}.$$

DEFINITION 3.2 The *projectivization of* $\mathcal{R}_R$ is defined as

$$\mathbb{P}_R = \mathcal{R}_R^{\odot_R}/\mathbb{F}^\times.$$

In particular, its elements are of the form

$$[l : m : n] = \{\lambda(l + mt + nt^2) \,|\, \lambda \in \mathbb{F}^\times\}, \quad \text{for } l + mt + nt^2 \in \mathcal{R}_R^{\odot_R}.$$

Analogously to the quadratic case, it is useful to define the *canonical representatives*:

- if $n \in \mathbb{F}^\times$, then $l + mt + nt^2$ is equivalent to $ln^{-1} + mn^{-1}t + t^2$ and the canonical representative is $[ln^{-1} : mn^{-1} : 1] \sim (l', m')$;
- if $n = 0$ and $m \in \mathbb{F}^\times$, then $l + mt$ is equivalent to $lm^{-1} + t$ and the canonical representative is $[lm^{-1} : 1 : 0] \sim (l', \alpha)$;
- finally, if $m = n = 0$, then the canonical representative is simply $[1 : 0 : 0] \sim (\alpha, \alpha)$.

Since $\odot_R$ consists of homogeneous polynomials, it is well defined also on $\mathbb{P}_R$ and determines a commutative group with identity $[1:0:0]$ and inverse of $[l:m:n]$ given by $[l^2-Rmn : Rn^2-lm : m^2-ln]$.

The operation $\odot_R$ over $\mathbb{P}_R$ is easier to be described with canonical representatives, so that the explicit formulas are

$$
\begin{aligned}
&(l_1,m_1)\odot_R(\alpha,\alpha)=(l_1,m_1),\\
&(l_1,\alpha)\odot_R(l_2,\alpha)=(l_1l_2,l_1+l_2),\\
&(l_1,m_1)\ \odot_R\ (l_2,\ \alpha) = \begin{cases} \left(\frac{l_1l_2+R}{m_1+l_2},\frac{l_1+m_1l_2}{m_1+l_2}\right), & \text{if } m_1+l_2\neq 0,\\ \left(\frac{l_1l_2+R}{l_1-m_1^2},\alpha\right), & \text{if } m_1=-l_2, l_1\neq m_1^2,\\ (\alpha,\alpha), & \text{otherwise,}\end{cases} \qquad (3.2)\\
&(l_1,m_1)\ \odot_R\ (l_2,m_2) = \begin{cases} \left(\frac{l_1l_2+R(m_1+m_2)}{l_1+l_2+m_1m_2},\frac{l_1m_2+m_1l_2+R}{l_1+l_2+m_1m_2}\right), & \text{if } l_1+l_2+m_1m_2\neq 0,\\ \left(\frac{l_1l_2+R(m_1+m_2)}{l_1m_2+m_1l_2+R},\alpha\right), & \text{if } \begin{cases} l_1+l_2+m_1m_2=0,\\ l_1m_2+m_1l_2+R\neq 0,\end{cases}\\ (\alpha,\alpha), & \text{otherwise.}\end{cases}
\end{aligned}
$$

The inverses are

$$
\begin{gathered}
[l:m:1]\odot_R[l^2-Rm : R-lm : m^2-l]=[1:0:0],\\
[l:1:0]\odot_R[l^2:-l:1]=[1:0:0],
\end{gathered}
$$

or using the short notation

$$
\begin{aligned}
(l,\alpha)^{-1}&=(l^2,-l),\\
(m^2,m)^{-1}&=(-m,\alpha),\\
(l,m)^{-1}&=\left(\frac{l^2-Rm}{m^2-l},\frac{R-lm}{m^2-l}\right),\quad \text{for } l\neq m^2, m\neq\alpha.
\end{aligned}
$$

The characterization of $\mathbb{P}_R$ depends on the parameter $R\in\mathbb{F}^\times$:

1. if R is a non–cube, then the null polynomial is the one and only with zero norm and

$$
\begin{aligned}
\mathbb{P}_R&=\{[l:m:1],[l:1:0],[1:0:0]\mid l,m\in\mathbb{F}\}\\
&\sim(\mathbb{F}\times\mathbb{F})\cup(\mathbb{F}\times\{\alpha\})\cup\{(\alpha,\alpha)\};
\end{aligned}
\qquad (3.3)
$$

2. if R is a cube and $\{1,\omega,\omega^2\}\subset\mathbb{F}$, then $\mathbb{F}$ contains all the cubic roots of R that, when denoting one of them with s, are $\{s,s\omega,s\omega^2\}$. In this case, t^3-R can be completely decomposed in three factors which are zero–divisors in $\mathcal{R}_R$, so that $\mathcal{R}_R^{\odot_R}$ contains all but the multiples of these 3 polynomials, i.e.,

$$
\mathcal{R}_R^{\odot_R}=\mathcal{R}_R\setminus\langle t-s,t-s\omega,t-s\omega^2\rangle.
$$

Thus, in the explicit description of $\mathbb{P}_R$, the only elements to be excluded are those in $\langle[-s:1:0],[-s\omega:1:0],[-s\omega^2:1:0]\rangle$.

The multiples of $[-s:1:0]$ are, for any $l \in \mathbb{F}$,

$$[-s:1:0] \odot_R [l:1:0] = [-ls:-s+l:1],$$

and, for any $l', m' \in \mathbb{F}$ with $[l':m':1] \neq [s^2:s:1]$,

$$\begin{aligned}
[-s:1:0] \odot_R [l':m':1] &= [-l's+s^3 : -m's+l' : -s+m'] \\
&= \begin{cases} [-s(l'-s^2) : l'-s^2 : 0], & \text{if } m' = s, \\ \left[-\left(\frac{l'-s^2}{m'-s}\right)s : \left(\frac{l'-s^2}{m'-s}\right) - s : 1\right], & \text{otherwise} \end{cases} \\
&= \begin{cases} [-s:1:0], & \text{if } m' = s, \\ [-ls:l-s:1], & \text{with } l = \frac{l'-s^2}{m'-s} \text{ otherwise.} \end{cases}
\end{aligned}$$

Analogous results are obtained for the multiples of $[-s\omega:1:0]$ and $[-s\omega^2:1:0]$, so that

$$\begin{aligned}
\langle[-s:1:0]\rangle &= \{[-s:1:0],[-ls:l-s:1] \mid l \in \mathbb{F}\}, \\
\langle[-s\omega:1:0]\rangle &= \{[-s\omega:1:0],[-ls\omega:l-s\omega:1] \mid l \in \mathbb{F}\}, \\
\langle[-s\omega^2:1:0]\rangle &= \{[-s\omega^2:1:0],[-ls\omega^2:l-s\omega^2:1] \mid l \in \mathbb{F}\}.
\end{aligned}$$

In order to characterize their intersections, if $0 \leqslant i < j \leqslant 2$, then $[-ls\omega^i : l-s\omega^i : 1] = [-l's\omega^j : l'-s\omega^j : 1]$ if and only if

$$\begin{cases} l = l'\omega^{j-i}, \\ l - l' = s(\omega^i - \omega^j) \end{cases} \Leftrightarrow \begin{cases} l = -s\omega^j, \\ l' = -s\omega^i. \end{cases}$$

This means that, by exploiting the identity $1 + \omega + \omega^2 = 0$,

$$\begin{aligned}
\langle[-s:1:0]\rangle \cap \langle[-s\omega:1:0]\rangle &= \{[s^2\omega : s\omega^2 : 1]\}, \\
\langle[-s\omega:1:0]\rangle \cap \langle[-s\omega^2:1:0]\rangle &= \{[s^2 : s : 1]\}, \\
\langle[-s\omega^2:1:0]\rangle \cap \langle[-s:1:0]\rangle &= \{[s^2\omega^2 : s\omega : 1]\}.
\end{aligned}$$

Thus, the list of the elements not in $\mathbb{P}_R$ can be obtained from the union of the three sets of the multiples of each zero–divisor while considering that three elements are obtained twice. In particular, the duplicates can be removed by excluding for each $0 \leqslant i \leqslant 2$ the element with second coordinate $m = s\omega^{i+2}$, i.e.,

$$\begin{aligned}
\mathbb{P}_R = \{&[l:m:1],[l:1:0],[1:0:0] \mid l, m \in \mathbb{F}\} \\
&\setminus \cup_{i=0}^{2} \{[-s\omega^i : 1 : 0], [-s\omega^i(m + s\omega^i) : m : 1] \mid m \neq s\omega^{i+2}\};
\end{aligned} \tag{3.4}$$

3. if R is a cube and $\{\omega, \omega^2\} \not\subset \mathbb{F}$, then only one root s of R is in $\mathbb{F}$, so that there are only two zero–divisors

$$t^3 - R = (t-s)(t^2 + st + s^2).$$

This means that the invertible elements in $\mathcal{R}_R$ are

$$\mathcal{R}_R^{\odot_R} = \mathcal{R}_R \setminus \langle t - s, t^2 + st + s^2 \rangle.$$

Thus, in the explicit description of $\mathbb{P}_R$, the only elements to be excluded are those in $\langle [-s : 1 : 0], [s^2 : s : 0] \rangle$.

As in the previous scenario,

$$\begin{aligned} \langle [-s : 1 : 0] \rangle &= \{ [-s : 1 : 0], [-ls : l - s : 1] \mid l \in \mathbb{F} \} \\ &= \{ [-s : 1 : 0], [-s(m + s) : m : 1] \mid m \in \mathbb{F} \}, \end{aligned}$$

while, the second zero–divisor does not have any new multiple since

$$[s^2 : s : 1] \odot_R [-s : 1 : 0] = [0 : 0 : 0],$$

and for every $[l : m : n]$ non multiple of $[-s : 1 : 0]$

$$[s^2 : s : 1] \odot_R [l : m : n] = [s^2 : s : 1].$$

In conclusion

$$\begin{aligned} \mathbb{P}_R = &\{ [l : m : 1], [l : 1 : 0], [1 : 0 : 0] \mid l, m \in \mathbb{F} \} \\ &\setminus \{ [-s : 1 : 0], [-s(m + s) : m : 1], [s^2 : s : 1] \mid m \in \mathbb{F} \}. \end{aligned} \tag{3.5}$$

However, differently from the quadratic case, the group isomorphism between $(\mathbb{P}_R, \odot_R)$ and $(\mathcal{C}_R, \odot_R)$ is still unknown.

3.2 THE PELL CUBIC OVER FINITE FIELDS

Despite a group isomorphism for the case with a generic field $\mathbb{F}$ is missing, the projectivization $\mathbb{P}_R$ can still be exploited to give a complete characterization of the Pell cubic over a finite field $\mathbb{F}_q$ with $q = p^k$ and p odd prime, generalizing the results in Section 2.2.

This characterization depends on the parameter $r \in \mathbb{F}_q^\times$ and, as observed in the previous section, there are three different scenarios. This is confirmed also by the value of $\gcd(3, q - 1)$ in the extended Euler criterion for cubes in a finite field [1]:

$$r \in \mathbb{F}_q \text{ is a cube} \Leftrightarrow r^{(q-1)/\gcd(3,q-1)} = 1. \tag{3.6}$$

3.2.1 r *non–cube*

From Equation 3.6, a finite field $\mathbb{F}_q$ contains a non–cube element r if and only if $\gcd(3, q - 1) > 1$, i.e., $q \equiv 1 \pmod 3$, in which case $(q - 1)/3 = \lfloor q/3 \rfloor$ and

$$\begin{cases} r^{(q-1)/3} \neq 1, \\ r^{q-1} = 1 \end{cases} \Leftrightarrow r^{\lfloor q/3 \rfloor} = \omega, \text{ primitive cubic root of unity.}$$

In this case, the polynomial $t^3 - r$ is irreducible over $\mathbb{F}_q$, so that

$$\mathcal{R}_r = \mathbb{F}_q[t]/\langle t^3 - r\rangle \cong \mathbb{F}_{q^3},$$

and it is possible to obtain a result analogous to Theorem 2.3.

THEOREM 3.1 If r is a non–cube in $\mathbb{F}_q$, then $\left(\mathcal{C}_r, \odot_r\right)$ is a cyclic group of order $q^2 + q + 1$.

Proof. Clearly, $\mathcal{R}_r^{\odot_r} \cong \mathbb{F}_{q^3}^{\times}$ has $q^3 - 1$ elements. If $G \subset \mathbb{F}_{q^3}^{\times}$ denotes its multiplicative subgroup of order $q^2 + q + 1$, then $x + yt + zt^2 \in G$ if and only if the exponentiation $(x + yt + zt^2)^{q^2+q+1} = 1$ and

$$\begin{aligned}(x + yt + zt^2)^{q^2+q+1} &= (x + yt + zt^2)^{q^2}(x + yt + zt^2)^{q}(x + yt + zt^2) \\ &= (x+yt^q+zt^{2q})^{q}(x+yt^q+zt^{2q})(x + yt + zt^2),\end{aligned}$$

where

$$t^q = (t^3)^{(q-1)/3}t = r^{\lfloor q/3\rfloor}t = \omega t, \quad \omega^q = (\omega^3)^{(q-1)/3}\omega = \omega,$$

so that $(x + yt + zt^2)^{q^2+q+1}$ becomes

$$\begin{aligned}&(x + y\omega t + z\omega^2 t^2)^{q}(x + y\omega t + z\omega^2 t^2)(x + yt + zt^2) \\ &= (x + y\omega^q t^q + z\omega^{2q} t^{2q})(x + y\omega t + z\omega^2 t^2)(x + yt + zt^2) \\ &= (x + y\omega^2 t + z\omega t^q)(x + y\omega t + z\omega^2 t^2)(x + yt + zt^2) \\ &= x^3 - 3rxyz + ry^3 + r^2 z^3.\end{aligned}$$

Thus, $x + yt + zt^2 \in G$ if and only if $(x, y, z) \in \mathcal{C}_r$. This association is a group isomorphism between G and $\left(\mathcal{C}_r, \odot_r\right)$, hence the Pell cubic is a cyclic group of order $q^2 + q + 1$. □

When considering $\mathbb{P}_r$, since there are no cubic roots of r in $\mathbb{F}_q$, then $\#\mathbb{P}_r = q^2 + q + 1$ from Equation 3.3. This is confirmed also by

$$\left(\mathbb{P}_r, \odot_r\right) = \mathcal{R}_r^{\odot_r}/\mathbb{F}_q^{\times} \cong \mathbb{F}_{q^3}^{\times}/\mathbb{F}_q^{\times},$$

which proves also that $\mathbb{P}_r$ is cyclic because quotient of cyclic groups. In addition, it is possible to obtain the following result.

THEOREM 3.2 If $q \equiv 1 \pmod 3$ and $r \in \mathbb{F}_q^{\times}$ is a non–cube, then the following map is a group isomorphism between $\mathbb{P}_r$ and $\mathcal{C}_r$:

$$\begin{aligned}\psi_r' : \left(\mathbb{P}_r, \odot_r\right) &\xrightarrow{\sim} \left(\mathcal{C}_r, \odot_r\right), \\ [l : m : n] &\longmapsto \mathcal{N}_r(l, m, n)^{\lfloor q/3\rfloor - 1}(l, m, n)^{\odot_r 3}.\end{aligned}$$

Proof. In order for ψ_r' to be a group isomorphism, it must be

- well defined: for any $[l : m : n] \in \mathbb{P}_r$, $\lambda \in \mathbb{F}_q^{\times}$,

$$\begin{aligned}\psi_r'([\lambda l : \lambda m : \lambda n]) &= \left(\lambda^3 \mathcal{N}_r(l, m, n)\right)^{(q-4)/3}\left(\lambda^3 (l, m, n)^{\odot_r 3}\right) \\ &= \lambda^{q-1}\psi_r'([l : m : n]) = \psi_r'([l : m : n]).\end{aligned}$$

Moreover, $\psi'_R(\mathbb{P}_R) \subseteq \mathcal{C}_R$ because, for any $[l : m : n] \in \mathbb{P}_R$,

$$\begin{aligned}\mathcal{N}_R\big(\psi'_R([l : m : n])\big) &= \mathcal{N}_R(l, m, n)^{q-4}\mathcal{N}_R(l, m, n)^3 \\ &= \mathcal{N}_R(l, m, n)^{q-1} = 1;\end{aligned}$$

- a group homomorphism: given $[l_1 : m_1 : n_1], [l_2 : m_2 : n_2] \in \mathbb{P}_R$, by denoting their product as $[l : m : n]$, the resulting image is

$$\begin{aligned}\psi'_R([l : m : n]) &= \mathcal{N}_R(l, m, n)^{\lfloor q/3 \rfloor - 1}(l, m, n)^{\odot_R 3} \\ &= \mathcal{N}_R(l_1, m_1, n_1)^{\lfloor q/3 \rfloor - 1}\mathcal{N}_R(l_2, m_2, n_2)^{\lfloor q/3 \rfloor - 1} \\ &\quad (l_1, m_1, n_1)^{\odot_R 3} \odot_R (l_2, m_2, n_2)^{\odot_R 3} \\ &= \psi'_R(l_1, m_1, n_1) \odot_R \psi'_R(l_2, m_2, n_2);\end{aligned}$$

- injective: for any $[l : m : n] \in \mathbb{P}_R$, $\psi'_R([l : m : n]) = (1, 0, 0)$ means

$$\begin{cases} \mathcal{N}_R(l, m, n)^{\lfloor q/3 \rfloor - 1}(l^3 + 6Rlmn + Rm^3 + R^2n^3) = 1, \\ \mathcal{N}_R(l, m, n)^{\lfloor q/3 \rfloor - 1}(3l^2m + 3Rln^2 + 3Rm^2n) = 0, \\ \mathcal{N}_R(l, m, n)^{\lfloor q/3 \rfloor - 1}(3l^2n + 3lm^2 + 3Rmn^2) = 0, \end{cases}$$

  with $\mathcal{N}_R(l, m, n) \neq 0$, so that:

  - if $m, n \neq 0$, then taking the second and third equation

$$\begin{cases} n(l^2m + Rln^2 + Rm^2n) = 0, \\ m(l^2n + lm^2 + Rmn^2) = 0 \end{cases} \Leftrightarrow l(Rn^3 - m^3) = 0,$$

    which is satisfied only when $l = 0$ since R is not a cube. However, this implies $m = 0$ or $n = 0$, i.e., a contradiction;
  - if $m \neq n = 0$, then from the third equation $lm^2 = 0$, i.e., $l = 0$, so that the point is $[l : m : n] = [0 : 1 : 0]$ and the first equation remains $R^{\lfloor q/3 \rfloor} = 1$, which is not true because of the generalized Euler criterion;
  - if $n \neq m = 0$, then from the second equation $Rln^2 = 0$, i.e., $l = 0$, so that the point is $[l : m : n] = [0 : 0 : 1]$ and the first equation remains $R^{2\lfloor q/3 \rfloor} = 1$. Thus, $r^{\lfloor q/3 \rfloor} = \pm 1$, but $r^{\lfloor q/3 \rfloor} = -1$ is not valid since it implies $r^{q-1} = -1$, while $r^{\lfloor q/3 \rfloor} = 1$ is in contradiction with the generalized Euler criterion;
  - $m = n = 0$ is the only remaining option, which implies that $\ker(\psi'_R) = \{[1 : 0 : 0]\}$;

- surjective: this is straightforward because ψ'_R is an injection between two finite groups with same cardinality $q^2 + q + 1$.

In conclusion, ψ'_R is a group isomorphism. □

Since this group isomorphism gives a parametrization of $\mathcal{C}_R$, it allows to find all the solutions of the cubic Pell equation over $\mathbb{F}_q$. Indeed, it is sufficient to evaluate ψ'_R over all the elements of $\mathbb{P}_R$, described explicitly in Equation 3.3. However, the explicit inverse is hard to find, so that it is difficult to obtain the point of $\mathbb{P}_R$ related to a given point of $\mathcal{C}_R$.

Example 3.1. Considering $q = 7$ and $R = 2$, which is not a cube in $\mathbb{F}_7$, the previous results assure that the cubic Pell equation

$$x^3 + 2y^3 + 4z^3 - 6xyz \equiv 1 \pmod 7,$$

admits $q^2 + q + 1 = 57$ solutions and ψ'_2 allows to find all of them as

$$\begin{gathered}\psi'_2([l : m : 1]), \quad \forall\, l, m \in \mathbb{F}_7,\\ \psi'_2([l : 1 : 0]), \quad \forall\, l \in \mathbb{F}_7,\\ \psi'_2([1 : 0 : 0]) = (1, 0, 0).\end{gathered}$$

For instance, a random solution of the cubic Pell equation can be found by taking randomly $l, m \in \mathbb{F}_7$, e.g., $l = 3$ and $m = 5$, and evaluating

$$\psi'_2([3 : 5 : 1]) = (5, 4, 4).$$

It is easy to check that

$$5^3 + 2 \cdot 4^3 + 4 \cdot 4^3 - 6 \cdot 5 \cdot 4 \cdot 4 \equiv 1 \pmod 7.$$

Similarly, when taking $l = 4$ and $[4 : 1 : 0] \in \mathbb{P}_2$,

$$\psi'_2([4 : 1 : 0]) = (2, 4, 1),$$

is another solution of the cubic Pell equation.

It is noteworthy that for large values of q this method for finding all the solutions of the cubic Pell equation is not efficient, since it has complexity $O(q^2)$, even if it is surely better than an exhaustive search that has complexity $O(q^3)$.

However, for large values of q it is really interesting to use the above method for generating random solutions of the cubic Pell equation since, exploiting ψ'_R as in the previous example, it is always possible to generate different solutions.

3.2.2 R *cube with three roots in* $\mathbb{F}_q$

If $q \equiv 1 \pmod 3$, then $\mathbb{F}_q$ contains both the primitive cubic roots of unity ω, ω^2. In addition, if R is a cube, fixed a cubic root $s \in \mathbb{F}_q^\times$ of R, then the other two cubic roots are $\omega s, \omega^2 s$ and $\{s, \omega s, \omega^2 s\} \subseteq \mathbb{F}_q^\times$.

In this case, with a proof analogous to Theorem 2.4, the following result holds.

THEOREM 3.3 If $q \equiv 1 \pmod 3$ and $r \in \mathbb{F}_q^\times$ is a cube, then $\left(\mathcal{C}_r, \odot_r\right)$ is isomorphic to $\mathbb{F}_q^\times \times \mathbb{F}_q^\times$.

Proof. Fixed a cubic root $s \in \mathbb{F}_q^\times$ of r, the norm $\mathcal{N}_r$ of $(x,y,z) \in \mathcal{C}_r$ can be factorized as

$$\begin{aligned} 1 &= x^3 + ry^3 + r^2z^3 - 3rxyz \\ &= (x + \omega sy + \omega^2 s^2 z)(x + \omega^2 sy + \omega s^2 z)(x + sy + s^2 z) = uvw, \end{aligned}$$

so that

$$x = \frac{w+v+u}{3}, \quad y = \frac{w+\omega v+\omega^2 u}{3s}, \quad z = \frac{w+\omega^2 v+\omega u}{3s^2},$$

is a bijective correspondence between $(x,y,z) \in \mathcal{C}_r$ and $(u,v,w) \in \mathbb{F}_q^3$ such that $uvw = 1$. This equation has exactly $(q-1)^2$ solutions in $\mathbb{F}_q^3$ and, in particular, a unique solution for each $(u,v) \in \mathbb{F}_q^\times \times \mathbb{F}_q^\times$. Thus,

$$\begin{gathered} \left(\mathcal{C}_r, \odot_r\right) \xleftrightarrow{\sim} \mathbb{F}_q^\times \times \mathbb{F}_q^\times, \\ (x,y,z) \longmapsto (x + \omega sy + \omega^2 s^2 z, x + \omega^2 sy + \omega s^2 z), \\ \left(\frac{1+uv^2+u^2v}{3uv}, \frac{1+\omega uv^2+\omega^2 u^2 v}{3suv}, \frac{1+\omega^2 uv^2+\omega u^2 v}{3s^2 uv}, \right) \longmapsto (u,v), \end{gathered}$$

is a group isomorphism. □

When considering $\mathbb{P}_r$, it is clear from Equation 3.4 that

$$\#\mathbb{P}_r = q^2 + q + 1 - 3q = (q-1)^2.$$

This is confirmed by the following result, obtained analogously to Theorem 2.5.

THEOREM 3.4 If $q \equiv 1 \pmod 3$ and $r \in \mathbb{F}_q^\times$ is a cube, then $\left(\mathbb{P}_r, \odot_r\right)$ is isomorphic to $\mathbb{F}_q^\times \times \mathbb{F}_q^\times$.

Proof. Fixed s cubic root of r in $\mathbb{F}_q$, $t^3 - r$ is reducible over $\mathbb{F}_q$ as

$$t^3 - r = (t-s)(t-\omega s)(t-\omega^2 s).$$

Thus, the Chinese remainder theorem gives the ring isomorphism

$$\begin{gathered} \mathcal{R}_r = \mathbb{F}_q[t]/\langle t^3 - r\rangle \xrightarrow{\sim} \mathbb{F}_q[t]/\langle t-s\rangle \times \mathbb{F}_q[t]/\langle t-\omega s\rangle \times \mathbb{F}_q[t]/\langle t-\omega^2 s\rangle, \\ x + yt + zt^2 \mapsto (x+sy+s^2z, x+\omega sy+\omega^2 s^2 z, x+\omega^2 sy+\omega s^2 z). \end{gathered}$$

In addition, $\mathbb{F}_q[t]/\langle t-s\rangle \cong \mathbb{F}_q[t]/\langle t-\omega s\rangle \cong \mathbb{F}_q[t]/\langle t-\omega^2 s\rangle \cong \mathbb{F}_q$, and when passing to the quotients there is the map

$$\begin{gathered} \left(\mathbb{P}_r, \odot_r\right) = \mathcal{R}_r^{\odot_r}/\mathbb{F}_q^\times \xleftrightarrow{\sim} (\mathbb{F}_q^\times \times \mathbb{F}_q^\times \times \mathbb{F}_q^\times)/\mathbb{F}_q^\times \cong \mathbb{F}_q^\times \times \mathbb{F}_q^\times, \\ [l:m:n] \longmapsto \left(\frac{l+\omega sm+\omega^2 s^2 n}{l+sm+s^2 n}, \frac{l+\omega^2 sm+\omega s^2 n}{l+sm+s^2 n}\right), \\ [s^2(1+v+u) : s(1+\omega v+\omega^2 u) : 1+\omega^2 v+\omega u] \longmapsto (u,v), \end{gathered}$$

which is a group isomorphism. □

Composing the obtained results gives an explicit group isomorphism between $\mathbb{P}_R$ and $\mathcal{C}_R$ for R cube in $\mathbb{F}_q$ with $q \equiv 1 \pmod 3$:

$$\begin{aligned}\psi''_R : \left(\mathbb{P}_R, \odot_R\right) &\xrightarrow{\sim} \left(\mathcal{C}_R, \odot_R\right),\\ [l : m : n] &\mapsto \left(\frac{l^3 + 2s^2l(m^2+smn+s^2n^2) + s^4mn(m+sn)}{\mathcal{N}_R(l,m,n)},\right.\\ &\qquad \frac{s^2m^3 + 2m(l^2+s^2ln+s^4n^2) + sln(l+s^2n)}{\mathcal{N}_R(l,m,n)},\\ &\qquad \left.\frac{s^5n^3 + 2sn(l^2+slm+s^2m^2) + lm(l+sm)}{s\mathcal{N}_R(l,m,n)}\right),\end{aligned}$$

where it is interesting to notice that the sum of the numerators is $(l + sm + s^2n)^3$. The inverse group homomorphism is the map

$$\begin{aligned}&(\psi''_R)^{-1} : \left(\mathcal{C}_R, \odot_R\right) \xrightarrow{\sim} \left(\mathbb{P}_R, \odot_R\right),\\ &(x, y, z) \longmapsto \left[s^2(1+2x-sy-s^2z) : s(1-x+2sy-s^2z) : 1-x-sy+2s^2z\right].\end{aligned}$$

This group isomorphism allows to find all the solutions of the cubic Pell equation: it is sufficient to evaluate ψ''_R over all the elements of $\mathbb{P}_R$ described explicitly in Equation 3.4. In addition, differently from the previous case, the explicit inverse of the group isomorphism can be used to describe each point of the Pell cubic with two thirds of the size with respect to the classical notation for the points in $\mathbb{F}_q^3$.

Example 3.2. When $q = 13$ and $R = 5$, which is the cube of $\{7, 8, 11\}$ in $\mathbb{F}_{13}$, the previous results assures that there are $(q-1)^2 = 144$ solutions of the cubic Pell equation

$$x^3 + 5y^3 - z^3 - 2xyz \equiv 1 \pmod{13}.$$

In this case, ψ''_5 allows to find all of them as

$$\begin{gathered}\psi''_5([l : m : 1]), \quad \forall\, m \in \mathbb{F}_{13},\ l \in \mathbb{F}_{13} \setminus \{-7m+3, -8m+1, -11m+9\},\\ \psi''_5([l : 1 : 0]), \quad \forall\, l \in \mathbb{F}_{13} \setminus \{-7, -8, -11\},\\ \psi''_5([1 : 0 : 0]) = (1, 0, 0).\end{gathered}$$

For instance, a random solution of the cubic Pell equation can be found by taking randomly $m \in \mathbb{F}_{13}$, e.g., $m = 3$, and another element $l \in \mathbb{F}_{13} \setminus \{8, 3, 2\}$, e.g., $l = 9$, and evaluating

$$\psi''_5([9 : 3 : 1]) = (3, 4, 3).$$

It is easy to check that

$$3^3 + 5 \cdot 4^3 - 3^3 - 2 \cdot 3 \cdot 4 \cdot 3 \equiv 1 \pmod{13}.$$

Similarly, when taking $l = 4 \notin \{6, 5, 2\}$ and $[4 : 1 : 0] \in \mathbb{P}_5$, so that

$$\psi''_5([4 : 1 : 0]) = (10, 4, 9),$$

is another solution of the cubic Pell equation.

3.2.3 r *cube with one root in* $\mathbb{F}_q$

If $q \not\equiv 1 \pmod 3$, then $\mathbb{F}_q$ does not contain any non–trivial cubic root of unity. In addition, each $r \in \mathbb{F}_q^\times$ is a cube and has only one cubic root s in $\mathbb{F}_q$.

In this case, Equation 3.5 holds and the projectivization $\mathbb{P}_r$ has

$$\#\mathbb{P}_r = q^2 + q + 1 - (q+2) = q^2 - 1,$$

unless there is a $m \in \mathbb{F}_q$ such that

$$[-s(m+s) : m : 1] = [s^2 : s : 1] \Leftrightarrow 3s^2 = 0,$$

which is satisfied only when $q = 3^k$, in which case $\#\mathbb{P}_r = q^2$. The result for generic q is also confirmed by the following statement, obtained analogously to Theorem 2.5.

THEOREM 3.5 If $q \equiv 2 \pmod 3$ and $r \in \mathbb{F}_q^\times$, then $\left(\mathbb{P}_r, \odot_r\right)$ is a cyclic group of order $q^2 - 1$.

Proof. Given s cubic root of r in $\mathbb{F}_q$, $t^3 - r$ is reducible over $\mathbb{F}_q$ as

$$t^3 - r = (t-s)(t^2 + st + s^2),$$

so that, using the Chinese remainder theorem, there is the ring isomorphism

$$\begin{aligned} \mathcal{R}_r = \mathbb{F}_q[t]/\langle t^3 - r\rangle &\xrightarrow{\sim} \mathbb{F}_q[t]/\langle t-s\rangle \times \mathbb{F}_q[t]/\langle t^2+st+s^2\rangle, \\ x + yt + zt^2 &\longmapsto \left(x + sy + s^2 z, x - s^2 z + (y - sz)t\right). \end{aligned}$$

In addition, $\mathbb{F}_q[t]/\langle t-s\rangle \cong \mathbb{F}_q$ and $\mathbb{F}_q[t]/\langle t^2+st+s^2\rangle \cong \mathbb{F}_{q^2}$, and when passing to the quotients there is

$$\begin{aligned} \left(\mathbb{P}_r, \odot_r\right) = \mathcal{R}_r^{\odot_r}/\mathbb{F}_q^\times &\xrightarrow{\sim} (\mathbb{F}_q^\times \times \mathbb{F}_{q^2}^\times)/\mathbb{F}_q^\times \cong \mathbb{F}_{q^2}^\times, \\ [l : m : n] &\longmapsto \left(\frac{l - s^2 n}{l + sm + s^2 n}, \frac{m - sn}{l + sm + s^2 n}\right), \\ [s^2(1 - sv + 2u) : s(1 + 2sv - u) : 1 - sv - u] &\longleftarrow\!\shortmid (u, v), \end{aligned}$$

which is a group isomorphism. □

The relation with the Pell cubic when $q = p^k$ and $p \neq 3$ is given by the following result.

THEOREM 3.6 If $q \equiv 2 \pmod 3$ and $r \in \mathbb{F}_q^\times$, then the following map is a group isomorphism between $\mathbb{P}_r$ and $\mathcal{C}_r$:

$$\begin{aligned} \psi_r''' : \left(\mathbb{P}_r, \odot_r\right) &\xrightarrow{\sim} \left(\mathcal{C}_r, \odot_r\right), \\ [l : m : n] &\longmapsto \mathcal{N}_r(l, m, n)^{\lfloor q/3 \rfloor}(l, m, n). \end{aligned}$$

Proof. In order for ψ_R''' to be a group isomorphism, it must be

- well defined: for any $[l : m : n] \in \mathbb{P}_R$, $\lambda \in \mathbb{F}_q^\times$,
$$\begin{aligned}\psi_R'''([\lambda l : \lambda m : \lambda n]) &= \left(\lambda^3 \mathcal{N}_R(l,m,n)\right)^{(q-2)/3} \lambda(l,m,n) \\ &= \lambda^{q-1} \psi_R'''([l : m : n]) = \psi_R'''([l : m : n]),\end{aligned}$$
and $\psi_R'''(\mathbb{P}_R) \subseteq \mathcal{C}_R$ because, for any $[l : m : n] \in \mathbb{P}_R$,
$$\begin{aligned}\mathcal{N}_R\left(\psi_R'''([l : m : n])\right) &= \mathcal{N}_R(l,m,n)^{q-2}\, \mathcal{N}_R(l,m,n) \\ &= \mathcal{N}_R(l,m,n)^{q-1} = 1;\end{aligned}$$
- a group homomorphism: given $[l_1 : m_1 : n_1], [l_2 : m_2 : n_2] \in \mathbb{P}_R$, by denoting their product as $[l : m : n]$, the resulting image is
$$\begin{aligned}\psi_R'''([l : m : n]) &= \mathcal{N}_R(l,m,n)^{\lfloor q/3 \rfloor}(l,m,n) \\ &= \mathcal{N}_R(l_1,m_1,n_1)^{\lfloor q/3 \rfloor} \mathcal{N}_R(l_2,m_2,n_2)^{\lfloor q/3 \rfloor} \\ &\quad (l_1,m_1,n_1) \odot_R (l_2,m_2,n_2) \\ &= \psi_R'''(l_1,m_1,n_1) \odot_R \psi_R'''(l_2,m_2,n_2);\end{aligned}$$
- injective: for any $[l : m : n] \in \mathbb{P}_R$, $\mathcal{N}_R(l,m,n) \neq 0$ and
$$\begin{aligned}\psi_R'''([l : m : n]) = (1,0,0) &\Leftrightarrow \begin{cases} \mathcal{N}_R(l,m,n)^{\lfloor q/3 \rfloor} l = 1, \\ \mathcal{N}_R(l,m,n)^{\lfloor q/3 \rfloor} m = 0, \\ \mathcal{N}_R(l,m,n)^{\lfloor q/3 \rfloor} n = 0 \end{cases} \\ &\Leftrightarrow \begin{cases} (l^3)^{(q-2)/3} l = 1, \\ m = 0, \\ n = 0 \end{cases} \\ &\Leftrightarrow [l : m : n] = [1 : 0 : 0];\end{aligned}$$
- surjective:
  - $x^3 = 1$ admits only the solution $x = 1$, so that $(1,0,0)$ is the only point of $\mathcal{C}_R$ with $y = z = 0$, as well as $[1 : 0 : 0]$ in $\mathbb{P}_R$;
  - if $z = 0$ but $y \neq 0$, then the preimage of $(x,y,0)$ is of the form $[l : 1 : 0]$ and
$$\begin{cases} x = (l^3 + R)^{\lfloor q/3 \rfloor} l, \\ y = (l^3 + R)^{\lfloor q/3 \rfloor} \end{cases} \Rightarrow l = \frac{x}{y};$$
  - if $z \neq 0$ then
$$\begin{cases} x = \mathcal{N}_R(l,m,1)^{\lfloor q/3 \rfloor} l, \\ y = \mathcal{N}_R(l,m,1)^{\lfloor q/3 \rfloor} m, \\ z = \mathcal{N}_R(l,m,1)^{\lfloor q/3 \rfloor} \end{cases} \Rightarrow \begin{cases} l = x/z, \\ m = y/z. \end{cases}$$

In conclusion, ψ_R''' is a group isomorphism. □

This proof gives also as inverse of ψ_R''' the classical projectivization

$$\begin{aligned}(\psi_R''')^{-1} : \left(\mathcal{C}_R, \odot_R\right) &\xrightarrow{\sim} \left(\mathbb{P}_R, \odot_R\right),\\ (1,0,0) &\longmapsto [1:0:0],\\ (x,y,0) &\longmapsto [x/y:1:0],\\ (x,y,z) &\longmapsto [x/z:y/z:1].\end{aligned}$$

Thanks to the group isomorphism ψ_R''', the properties of $\left(\mathbb{P}_R, \odot_R\right)$ are inherited by $\left(\mathcal{C}_R, \odot_R\right)$, i.e, it is cyclic with q^2-1 elements. In addition, it is possible to find all the solutions of the cubic Pell equation by simply evaluating ψ_R''' over all the elements of $\mathbb{P}_R$, which are described explicitly in Equation 3.5. As in the previous case, the explicit inverse can be used to describe each point of the Pell cubic with two thirds of the size of points in $\mathbb{F}_q^3$.

Example 3.3. Considering $q = 11$ and $R = 9$, which is the cube of 4 in $\mathbb{F}_{11}$, the previous results assures that the cubic Pell equation

$$x^3 + 9y^3 + 4z^3 + 6xyz \equiv 1 \pmod{11},$$

admits $q^2 - 1 = 120$ solutions and ψ_9''' allows to find all of them as

$$\begin{gathered}\psi_9'''([l:m:1]), \quad \forall\, m \in \mathbb{F}_{11},\, l \in \mathbb{F}_{11} \setminus \{-4m+5\},\, (l,m) \neq (5,4),\\ \psi_9'''([l:1:0]), \quad \forall\, l \in \mathbb{F}_{11} \setminus \{-4\},\\ \psi_9'''([1:0:0]) = (1,0,0).\end{gathered}$$

For instance, a random solution of the cubic Pell equation can be found by taking randomly $m \in \mathbb{F}_{11}$, e.g., $m = 2$, and another element $l \in \mathbb{F}_{11} \setminus \{8\}$, e.g., $l = 7$, and evaluating

$$\psi_9'''([7:2:1]) = (9,1,6).$$

It is easy to check that

$$9^3 + 9 \cdot 1^3 + 4 \cdot 6^3 + 6 \cdot 9 \cdot 1 \cdot 6 \equiv 1 \pmod{11}.$$

Similarly, when taking $l = 3 \neq 7$ and $[3:1:0] \in \mathbb{P}_9$,

$$\psi_9'''([3:1:0]) = (4,5,0),$$

is another solution of the cubic Pell equation.

When $q = 3^k$, the Pell cubic is no more a cyclic group and the group isomorphism has a different form with respect to ψ_R'''.

THEOREM 3.7 If $q = 3^k$ and $R \in \mathbb{F}_q^\times$, then the following map is a group isomorphism between $\mathbb{P}_R$ and $\mathcal{C}_R$

$$\begin{aligned}\tilde{\psi}_R''' : \left(\mathbb{P}_R, \odot_R\right) &\xrightarrow{\sim} \left(\mathcal{C}_R, \odot_R\right),\\ [l:m:n] &\longmapsto \mathcal{N}_R(l,m,n)^{q/3-1}(l,m,n)^{\odot_R 2}.\end{aligned}$$

Proof. In order for $\tilde{\psi}_R'''$ to be a group isomorphism, it must be

- well defined: for any $[l : m : n] \in \mathbb{P}_R$, $\lambda \in \mathbb{F}_q^\times$,

$$\begin{aligned}\tilde{\psi}_R'''([\lambda l : \lambda m : \lambda n]) &= \left(\lambda^3 \mathcal{N}_R(l,m,n)\right)^{q/3-1} \lambda^2 (l,m,n)^{\odot_R 2} \\ &= \lambda^{q-1} \tilde{\psi}_R'''([l : m : n]) = \tilde{\psi}_R'''([l : m : n]),\end{aligned}$$

  and $\tilde{\psi}_R'''(\mathbb{P}_R) \subseteq \mathcal{C}_R$ because, for any $[l : m : n] \in \mathbb{P}_R$,

$$\begin{aligned}\mathcal{N}_R\left(\tilde{\psi}_R'''([l : m : n])\right) &= \mathcal{N}_R(l,m,n)^{q-3} \mathcal{N}_R(l,m,n)^2 \\ &= \mathcal{N}_R(l,m,n)^{q-1} = 1;\end{aligned}$$

- a group homomorphism: given $[l_1 : m_1 : n_1], [l_2 : m_2 : n_2] \in \mathbb{P}_R$, by denoting their product as $[l : m : n]$, the resulting image is

$$\begin{aligned}\tilde{\psi}_R'''([l : m : n]) &= \mathcal{N}_R(l,m,n)^{q/3-1} (l,m,n)^{\odot_R 2} \\ &= \mathcal{N}_R(l_1,m_1,n_1)^{q/3-1} \mathcal{N}_R(l_2,m_2,n_2)^{q/3-1} \\ &\quad (l_1,m_1,n_1)^{\odot_R 2} \odot_R (l_2,m_2,n_2)^{\odot_R 2} \\ &= \tilde{\psi}_R'''(l_1,m_1,n_1) \odot_R \tilde{\psi}_R'''(l_2,m_2,n_2);\end{aligned}$$

- injective: for any $[l : m : n] \in \mathbb{P}_R$, $\tilde{\psi}_R'''([l : m : n]) = (1,0,0)$ means

$$\begin{cases} \mathcal{N}_R(l,m,n)^{q/3-1}(l^2 + 2Rmn) = 1, \\ \mathcal{N}_R(l,m,n)^{q/3-1}(Rn^2 + 2lm) = 0, \\ \mathcal{N}_R(l,m,n)^{q/3-1}(m^2 + 2ln) = 0, \end{cases}$$

  with $\mathcal{N}_R(l,m,n) \neq 0$, so that:

  - if $m, n \neq 0$, then taking the second and third equation gives

$$\begin{cases} n(Rn^2 + 2lm) = 0, \\ m(m^2 + 2ln) = 0 \end{cases} \Leftrightarrow Rn^3 - m^3 = 0.$$

    The system is satisfied only when $m = sn$ with $s \in \mathbb{F}_q^\times$ cubic root of R, which implies $[l : m : n] = [l/n : s : 1]$ that, if substituted in the third equation, gives

$$s^2 + 2l/n = 0 \Leftrightarrow l/n = s^2.$$

    However, this is a contradiction since $[s^2 : s : 1] \notin \mathbb{P}_R$;
  - if $m \neq n = 0$, then the third equation gives $m^2 = 0$;
  - if $n \neq m = 0$, then the second equation gives $Rn^2 = 0$;
  - $m = n = 0$ is the only remaining option, which implies that $\ker(\tilde{\psi}_R''') = \{[1 : 0 : 0]\}$;

- surjective: it is important to notice that for each $(x, y, z) \in \mathcal{C}_R$

$$1 = x^3 + Ry^3 + R^2z^3 = (x + sy + s^2z)^3,$$

with $s \in \mathbb{F}_q^\times$ cubic root of R. This implies that each point of the Pell cubic can be identified by the pair $(y, z) \in \mathbb{F}_q^2$, since the first coordinate can be evaluated as $x = 1 - sy - s^2z$. Thus, $\mathcal{C}_R$ has q^2 elements and $\tilde{\psi}_R'''$ is an injection between two finite groups with same cardinality, which means that it is also surjective.

In conclusion, $\tilde{\psi}_R'''$ is a group isomorphism. □

3.3 GENERALIZED PELL CUBIC

In this section, as for the quadratic case in Section 2.3, a generalization of the cubic Pell equation and the resulting generalized Pell cubic are introduced. In addition, an explicit group isomorphism between the standard Pell cubic from Section 3.1 and a generalized Pell cubic is obtained.

DEFINITION 3.3 Given the parameters $R, Q \in \mathbb{F}^\times$, the solutions of a generalized cubic Pell equation $x^3 + Ry^3 + R^2z^3 - 3Rxyz = Q$ correspond to the elements of $\mathcal{R}_R$ with norm $\mathcal{N}_R$ equal to Q, which are the points of the *generalized Pell cubic with parameter* R *and norm* Q:

$$\mathcal{C}_{R,Q} = \{(x, y, z) \in \mathbb{F}^3 \,|\, x^3 + Ry^3 + R^2z^3 - 3Rxyz = Q\}.$$

The product $\odot_R$ does not give a group structure on $\mathcal{C}_{R,Q}$, but it can be exploited to define the *generalized cubic Brahmagupta product with identity* $(a, b, c) \in \mathcal{C}_{R,Q}$ as

$$\begin{aligned}&(x_1, y_1, z_1) \odot_{R,Q,a,b,c} (x_2, y_2, z_2)\\ &= \frac{1}{Q}(a^2 - Rbc, Rc^2 - ab, b^2 - ac) \odot_R (x_1, y_1, z_1) \odot_R (x_2, y_2, z_2).\end{aligned}$$

In the following, the product of the conjugates of an element is denoted by $\overline{(a, b, c)}$. Clearly, the identity point for $\odot_{R,Q,a,b,c}$ is the chosen $(a, b, c) \in \mathcal{C}_{R,Q}$, the inverse of a point $(x, y, z) \in \mathcal{C}_{R,Q}$ is the point

$$\frac{1}{Q}(a, b, c) \odot_R (a, b, c) \odot_R \overline{(x, y, z)},$$

and $\left(\mathcal{C}_{R,Q}, \odot_{R,Q,a,b,c}\right)$ is a commutative group.

When $Q = 1$ and the chosen identity point is $(a, b, c) = (1, 0, 0)$, the product $\odot_{R,Q,a,b,c}$ coincides with the classical $\odot_R$.

Despite the introduction of the new parameter $Q \in \mathbb{F}^\times$, it is possible to obtain the explicit group isomorphism between two generalized Pell cubics with same Q by exploiting the definition of $\odot_{R,Q,a,b,c}$.

THEOREM 3.8 Given $r, q \in \mathbb{F}^\times$ and a point $(a,b,c) \in \mathcal{C}_{r,q}$, the following map is a group isomorphism between $\mathcal{C}_r$ and $\mathcal{C}_{r,q}$:

$$\begin{aligned} \upsilon_{r,q}^{a,b,c} : \big(\mathcal{C}_r, \odot_r\big) &\xrightarrow{\sim} \big(\mathcal{C}_{r,q}, \odot_{r,q,a,b,c}\big), \\ (x,y,z) &\longmapsto (a,b,c) \odot_r (x,y,z). \end{aligned}$$

Proof. In order for $\upsilon_{r,q}^{a,b,c}$ to be a group isomorphism, it must be

- well defined: $\upsilon_{r,q}^{a,b,c}(\mathcal{C}_r) \subseteq \mathcal{C}_{r,q}$ since, for any $(x,y,z) \in \mathcal{C}_r$,

$$\mathcal{N}_r\big((a,b,c) \odot_r (x,y,z)\big) = \mathcal{N}_r(a,b,c)\,\mathcal{N}_r(x,y,z) = q;$$

- a group homomorphism: for any $(x_1,y_1,z_1), (x_2,y_2,z_2) \in \mathcal{C}_r$,

$$\begin{aligned} &\upsilon_{r,q}^{a,b,c}\big((x_1,y_1,z_1) \odot_r (x_2,y_2,z_2)\big) \\ &= (a,b,c) \odot_D (x_1,y_1,z_1) \odot_r (x_2,y_2,z_2) \\ &= \frac{(a,b,c) \odot_r \overline{(a,b,c)}}{q} \odot_r (a,b,c) \odot_r (x_1,y_1,z_1) \odot_r (x_2,y_2,z_2) \\ &= \upsilon_{r,q}^{a,b,c}(x_1,y_1,z_1) \odot_{r,q,a,b,c} \upsilon_{r,q}^{a,b,c}(x_2,y_2,z_2); \end{aligned}$$

- injective: for any $(x,y,z) \in \mathcal{C}_r$,

$$\upsilon_{r,q}^{a,b,c}(x,y,z) = (a,b,c) \Leftrightarrow (x,y,z) = (1,0,0);$$

- surjective: for any $(x,y,z) \in \mathcal{C}_{r,q}$,

$$\begin{aligned} (x,y,z) &= \frac{(a,b,c) \odot_r \overline{(a,b,c)}}{q} \odot_r (x,y,z) \\ &= (a,b,c) \odot_r (1,0,0) \odot_{r,q,a,b,c} (x,y,z) \\ &= \upsilon_{r,q}^{a,b,c}\big((1,0,0) \odot_{r,q,a,b,c} (x,y,z)\big), \end{aligned}$$

  where $(1,0,0) \odot_{r,q,a,b,c} (x,y,z) \in \mathcal{C}_r$ because

$$\mathcal{N}_r\big((1,0,0) \odot_{r,q,a,b,c} (x,y,z)\big) = 1.$$

Thus, $\upsilon_{r,q}^{a,b,c}$ is a group isomorphism and

$$\begin{aligned} (\upsilon_{r,q}^{a,b,c})^{-1} : \big(\mathcal{C}_{r,q}, \odot_{r,q,a,b,c}\big) &\xrightarrow{\sim} \big(\mathcal{C}_r, \odot_r\big), \\ (x,y,z) &\longmapsto (1,0,0) \odot_{r,q,a,b,c} (x,y,z), \end{aligned}$$

is the inverse group homomorphism. □

By composing $(\upsilon_{r,q}^{a,b,c})^{-1}$ with $\upsilon_{r,q'}^{a',b',c'}$, it is possible to obtain an explicit group isomorphism between generalized Pell cubics with same parameter r:

$$\begin{aligned} \big(\mathcal{C}_{r,q}, \odot_{r,q,a,b,c}\big) &\xrightarrow{\sim} \big(\mathcal{C}_{r,q'}, \odot_{r,q',a',b',c'}\big), \\ (x,y,z) &\longmapsto (a',b',c') \odot_{r,q,a,b,c} (x,y,z). \end{aligned} \tag{3.7}$$

Since the group $(\mathbb{P}_R, \odot_R)$ introduced in Definition 3.2 is independent of the choice of the parameter $Q \in \mathbb{F}^\times$ and of the identity point $(a, b, c) \in \mathcal{C}_{R,Q}$, all the results in Section 3.2 can be adapted to generalized Pell cubics. In particular, taking ψ_R as one of the group isomorphisms ψ'_R, ψ''_R, ψ'''_R or $\tilde{\psi}'''_R$ (depending on the parameter R and the value of $q \pmod 3$) and composing it with $\upsilon_{R,Q}^{a,b,c}$ results in a group isomorphism between $\mathbb{P}_R$ and $\mathcal{C}_{R,Q}$:

$$\begin{aligned}\psi_{R,Q}^{a,b,c} : (\mathbb{P}_R, \odot_R) &\xrightarrow{\sim} (\mathcal{C}_{R,Q}, \odot_{R,Q,a,b,c}) \\ [l : m : n] &\longmapsto (a, b, c) \odot_R \psi_R([l : m : n]).\end{aligned}$$

The inverse group homomorphism is

$$\begin{aligned}(\psi_{R,Q}^{a,b,c})^{-1} : (\mathcal{C}_{R,Q}, \odot_{R,Q,a,b,c}) &\xrightarrow{\sim} (\mathbb{P}_R, \odot_R), \\ (x, y, z) &\longmapsto \psi_R^{-1}\big((1, 0, 0) \odot_{R,Q,a,b,c} (x, y, z)\big).\end{aligned}$$

When the inverse of ψ_R is explicitly known, like for ψ''_R and ψ'''_R, this parametrization and its inverse can be used as an alternative way to obtain the group isomorphism in Equation 3.7.

All the results in Section 3.2 can be adapted for generalized Pell cubics in the case of finite fields.

In addition, if $R \neq R'$ are both cubes in $\mathbb{F}_q$, then there is $\rho \in \mathbb{F}_q^\times$ such that $R = \rho^3 R'$, so that the following map is a group isomorphism between $\mathcal{C}_R$ and $\mathcal{C}_{R'}$:

$$\begin{aligned}\rho_{R,R'} : (\mathcal{C}_R, \odot_R) &\xrightarrow{\sim} (\mathcal{C}_{R'}, \odot_{R'}), \\ (x, y, z) &\longmapsto (x, \rho y, \rho^2 z).\end{aligned}$$

This can be also a group isomorphism between projectivizations

$$\begin{aligned}\rho_{R,R'} : (\mathbb{P}_R, \odot_R) &\xrightarrow{\sim} (\mathbb{P}_{R'}, \odot_{R'}), \\ [l : m : 1] &\longmapsto [l/\rho^2 : m/\rho : 1], \\ [l : 1 : 0] &\longmapsto [l/\rho : 1 : 0].\end{aligned}$$

Thus, the composition of $(\upsilon_{R,Q}^{a,b,c})^{-1}$, $\rho_{R,R'}$ and $\upsilon_{R',Q'}^{a',b',c'}$ results in an explicit group isomorphism between two generalized Pell cubics $(\mathcal{C}_{R,Q}, \odot_{R,Q,a,b,c})$ and $(\mathcal{C}_{R',Q'}, \odot_{R',Q',a',b',c'})$.

3.4 EXPONENTIATION AND EXTENDED RÉDEI POLYNOMIALS

In this section, the results obtained in Section 2.4 for the quadratic case are adapted for the exponentiation with respect to $\odot_R$. After considering possible implementations of the square–multiply algorithm, an alternative approach is obtained exploiting an extension of Rédei polynomials and rational functions.

```
BrahSquaMult(R, x, y, z, k):
1.  a, b, c = 1, 0, 0
2.  kbin = binary(k)
3.  for bit in kbin:
4.      a, b, c = a^2 + 2Rbc, Rc^2 + 2ab, b^2 + 2ac
5.      if bit == 1:
6.          a, b, c = ax + R(bz + cy), ay + bx + Rcz, az + by + cx
7.  return a, b, c
```

Figure 4: square–multiply algorithm with $\odot_R$ on $\mathcal{C}_R$.

As can be easily observed in Equation 3.1, the operation $\odot_R$ on $\mathcal{C}_R$ requires 11 products and 6 additions in $\mathbb{F}$, while the version with the canonical representatives in $\mathbb{P}_R$ introduced in Equation 3.2 requires at most 1 inversion, 5 products and 6 additions in $\mathbb{F}$. However, the inversion is largely more expensive than the additional 6 products required in Equation 3.1. Therefore, in a comparison of square–multiply implementations, the first one is the most efficient. Figure 4 describes the algorithm `BrahSquaMult` for the square–multiply exponentiation over $\mathcal{C}_R$.

In the following the Rédei polynomials and rational functions are extended to the cubic case in order to let them exploitable for the exponentiation over $\mathcal{C}_R$ and $\mathbb{P}_R$, respectively.

DEFINITION 3.4 Given a parameter $R \in \mathbb{F}^\times$ and $t^3 = R$ not necessarily in $\mathbb{F}$, the *extended Rédei polynomials* $(A_k)_{k\geqslant 0}$, $(B_k)_{k\geqslant 0}$ and $(C_k)_{k\geqslant 0}$ result from

$$\begin{aligned}&(l + mt + t^2)^k\\ &= A_k(R, l, m) + B_k(R, l, m)t + C_k(R, l, m)t^2, \quad \text{for } k \geqslant 0.\end{aligned}$$

These three sequences clearly correspond to the coordinates resulting from $[l : m : 1]^{\odot_R k} \in \mathbb{P}_R$.

The *extended Rédei rational functions* are obtained for $k > 0$ as

$$\begin{aligned}P_k(R, l, m) &= \frac{A_k(R, l, m)}{C_k(R, l, m)},\\ Q_k(R, l, m) &= \frac{B_k(R, l, m)}{C_k(R, l, m)}.\end{aligned}$$

They are the canonical representatives of $(l, m)^{\odot_R k} \in \mathbb{P}_R$.

On the other hand, as for the quadratic case, considering the coordinates of the point $(x, y, z)^{\odot_R k} \in \mathcal{C}_R$, the *generalized extended Rédei polynomials* can be defined as

$$\begin{aligned}&(x + yt + zt^2)^k\\ &= a_k(R, x, y, z) + b_k(R, x, y, z)t + c_k(R, x, y, z)t^2, \quad \text{for } k \geqslant 0.\end{aligned}$$

In particular, there is an explicit form for the extended Rédei polynomials given by

$$A_k(R,l,m) = \sum_{\substack{h+i+j=k,\\ i+2j\equiv 0 \pmod 3}} \binom{k}{h,i,j} R^{(i+2j)/3} l^h m^i,$$

$$B_k(R,l,m) = \sum_{\substack{h+i+j=k,\\ i+2j\equiv 1 \pmod 3}} \binom{k}{h,i,j} R^{(i+2j-1)/3} l^h m^i,$$

$$C_k(R,l,m) = \sum_{\substack{h+i+j=k,\\ i+2j\equiv 2 \pmod 3}} \binom{k}{h,i,j} R^{(i+2j-2)/3} l^h m^i,$$

while the generalized extended Rédei polynomials are

$$a_k(R,x,y,z) = \sum_{\substack{h+i+j=k,\\ i+2j\equiv 0 \pmod 3}} \binom{k}{h,i,j} R^{(i+2j)/3} x^h y^i z^j,$$

$$b_k(R,x,y,z) = \sum_{\substack{h+i+j=k,\\ i+2j\equiv 1 \pmod 3}} \binom{k}{h,i,j} R^{(i+2j-1)/3} x^h y^i z^j,$$

$$c_k(R,x,y,z) = \sum_{\substack{h+i+j=k,\\ i+2j\equiv 2 \pmod 3}} \binom{k}{h,i,j} R^{(i+2j-2)/3} x^h y^i z^j,$$

where the multinomial coefficients are $\binom{k}{h,i,j} = \frac{k!}{h!\,i!\,j!}$.

It also possible to use a matrix notation, so that

$$\begin{pmatrix} A_k(R,l,m) \\ B_k(R,l,m) \\ C_k(R,l,m) \end{pmatrix} = \begin{pmatrix} l & R & Rm \\ m & l & R \\ 1 & m & l \end{pmatrix}^k \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} a_k(R,x,y,z) \\ b_k(R,x,y,z) \\ c_k(R,x,y,z) \end{pmatrix} = \begin{pmatrix} x & Rz & Ry \\ y & x & Rz \\ z & y & x \end{pmatrix}^k \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

The modified More algorithm for the quadratic case described in Figure 3 can be adapted for the evaluation of the extended Rédei rational functions P_k, Q_k i.e., for evaluating the exponentiation of a canonical representative $(l,m)^{\odot_R k} \in \mathbb{P}_R$. The obtained algorithm is detailed in Figure 5 under the denomination `ModMore` and has complexity $O(\log k)$ considering additions and multiplications over $\mathbb{F}$. In a comparison with Figure 4, the two algorithms have the same number of operations at each step except for step 6, where `BrahSquaMult` requires 3 additional products, and step 7, where `ModMoore` requires two final inversions.

```
ModMore(R, l, m, k):
1.  A, B, C = 1, 0, 0
2.  kbin = binary(k)
3.  for bit in kbin:
4.      A, B, C = A^2 + 2RBC, RC^2 + 2AB, B^2 + 2AC
5.      if bit == 1:
6.          A, B, C = Al + R(B + Cm), Am + Bl + RC, A + Bm + Cl
7.  return A/C, B/C
```

Figure 5: Modified More algorithm for the exponentiation over $\mathbb{P}_R$ using extended Rédei rational functions.

Thus, from the point of view of performance, the two algorithms are comparable. The main advantage in evaluating the exponentiation of a canonical representative in $\mathbb{P}_R$ through the modified More algorithm is that the size of the data is two thirds of that of the points on the Pell cubic $\mathcal{C}_R$.

Part II

HOW IS THE PELL EQUATION USED IN CRYPTOGRAPHY?

Now that the theory and formalism of the Pell equation have been introduced, their applications to cryptography are discussed. After introducing in Chapter 4 the basic concepts of cryptography and the state of the art in the use of the Pell equation in cryptosystems, the quadratic Pell equation is exploited to obtain new cryptosystems in Chapter 5. Analogously, other cryptosystems based on the cubic Pell equation are introduced in Chapter 6.

4

PUBLIC–KEY CRYPTOGRAPHY

After introducing all the required definitions and results concerning the quadratic and cubic Pell equation, this chapter focuses on their use in cryptography. In particular, Section 4.1 introduces the basic concepts behind classical and modern cryptography. Then two of the mainly used cryptosystems are addressed in order to adapt them for exploiting Pell equations. In Section 4.2, the classical RSA cryptosystem is described together with some RSA–like schemes from the state of the art that can be considered as initial points for the following work. In the same way, Section 4.4 and Section 4.5 focus on the ElGamal cryptosystem and on the DSA, respectively, since they can be formulated also using Pell conics and cubics, as will be pointed out in the following chapters.

4.1 CLASSICAL AND MODERN CRYPTOGRAPHY

Cryptography is one of the two branches under the discipline *cryptology*. Its main purpose is to study the ways in which two or more parties can talk to each other in an insecure environment with the requirement that nobody else can understand what is being said. Its counterpart is the *cryptanalysis*, which focuses on breaking the systems developed by cryptographers.

From a mathematical point of view, cryptography models the systems used to hide information in a message, called *cryptosystem*, using the following sets:

- $\mathcal{P}$ containing the comprehensible messages, called *plaintexts*;
- $\mathcal{C}$ containing the unintelligible messages, called *ciphertexts*;
- $\mathcal{K}$ containing the *keys* for obtaining ciphertexts from plaintexts and vice versa, they are usually generated by $\mathtt{Gen} : \mathbb{N} \to \mathcal{K}$;
- $\mathcal{E} = \{\mathtt{Enc}_k : \mathcal{P} \to \mathcal{C},\ k \in \mathcal{K}\}$ set of the *encryption functions*;
- $\mathcal{D} = \{\mathtt{Dec}_k : \mathcal{C} \to \mathcal{P},\ k \in \mathcal{K}\}$ set of the *decryption functions*.

Classical cryptography is only based on the secrecy of the used key that, if known, allows to obtain the plaintext from the ciphertext. This is the principle behind the modern *symmetric* or *private-key* cryptography, so-called because the encryption key is the same used for the decryption, i.e., given $k \in \mathcal{K}$, for each $p \in \mathcal{P}$,

$$\mathtt{Dec}_k(\mathtt{Enc}_k(p)) = p.$$

The main advantages of this kind of cryptosystems are the low computational costs and the easy way to define the security level, which is simply given by the length of the key. However, they require to share secretly the key using a secure channel and, if the information exchange is among n different parties, then the required keys are

$$\binom{n}{2} = \frac{n(n-1)}{2} \approx \frac{n^2}{2}.$$

They are useful for sharing large amount of data among few users.

After the revolution of computers, the number of users dramatically increased while the secure channels to share the secret keys became hardly available. Those are the main reasons behind the introduction in the mid-1970s of the *asymmetric* or *public–key* cryptography, in which the key for the decryption is secret and different from the one used in the encryption that is public. Thus, given $pk \in \mathcal{K}$ there is $sk \in \mathcal{K}$ such that, for each $p \in \mathcal{P}$,

$$\mathtt{Dec}_{sk}(\mathtt{Enc}_{pk}(p)) = p.$$

This idea was firstly introduced in 1976, when Diffie and Hellman published an history changing article titled *New directions in cryptography* [23], in which they introduced a method for exchanging through a public channel a secret key that then can be used in a symmetric cryptosystem. Since then, actual asymmetric cryptosystems with different purposes were developed. In particular, the main uses addressed are:

- *secrecy*: the information in message can be sent secretly if encrypted using the public key of the receiver, so that it can be retrieved using the secret key in its possession. The resulting cryptosystems are called *Public-Key Encryption* (PKE) *schemes*;
- *data integrity*, *authentication* and *non–repudiation*: it is important to guarantee that a message has not been modified while in transit, as well as assure the identity of the sender that must not be able to repudiate it. All these objectives can be achieved by signing the message using the secret key, so that anyone can use the related public key to check its integrity and the identity on the signer. The resulting cryptosystems are called *digital signatures*.

Public–key cryptography resolves the problems of symmetric cryptography, since the secure channel is no more needed and the number of key required in a conversation among n users is simply $2n$ (secret and public key for each one). However, computational costs are higher than in the symmetric case and security is lower since it relies on mathematical problems that are thought to be difficult, but not impossible, to solve. Therefore, public–key cryptosystems are useful when small amounts of data need to be encrypted and sent to many users via an insecure channel.

Gen(n):	Enc(m, pk):
1. $p, q \leftarrow_\$ \{0,1\}^{n/2}$ primes	REQUIRE: $m < N$
2. $N = pq$	1. $c = m^e \pmod N$
3. $\varphi(N) = (p-1)(q-1)$	2. return c
4. $e \leftarrow_\$ \mathbb{Z}^\times_{\varphi(N)}$	Dec(c, pk, sk):
5. $d = e^{-1} \pmod{\varphi(N)}$	1. $m = c^d \pmod N$
6. $sk = (p, q, d)$	2. return m
7. $pk = (N, e)$	
8. return pk, sk	

Figure 6: RSA PKE scheme.

4.2 RSA CRYPTOSYSTEM

This section focuses on one of the first and still largely used cryptosystems. This is a PKE scheme introduced in 1977 by Rivest, Shamir and Adleman (RSA) [52]. In particular, RSA is detailed in Figure 6.

In the key generation algorithm (Gen), given the bit–length n of the modulus, two primes p, q of $n/2$ bits are generated and used to obtain $N = pq$ and its Euler totient function $\varphi(N)$ (steps 1-3). Step 4 takes a random public exponent e, which is inverted modulo $\varphi(N)$ in step 5 in order to obtain the secret exponent d. Finally, the public key consists of N and e while the secret key contains p, q and d. In some formulations, the factors of N are not part of the secret key, since the decryption does not necessarily requires them. However, they can be used to reduce the computational costs by performing calculations with smaller moduli thanks to the Chinese remainder theorem.

The encryption (Enc) takes a message $m \in \mathbb{Z}_N$ and evaluates the ciphertext c as m raised to the public exponent e modulo N. In practice, m must be invertible in $\mathbb{Z}_N$, otherwise N can be efficiently factorized through the evaluation of $\gcd(m, N)$.

The decryption (Dec) requires only the exponentiation of the ciphertext $c \in \mathbb{Z}_N$ to the secret exponent d modulo N. If p and q have not been discarded, it is possible to improve the performance by evaluating $m_p = c^d \pmod p$ and $m_q = c^d \pmod q$ and applying the Chinese remainder theorem to retrieve $m \in \mathbb{Z}_N$.

The cryptosystem is correct, i.e., the retrieved message is the initial one, thanks to the generalized Euler theorem.

THEOREM 4.1 If N is a square–free integer and $k \equiv 1 \pmod{\varphi(N)}$, then each $m \in \mathbb{Z}$ satisfies $m^k \equiv m \pmod N$.

Indeed, $m = \mathsf{Dec}(\mathsf{Enc}(m, pk), pk, sk)$ because

$$c^d \equiv (m^e)^d \equiv m^{ed \pmod{\varphi(N)}} \pmod N,$$

where $ed \equiv 1 \pmod{\varphi(N)}$ and the theorem can be directly applied.

The security of RSA can be studied for different attacks:

- in a key recovery attack (obtaining d from N and e), since the secret exponent is $e^{-1} \pmod{\varphi(N)}$, it is hard when the modulus is unknown and find $\varphi(N)$ is equivalent to factorize N;
- a message recovery attack, i.e., retrieve m knowing c, N and e, is an instance of discrete e–th root, which is hard unless the factorization of N is known.

In conclusion, the security of RSA relies on the *Integer Factorization Problem* (IFP), i.e., finding the prime factors p, q of N. The best known classical algorithm for solving an IFP is the general number field sieve, which requires sub-exponential time. However, for some bad choices of p and q the resulting instances of the IFP can be easily solved:

- if $|p - q| < 2\sqrt[4]{N}$ then the two primes are close to $\sqrt{N}$ and it is possible to exploit the Fermat factorization method:
  1. take $a = \sqrt{N}$;
  2. if $a^2 - N$ is not a square, then take $a + 1$ and retry;
  3. else, $b^2 = a^2 - N$ and $N = (a + b)(a - b)$;
- if $p - 1$ (or $q - 1$) has only small prime factors, then there is the Pollard $p - 1$ algorithm:
  1. take $k = 2^{k_1} 3^{k_2} \cdots p_l^{k_l}$ and a coprime with N;
  2. evaluate $b = \gcd(a^k - 1, N)$;
  3. if $b = 1$, then increase k and retry;
  4. if $b = N$, then decrease k and retry;
  5. else, b is a factor of N.

  Since any $a \in \mathbb{Z}_N$ satisfies $a^{h(p-1)} \equiv 1 \pmod{p}$ and $p - 1$ has only small factors, a $k = h(p - 1)$ can be easily found.

Other attentions should be put in the choice of the exponents e and d. Generally, it is best to take $e = 2^k + 1$ in order to reduce the number of bits that need to be published. However, if e is too small, then in the encryption of all the messages $m < \sqrt[e]{N}$ there is no reduction modulo N and a simple integer e–th root returns the plaintext.

Also a small secret exponent d can be problematic since there is the Wiener attack [58]. The idea is that, if $q < p < 2q$ and $d < \sqrt[3]{N}/3$, then among the convergents of the continued fraction of e/N there is the secret exponent d that can be obtained in polynomial time.

In addition, it is important to notice that the RSA encryption is *deterministic*, i.e., encrypting the same message with the same public key returns always the same ciphertext. This is not a suitable behaviour for a cryptosystem, since an adversary could easily create a dictionary of all the associations plaintext-ciphertext for the given public key.

Gen(n):	Enc(m, pk):
1. $p, q \leftarrow_{\$} \{0,1\}^{n/2}$ primes	REQUIRE: $m < N$
2. $N = pq$	1. $M = \phi_D(m) = \left(\frac{m^2+D}{m^2-D}, \frac{2m}{m^2-D}\right) \in \mathcal{C}_D$
3. $D \leftarrow_{\$} \mathbb{Z}_N$	2. $C = M^{\otimes_D e} \in \mathcal{C}_D$
4. $\tilde{\varphi}_D(N) = (p - \left(\frac{D}{p}\right))(q - \left(\frac{D}{q}\right))$	3. return C
5. $e \leftarrow_{\$} \mathbb{Z}^{\times}_{\tilde{\varphi}_D(N)}$	Dec(C, pk, sk):
6. $d = e^{-1} \pmod{\tilde{\varphi}_D(N)}$	1. $M = C^{\otimes_D d} \in \mathcal{C}_D$
7. $sk = (p, q, d)$	2. $m = \phi_D^{-1}(x, y) = \frac{x+1}{y} \in \mathbb{Z}_N$
8. $pk = (N, e, D)$	3. return m
9. return pk, sk	

Figure 7: RSA on $(\mathcal{C}_D, \otimes_D)$ with fixed D.

In order to avoid this problem, a padding function is adopted in order to hid the message behind random information that could be removed easily removed by the receiver. A standard choice is the Optimal Asymmetric Encryption Padding (OAEP) [10].

4.3 RSA WITH PELL CONICS AND CUBICS

The use of conics in cryptography has been widely studied, and the quadratic Pell equation has already some applications in cryptosystems like RSA.

The Pell conic can be adopted in RSA because the Brahmagupta product with parameter D satisfies an analogous of the generalized Euler theorem.

However, since RSA works with $\mathbb{Z}_N$ that is not a field, the quadratic character adopted in Chapter 2 is no more well defined and must be replaced by the Jacobi symbol $\left(\frac{D}{N}\right)$. This is still a correct notation on $\mathbb{Z}_p$ and $\mathbb{Z}_q$, since they are fields and the Jacobi symbols are actually Legendre symbols that coincide with χ_p and χ_q, respectively. It is important to remember that for $N = pq$, $\left(\frac{D}{N}\right) = \left(\frac{D}{p}\right)\left(\frac{D}{q}\right)$ so that

$$\left(\frac{D}{N}\right) = \begin{cases} -1, & \text{then } D \text{ is a quadratic non–residue } \mathbb{Z}_N, \\ 0, & \text{then } D \equiv 0 \pmod{N}, \\ 1, & \text{then } D \text{ could be a quadratic residue or not.} \end{cases}$$

THEOREM 4.2 [37] If $N = \prod_{i=1}^{l} p_i$ is a square–free integer, given $D \in \mathbb{Z}_N$, $\tilde{\varphi}_D(N) = \prod_{i=1}^{l}(p_i - \left(\frac{D}{p_i}\right))$ and $k \equiv 1 \pmod{\tilde{\varphi}_D(N)}$, then each point $M \in \mathcal{C}_D$ satisfies $M^{\otimes_D k} = M$. Analogously, each canonical representative $m \in \mathbb{P}_D$ satisfies $m^{\otimes_D k} \equiv m \pmod{N}$.

The standard RSA resulting from the use of $\mathcal{C}_D$ is introduced in [37]. The proposed scheme is described in Figure 7 where, instead of using $\mathbb{Z}_N$, the Pell conic over $\mathbb{Z}_N$ with parameter D is adopted.

Gen(n):	Enc(msg, pk):
1. $p, q \leftarrow_{\$} \{0,1\}^{n/2}$ primes	REQUIRE: $msg < N^2$
2. $N = pq$	1. $(x, y) \leftarrow\!\shortmid msg$
3. $\tilde{\varphi}(N) = (p^2-1)(q^2-1)$	2. $D = \frac{x^2-1}{y^2} \in \mathbb{Z}_N$
4. $e \leftarrow_{\$} \mathbb{Z}^{\times}_{\tilde{\varphi}(N)}$	3. $C = (x, y)^{\otimes_D e} \in \mathcal{C}_D$
5. $d = e^{-1} \pmod{\tilde{\varphi}(N)}$	4. return C, D
6. $sk = (p, q, d)$	Dec(C, D, pk, sk):
7. $pk = (N, e)$	1. $M = C^{\otimes_D d} \in \mathcal{C}_D$
8. return pk, sk	2. $msg \leftarrow\!\shortmid M$
	3. return msg

Figure 8: RSA on $(\mathcal{C}_D, \otimes_D)$ with generic D depending on msg.

The key generation algorithm (Gen) takes an n bits long modulus $N = pq$ in steps 1-2, while D and the cardinality $\tilde{\varphi}_D(N)$ are obtained in steps 3-4. Then the algorithm continues as in the classical RSA with $\tilde{\varphi}_D(N)$ instead of $\varphi(N)$. The private key is the same generated for the classical RSA, while the public key contains the modulus N, the exponent e and also the parameter D for the Pell conic.

In the encryption algorithm (Enc), the message $m \in \mathbb{Z}_N$ is encoded in step 1 into a point of the Pell conic using ϕ_D and the ciphertext is simply its power to e over $\mathcal{C}_D$.

The decryption algorithm (Dec) works backwards, i.e., after obtaining the initial point through the power of the ciphertext to d over $\mathcal{C}_D$, the message is retrieved using the inverse of ϕ_D.

As the author notices, this is simply a scholastic cryptosystem. In practice, its computational costs are doubled with respect to those of the classical RSA. Also the size of the ciphertext is doubled, while the public key includes also D, so that more information needs to be sent. Furthermore, despite this disadvantages, there is no gain in security since the IFP has the same difficulty of an analogous instance of RSA.

An alternative approach was proposed in [32] and is described in Figure 8. The main difference with the previous cryptosystem is that the Pell conic is not fixed in the key generation algorithm, but it depends on the point obtained from the message.

This is why, in the key generation algorithm (Gen), after obtaining the modulus $N = pq$ in steps 1-2, step 3 considers as cardinality the product of the possible orders of the Pell conic over $\mathbb{Z}_p$ and $\mathbb{Z}_q$, i.e.,

$$\tilde{\varphi}(N) = (p-1)(p+1)(q-1)(q+1).$$

Then, the algorithm continues as the standard RSA.

The encryption algorithm (Enc) takes a message that has double length with respect to the previous cases. This is encoded in step 1 into a point of $\mathbb{Z}_N^2$, from which the parameter D of the related Pell conic is obtained in step 2. The ciphertext contains the e–th power of the point of $\mathcal{C}_D$ obtained in step 3 and also the parameter D.

Gen(n):	Enc(msg, pk):
1. $p, q \leftarrow_\$ \{0,1\}^{n/2}$ primes	REQUIRE: $msg < N^2$
2. $N = pq$	1. $(x, y) \leftarrow\!\shortmid msg$
3. $\tilde{\varphi}(N) = (p+1)(q+1)$	2. $D = \frac{x^2-1}{y^2} \in \mathbb{Z}_N$ quadratic non–residue
4. $e \leftarrow_\$ \mathbb{Z}^\times_{\tilde{\varphi}(N)}$	3. $m = \phi_D^{-1}(x, y) = \frac{x+1}{y} \in \mathbb{P}_D$
5. $d = e^{-1} \pmod{\tilde{\varphi}(N)}$	4. $c = m^{\otimes_D e} \in \mathbb{P}_D$
6. $sk = (p, q, d)$	5. return c, D
7. $pk = (N, e)$	Dec(c, D, pk, sk):
8. return pk, sk	1. $m = c^{\otimes_D d} \in \mathbb{P}_D$
	2. $msg \leftarrow\!\shortmid \phi_D(m) = \left(\frac{m^2+D}{m^2-D}, \frac{2m}{m^2-D}\right)$
	3. return msg

Figure 9: RSA–like cryptosystem using ϕ_D with D non–square depending on msg.

In this way, the receiver can easily obtain the plaintext using Dec on the received point, which involves the d–th power over $\mathcal{C}_D$ and the decoding from the obtained point to the original message.

With respect to the classical RSA, the keys have the same size but the bit–length of plaintext and ciphertext are doubled. However, also the computational costs are doubled because it is required the power of a point of the Pell conic. Considering that again the security is not increased, this cryptosystem is equivalent to the classical RSA.

A different enhancement of RSA was described in [36], where the concept of RSA–like cryptosystems obtained exploiting a group isomorphism is introduced. The idea is to encode the message as an element of a group $(\mathcal{G}, \otimes)$, e.g., a curve, then exploit an explicit group isomorphism to another $(\mathcal{G}', \odot)$, so that the exponentiation required in the decryption is computationally lighter.

Following this approach, Pell conics have been exploited in [48] and [11], for D quadratic residue and non–residue, respectively. In particular, the second proposal adopts ϕ_D, i.e., the group isomorphism between $\mathbb{P}_D$ and $\mathcal{C}_D$, to obtain the RSA–like cryptosystem described in Figure 9. In addition, since the formulation works in $\mathbb{P}_D$, the authors adopt the modified More algorithm based on Rédei rational functions, described in Figure 3.

Differently from the previous cryptosystems, in the key generation algorithm, after obtaining the modulus $N = pq$ in steps 1-2, the cardinality of the curve is fixed in step 3 as $(p+1)(q+1)$, which means the required D is a quadratic non–residue in both $\mathbb{Z}_p$ and $\mathbb{Z}_q$. Then, the algorithm continues as the classical RSA.

The encryption is similar to the previous one but, instead of working on the Pell conic, the inverse of ϕ_D is exploited to pass to the canonical representatives in $\mathbb{P}_D$ in step 3. When the plaintext length is fixed to $2n$, this allows to reduce the ciphertext size from $3n$ of the proposals in [48] to $2n$.

Gen(n):	Enc(msg, pk):
1. $p, q \leftarrow_{\$} \{0,1\}^{n/2}$ primes	REQUIRE: $msg < N^2$
2. $p, q \equiv 1 \pmod 3$	1. $(l, m) \hookleftarrow msg$
3. $N = pq$	2. $c = (l, m)^{\odot_R e} \in \mathbb{P}_R$
4. $R \leftarrow_{\$} \mathbb{Z}_N$ non–cube in $\mathbb{Z}_p$ and $\mathbb{Z}_q$	3. return c
5. $\tilde{\varphi}_R(N) = (p^2+p+1)(q^2+q+1)$	Dec(c, pk, sk):
6. $e \leftarrow_{\$} \mathbb{Z}^{\times}_{\tilde{\varphi}_R(N)}$	1. $(l, m) = c^{\odot_R d} \in \mathbb{P}_R$
7. $d = e^{-1} \pmod{\tilde{\varphi}_R(N)}$	2. $msg \hookleftarrow (l, m)$
8. $sk = (p, q, d)$	3. return msg
9. $pk = (N, e, R)$	
10. return pk, sk	

Figure 10: RSA on $\left(\mathbb{P}_R, \odot_R\right)$ with fixed R non–cube.

Finally, the decryption algorithm (Dec) retrieves the element of $\mathbb{P}_D$ and then the message using ϕ_D in the version from Equation 2.4.

As observed in [11], in a comparison with the classical RSA in terms of computational costs of Dec when the plaintext length is fixed to $2n$, the proposal in Figure 9 is better because it requires only 1 exponentiation in $\mathbb{P}_D$, plus 3 multiplications and 1 inversion in $\mathbb{Z}_N$, instead of two exponentiation in $\mathbb{Z}_N$. In the same scenario, the described formulation is also better than the proposals from [48], which require 1 exponentiation in $\mathbb{Z}_N$ and a comparable number of multiplications in $\mathbb{Z}_N$, but at least two inversions.

Despite these advantages, the cryptosystem in Figure 9 presents an important problem that will be described in Section 5.1, where a solution is proposed.

Also the cubic Pell equation can be exploited in cryptosystems based on RSA. In this case, the results in Section 3.2 give different cases since for any p prime and $R \in \mathbb{Z}_p^{\times}$

$$\tilde{\varphi}_R(p) = \begin{cases} p^2+p+1, & \text{for } p \equiv 1 \pmod 3 \text{ and R non–cube,} \\ (p-1)^2, & \text{for } p \equiv 1 \pmod 3 \text{ and R cube,} \\ p^2-1, & \text{for } p \equiv 2 \pmod 3, \\ p^2, & \text{for } p \equiv 0 \pmod 3. \end{cases}$$

Despite this difference, a formulation of the generalized Euler theorem can proved analogously as in the quadratic case.

THEOREM 4.3 If $N = \prod_{i=1}^{l} p_i$ is a square–free integer, given $R \in \mathbb{Z}_N$, $\tilde{\varphi}_R(N) = \prod_{i=1}^{l} \tilde{\varphi}_R(p_i)$ and $k \equiv 1 \pmod{\tilde{\varphi}_R(N)}$, then each $M \in \mathcal{C}_R$ satisfies $M^{\odot_R k} = M$. Analogously, each canonical representative in the projectivization $(l, m) \in \mathbb{P}_R$ satisfies $(l, m)^{\odot_R k} = (l, m)$.

In [46], RSA is formulated using the projectivization related to a Pell cubic with R non–cube, as described in Figure 10.

In order to have a non–cube parameter, steps 1-2 of the key generation algorithm (Gen) take $p, q \equiv 1 \pmod 3$, the modulus $N = pq$ of n bits is obtained in step 3 and, after choosing in step 4 the parameter r non–cube in both $\mathbb{Z}_p$ and $\mathbb{Z}_q$, the cardinality of the cubic is evaluated in step 5 according to the generalized Euler theorem. Then, the algorithm continues as the classical RSA but, as for the quadratic version in Figure 7, the public key contains also the parameter r.

The encryption algorithm (Enc) takes a message $2n$ bits long which is converted in step 1 into a canonical representative in $\mathbb{P}_r$. Thus, its e–th power is evaluated in step 2 and the result is the ciphertext.

The decryption algorithm is straightforward since simply evaluates the d–th power of the ciphertext in $\mathbb{P}_r$ and retrieve the message from the obtained element.

Clearly, this is not a RSA–like cryptosystems since no group isomorphism is exploited. In particular, this cryptosystem is analogous to the one described in Figure 7 but with the cubic projectivization. It presents no advantages with respect to the RSA–like cryptosystem in Figure 9 the message and ciphertext length are the same as well as the security order, but the computational costs are clearly higher. The authors explicitly observe that finding the group isomorphism would allow to improve their scheme by obtaining the cubic version of the quadratic cases described before. Exploiting the results from Chapter 3, this problem is tackled in Section 6.1.

4.4 ELGAMAL CRYPTOSYSTEM

One of the other main cryptosystems on which is based modern cryptography was introduced in 1985 by ElGamal [24]. This PKE scheme is based on the same concept of the Diffie-Hellman key exchange: taking a cyclic group $(\mathcal{G}, \otimes)$ and a generator $g \in \mathcal{G}$, it is easy to evaluate $h = g^{\otimes x} \in \mathcal{G}$ but knowing only g and h it is difficult to obtain the used exponent x. This problem is called Discrete Logarithm Problem (DLP) and, together with the IFP, constitutes the foundations of the security in nowadays public–key cryptography.

The cryptosystem proposed by ElGamal is presented in Figure 11.

The algorithm for the key generation (Gen) takes as input an integer n representing the bit–length of the order q of the cyclic group to be used. Usually q is taken as prime as possible in order to avoid possible small subgroups that constitute easy instances of the DLP. In steps 1-2, the order and the group $(\mathcal{G}, \otimes)$ are chosen, generally there are recommended options depending on the order q. Then, in step 3, a generator $g \in \mathcal{G}$ is taken. After choosing a random exponent sk as secret key in step 4, a public point $h \in \mathcal{G}$ is obtained in step 5 as the sk–th power of the generator g. In conclusion, the public key contains all the information about the cyclic group and the elements $g, h \in \mathcal{G}$.

Gen(n):	Enc(msg, pk):
1. $q \leftarrow_\$ \{0,1\}^n$	REQUIRE: $msg < q$
2. $(\mathcal{G}, \otimes)$ cyclic of order q	1. $m \leftarrow\!\shortmid msg$
3. $g \leftarrow_\$ \mathcal{G}$ generator	2. $r \leftarrow_\$ \{1, \ldots, q-1\}$
4. $sk \leftarrow_\$ \{1, \ldots, q-1\}$	3. $c_1 = g^{\otimes r} \in \mathcal{G}$
5. $h = g^{\otimes sk} \in \mathcal{G}$	4. $c_2 = h^{\otimes r} \otimes m \in \mathcal{G}$
6. $pk = (q, \mathcal{G}, \otimes, g, h)$	5. return c_1, c_2
7. return pk, sk	Dec(c_1, c_2, pk, sk):
	1. $m = (c_1^{\otimes sk})^{-1} \otimes c_2 \in \mathcal{G}$
	2. $msg \leftarrow\!\shortmid m$
	3. return msg

Figure 11: ElGamal PKE scheme.

The encryption algorithm (Enc) takes a message msg smaller than q which is encoded in step 1 into an element $m \in \mathcal{G}$. After taking a random exponent r in step 2, the ciphertext is obtained as two elements $c_1, c_2 \in \mathcal{G}$: the first one is evaluated in step 3 as the r–th power of the public generator g, while c_2 is determined in step 4 as the group operation between $h^{\otimes r}$ and the element m representing the message.

During the decryption algorithm (Dec), the element m is retrieved in step 1 by evaluating the group operation between the inverse of $c_1^{\otimes sk}$ and c_2. Finally, the original message is recovered in step 2.

The cryptosystem is correct since

$$\begin{aligned}(c_1^{\otimes sk})^{-1} \otimes c_2 &= ((g^{\otimes r})^{\otimes sk})^{-1} \otimes h^{\otimes r} \otimes m \\ &= (h^{\otimes r})^{-1} \otimes h^{\otimes r} \otimes m = m.\end{aligned}$$

The security for ElGamal can be studied considering two different scenarios:

- a key recovery attack, i.e., obtain sk knowing only pk, involves solving the DLP instance given by $h = g^{\otimes sk} \in \mathcal{G}$;
- a message recovery attack, i.e., retrieve m knowing the public key and the ciphertext, requires to find the exponent r from the equation $c_1 = g^{\otimes r} \in \mathcal{G}$, which is another instance of DLP.

In conclusion, the security of ElGamal relies clearly on the DLP.

It is noteworthy that, differently from RSA, the ElGamal is a *probabilistic* cryptosystem since each encryption of the same message with the same public key depends on the chosen random exponent r, i.e., different runs of Enc with same inputs return different outputs. This means that the ElGamal cryptosystem does not require the use of a padding function in order to satisfy the minimal security criteria.

Gen(l, n):	Sig(msg, pk, sk):
1. $q \leftarrow_\$ \{0,1\}^n$ prime	1. $r \leftarrow_\$ \{2, \ldots, q-1\}$
2. $p \leftarrow_\$ \{0,1\}^l$ prime, $p - 1 = dq$	2. $s_1 = (g^r \pmod p) \pmod q$
3. $h \leftarrow_\$ \{2, \ldots, p-2\}$	3. $s_2 = r^{-1}(H(msg) + sk \cdot s_1) \pmod q$
4. $g = h^d \pmod p$	4. return s_1, s_2
5. $sk \leftarrow_\$ \{2, \ldots, q-1\}$	Ver(msg, s_1, s_2, pk):
6. $y = g^{sk} \pmod p$	1. $u_1 = H(msg) \cdot s_2^{-1} \pmod q$
7. $pk = (p, q, g, y)$	2. $u_2 = s_1 \cdot s_2^{-1} \pmod q$
8. return pk, sk	3. $v = (g^{u_1} \cdot y^{u_2} \pmod p) \pmod q$
	4. return $s_1 = v$?

Figure 12: Digital Signature Algorithm (DSA).

4.5 DIGITAL SIGNATURE ALGORITHM AND ECDSA

Among all the cryptosystems classified as DLP–based, the mainly used is the Digital Signature Algorithm (DSA). There are two classical formulations: the original one is less efficient and is classified as Finite Field Cryptography (FFC) [27], while the most efficient and widely adopted is among the cryptosystems in Elliptic Curve Cryptography (ECC), namely the ECDSA [56]. Before exploring the possible formulations of DSA with the introduced groups on the Pell conics and cubics, the classical versions are here described.

The Digital Signature Algorithm (DSA) is detailed in Figure 12.

The key generation algorithm is quite similar to the one for ElGamal with $\mathcal{G} = \mathbb{Z}_p$, but a multiplicative subgroup of order q is required. The input is a pair of integers $(l, m) \in \{(1024, 160), (2048, 224), (3072, 256), (7680, 384), (15360, 512)\}$ depending on the standard security strengths from [8]. The public parameters, that may be shared between different users, are p, q and the generator $g \in \mathbb{Z}_p$ of order q obtained in steps 1-4. The secret key is an integer smaller than q, that is used as exponent for $g \in \mathbb{Z}_p$ to obtain the public key y in step 6.

In the signature algorithm, after taking a random integer $r < q$ in step 1, the message is signed through two values: s_1 is obtained in step 2 as the r-power of g in $\mathbb{Z}_p$ then reduced modulo q; s_2 is obtained in step 3 as the digest of msg through a public hash H plus $sk \cdot s_1$, all multiplied by the inverse of r in $\mathbb{Z}_q$.

The verification algorithm evaluates two exponents in $u_1, u_2 \in \mathbb{Z}_q$ in steps 1-2, then compares the value v with the received s_1 so that, if the signature is valid, then it returns True since

$$\begin{aligned} v &= \left(g^{u_1} \cdot y^{u_2} \pmod p\right) \pmod q \\ &= \left(g^{H(msg)\cdot s_2^{-1} \pmod q} \cdot (g^{sk})^{s_1 \cdot s_2^{-1} \pmod q} \pmod p\right) \pmod q \\ &= \left(g^{s_2^{-1}(H(msg)+sk\cdot s_1) \pmod q} \pmod p\right) \pmod q \\ &= \left(g^{s_2^{-1}\cdot r \cdot s_2 \pmod q} \pmod p\right) \pmod q = s_1. \end{aligned}$$

Gen(n):	Sig(msg, pk, sk):
1. $q \leftarrow_\$ \{0,1\}^n$ prime	1. $r \leftarrow_\$ \{2,\dots,q-1\}$
2. $E(\mathbb{Z}_p)$ elliptic curve, p prime	2. $S = r \cdot G \in E(\mathbb{Z}_p)$
3. $G \leftarrow_\$ E(\mathbb{Z}_p)$ of order q	3. $s_1 = x_S \pmod q$
4. $sk \leftarrow_\$ \{2,\dots,q-1\}$	4. $s_2 = r^{-1}(H(msg) + sk \cdot s_1) \pmod q$
5. $P = sk \cdot G \in E(\mathbb{Z}_p)$	5. return s_1, s_2
6. $pk = (p, q, E(\mathbb{Z}_p), G, P)$	Ver(msg, s_1, s_2, pk):
7. return pk, sk	1. $u_1 = H(msg) \cdot s_2^{-1} \pmod q$
	2. $u_2 = s_1 \cdot s_2^{-1} \pmod q$
	3. $V = u_1 \cdot G + u_2 \cdot P \in E(\mathbb{Z}_p)$
	4. return $s_1 \equiv x_V \pmod q$?

Figure 13: DSA with Elliptic Curves (ECDSA).

DSA was a standard proposed by the U.S. National Institute of Standard and Technology (NIST) in 1991, and among the first public responses to the proposal, Vanstone introduced the possibility of a more efficient DSA with Elliptic Curves (ECDSA). The resulting cryptosystem is described in Figure 13.

Clearly, the main difference with DSA is the adopted cyclic group. The key generation algorithm generates in steps 1-3 the public parameters consisting of a prime q, the elliptic curve $E(\mathbb{Z}_p)$ and a generator G of the cyclic group of order q. Since secure elliptic curves are not easy to generate randomly, some chosen ones are considered standard, e.g., [16, 26]. Then, the algorithm takes the secret key and evaluate the point for the public key as in DSA.

The signature algorithm is very similar to the one for DSA: a random factor r taken in step 1 is used in step 2 to obtain a point S on the curve. Here, instead of the double reduction modulo p and q in Figure 12, the x coordinate of S is reduced modulo q in order to obtain the first element s_1 of the signature in step 3. Finally, s_2 is obtained as in DSA.

The decryption algorithm is again analogous to DSA: after evaluating u_1 and u_2 in steps 1-2 in the same way as in Figure 12, the point V used to verify the signature is obtained in step 3. The algorithm is correct since

$$\begin{aligned} V &= u_1 \cdot G + u_2 \cdot P \\ &= \left(H(msg) \cdot s_2^{-1} \pmod q\right) \cdot G + \left(s_1 \cdot s_2^{-1} \pmod q\right) \cdot sk \cdot G \\ &= \left(s_2^{-1}(H(msg) + sk \cdot s_1) \pmod q\right) \cdot G \\ &= \left(s_2^{-1} \cdot r \cdot s_2 \pmod q\right) \cdot G = S. \end{aligned}$$

The success of ECDSA is due mainly to its shorter key lengths with respect to FFC and RSA signatures, as well as for its advantages of performance and scalability [8].

5

NEW CRYPTOSYSTEMS WITH THE PELL CONIC

This chapter focuses on the construction of new cryptosystems that exploit the Pell conic.

Firstly, after some considerations about the RSA–like cryptosystem on the Pell conic described in Figure 9 and introduced in [11], an alternative version without the constraint on D of being a quadratic non–residue is obtained in Section 5.1.

Then, the cyclic structure of the Pell conic over a finite field $\mathbb{F}_q$ and the group isomorphisms obtained in Section 2.2 are exploited to obtain different ElGamal formulations, considering that:

- as observed in Section 2.2.2, when $\chi_q(D) = 1$, the Pell conic is isomorphic to $\mathbb{F}_q^\times$, so that its applications in cryptography are not different from FFC. On the other hand, when D is a non–square in $\mathbb{F}_q$, $\mathcal{C}_D$ is isomorphic to the subgroup of $\mathbb{F}_{q^2}^\times$ of order $q+1$, so that its applications in cryptography are not trivial;
- in this case, it is also important to avoid small subgroups since they give easier DLP instances. Thus, the best option is to have $q = 2p-1$ with p prime, so that only the elements in the trivial subgroup of order 2 must be avoided;
- from a computational point of view, it is not useful to use a generalized Pell conic $\left(\mathcal{C}_{D,Q}, \otimes_{D,Q,a,b}\right)$ in a formulation of ElGamal since the group isomorphism between $\mathcal{C}_D$ and $\mathcal{C}_{D,Q}$ ($\tau_{D,Q}^{a,b}$) reduces the complexity of the DLP over $\mathcal{C}_{D,Q}$ to the DLP over $\mathcal{C}_D$, while the computational costs would arise.

In particular, three different ElGamal formulations are proposed: the first one is introduced in Section 5.2 and exploits directly the cyclic group $\left(\mathcal{C}_D, \otimes_D\right)$. Since the projectivization $\left(\mathbb{P}_D, \otimes_D\right)$ is also a cyclic group, isomorphic to the Pell conic, it can be used in a second formulation of ElGamal that is described in Section 5.3. The third option is a new formulation of ElGamal that requires ϕ_D and $\delta_{D,D'}$, i.e., the group isomorphism between $\mathbb{P}_D$ and $\mathcal{C}_D$ and group isomorphism between $\mathcal{C}_D$ and $\mathcal{C}_{D'}$, respectively, where $\chi_q(D) = \chi_q(D')$ and is defined in Section 5.4.

A last proposal for a digital signature scheme, inspired by DSA and ECDSA, is introduced in Section 5.5.

Finally, in Section 5.6, some comparisons in terms of security, data–size and performance among the proposals and with the classical schemes are described.

Gen(n):	Enc(msg, pk):
1. $p, q \leftarrow_\$ \{0,1\}^{n/2}$ primes	REQUIRE: $msg < N^2$
2. $N = pq$	1. $(x, y) \leftarrow\!\shortmid msg$
3. $\tilde{\varphi}(N) = (p^2-1)(q^2-1)$	2. $D = \frac{x^2-1}{y^2} \in \mathbb{Z}_N$
4. $e \leftarrow_\$ \mathbb{Z}^\times_{\tilde{\varphi}(N)}$	3. $m = \phi_D^{-1}(x, y) = \frac{x+1}{y} \in \mathbb{P}_D$
5. $d = e^{-1} \pmod{\tilde{\varphi}(N)}$	4. $c = m^{\otimes_D e} \in \mathbb{P}_D$
6. $sk = (p, q, d)$	5. return c, D
7. $pk = (N, e)$	Dec(c, D, pk, sk):
8. return pk, sk	1. $m = c^{\otimes_D d} \in \mathbb{P}_D$
	2. $msg \leftarrow\!\shortmid \phi_D(m) = \left(\frac{m^2+D}{m^2-D}, \frac{2m}{m^2-D}\right)$
	3. return msg

Figure 14: RSA–like cryptosystem using ϕ_D with generic D depending on msg.

5.1 ALTERNATIVE RSA–LIKE CRYPTOSYSTEM

The RSA–like cryptosystem from [11] in Figure 9 is theoretically correct, but in a real implementation has a construction problem.

In the algorithm Gen, step 3 fixes the cardinality of the Pell conic as $\tilde{\varphi}(N) = (p+1)(q+1)$. This means that the parameter D is constrained to be a quadratic non–residue in both $\mathbb{Z}_p$ and $\mathbb{Z}_q$. When considering the composite modulus N, this condition assures that D is a quadratic non–residue, but excludes the cases which are quadratic non–residue only modulo p or q. The problem arises in step 2 of the algorithm Enc, which requires to check that the parameter D of the Pell conic obtained from the point representing the message is a quadratic non–residue. Since the factorization of N is unknown by the sender, the best way to do so is to evaluate the Jacobi symbol $\left(\frac{D}{N}\right)$ knowing that it is equal to -1 if and only if D is a quadratic non–residue in $\mathbb{Z}_N$. However, when $\left(\frac{D}{N}\right) = \left(\frac{D}{p}\right)\left(\frac{D}{q}\right) = -1$, the cardinality of the conic should be $(p-1)(q+1)$ or $(p+1)(q-1)$, which is in contradiction with the cardinality $\tilde{\varphi}(N)$ fixed in the key generation at step 3.

In order to avoid this misbehaviour, a modification in one or both the algorithms is required, but:

- asking for $\left(\frac{D}{N}\right) = 1$ in the encryption is not the solution, since this is satisfied also by quadratic residues modulo N (and p, q);
- fixing $\tilde{\varphi}(N) = (p \pm 1)(q \mp 1)$ in the key generation results in a correct behaviour only for half of the values for D that are accepted in the encryption.

A working solution inspired by the cryptosystem in Figure 8 is described in Figure 14. The algorithms are analogous to those in Figure 9, but in the key generation step 3 takes $\tilde{\varphi}(N)$ as the product of the possible orders of the curve over $\mathbb{Z}_p$ and $\mathbb{Z}_q$, i.e., $(p^2-1)(q^2-1)$.

Gen(n):	Enc(msg, pk):
1. $p \leftarrow_\$ \{0,1\}^{n-1}$ prime	REQUIRE: $msg < q$
2. $q = 2p - 1$	1. $y \hookleftarrow msg$
3. $D \leftarrow_\$ \mathbb{F}_q$ with $\chi_q(D) = -1$	2. $x = \sqrt{1 + Dy^2} \in \mathbb{F}_q$
4. $G \leftarrow_\$ \mathcal{C}_D$ of order $q+1$	3. $r \leftarrow_\$ \{2, \dots, q\}$
5. $sk \leftarrow_\$ \{2, \dots, q\}$	4. $C_1 = G^{\otimes_D r} \in \mathcal{C}_D$
6. $H = G^{\otimes_D sk} \in \mathcal{C}_D$	5. $C_2 = H^{\otimes_D r} \otimes_D (x, y) \in \mathcal{C}_D$
7. $pk = (q, D, G, H)$	6. return C_1, C_2
8. return pk, sk	Dec(C_1, C_2, pk, sk):
	1. $(x, y) = (C_1^{\otimes_D sk})^{-1} \otimes_D C_2 \in \mathcal{C}_D$
	2. $msg \hookleftarrow y$
	3. return msg

Figure 15: ElGamal with $(\mathcal{C}_D, \otimes_D)$ of order $q+1$.

This means that the bit–length of secret exponent d is doubled ($2n$ instead of n), but it allows to accept any D in step 2 of the encryption.

In order to avoid an increment of the computational costs, the private exponent d can also be determined only in the decryption, since the receiver can easily evaluate the actual cardinality of the conic $\tilde{\varphi}_D(N) = (p - \left(\frac{D}{p}\right))(q - \left(\frac{D}{q}\right))$ from the received D. In this way, the exponentiation in step 1 has an exponent of n bits as in the cryptosystem in Figure 9.

5.2 ELGAMAL WITH THE PELL CONIC

The first cryptosystem is detailed in Figure 15 and consists in ElGamal with the cyclic group $(\mathcal{C}_D, \otimes_D)$.

The algorithm for the key generation (Gen) takes as input an integer n representing the bit–length of the cardinality q of the finite field. By following the considerations on the choice of q, in step 1 a prime p of $n-1$ bits is taken randomly such that $q = 2p - 1$ in step 2. Then, in step 3, the parameter $D \in \mathbb{F}_q$ with $\chi_q(D) = -1$ is taken, so that the order of the cyclic group $(\mathcal{C}_D, \otimes_D)$ is $q + 1 = 2p$. Since half of the elements in $\mathbb{F}_q^\times$ are non–squares, the search ends rapidly. In general, it is useful to take D small, so that the computational costs of $\otimes_D$ are lower. In step 4, a generator G of $\mathcal{C}_D$ is taken randomly. Since the generators are $\varphi(q+1) = \varphi(2p) = p - 1$, excluding the trivial subgroup of order 2, i.e., $\{(1,0), (-1,0)\}$, there is 50% of probability to take one of them at the first attempt. Then the algorithm proceeds as the classical ElGamal key generation: the secret key sk is a random exponent taken in step 5 and a public point $H \in \mathcal{C}_D$ is obtained in step 6 through the square–multiply algorithm with $\otimes_D$ (introduced in Figure 2). In conclusion, the public key contains the cardinality q, the parameter $D \in \mathbb{F}_q$ and the points $G, H \in \mathcal{C}_D$.

Gen(n):	Enc(msg, pk):
1. $p \leftarrow_\$ \{0,1\}^{n-1}$ prime	REQUIRE: $msg < q$
2. $q = 2p - 1$	1. $m \leftarrow\!\shortmid msg$
3. $D \leftarrow_\$ \mathbb{F}_q$ with $\chi_q(D) = -1$	2. $r \leftarrow_\$ \{2, \ldots, q\}$
4. $g \leftarrow_\$ \mathbb{P}_D$ of order $q+1$	3. $c_1 = g^{\otimes_D r} \in \mathbb{P}_D$
5. $sk \leftarrow_\$ \{2, \ldots, q\}$	4. $c_2 = h^{\otimes_D r} \otimes_D m \in \mathbb{P}_D$
6. $h = g^{\otimes_D sk} \in \mathbb{P}_D$	5. return c_1, c_2
7. $pk = (q, D, g, h)$	Dec(c_1, c_2, pk, sk):
8. return pk, sk	1. $m = (c_1^{\otimes_D sk})^{-1} \otimes_D c_2 \in \mathbb{P}_D$
	2. $msg \leftarrow\!\shortmid m$
	3. return msg

Figure 16: ElGamal with $(\mathbb{P}_D, \otimes_D)$ of order $q+1$.

The encryption algorithm (Enc) takes a message msg smaller than q which is used in step 1 to determine the y coordinate of a point of $\mathbb{F}_q^2$. The corresponding x such that $(x, y) \in \mathcal{C}_D$ is obtained in step 2. Since such a point could not exist, some bits of y can be kept variable by reducing the maximum length of the message msg. In the following steps, the ciphertext consisting of two points $C_1, C_2 \in \mathcal{C}_D$ is obtained: after taking a random exponent $r \leqslant q$ in step 3, it is used in step 4 to obtain C_1 through the square–multiply exponentiation with $\otimes_D$ and base the public generator G, while the second point C_2 is determined in step 5 as the Brahmagupta product of $H^{\otimes_D r}$ with the point (x, y) representing the message.

During the decryption algorithm (Dec), the point (x, y) is retrieved in step 1 as the Brahmagupta product of the inverse of $C_1^{\otimes_D sk}$ with C_2. From the obtained y, the original message is recovered in step 2.

From the point of view of security, since as observed in Theorem 2.3 the Pell conic is isomorphic to the multiplicative subgroup $G \subset \mathbb{F}_{q^2}^\times$ of order $q+1$, the DLP in $\mathcal{C}_D$ can be reduced to the DLP in G. Thus, the obtained algorithm is simply an alternative form for that with G.

5.3 ELGAMAL WITH THE PROJECTIVIZATION

The second cryptosystem is described in Figure 16 and consists in ElGamal with the cyclic group $(\mathbb{P}_D, \otimes_D)$.

The algorithm for the key generation (Gen) is very similar to the one in the previous scheme. The input is the bit–length n of the cardinality of the finite field, which is obtained as $q = 2p - 1$ with p random prime of $n-1$ bits (steps 1-2). The parameter $D \in \mathbb{F}_q$ with $\chi_q(D) = -1$ is taken in step 3, so that it is a non–square and the order of the cyclic group $(\mathbb{P}_D, \otimes_D)$ is $q + 1 = 2p$. The search for such a D ends rapidly because half of the elements in $\mathbb{F}_q^\times$ are non–squares. As in the previous cryptosystem, it is best to take a small parameter D in order to reduce the computational costs of the Brahmagupta product in $\mathbb{P}_D$.

In step 3, a random generator $g \in \mathbb{P}_D$ is taken among the possible $\varphi(q+1) = \varphi(2p) = p-1$ choices and, excluding $(1,0),(-1,0)$ that have period 2, there is 50% of probability to find it at the first attempt. Then, in step 4, a random exponent is taken as the secret key sk. The public key, instead, consists of the cardinality q, the parameter $D \in \mathbb{F}_q$, the chosen generator $g \in \mathbb{P}_D$ and h that is the power of g to sk with $\otimes_D$ (step 6), that can be implemented using the modified More algorithm introduced in Figure 3.

The encryption algorithm (`Enc`) takes a message msg smaller than q which is used in step 1 to determine the coordinate m of the canonical representative of an element $[m:1] \in \mathbb{P}_D$. In step 2, an exponent $r \leqslant q$ is chosen randomly. It is then used to obtain the ciphertext in steps 3-4, which consists of two canonical representatives $c_1, c_2 \in \mathbb{P}_D$ given by the power of g to r with $\otimes_D$ and the Brahmagupta product of $h^{\otimes_D r}$ with m, respectively.

During the decryption algorithm (`Dec`), the canonical representative m related to the message is retrieved as the Brahmagupta product of the inverse of $c_1^{\otimes_D sk}$ (which is simply its opposite) with c_2 in step 1. Finally, in step 2, the original message is recovered from m.

The security of this cryptosystem relies on the DLP in $\mathbb{P}_D$, which can be reduced to the DLP on $\mathbb{F}_{q^2}^\times/\mathbb{F}_q^\times$ because of the group isomorphism observed in Section 2.2.1. This is again isomorphic to the multiplicative subgroup $G \subset \mathbb{F}_{q^2}^\times$ of order $q+1$. The advantage with respect to the previous algorithm is that the elements require half the size to be stored, as will be observed in Section 5.6.

5.4 ELGAMAL WITH TWO PELL CONICS

The third cryptosystem is an alternative version of ElGamal with $(\mathbb{P}_D, \otimes_D)$. The new ideas adopted in this scheme are to exploit the group isomorphism between $\mathbb{P}_D$ and $\mathcal{C}_D$ (ϕ_D) to obtain a reduction in the data–size, and to use the group isomorphism between $\mathcal{C}_D$ and $\mathcal{C}_{D'}$ ($\delta_{D,D'}$) introduced in Section 2.2 in order to have the possibility to choose the message without the constraint given by the fixed parameter D. The resulting cryptosystem is described in Figure 17.

The key generation algorithm (`Gen`) takes as input the bit–length n of the cardinality q of the finite field, obtained in steps 1-2 from a randomly chosen prime p of $n-1$ bits such that $q = 2p-1$. The main difference with the previous key generations is in step 3 where, instead of taking a random D, the chosen parameter is the smallest non–square, so that the computational costs of $\otimes_D$ are decreased. Then, a generator $g \in \mathbb{P}_D$ is randomly chosen in step 4, as well as a secret exponent sk in step 5, which are used to obtain $h = g^{\otimes_D sk} \in \mathbb{P}_D$ in step 6, using the modified More algorithm from Figure 3. Finally, the public key contains the cardinality q, the parameter D and the elements g, h, while the secret key is simply the used exponent sk.

Gen(n):	Enc(msg, pk):
1. $p \leftarrow_{\$} \{0,1\}^{n-1}$ prime	REQUIRE: $msg < q^2$
2. $q = 2p - 1$	1. $(x, y) \leftarrow\!\shortmid msg$
3. $D \leftarrow_{\$} \mathbb{F}_q$ min with $\chi_q(D) = -1$	2. $D' = \frac{x^2-1}{y^2} \in \mathbb{F}_q$ with $\chi_q(D') = -1$
4. $g \leftarrow_{\$} \mathbb{P}_D$ of order $q+1$	3. $m = \phi_{D'}^{-1}(x, y) = \frac{x+1}{y} \in \mathbb{P}_{D'}$
5. $sk \leftarrow_{\$} \{2, \ldots, q\}$	4. $r \leftarrow_{\$} \{2, \ldots, q\}$
6. $h = g^{\otimes_D sk} \in \mathbb{P}_D$	5. $\delta = \sqrt{D/D'} \in \mathbb{F}_q$
7. $pk = (q, D, g, h)$	6. $c_1 = \delta_{D,D'}(g)^{\otimes_{D'} r} \in \mathbb{P}_{D'}$
8. return pk, sk	7. $c_2 = \delta_{D,D'}(h)^{\otimes_{D'} r} \otimes_{D'} m \in \mathbb{P}_{D'}$
	8. return c_1, c_2, D'
	Dec(c_1, c_2, D', pk, sk):
	1. $m = -c_1^{\otimes_{D'} sk} \otimes_{D'} c_2 \in \mathbb{P}_{D'}$
	2. $msg \leftarrow\!\shortmid \phi_{D'}(m) = \left(\frac{m^2+D'}{m^2-D'}, \frac{2m}{m^2-D'}\right)$
	3. return msg

Figure 17: ElGamal with $(\mathbb{P}_D, \otimes_D)$ of order $q+1$, ϕ_D and $\delta_{D,D'}$.

The encryption algorithm (Enc) is very different from the classical ElGamal. With respect to the previous algorithms, the maximum length of the message can be doubled because it is used in step 1 to obtain the coordinates of a point $(x, y) \in \mathbb{F}_q^2$. From this point, step 2 searches for a non–square $D' \in \mathbb{F}_q$ such that $(x, y) \in \mathcal{C}_{D'}$. If necessary, some of the bits of x can be kept variable so that such a D' can be found. Then, in step 3, the parameter m related to the point is obtained through the inverse of parametrization $\phi_{D'}$, specifically considering the canonical representative in $\mathbb{P}_{D'}$. In step 4 a random exponent $r \leqslant q$ is chosen. Now, since the public key contains parameters of points of $\mathcal{C}_D$, the isomorphism $\delta_{D,D'}$ is required, i.e., step 5 evaluates the coefficient δ such that $D = \delta^2 D'$. This is used in step 6 to obtain the base $g/\delta \in \mathbb{P}_{D'}$ that raised to r gives c_1 and in step 7 for the base $h/\delta \in \mathbb{P}_{D'}$ that raised to r and multiplied by m gives c_2. The ciphertext contains $c_1, c_2 \in \mathbb{P}_{D'}$ and the parameter D' used in the calculations.

The decryption algorithm (Dec) is analogous to the previous cases but, after evaluating in step 1 m as the product between the inverse of $c_1^{\otimes_{D'} sk}$ and c_2, the message is retrieved in step 2 using the point of $\mathcal{C}_{D'}$ obtained from the element $m \in \mathbb{P}_{D'}$ through $\phi_{D'}$ in the form with the canonical representative given in Equation 2.4.

As for ElGamal with the projectivization, the security is based on the DLP on $\mathbb{P}_D$ and $\mathbb{P}_{D'}$, which can be reduced to the DLP over the multiplicative subgroup $G \subset \mathbb{F}_{q^2}^{\times}$ of order $q+1$.

The advantages in adopting this algorithm are the halved data–size and the doubled length of the message without doubling the computational costs, as will be pointed-out in the last section.

Gen(l, n):	Sig(msg, pk, sk):
1. $p \leftarrow_{\$} \{0,1\}^n$ prime	1. $r \leftarrow_{\$} \{2, \dots, p-1\}$
2. $q \leftarrow_{\$} \{0,1\}^l$ prime, $q+1 = dp$	2. $s_1 = g^{\otimes_D r} \pmod p$
3. $D \leftarrow_{\$} \mathbb{F}_q$ with $\chi_q(D) = -1$	3. $s_2 = r^{-1}(H(msg) + sk \cdot s_1) \pmod p$
4. $h \leftarrow_{\$} \mathbb{P}_D$	4. return s_1, s_2
5. $g = h^{\otimes_D d} \in \mathbb{P}_D$ of order p	Ver(msg, s_1, s_2, pk):
6. $sk \leftarrow_{\$} \{2, \dots, p-1\}$	1. $u_1 = H(msg) \cdot s_2^{-1} \pmod p$
7. $y = g^{\otimes_D sk} \in \mathbb{P}_D$	2. $u_2 = s_1 \cdot s_2^{-1} \pmod p$
8. $pk = (p, q, D, g, y)$	3. $v = (g^{\otimes_D u_1} \otimes_D y^{\otimes_D u_2}) \pmod p$
9. return pk, sk	4. return $s_1 = v$?

Figure 18: DSA with $(\mathbb{P}_D, \otimes_D)$ of order $q+1$.

5.5 DSA WITH THE PELL CONIC

As for the classical ElGamal, the structure of cyclic group on the Pell conic or on the related projectivization can be exploited to formulate alternative DSA. In particular, the most interesting option to be considered is the non–trivial case with $D \in \mathbb{F}_q$ non–square. Since signature schemes require smaller data–size and working with $(\mathcal{C}_D, \otimes_D)$ requires larger data and comparable times with respect to $(\mathbb{P}_D, \otimes_D)$, only the formulation in the second case is introduced. The algorithms for the obtained DSA are described in Figure 18.

The key generation (Gen) takes a couple of integer values (l, n) as in the classical DSA. The first one is the bit–length of the field cardinality $q = dp - 1$ where p is a prime n bits long and d is an even integer (steps 1-2). Then a generator g of the subgroup of $\mathbb{P}_D$ of order p is obtained in steps 3-4 as the exponentiation of a generic element to d. Finally, the secret key is taken in step 6 as an integer smaller than p and used as exponent in step 7 to evaluate the public key y as power of the generator g. The values p, q, D and the generator $g \in \mathbb{P}_D$ can be considered as parameters of the system that can be shared among different users.

In the signature algorithm (Sig), a random exponent r is taken in step 1 and then used to obtain the first part of the signature s_1 as r–th power of the public generator $g \in \mathbb{P}_D$ reduced modulo p. The second part is obtained exactly as in DSA and ECDSA.

In order to verify a signature on the message msg, the algorithm Ver obtains $u_1, u_2 \in \mathbb{Z}_p$ as in the classical schemes. Then they are used to check the validity of s_1 and s_2 by verifying that

$$\begin{aligned} v &= \left(g^{\otimes_D u_1} \otimes_D y^{\otimes_D u_2}\right) \pmod p \\ &= \left(g^{\otimes_D H(msg) \otimes_D s_2^{-1} \pmod p} \cdot (g^{\otimes_D sk})^{\otimes_D s_1 \cdot s_2^{-1} \pmod p}\right) \pmod p \\ &= \left(g^{\otimes_D s_2^{-1}(H(msg) + sk \cdot s_1)}\right) \pmod p \\ &= \left(g^{\otimes_D s_2^{-1} \cdot r \cdot s_2 \pmod p}\right) \pmod p = s_1. \end{aligned}$$

Sec.	Cryptosystem	pk	sk	msg	c
80	RSA	1024	1024	1024	1024
	RSA–like	1024	3072	2048	2048
112	RSA	2048	2048	2048	2048
	RSA–like	2048	6144	4096	4096
128	RSA	3072	3072	3072	3072
	RSA–like	3072	9216	6144	6144
192	RSA	7680	7680	7680	7680
	RSA–like	7680	23040	15360	15360
256	RSA	15360	15360	15360	15360
	RSA–like	15360	46080	30720	30720

Table 1: Data–size in bits of the classical RSA and the RSA–like cryptosystem with ϕ_D introduced in Figure 14 for different security strengths.

5.6 SECURITY, DATA–SIZE AND PERFORMANCE

This section presents some practical results about security, data–size and computational costs for all the newly introduced cryptosystems based on the Pell conic.

The RSA–like cryptosystem described in Figure 14 is the only proposal whose security is based on the IFP. In particular, it shares the same security requirements as the classical RSA, i.e., their public keys have the same size, as described in Table 1 for the standard security strengths from [8]. The table shows a comparison of data–size for public and secret keys, maximum message length and ciphertext size in bits between the classical RSA and the RSA–like cryptosystem with ϕ_D introduced in Figure 14 for different security strengths. The secret key of the RSA–like cryptosystem requires triple the bits for RSA, but this is negligible since it remains stored by the receiver. On the other hand, the sizes of messages and ciphertexts are doubled so that, in a performance comparison of `Enc` and `Dec`, the RSA–like cryptosystem should be compared with a double run of the algorithms of RSA.

This is the idea adopted when collecting the time data in Table 2: the study uses simple Python implementations for both RSA and the RSA–like cryptosystem from Figure 14 for a serial run on the cluster of the DISMA at Politecnico of Turin, on a single CPU with 46G of RAM allocated. The displayed values are the average times in seconds for 10 randomly generated instances, whose times are taken as the minimum of 10 identical runs. In the times for the `Gen` algorithm, the generation of the primes factors p and q of N is not considered since it is a common procedure for all the cryptosystems. For the double use of RSA, the keys can be used multiple times, so that only a single run of `Gen` is considered.

n	Alg.	RSA	RSA $\times$ 2	RSA–like
1024	Gen	0.000478	0.000478	0.001163
	Enc	0.000004	0.000007	0.000601
	Dec	0.004465	0.008930	0.019687
2048	Gen	0.001098	0.001098	0.003044
	Enc	0.000003	0.000006	0.001253
	Dec	0.020171	0.040341	0.115604
3072	Gen	0.001949	0.001949	0.005771
	Enc	0.000003	0.000006	0.001480
	Dec	0.059177	0.118355	0.334385
7680	Gen	0.008427	0.008427	0.029038
	Enc	0.000003	0.000006	0.011386
	Dec	0.744808	1.489616	4.231494
15360	Gen	0.028525	0.028525	0.102918
	Enc	0.000003	0.000006	0.013337
	Dec	5.194125	10.388251	28.937031

Table 2: Average times in seconds for 10 random instances of RSA, RSA repeated two times and the RSA–like cryptosystem with ϕ_D introduced in Figure 14, depending on the bit–length n of N.

Despite the promising results in [11], the practical implementations do not respect the theoretical predictions. Indeed, all the times for the RSA–like cryptosystem collected in the last column are higher than those in columns 3 and 4. This is mainly due to the computational costs of the operation $\otimes_D$ in $\mathbb{P}_D$ that, despite the optimized implementation of the modified More algorithm introduced in Figure 3, is still less efficient with respect to the multiplication in the modular exponentiation required in the classical RSA.

When considering the DLP–based cryptosystems, the security depends on the adopted cyclic group. In particular, since in all the introduced schemes the parameter $D \in \mathbb{F}_q$ is a non–square, Theorem 2.3 gives an explicit group isomorphism between $(\mathcal{C}_D, \otimes_D)$ and the multiplicative subgroup $G \subset \mathbb{F}_{q^2}^{\times}$ of order $q+1$. In addition, this is true also for $(\mathbb{P}_D, \otimes_D)$ through ϕ_D. Thus, the DLP related to the Pell conic can be reduced to that in a finite field that, with respect to the standard security strengths from [8] for FFC, has halved size of q.

The comparison of the size of q for FFC, ECC and cryptosystems based on the Pell conic is detailed in Table 3. Despite the sizes of the fields in Pell-based cryptosystems are halved with respect to those in FFC, the sizes in ECC still remain the smallest. Given these security levels, it is possible to compare the proposed cryptosystems with the classical ones in terms of data–size and performance.

Sec.	FFC	ECC	$\mathcal{C}_D, \mathbb{P}_D$
80	1024	160	512
112	2048	224	1024
128	3072	256	1536
192	7680	384	3840
256	15360	512	7680

Table 3: Field size in bits for different DLP–based cryptosystems depending on the cyclic group and the classical security strength in bits.

Table 4 collects the size of the data involved in the various ElGamal formulations. In particular, the public key is divided in two parts: the public parameters, i.e., the data required for the description of the cyclic group and one of its generators, that can be used by different users, and the actual public key (the element h or the point H). The other collected values are the size of the secret key, the maximum message length and the bit–length of the ciphertext (that for ElGamal is a pair of elements or points). The considered formulations are, in order from top to bottom, the classical ElGamal scheme in Figure 11 with finite fields and with elliptic curves, and the cryptosystems based on the Pell conic described in Figure 15, Figure 16 and Figure 17, i.e., respectively, ElGamal with the cyclic group $(\mathcal{C}_D, \otimes_D)$ of order $q+1$, ElGamal with the cyclic group $(\mathbb{P}_D, \otimes_D)$ of order $q+1$ and the ElGamal cryptosystem that still works with the cyclic group $(\mathbb{P}_D, \otimes_D)$ of order $q+1$ but exploits the parametrization ϕ_D and the isomorphism $\delta_{D,D'}$. For each case, the table shows the data–size depending on the size n of the cardinality q of the related finite field (taken from Table 3) and the values for 80 bits of security strength.

The formulation with $\mathcal{C}_D$ in the third row has the same size of that in FFC (first row) in terms of parameters, public key and ciphertext, but the maximum message length is halved, so that in a fair comparison its encryption and decryption should be run twice, and the ciphertext length becomes the double of that in the first row. Despite this drawback, a performance comparison could be interesting since q has still halved size with respect to FFC.

Looking at the fourth row, i.e, at the formulation with $\mathbb{P}_D$, all the sizes are half of those for FFC, except for the bit–length of the parameters which is still smaller. Again, when fixing the same message length, two runs of `Enc` and `Dec` are required so that the size of the ciphertext is doubled and becomes equal to that in the first row. However, with respect to the previous formulation, the public key has half the size and calculations are still faster than in FFC since q is smaller.

Finally, when comparing the formulation in the fifth row with the classical FFC, its parameters and keys require half the bits, but the maximum message length is the same and the ciphertext is smaller. Thus, this is the best proposal also in terms of information encrypted.

Formulation	par	pk	sk	msg	c_1, c_2
FFC	$2n$	n	n	n	$2n$
	2048	1024	1024	1024	2048
ECC	$6n$	$2n$	n	n	$4n$
	960	320	160	160	640
$\mathcal{C}_D$	$4n$	$2n$	n	n	$4n$
	2048	1024	512	512	2048
$\mathbb{P}_D$	$3n$	n	n	n	$2n$
	1536	512	512	512	1024
ϕ_D, $\delta_{D,D'}$	$2n$	n	n	$2n$	$3n$
	1024	512	512	1024	1536

Table 4: Data–size in bits for ElGamal with FFC, ECC, $\mathcal{C}_D$, $\mathbb{P}_D$ and the alternative formulation, depending on the size n of q and for 80 bits of security.

Despite the formulation in ECC (second row) is competitive for its smallest data, when fixing the maximum message length, its parameters and keys maintain the smallest size, but the ciphertext length grows as that of the formulation with $\mathcal{C}_D$. In particular, the ratio with the length of the message is 4, while in the first and fourth row the ratio is 2 and the best value is 1.5 corresponding to the last proposal.

The last study concerns the performance of the ElGamal formulations and consists in collecting the elapsed times of a simple implementation in Python of each of the algorithms run on the cluster of the DISMA at Politecnico of Turin, on a single CPU with 46G of RAM allocated. For each level of security strength and cryptosystem, the times shown in Table 5 are the averages of 10 randomly generated instances, whose times are taken as the minimum of 10 identical runs. The formulations are compared for different security strengths each with maximum message length fixed. Following considerations from the analysis on the data–size, this results in repeating encryption and decryption k times for ECC, where k is the ratio between the message length for FFC and ECC (e.g., for 80 bits of security, $1024/160 = 6.4$ so that $k = 7$), and 2 times when using directly $\mathcal{C}_D$ or $\mathbb{P}_D$. When working with a finite field $\mathbb{F}_q$, the case with q prime is considered, and the bit–length n of q depends on the standard security strengths, assuming the values obtained in Table 3.

The times for the key generation algorithm do not take into account the generation of the public parameters, i.e., the cyclic group and one of its generators as described in steps 1-3 in Figure 11 or 1-4 in Figure 15, Figure 16 and Figure 17. This because they can be precomputed and used by different users, so that the times take into account only the generation of the keys, i.e., a single exponentiation.

Sec.	Alg.	FFC	ECC × k	$\mathcal{C}_D \times 2$	$\mathbb{P}_D \times 2$	$\phi_D, \delta_{D,D'}$
80	Gen	0.011079	0.028271	0.011713	0.009781	0.007524
	Enc	0.022311	0.393407	0.059983	0.040459	0.028152
	Dec	0.012183	0.194531	0.023631	0.020472	0.010203
112	Gen	0.074718	0.056586	0.073778	0.056865	0.038527
	Enc	0.149400	1.194561	0.364686	0.229299	0.164122
	Dec	0.077622	0.567866	0.148194	0.115962	0.057106
128	Gen	0.233983	0.075437	0.227347	0.171958	0.112873
	Enc	0.467730	1.818186	1.103675	0.689103	0.496599
	Dec	0.239429	0.903710	0.454805	0.347872	0.171190
192	Gen	3.188959	0.185410	2.811594	2.127992	1.372381
	Enc	6.372422	7.454103	13.791595	8.525471	6.291258
	Dec	3.218019	3.718247	5.630895	4.273549	2.103753
256	Gen	22.874051	0.365562	18.155630	13.841428	9.519104
	Enc	45.766954	22.052779	87.457496	55.563741	42.658508
	Dec	22.981310	10.965318	36.287580	27.792128	14.464945

Table 5: Average times in seconds for 10 random instances of ElGamal with FFC, ECC, $\mathcal{C}_D$, $\mathbb{P}_D$ and the alternative formulation, for fixed message length, depending on the security strength.

The key generation performs similarly for FFC and ElGamal with $\mathcal{C}_D$ (Figure 15) in columns 3 and 5, despite the latter is generally a bit faster. The formulation with $\mathbb{P}_D$ (Figure 15) in the sixth column is generally at the third place, beaten by the version that exploits the change of parameter D described in Figure 17 (column 7). ECC in the fourth column is less efficient at lower security strengths but, thanks to the good scalability of elliptic curves, works better than any other formulation starting from the third level of security strength (128 bits).

This advantage is a bit attenuated for encryption and decryption: in particular, ECC is the less efficient formulation for the first four levels and becomes comparable to the others for 128 bits of security. However, for the highest level, the good scalability of elliptic curves allows to increase the efficiency of the two algorithms, which become the fastest. Among the others, ElGamal with $\mathcal{C}_D$ is generally the worst option, followed by the formulation with $\mathbb{P}_D$. For the first four levels of security, the formulations in FFC and from Figure 17 are the most efficient, with encryption slightly better for the former and decryption more efficient for the latter.

In conclusion, considering the big advantage in key and ciphertext size, the new ElGamal using the group isomorphisms ϕ_D and $\delta_{D,D'}$ (Figure 17) seems to be a very powerful alternative for DLP–based PKE schemes.

Formulation	par	pk	sk	s_1, s_2
FFC	$2l+n$	l	n	$2n$
	2208	1024	160	320
ECC	$6n$	$2n$	n	$2n$
	960	320	160	320
$\mathbb{P}_D$	$3l/2+n$	$l/2$	n	$2n$
	1696	512	160	320

Table 6: Data–size in bits for DSA with FFC, ECC and $\mathbb{P}_D$, depending on the sizes l, n of q, p and for 80 bits of security strength.

Finally, the new formulation of DSA with $\mathbb{P}_D$ from Figure 18 is compared with the classical DSA in FFC and ECDSA, introduced in Figure 12 and Figure 13, respectively, through the same analysis carried out for the ElGamal formulations.

From the point of view of security, the results obtained for ElGamal can be adapted directly to the formulations of DSA, so that the sizes of the adopted fields depending on the classical security strengths $80, 112, 128, 192, 256$ bits are still the ones described in Table 3. In particular, column 2 can be related to the classical DSA, column 3 to ECDSA and column 4 to DSA with $\mathbb{P}_D$.

Looking at the data–size, there is a main difference with ElGamal since, instead of working in the cyclic group of order q that is l bits long, the classical formulation in FFC, as well as the new proposal, consider a subgroup of prime order p of n bits.

The standard pairs (l, n) for FFC [8], in order of increasing security strength, are $(1024, 160)$, $(2048, 224)$, $(3072, 256)$, $(7680, 384)$ and $(15360, 512)$. For the formulation with $\mathbb{P}_D$, the size of q is $l/2$ as observed for ElGamal, while the related values of n remain unchanged. Table 6 shows the size in bits of the involved data depending on the standard security parameters (l, n) and for 80 bits of security strength, which corresponds to $l = 1024$ and $n = 160$. All the cryptosystems share the same size for secret key and signature, since all the used subgroups have the same order independently of the chosen security strength. For the public parameters and keys, it is noteworthy that, despite ECC maintains the smallest sizes, the new formulation with $\mathbb{P}_D$ is better than the classical DSA in FFC.

The performance of the three digital signatures is studied, as for the ElGamal formulations, by collecting the elapsed times of simple Python implementations run on the cluster of the DISMA at Politecnico of Turin, on a single CPU with 46G of RAM allocated. For each level of security strength and formulation of DSA, Table 7 shows the average times of 10 randomly generated instances, whose times are taken as the minimum of 10 identical runs.

Sec.	Alg.	FFC	ECC	$\mathbb{P}_D$
80	Gen	0.002201	0.028173	0.003310
	Sig	0.002349	0.028242	0.003514
	Ver	0.004529	0.055046	0.007245
112	Gen	0.010515	0.056601	0.013278
	Sig	0.010660	0.056749	0.013333
	Ver	0.021193	0.112299	0.027623
128	Gen	0.021097	0.075419	0.029619
	Sig	0.021220	0.075467	0.029847
	Ver	0.042351	0.150855	0.060985
192	Gen	0.169084	0.185556	0.216012
	Sig	0.169643	0.186087	0.216910
	Ver	0.338843	0.371402	0.443050
256	Gen	0.797075	0.364633	0.983670
	Sig	0.797973	0.366812	0.989801
	Ver	1.593990	0.730208	2.014678

Table 7: Average times in seconds for 10 random instances of DSA with FFC, ECC and $\mathbb{P}_D$, depending on the security strength.

In the key generation, the times required to obtain the public parameters is not counted since they can be precomputed and used by different users. Thus, the collected times take into account only the generation of private and public key, i.e., a single exponentiation in the cyclic group, excluding steps 1-4 in Figure 12, steps 1-3 in Figure 13 and steps 1-5 in Figure 18.

For all the three algorithms, the time comparison is the same: in the first three levels of security strengths the FFC formulation is the best, followed by that with $\mathbb{P}_D$, while ECDSA is the less efficient. For 192 bits of security strength, the scalability of ECC makes it the second best. Finally, in the highest level, the most efficient, ECC becomes the most efficient scheme.

Thus, at lower levels of security, the classical DSA is suitable when there is no constraint on the public data–size, while the new proposal based on $\mathbb{P}_D$ could be a good alternative since the bit–length is reduced to almost 2/3 at the cost of a small loss in performance. When higher security is required, ECDSA is still the best option for both data–size and computational costs.

6

CRYPTOSYSTEMS WITH THE PELL CUBIC

As in the previous chapter for the quadratic case, it is interesting to investigate the applications of the Pell cubic to cryptosystems.

New RSA formulations can be obtained by exploiting Theorem 4.3 that allows to use any Pell cubic $\mathcal{C}_R$ or the related projectivizations $\mathbb{P}_R$. However, as observed for the quadratic cases in Section 4.3, generally there are no advantages with respect to the classical RSA. On the other hand, Section 6.1 investigates the idea of a RSA–like cryptosystem based on the Pell cubic, which can be obtained by improving the cryptosystem introduced in [46] and described in Figure 10. In addition to the theoretical interest, due to the fact that the application of the cubic generalization of the Pell equation is new in cryptography, exploiting the group isomorphisms obtained in Chapter 3 could bring to promising results from the point of view of data–size and performance, since these are the main advantages of RSA–like cryptosystem, as described in Section 4.3. Moreover, if the size reduction from a point in $\mathbb{F}_q^3$ to one in $\mathbb{F}_q^2$ observed in Section 3.2 can be efficiently exploited, it could be interesting to investigate further improvements from the combination with the results on the Pell conic.

Then, the idea of formulating the ElGamal cryptosystems with the groups related to the Pell cubic is considered. While the Pell conic is always a cyclic group, the results obtained in Section 3.2 assures that $\mathcal{C}_R$ is a cyclic group only when:

- $q \equiv 1 \pmod 3$ and the parameter $r \in \mathbb{F}_q$ is a non–cube. This is the most interesting option since the cyclic group $\mathcal{C}_R$ is isomorphic to the subgroup of $\mathbb{F}_{q^3}^\times$ of order q^2+q+1, so that its applications in cryptography are not trivial. However, the group isomorphism ψ_R' that allows to obtain efficiently the points of the cubic from the elements of the projectivization has unknown inverse, so that computations are complicated;

- $q \equiv 2 \pmod 3$ with any parameter $r \in \mathbb{F}_q$ (that are all cubes), in which case the order is q^2-1. Despite in this case the group is isomorphic to $\mathbb{F}_{q^2}^\times$, the group isomorphism ψ_R''' has explicit inverse so that it could be interesting to investigate this formulation in comparison with the one from FFC.

These cases are studied in Section 6.2 for ElGamal with the Pell cubic and in Section 6.3 with the related projectivizations. Finally, in Section 6.4, the group isomorphisms ψ_R''' and $\rho_{R,R'}$ are used as in Section 5.4 to obtain a cryptosystem that has no constraints given by a fixed parameter r, generalizing Figure 17.

$\underline{\texttt{Gen}}(n)$:	$\underline{\texttt{Enc}}(msg, pk)$:
1. $p, q \leftarrow_{\$} \{0,1\}^{n/2}$ primes	REQUIRE: $msg < N^3$
2. $p, q \equiv 1 \pmod 3$	1. $(x, y, z) \hookleftarrow msg$
3. $N = pq$	2. $R = s^3 \hookleftarrow (x, y, z) \in \mathcal{C}_R$
4. $\tilde{\varphi}(N) = (p-1)^2(q-1)^2$	3. $(l, m) = (\psi_R'')^{-1}(x, y, z) \in \mathbb{P}_R$
5. $e \leftarrow_{\$} \mathbb{Z}^{\times}_{\tilde{\varphi}(N)}$	4. $c = (l, m)^{\odot_R e} \in \mathbb{P}_R$
6. $d = e^{-1} \pmod{\tilde{\varphi}(N)}$	5. return c, R
7. $sk = (p, q, d)$	$\underline{\texttt{Dec}}(c, R, pk, sk)$:
8. $pk = (N, e)$	1. $(l, m) = c^{\odot_R d} \in \mathbb{P}_R$
9. return pk, sk	2. $msg \hookleftarrow \psi_R''(l, m)$
	3. return msg

Figure 19: RSA–like cryptosystem using ψ_R'' with R cube depending on msg.

Then, analogously to the quadratic case, a digital signature scheme with $\mathbb{P}_R$ inspired by DSA and ECDSA, is introduced in Section 6.5.

Finally, Section 6.6 focuses on the comparisons in terms of security, data–size and performance among the proposals based on the Pell cubic with the best options based on the Pell conic and with the classical schemes.

6.1 RSA-LIKE CRYPTOSYSTEM WITH THE PELL CUBIC

Thanks to the explicit group isomorphisms for $\mathcal{C}_R$ obtained in Section 3.2, it is possible to formulate new RSA–like cryptosystems.

For RSA there is no need for a cyclic group, since its correctness is assured by the generalized Euler theorem and Theorem 4.3 is the version for the Pell cubic. However, a RSA–like cryptosystem requires an explicit group isomorphism that works as parametrization for the cubic, with an explicit inverse. In the following, since RSA uses $\mathbb{Z}_N$ with $N = pq$ and p, q primes, the finite fields are $\mathbb{Z}_p$ and $\mathbb{Z}_q$.

As observed in Section 3.2, despite having the theoretical proofs for all the cases generated by the choice of p (or q) and the parameter R, the only cases for which the group isomorphism between $\mathbb{P}_R$ and $\mathcal{C}_R$ as well as its inverse are explicitly known are when $p \not\equiv 0 \pmod 3$ and $R \in \mathbb{Z}_p$ is a cube. In particular, there are only two possible choices of parameters that satisfy this property: the one described in Section 3.2.2 where R is a cube with 3 roots and the group isomorphism is ψ_R'', and the case studied in Section 3.2.3 in which any $R \in \mathbb{Z}_p$ is a cube with 1 root and the group isomorphism is ψ_R'''.

The first case can occur only when $p \equiv 1 \pmod 3$ and $R \in \mathbb{Z}_p$ is a cube. In particular, if also $q \equiv 1 \pmod 3$ and $R \in \mathbb{Z}_q$ is a cube, then R is a cube with 9 roots in $\mathbb{Z}_N$ with $N = pq$, because of the Chinese remainder theorem. The RSA–like cryptosystem resulting from this case is described in Figure 19.

In the key generation algorithm (Gen), after steps 1-2 which take the primes p, q so that in step 3 the modulus $N = pq$ is n bits long, the cardinality of $\mathcal{C}_R$ (and $\mathbb{P}_R$) is evaluated as $\tilde{\varphi}(N) = (p-1)^2(q-1)^2$, obtained from Theorem 4.3. Then the algorithm continues as in the classical RSA.

Choosing $2n$ as maximum message length results in an easy encoding of the plaintext in a canonical representative of $\mathbb{P}_R$, but also in having no advantages with respect to the quadratic version introduced in Figure 14, as for RSA with $\mathbb{P}_R$ from Figure 10.

Thus, it is best for the encryption algorithm (Enc) to take a message of $3n$ bits and encode it into a point of $\mathbb{Z}_N^3$ in step 1. The idea is to obtain in step 2 the parameter R of the related Pell cubic from the cubic Pell equation

$$z^3R^2 + (y^3 - 3xyz)R + (x^3 - 1) = 0, \tag{6.1}$$

but evaluate the square root modulus N of the determinant is not efficient because the sender does not know the factors of N. In addition, in order to evaluate the inverse of ψ_R'', the cubic root s of R in $\mathbb{Z}_N$ is required and this is another inefficient calculation if the factors of N are unknown. In the remaining part of the algorithm, the related canonical representative in $\mathbb{P}_R$ is obtained in step 3 and the ciphertext contains its e–th power and the parameter R.

The decryption algorithm (Dec) works backwards and has no inefficient parts since, after obtaining the d–th power of the ciphertext, the receiver can easily evaluate the cubic root of R and use ψ_R'' to retrieve the point and hence the message.

In conclusion, it could be interesting to compare the obtained RSA–like cryptosystem with the classical RSA, or the RSA with $\mathbb{P}_R$, or the RSA–like cryptosystem with the Pell conic described in Figure 14. However, the problem in the encryption seems to be unsolvable.

The second case in which the group isomorphism and its inverse are explicitly known consists in having $p, q \equiv 2 \pmod 3$, so that each $R \in \mathbb{Z}_N$ is a cube with one root. The RSA–like cryptosystem obtained with this choice of parameters is described in Figure 20.

As before, in the key generation algorithm (Gen), after obtaining $N = pq$ of n bits in steps 1-3, the results in Section 3.2.3 assure that the cardinality of $\mathcal{C}_R$ (and $\mathbb{P}_R$) is $\tilde{\varphi}(N) = (p^2-1)(q^2-1)$ as evaluated in step 4. Then, the algorithm continues as in the classical RSA by taking the public exponent and its secret inverse modulo $\tilde{\varphi}(N)$.

The encryption algorithm (Enc) takes a message of $3n$ bits and encodes it into a point of $\mathbb{Z}_N^3$ in step 1. Step 2 focuses on obtaining the parameter R of the related Pell cubic from Equation 6.1. However, evaluating the square root modulo N of the determinant is still inefficient. Differently from Figure 19, the inverse of ψ_R''' in step 3 does not require to know the cubic root of R. In the end, the ciphertext contains the e–th power of the canonical representative in $\mathbb{P}_R$ related to the point and the parameter R.

Gen(n):	Enc(msg, pk):
1. $p, q \leftarrow_{\$} \{0,1\}^{n/2}$ primes	REQUIRE: $msg < N^3$
2. $p, q \equiv 2 \pmod 3$	1. $(x, y, z) \leftarrow\!\shortmid msg$
3. $N = pq$	2. $R \leftarrow\!\shortmid (x, y, z) \in \mathcal{C}_R$ (cube)
4. $\tilde{\varphi}(N) = (p^2-1)(q^2-1)$	3. $(l, m) = (\psi'''_R)^{-1}(x, y, z) \in \mathbb{P}_R$
5. $e \leftarrow_{\$} \mathbb{Z}^{\times}_{\tilde{\varphi}(N)}$	4. $c = (l, m)^{\odot_R e} \in \mathbb{P}_R$
6. $d = e^{-1} \pmod{\tilde{\varphi}(N)}$	5. return c, R
7. $sk = (p, q, d)$	Dec(c, R, pk, sk):
8. $pk = (N, e)$	1. $(l, m) = c^{\odot_R d} \in \mathbb{P}_R$
9. return pk, sk	2. $msg \leftarrow\!\shortmid \psi'''_R(l, m)$
	3. return msg

Figure 20: RSA–like cryptosystem using ψ'''_R with R cube depending on msg.

Again, the decryption algorithm (Dec) has no problematic parts since, after obtaining the d–th power of the ciphertext, the receiver can easily use ψ''_R to retrieve the point and hence the original message from its coordinates.

In conclusion, the obtained cryptosystem could be interesting when compared with the other RSA or RSA–like cryptosystems with Pell conic or cubic but, as for the previous formulation, the problem in the encryption seems to be unsolvable.

6.2 ELGAMAL WITH THE PELL CUBIC

This section addresses the formulation of the ElGamal cryptosystem with the group $(\mathcal{C}_R, \odot_R)$. As observed at the beginning of the chapter, there are two possible cases that result in two different cryptosystems, both with pros and cons.

In Figure 21, the ElGamal formulation with $q \equiv 1 \pmod 3$ and parameter $R \in \mathbb{F}_q$ non–cube is described.

The algorithm for the key generation (Gen) takes as input the bit–length n of the cardinality q of the finite field. In step 1, it is required that the remainder of q divided by 3 is 1 and, in addition, step 2 supposes that the obtained order $q^2 + q + 1$ of the cyclic group $\mathcal{C}_R$, which is clearly divisible by 3, has only another prime factor p. With this constraint, there is only one small subgroup of order 3, so that it is easily avoidable. Then, in step 3, the parameter $R \in \mathbb{F}_q$ is taken, while checking if it is not a cube by using the extended Euler criterion introduced in Equation 3.6. Since one third of the elements in $\mathbb{F}^{\times}_q$ are non–cubes, the search ends rapidly. In general, it useful to take R small, so that the computational costs of $\odot_R$ are reduced. In step 4, a generator G of $\mathcal{C}_R$ is taken randomly. Since the points of the cubic are not easy to find, it is useful to take randomly $l, m \in \mathbb{F}_q$ and exploit the parametrization $\psi'_R([l : m : 1]) = G \in \mathcal{C}_R$. It is easy to check when G is a generator since 3 and p are the only divisors of the order of $\mathcal{C}_D$.

Gen(n):	Enc(msg, pk):
1. $q \leftarrow_\$ \{0,1\}^n$, $q \equiv 1 \pmod 3$	REQUIRE: $msg < q^2$
2. $p = (q^2+q+1)/3$ prime	1. $y, z \leftarrow\!\shortmid msg$
3. $r \leftarrow_\$ \mathbb{F}_q$ non–cube	2. $x \leftarrow\!\shortmid y, z, r$
4. $G \leftarrow_\$ \mathcal{C}_r$ of order q^2+q+1	3. $r \leftarrow_\$ \{2, \ldots, q^2+q\}$
5. $sk \leftarrow_\$ \{2, \ldots, q^2+q\}$	4. $C_1 = G^{\odot_r r} \in \mathcal{C}_r$
6. $H = G^{\odot_r sk} \in \mathcal{C}_r$	5. $C_2 = H^{\odot_r r} \odot_r (x, y, z) \in \mathcal{C}_r$
7. $pk = (q, r, G, H)$	6. return C_1, C_2
8. return pk, sk	Dec(C_1, C_2, pk, sk):
	1. $(x, y, z) = (C_1^{\odot_r sk})^{-1} \odot_r C_2 \in \mathcal{C}_r$
	2. $msg \leftarrow\!\shortmid y, z$
	3. return msg

Figure 21: ElGamal with $(\mathcal{C}_r, \odot_r)$ of order q^2+q+1.

In particular, since there are $\varphi(q^2+q+1) = \varphi(3p) = 2(p-1)$ generators, excluding the trivial subgroup of order 3, there is $2/3$ of probability to take one of them at the first attempt. Then the algorithm proceeds as the classical ElGamal key generation: the secret key sk is a random exponent taken in step 5, while a public point $H \in \mathcal{C}_r$ is obtained in step 6 through the square–multiply algorithm with $\odot_r$ (introduced in Figure 4). In conclusion, the public key contains the cardinality q, the parameter $r \in \mathbb{F}_q$ and the points $G, H \in \mathcal{C}_r$.

The encryption algorithm (Enc) takes a message that needs to be encoded into a point on the cubic. However, ψ_r' is not exploitable because its inverse should be used in the decryption but it is not known explicitly. Thus, msg is taken smaller than q^2 and determines in step 1 the y and z coordinates of a point of $\mathbb{F}_q^2$. The corresponding x such that $(x, y, z) \in \mathcal{C}_r$ is then obtained in step 2. Since the computation of the roots of $x^3 - (3ryz)x + (ry^3 + r^2z^3 - 1) = 0$ is not efficient, this step needs a difficult search. In the following steps, the ciphertext consisting of two points $C_1, C_2 \in \mathcal{C}_r$ is obtained: after taking a random exponent $r < q^2+q+1$ in step 3, it is used in step 4 to obtain C_1 through the square–multiply exponentiation with $\odot_r$ and base the public generator G, while the second point C_2 is determined in step 5 as the cubic Brahmagupta product of $H^{\odot_r r}$ with the point (x, y, z) representing the message.

In the decryption algorithm (Dec), (x, y, z) is retrieved as the cubic Brahmagupta product of the inverse of $C_1^{\odot_r sk}$ with C_2 in step 1. From the obtained y and z coordinates, the original message is recovered in step 2.

From the point of view of security, since as observed in Theorem 3.1 the Pell cubic with r non–cube is isomorphic to the multiplicative subgroup $G \subset \mathbb{F}_{q^3}^\times$ of order q^2+q+1, the DLP in $\mathcal{C}_r$ can be reduced to the DLP in G. Thus, the obtained cryptosystem is an alternative formalization for that with G.

Gen(n):	Enc(msg, pk):
1. $q \leftarrow_{\$} \{0,1\}^n$, $q \equiv 2 \pmod 3$	REQUIRE: $msg < q^2$
2. $R \leftarrow_{\$} \mathbb{F}_q$ (cube)	1. $l, m \mapsto msg$
3. $G \leftarrow_{\$} \mathcal{C}_R$ of order q^2-1	2. $(x,y,z) = \psi_R'''(l,m)$
4. $sk \leftarrow_{\$} \{2,\dots,q^2-2\}$	3. $r \leftarrow_{\$} \{2,\dots,q^2-2\}$
5. $H = G^{\odot_R sk} \in \mathcal{C}_R$	4. $C_1 = G^{\odot_R r} \in \mathcal{C}_R$
6. $pk = (q, R, G, H)$	5. $C_2 = H^{\odot_R r} \odot_R (x,y,z) \in \mathcal{C}_R$
7. return pk, sk	6. return C_1, C_2
	Dec(C_1, C_2, pk, sk):
	1. $(x,y,z) = (C_1^{\odot_R sk})^{-1} \odot_R C_2 \in \mathcal{C}_R$
	2. $msg \mapsto (\psi_R''')^{-1}(x,y,z)$
	3. return msg

Figure 22: ElGamal with $(\mathcal{C}_R, \odot_R)$ of order q^2-1.

The other case with $q \equiv 2 \pmod 3$ and parameter $R \in \mathbb{F}_q$ cube is described in Figure 22.

As in the previous cryptosystem, the algorithm for the key generation (Gen) takes as input the bit–length n of the cardinality q of the finite field. Step 1 takes the required q, checking that it gives remainder 2 when divided by 3. Differently from before, it is not possible to have an order with only two prime factors. Indeed, the cyclic group has $q^2-1 = (q-1)(q+1)$ elements, where $3 \mid q+1$ and $8 \mid (q-1)(q+1)$. The best choice should be a prime q such that $q-1 = 2p_1$ and $p_1+1 = 6p_2$ with p_1, p_2 primes, so that

$$q^2-1 = (q-1)(q+1) = 2p_1(2p_1+2) = 24p_1p_2,$$

but such a q is quite rare: among the primes up to 2^{20}, only 312 satisfy this property. Thus, for simplicity of formulation and implementation, no other constraints on the choice of q are added. In step 2, the parameter $R \in \mathbb{F}_q$ is taken among all the elements of $\mathbb{F}_q$ since they are all cubes. In general, it useful to take R small, so that the computational costs of $\odot_R$ are reduced. In step 3, a generator G of $\mathcal{C}_R$ is taken randomly. Again the parametrization ψ_R''' is useful to find easily a point on the cubic given two values $l, m \in \mathbb{F}_q$. Differently from the previous case, there is no assurance that the obtained point is a generator without knowing a factorization of the order q^2-1, but at least checking that $G^{\odot_R (q^2-1)/3}$ and $G^{\odot_R (q^2-1)/2}$ are not the identity point $(1,0,0)$ allows to exclude small subgroups. Then the algorithm proceeds as the classical ElGamal key generation, i.e., in step 4 a random exponent is taken as secret key sk, while a public point $H \in \mathcal{C}_R$ is obtained in step 5 as the exponentiation of G to sk through the square–multiply algorithm with $\odot_R$. In conclusion, the public key contains the cardinality q, the parameter $R \in \mathbb{F}_q$ and the points $G, H \in \mathcal{C}_R$.

The encryption algorithm (Enc) takes a message $msg < q^2$ and encodes it into a point on the cubic. In order to do so, ψ_R''' can be exploited since its inverse is explicitly known. Thus, a pair $l, m \in \mathbb{F}_q$ is obtained from msg in step 1 and $(x, y, z) = \psi_R'''([l : m : 1]) \in \mathcal{C}_R$ is evaluated in step 2. As observed in Equation 3.5, there are $q + 1$ invalid pairs but the probability to obtain one of them is about $1/q$, which is quite low for large q. As in all ElGamal formulations, the ciphertext consists of two elements of the group, i.e., $C_1, C_2 \in \mathcal{C}_R$. They are obtained from a random exponent $r < q^2 - 1$ chosen in step 3, which is then used in step 4 to obtain C_1 through the square–multiply exponentiation with $\odot_R$ and base the public generator G. The second point C_2 is determined in step 5 as the cubic Brahmagupta product of $H^{\odot_R r}$ with the point (x, y, z) representing the message.

In the decryption algorithm (Dec), (x, y, z) is retrieved as the cubic Brahmagupta product of the inverse of $C_1^{\odot_R sk}$ with C_2 in step 1. From the obtained point (x, y, z), the original message is recovered from the element in the projectivization returned by the inverse of the group isomorphism ψ_R'''.

From the point of view of security, the DLP over $\mathcal{C}_R$ with R cube with one root over $\mathbb{F}_q$ can be reduced thanks to the explicit group isomorphism in Theorem 3.5 and ψ_R''' to the DLP in $\mathbb{F}_{q^2}^\times$.

The assessments concerning data–size and computational costs for both the cryptosystems introduced in this section are tackled in Section 6.6 in comparison with the other proposals.

6.3 ELGAMAL WITH THE PROJECTIVIZATION

This section addresses the ElGamal formulation obtained by exploiting the cyclic group $(\mathbb{P}_R, \odot_R)$, i.e., the projectivization related to the Pell cubic. As for the previous scenario, there are only two possible formulations depending on the choice of q and R.

Figure 23 describes the cryptosystem resulting from the case with $q \equiv 1 \pmod 3$ and $R \in \mathbb{F}_q$ non–cube.

The algorithm for the key generation (Gen) is very similar to the one in Figure 21. The input is an integer n representing the bit–length of the cardinality q of the finite field. This value is obtained as before in step 1 such that the remainder of its division by 3 is 1. Again, the order $q^2 + q + 1$ is a multiple of 3, so that it is best if it has only another prime factor p, as checked in step 2. As before, the parameter $R \in \mathbb{F}_q$ is taken in step 3 among the non–cubes by exploiting the generalized Euler criterion from Equation 3.6. The search ends rapidly because one third of the elements in $\mathbb{F}_q^\times$ are non–cubes and, generally, it is useful to take R small in order to decrease the computational costs of the product $\odot_R$ over $\mathbb{P}_R$. In step 4, a random generator $g \in \mathbb{P}_R$ is taken among the $\varphi(q^2 + q + 1) = \varphi(3p) = 2(p - 1)$ generators of the cyclic group. Thus, there is $2/3$ of probability to find it at the first attempt.

Gen(n):	Enc(msg, pk):
1. $q \leftarrow_\$ \{0,1\}^n$, $q \equiv 1 \pmod 3$	REQUIRE: $msg < q^2+q+1$
2. $p = (q^2+q+1)/3$ prime	1. $(l, m) \leftarrow\!\shortmid msg$
3. $r \leftarrow_\$ \mathbb{F}_q$ non–cube	2. $r \leftarrow_\$ \{2, \ldots, q^2+q\}$
4. $g \leftarrow_\$ \mathbb{P}_r$ of order q^2+q+1	3. $c_1 = g^{\odot_r r} \in \mathbb{P}_r$
5. $sk \leftarrow_\$ \{2, \ldots, q^2+q\}$	4. $c_2 = h^{\odot_r r} \odot_r (l, m) \in \mathbb{P}_r$
6. $h = g^{\odot_r sk} \in \mathbb{P}_r$	5. return c_1, c_2
7. $pk = (q, r, g, h)$	Dec(c_1, c_2, pk, sk):
8. return pk, sk	1. $(l, m) = (c_1^{\odot_r sk})^{-1} \odot_r c_2 \in \mathbb{P}_r$
	2. $msg \leftarrow\!\shortmid (l, m)$
	3. return msg

Figure 23: ElGamal with $(\mathbb{P}_r, \odot_r)$ of order q^2+q+1.

Then, in step 5, a random exponent is taken as the secret key sk. The public key, instead, consists of the cardinality q, the parameter $r \in \mathbb{F}_q$, the chosen generator $g \in \mathbb{P}_r$ and h that is the power of g to sk with $\odot_r$ (step 6), that can be implemented using the modified More algorithm introduced in Figure 5.

The encryption algorithm (Enc) takes a message msg smaller than q^2+q+1 which is used in step 1 to determine the coordinates l, m of the canonical representative of an element in $\mathbb{P}_r$. In step 2, an exponent $r < q^2+q+1$ is chosen randomly and is then used to obtain the ciphertext in steps 3-4. This consists of two canonical representatives $c_1, c_2 \in \mathbb{P}_r$ given by the power of g to r with $\odot_r$ and the Brahmagupta product of $h^{\odot_r r}$ with (l, m), respectively.

During the decryption algorithm (Dec), the canonical representative (l, m) related to the message is retrieved as the cubic Brahmagupta product of the inverse of $c_1^{\odot_r sk}$ with c_2 in step 1. Finally, in step 2, the original message is recovered from (l, m).

All exponentiations can be obtained exploiting the modified More algorithm introduced in Figure 5.

The security of this cryptosystem relies on the DLP in $\mathbb{P}_r$, which can be reduced to the DLP on $\mathbb{F}_{q^3}^\times/\mathbb{F}_q^\times$ because of the group isomorphism observed in Section 3.2.1. This is again isomorphic to the multiplicative subgroup $G \subset \mathbb{F}_{q^3}^\times$ of order q^2+q+1. In addition to the easier way to encode the message, which allows to encrypt also longer messages, the advantage with respect to the formulation in Figure 21 is that the elements require 2/3 of the size to be stored, as will be pointed out in Section 6.6 where comparisons of data–size and performance among the different formulations are addressed.

As observed in Section 3.2.3, $\mathbb{P}_r$ is cyclic also for $q \equiv 2 \pmod 3$ and $r \in \mathbb{F}_q$ cube, and Figure 24 describes the resulting cryptosystem.

The algorithm for the key generation (Gen) takes as input the bit–length n of the cardinality q of the finite field, which is obtained in step 1 while checking that it gives remainder 2 when divided by 3.

$\underline{\texttt{Gen}}(n, k)$:	$\underline{\texttt{Enc}}(msg, pk)$:
1. $q \leftarrow_\$ \{0,1\}^n$, $q \equiv 2 \pmod 3$	REQUIRE: $msg < q^2 - 1$
2. $r \leftarrow_\$ \mathbb{F}_q$ (cube)	1. $(l, m) \leftarrow\!\shortmid msg$
3. $g \leftarrow_\$ \mathbb{P}_R$ of order $q^2 - 1$	2. $r \leftarrow_\$ \{2, \dots, q^2 - 2\}$
4. $sk \leftarrow_\$ \{2, \dots, q^2 - 2\}$	3. $c_1 = g^{\odot_R r} \in \mathbb{P}_R$
5. $h = g^{\odot_R sk} \in \mathbb{P}_R$	4. $c_2 = h^{\odot_R r} \odot_R (l, m) \in \mathbb{P}_R$
6. $pk = (q, r, g, h)$	5. return c_1, c_2
7. return pk, sk	$\underline{\texttt{Dec}}(c_1, c_2, pk, sk)$:
	1. $(l, m) = (c_1^{\odot_R sk})^{-1} \odot_R c_2 \in \mathbb{P}_R$
	2. $msg \leftarrow\!\shortmid (l, m)$
	3. return msg

Figure 24: ElGamal with $(\mathbb{P}_R, \odot_R)$ of order $q^2 - 1$.

As observed for the case with the cubic, the cyclic group has order $q^2 - 1 = (q-1)(q+1)$, where $3 \mid q+1$ and $8 \mid (q-1)(q+1)$, but there are other prime factors depending on the value of q and having the best factorization means asking too many constraints. In step 2, the parameter $r \in \mathbb{F}_q$ is taken among all the elements of $\mathbb{F}_q$ which are all cubes. In general, it useful to take r small, so that the computational costs of $\odot_R$ are reduced. In step 3, a generator g of $\mathbb{P}_R$ is taken randomly. As with the Pell cubic, without a factorization of $q^2 - 1$ there is no assurance that the obtained element is a generator, but checking that $g^{\odot_R (q^2-1)/3}$ and $g^{\odot_R (q^2-1)/2}$ are not (α, α) allows to exclude small subgroups. From there, the algorithm is an implementation of the classical ElGamal key generation, since in step 4 a random exponent is taken as secret key sk, while a public point $H \in \mathbb{P}_R$ is obtained in step 5 as the exponentiation of G to sk. Here the modified More algorithm for the cubic case can be exploited. In conclusion, the public key contains the cardinality q, the parameter $r \in \mathbb{F}_q$ and $g, h \in \mathbb{P}_R$.

The encryption algorithm (Enc) takes a message msg smaller than q^2 and encodes it into the canonical representative of an element in $\mathbb{P}_R$. The element $(l, m) \in \mathbb{F}_q^2$ obtained in step 1 should not be one of the $q + 1$ invalid pairs described in Equation 3.5, but the probability to obtain one of them is about $1/q$, which is quite low for large q. As always, the ciphertext in an ElGamal cryptosystem consists of two elements of the cyclic group, i.e., $c_1, c_2 \in \mathbb{P}_R$. They are obtained from a random exponent $r < q^2 - 1$ chosen in step 2, which is then used in step 3 to obtain c_1 through the cubic modified More algorithm for the exponentiation with $\odot_R$ with base the public generator g. The second point c_2 is determined in step 4 as the cubic Brahmagupta product of $h^{\odot_R r}$ with the element (l, m) representing the message.

During the decryption algorithm (Dec), the element (l, m) is retrieved as the cubic Brahmagupta product of the inverse of $c_1^{\odot_R sk}$ with c_2 in step 1. From the obtained pair, in step 2 the original message is recovered.

Gen(n):	Enc(msg, pk):
1. $q \leftarrow_\$ \{0,1\}^n$, $q \equiv 2 \pmod 3$	REQUIRE: $msg < q^3$
2. $R \leftarrow_\$ \mathbb{F}_q$ small cube	1. $(x, y, z) \leftarrow\!\shortmid msg$
3. $g \leftarrow_\$ \mathbb{P}_R$ of order $q^2 - 1$	2. $R' \leftarrow\!\shortmid (x, y, z) \in \mathcal{C}_{R'}$
4. $sk \leftarrow_\$ \{2, \ldots, q^2 - 2\}$	3. $(l, m) = (\psi'''_{R'})^{-1}(x, y, z) \in \mathbb{P}_{R'}$
5. $h = g^{\odot_R sk} \in \mathbb{P}_R$	4. $r \leftarrow_\$ \{2, \ldots, q^2 - 2\}$
6. $pk = (q, R, g, h)$	5. $\rho = \sqrt[3]{R/R'} \in \mathbb{F}_q$
7. return pk, sk	6. $c_1 = \rho_{R,R'}(g)^{\odot_{R'} r} \in \mathbb{P}_{R'}$
	7. $c_2 = \rho_{R,R'}(h)^{\odot_{R'} r} \odot_{R'} (l, m) \in \mathbb{P}_{R'}$
	8. return c_1, c_2, R'
	Dec(c_1, c_2, R', pk, sk):
	1. $(l, m) = (c_1^{\odot_{R'} sk})^{-1} \odot_{R'} c_2 \in \mathbb{P}_{R'}$
	2. $msg \leftarrow\!\shortmid \psi'''_{R'}(l, m)$
	3. return msg

Figure 25: ElGamal with $(\mathcal{C}_R, \odot_R)$ of order $q^2 - 1$, ψ'''_R and $\rho_{R,R'}$.

Again, the modified More algorithm introduced in Figure 5 is useful to efficiently obtain the required exponentiations.

From the point of view of security, the DLP over $\mathbb{P}_R$ with R cube with one root over $\mathbb{F}_q$ can be reduced to the DLP over $\mathbb{F}_{q^2}^\times$ because of the group isomorphism obtained in Theorem 3.5. Thus, it can be compared with the second cryptosystem introduced in the previous section, which is very similar but requires more computations (because of the use of ψ'''_R) and has higher data–size since canonical representatives of $\mathbb{P}_R$ require only a pair of values instead of 3.

6.4 ELGAMAL WITH TWO PELL CUBICS

The last formulation of the ElGamal cryptosystem with the Pell cubic wants to reproduce the alternative version described for the quadratic case in Figure 17. The idea is to remove the constraint on the message given by the fixed Pell cubic by exploiting both the group isomorphism with the projectivization and that with another cubic with different parameter R. The only case in which these two maps and their inverses are explicitly described is when $q \equiv 2 \pmod 3$ and $R \in \mathbb{F}_q$ is a cube. The obtained cryptosystem is described in Figure 25.

The key generation algorithm (Gen) is practically the same of the cryptosystem with the projectivization of order $q^2 - 1$: it takes as input the bit–length n of the cardinality q of the finite field. In step 1, the required q is chosen such that it is congruent to 2 modulo 3. The cyclic group has order $q^2 - 1$ with $3 \mid q + 1$ and $8 \mid q^2 - 1$ but, as observed in the other analogous cases, it is difficult to have the best factorization, i.e., $24p_1p_2$, so that no other constraints are considered. In step 3, a small cube $R \in \mathbb{F}_q$ is chosen randomly. This allows to reduce the computational costs of $\odot_R$ when obtaining the public key.

Specifically when, after taking a generator $g \in \mathbb{P}_R$ in step 4 and a secret exponent $sk < q^2 - 1$ in step 5, $h = g^{\odot_R sk} \in \mathbb{P}_R$ is obtained in step 6. In the end, the public key contains the cardinality q and the elements g, h, while the secret key is simply the used exponent sk.

The encryption algorithm (`Enc`) is very different from that of the classical ElGamal. With respect to the previous algorithms with cyclic groups of the same order, the maximum length of the message can be increased because in step 1 msg is encoded into the coordinates of a point $(x, y, z) \in \mathbb{F}_q^3$. From this point, step 2 searches for a cube $R' \in \mathbb{F}_q$ such that $(x, y, z) \in \mathcal{C}_{R'}$. This can be obtained by solving the quadratic equation obtained from the constraint given by the unitary norm. If necessary, some of the bits of x can be kept variable so that such a R' can be found. Then, in step 3, the canonical representative (l, m) related to the point is obtained through the inverse of parametrization $\psi'''_{R'}$. In step 4 a random exponent $r < q^2 - 1$ is chosen. Now, since the public key contains parameters of points on $\mathcal{C}_R$, the isomorphism $\rho_{R,R'}$ is required. Thus, step 5 evaluates the coefficient ρ such that $R = \rho^3 R'$, which is easy to find using the inverse of $3 \in \mathbb{Z}_q^\times$ that exists since $\gcd(3, q - 1) = 1$. The found value is used in step 6 to obtain the base $\rho_{R,R'}(g) \in \mathbb{P}_{R'}$ that raised to r gives c_1 and in step 7 for the base $\rho_{R,R'}(h) \in \mathbb{P}_{R'}$ that raised to r and multiplied by (l, m) gives c_2. The ciphertext contains $c_1, c_2 \in \mathbb{P}_{R'}$ and the parameter R' used in the calculations.

In the decryption algorithm (`Dec`), after evaluating in step 1 (l, m) as the product between the inverse of $c_1^{\odot_{R'} sk}$ and c_2, the message is retrieved in step 2 using the point of $\mathcal{C}_{R'}$ obtained $\psi'''_{R'}$.

As for the previous ElGamal formulations a key-recovery attack should solve the DLP on $\mathbb{P}_R$, while the security against message recovery is based on the DLP on $\mathbb{P}_{R'}$. However, they can both be reduced to the DLP over $\mathbb{F}_{q^2}^\times$.

The advantages in adopting this version are the data–size reduced to 2/3 with respect to the cryptosystems on the Pell cubic as well as the increased length of the message without doubling the computational costs.

6.5 DSA WITH THE PELL CUBIC

As for the classical ElGamal and with the Pell conic, when the Pell cubic or the related projectivization are cyclic groups they can be exploited to formulate alternative DSA. In particular, between the two possible cases in which the groups are cyclic, the non–trivial choice of parameters, i.e., $q \equiv 1 \pmod 3$ and $R \in \mathbb{F}_q$ non–cube is preferable. Since signature schemes require smaller data–size and working with $(\mathcal{C}_R, \odot_R)$ requires larger data and comparable times with respect to $(\mathbb{P}_R, \odot_R)$, only the formulation with the projectivization is introduced. The obtained DSA is described in Figure 26.

Gen(l):	Sig(msg, pk, sk):
1. $q \leftarrow_{\$} \{0,1\}^l$, $q \equiv 1 \pmod 3$	1. $r \leftarrow_{\$} \{2,\dots,p-1\}$
2. $p = (q^2+q+1)/3$ prime	2. $(l,m) = g^{\odot_R r} \in \mathbb{P}_R$
3. $R \leftarrow_{\$} \mathbb{F}_q$ non–cube	3. $s_1 = l \pmod p$
4. $g \leftarrow_{\$} \mathbb{P}_R$ of order p	4. $s_2 = r^{-1}(H(msg) + sk \cdot s_1) \pmod p$
5. $sk \leftarrow_{\$} \{2,\dots,p-1\}$	5. return s_1, s_2
6. $y = g^{\odot_R sk} \in \mathbb{P}_R$	Ver(msg, s_1, s_2, pk):
7. $pk = (q, R, g, y)$	1. $u_1 = H(msg) \cdot s_2^{-1} \pmod p$
8. return pk, sk	2. $u_2 = s_1 \cdot s_2^{-1} \pmod p$
	3. $(l,m) = g^{\odot_R u_1} \odot_R y^{\odot_R u_2} \in \mathbb{P}_R$
	4. return $s_1 = l \pmod p$?

Figure 26: DSA with $(\mathbb{P}_R, \odot_R)$ of order $q^2+q+1 = 3p$.

The key generation algorithm (Gen) takes as input the size l (adopting the same notation of the other DSA) of the cardinality q of the finite field. In step 1, q is taken such that $q \equiv 1 \pmod 3$. Differently from the formulation in FFC or with the Pell conic where it is easy to obtain the order p of the adopted subgroup with fixed length of n bits, this is very difficult in this case. The most efficient option is taking $p = (q^2+q+1)/3$, as in step 2, but the resulting p is $2l$ bits long and, as will be observed in the following section, this choice gives large signatures and increases the computational costs. After choosing in step 3 a non–cube parameter $R \in \mathbb{F}_q$, the following steps the algorithm works as in the classical DSA: a generator g of order p is taken in step 4, as well as a secret exponent $sk < p$ in step 5, and they give the actual public key $h = g^{\odot_R sk}$. The values p, q, R and the generator $g \in \mathbb{P}_R$ are parameters of the system that can be shared among different users.

The signature algorithm (Sig) is analogous to the one for ECDSA: a random exponent r is taken in step 1 and then used to obtain a element $(l,m) \in \mathbb{P}_R$ from the public generator g. The first part of the signature s_1 is the coordinate $l \pmod p$ (despite the reduction is not required since $p > q$), while the second part is obtained exactly as in DSA and ECDSA.

For the verification through Ver of a signature on the message msg, the algorithm obtains $u_1, u_2 \in \mathbb{Z}_p$ as in the classical schemes. Then they are used to check the validity of (s_1, s_2) by verifying that

$$\begin{aligned}(l,m) &= g^{\odot_R u_1} \odot_R y^{\odot_R u_2} \\ &= g^{\odot_R H(msg)\cdot s_2^{-1} \pmod p} \odot_R (g^{\odot_R sk})^{\odot_R s_1 \cdot s_2^{-1} \pmod p} \\ &= g^{\odot_R s_2^{-1}(H(msg)+sk\cdot s_1) \pmod p} \\ &= g^{\odot_R s_2^{-1}\cdot r \cdot s_2 \pmod p} \equiv (s_1, m) \pmod p .\end{aligned}$$

Cryptosystem	pk	sk	msg	c
RSA	1024	1024	1024	1024
RSA–like $\mathcal{C}_D$	1024	3072	2048	2048
RSA–like $\mathcal{C}_R$	1024	3072	3072	3072

Table 8: Data–size in bits of the classical RSA and the RSA–like cryptosystems with $\mathcal{C}_D$ and $\mathcal{C}_R$ for 80 bits of security strength.

6.6 SECURITY, DATA–SIZE AND PERFORMANCE

As for the quadratic case, the proposed cryptosystems are studied in terms of security, data–size and performance.

All the cryptosystems with security based on the IFP share the same security requirements as the classical RSA. Thus, when comparing the size of the data required in each cryptosystem the public key can have the same size for fixed security level. This is described in Table 8, which compares the new RSA–like cryptosystems based on the Pell cubic, introduced in Figure 19 and Figure 20, with the classical RSA and the RSA–like cryptosystem with $\mathcal{C}_D$ from Section 5.1. Potentially, the versions that adopt the Pell cubic have tripled maximum message length, so that should be compared with three runs of the classical RSA. However, as observed in Section 6.1, it is impossible to conduct a performance analysis since the two proposals based on the Pell cubic miss an efficient implementation. This is because both the encryption algorithms require to obtain the parameter r for the Pell cubic that contains the point of $\mathbb{F}_q^3$ in which the message is encoded, and this requires to evaluate a square root modulo N with unknown factorization. In alternative it is possible to reduce the maximum message length to the size adopted for the quadratic case, but this results obviously in higher computational costs with respect to the RSA–like cryptosystem with $\mathcal{C}_D$ from Section 5.1.

On the other hand, a comparison of the DLP–based cryptosystems with the cyclic groups related to the Pell cubic with the classical versions and with those with the Pell conic could be very interesting.

Firstly, it is important to notice that the size of the cardinality of the underlying finite fields depends on the considered case:

- if $q \equiv 1 \pmod 3$ and $r \in \mathbb{F}_q$ is a non–cube, then the cyclic groups are isomorphic to the multiplicative subgroup $G \subset \mathbb{F}_{q^3}^\times$ of order $q^2 + q + 1$. Thus, the bit–length of q can be $1/3$ of that in FFC;

- if $q \equiv 2 \pmod 3$ and $r \in \mathbb{F}_q$ is a cube, then the cyclic groups are isomorphic to the $\mathbb{F}_{q^2}^\times$ of order $q^2 - 1$ and the bit–length of q is half of that in FFC. In this case, the field size is the same as in the formulations based on the Pell conic.

Sec.	FFC	ECC	$\mathcal{C}_R \cong G$	$\mathcal{C}_R \cong \mathbb{F}_{q^2}^\times$
80	1024	160	341	512
112	2048	224	682	1024
128	3072	256	1024	1536
192	7680	384	2560	3840
256	15360	512	5120	7680

Table 9: Field size in bits for different DLP–based cryptosystems depending on the cyclic group and the classical security strength in bits.

The resulting sizes of q for the classical cryptosystems and the new formulations, depending on the standard security strengths, are collected in Table 9. Despite the reduction with respect to the size of FFC, ECC still maintains the smallest size. Given these security levels, it is possible to compare the proposed cryptosystems with the classical ones in terms of data–size and performance.

In the following analysis, some formulations with the cyclic groups related to the Pell cubic are not considered, in particular:

- when the order is $q^2 + q + 1$, i.e., the cryptosystem with $\mathcal{C}_R$ in Figure 21, in the encryption algorithm, requires to encode the message into a point on the Pell cubic and an efficient encoding is unknown;
- when the order is $q^2 - 1$, i.e., for the cryptosystem with $\mathcal{C}_R$ in Figure 22, the message can be encoded into a point by exploiting the group isomorphism ψ_R''', but working directly on the projectivization as in Figure 24 is more efficient;
- in the same context above, since $\mathbb{P}_R$ is isomorphic to $\mathbb{F}_{q^2}^\times$, the cryptosystem in Figure 24 has the same security of the ones with the Pell conic, which however have clearly lower computational costs.

Thus, the only ElGamal formulations considered in the analysis of data–size and performance are the one in Figure 23 with the group $(\mathbb{P}_R, \odot_R) \cong G \subset \mathbb{F}_{q^3}^\times$ of order $q^2 + q + 1$ and the alternative formulation in Figure 25 that exploits the group isomorphisms ψ_R''' and $\rho_{R,R'}$.

The first study concerns the size of the data involved in the different ElGamal formulations. In Table 10, the public key is divided in the public parameters that can be used by different users, i.e., the data required for the description of the cyclic group and one of its generators, and the element h or the point H representing the actual public key. The other columns contain, from left to right, the size of the secret key, the maximum message length and the bit–length of the ciphertext.

Formulation	par	pk	sk	msg	c_1, c_2
FFC	$2n$	n	n	n	$2n$
	2048	1024	1024	1024	2048
ECC	$6n$	$2n$	n	n	$4n$
	960	320	160	160	640
ϕ_D, $\delta_{D,D'}$	$2n$	n	n	$2n$	$3n$
	1024	512	512	1024	1536
$\mathbb{P}_R$	$4n$	$2n$	$2n$	$2n$	$4n$
	1364	682	682	682	1364
ψ_R''', $\rho_{R,R'}$	$3n$	$2n$	$2n$	$3n$	$5n$
	1536	1024	1024	1536	2560

Table 10: Data–size in bits for ElGamal with FFC, ECC, $\mathbb{P}_R$ and the alternative formulations with the Pell conic and cubic, depending on the size n of q and for 80 bits of security strength.

For sake of completeness, in addition to the classical versions in FFC and ECC, there are also the results related to the best cryptosystem with the Pell conic, namely the alternative formulation from Figure 17. These formulations are compared with those based on the Pell cubic from Figure 23 and Figure 25, in rows 4 and 5, respectively. For each formulation, the table shows the data–size depending on the size n of the cardinality q of the adopted finite field, as described in Table 9, and the values for 80 bits of security strength.

The formulation with $\mathbb{P}_R$ in the fourth row has size of q taken from the fourth column in Table 9. Despite this is one third of the size in FFC, the ratio between the size of parameters or ciphertext and the other data is still 2, so that the two formulations are comparable.

The results in the last row, corresponding to the alternative formulation with the Pell cubic, are similar to those for the analogous formulation in the quadratic case, collected in the third row. In particular, while for the former cryptosystem the parameters and the message are shorter in relation to the length of the keys, i.e., the ratio is $3/2$ instead of 2, the quadratic formulation is better for the ciphertext length with respect to the message size since the ratio is $3/2$ that is smaller than $5/3$. Thus, the alternative formulation that exploits the group isomorphisms ϕ_D and $\delta_{D,D'}$ from Figure 17 is still the best option in terms of information encrypted, while ECC maintains the smallest parameters and keys.

Finally, the new ElGamal formulations are compared in terms of performance with the classical versions and the best for the quadratic case. The study collects the elapsed times of a simple implementation in Python of each of the algorithms run on the cluster of the DISMA at Politecnico of Turin, on a single CPU with 46G of RAM allocated.

Sec.	Alg.	FFC $\times 3$	ECC $\times k$	$\phi_D, \delta_{D,D'} \times 3$	$\mathbb{P}_R \times 5$	$\psi_R''', \rho_{R,R'} \times 2$
80	Gen	0.011079	0.028271	0.007524	0.014578	0.027371
	Enc	0.066933	1.124020	0.084456	0.142804	0.208351
	Dec	0.036549	0.555804	0.030609	0.073841	0.083358
112	Gen	0.074718	0.056586	0.038527	0.075820	0.150707
	Enc	0.448200	3.344770	0.492367	0.757254	1.278954
	Dec	0.232865	1.590024	0.171319	0.385625	0.491019
128	Gen	0.233983	0.075437	0.112873	0.231408	0.445202
	Enc	1.403191	5.454558	1.489797	2.309788	3.933183
	Dec	0.718288	2.711129	0.513570	1.169162	1.486196
192	Gen	3.188959	0.185410	1.372381	2.799048	5.399148
	Enc	19.117265	22.362310	18.873775	27.945614	50.413716
	Dec	9.654056	11.154742	6.311260	14.064906	18.491061
256	Gen	22.874051	0.365562	9.519104	18.526002	36.211894
	Enc	137.300863	65.423245	127.975524	184.850919	337.483436
	Dec	68.943929	32.530443	43.394834	92.752310	124.280203

Table 11: Average times in seconds for 10 random instances of ElGamal with FFC, ECC, $\mathbb{P}_R$ and the alternative formulations with the Pell conic and cubic, for fixed message length, depending on the security strength.

For each case, Table 11 shows the average elapsed times of 10 randomly generated instances, whose times are taken as the minimum of 10 identical runs. In order to have a fair comparison, besides the same security strength, the maximum message length is fixed. Since the biggest size is the one for the alternative formulation in Figure 25 but it is not a multiple of the others, the encryption and decryption algorithms in the last column are run twice, so that the other cryptosystems can encrypt messages of similar size with different quantities of multiple runs. In particular, following the analysis on the data–size, this means repeating encryption and decryption 3 times for FFC, k times for ECC, where k is the ratio between the message length for the last column and ECC (e.g., for 80 bits of security, $3072/160 = 19.2$ so that $k = 20$), 3 times for the alternative formulation in the quadratic case and 2 times when using directly $\mathbb{P}_R$. When working with a finite field $\mathbb{F}_q$, the case with q prime is considered, and the bit–length n of q depends on the standard security strengths, assuming the values obtained in Table 9.

Since the parameters for the description of the cyclic group and one of its generators can be precomputed and shared among different users, the times for the key generation algorithm do not take into account these steps. In particular, this means excluding steps 1-4 in Figure 23 and steps 1-3 in Figure 25.

The times required for the single exponentiation in the key generation scale very well for ECC, starting as the worst and then passing to second and first place in the following levels of security. The other formulations maintain generally the same order: the case based on the Pell conic in the third column is the fastest, followed by the classical FFC and the new formulation with $\mathbb{P}_R$ that are comparable, and finally by the alternative version in the last column.

In addition to the considerations about the encryption and decryption algorithms from Section 5.6, the formulations based on the Pell cubic are among the slowest. In both algorithms, for the first three levels of security, ElGamal with $\mathbb{P}_R$ is third while the alternative formulation in the last column is fourth. In the two higher levels, they maintain the same order but are at the fourth and fifth place, respectively, because ECC is less slowed down by the required repetitions.

In conclusion, the formulations based on the Pell cubic are not comparable with the other ElGamal cryptosystems. However, they remain interesting mainly from the theoretical point of view, especially because finding the missing explicit inverse of the group isomorphisms could result in useful enhancements.

In the last part of this section, the new formulation of DSA with the projectivization related to the Pell cubic with parameter R non–cube introduced in Figure 26 is compared with the classical DSA with FFC and ECDSA, and also with the formulation with $\mathbb{P}_D$ from Figure 18.

From the point of view of security, since the adopted cyclic group is the same used in the ElGamal formulation, the results are still those obtained in the fourth column of Table 9. In particular, the size of the cardinality q of the finite field is one third of the size l of q for FFC, but ECC still works with smaller fields. In addition, the fifth column can be related also to the size of the finite field exploited with the Pell conic. These results allow to obtain balanced comparisons among the different DSA formulations.

Firstly, looking at the size of the involved data, Table 12 extends Table 6 with the results for the formulation with $\mathbb{P}_R$. In particular, the table shows the size in bits of the involved data depending on the pairs (l, n) representing the size of q and p, respectively, taken among the standard values $(1024, 160)$, $(2048, 224)$, $(3072, 256)$, $(7680, 384)$ and $(15360, 512)$, which correspond to the standard levels of security strength.

In addition, each row contains the values for 80 bits of security strength, which corresponds to $l = 1024$ and $n = 160$. As observed in Section 6.5, when working with the Pell cubic it is difficult to obtain an order p for the adopted subgroup with fixed length of n bits, and the most efficient option is taking $p = (q^2 + q + 1)/3$. With this choice, p is $2l$ bits long so that, differently from the first and third row, the values depend only on the size l of q.

Formulation	par	pk	sk	s_1, s_2
FFC	$2l+n$	l	n	$2n$
	2208	1024	160	320
ECC	$6n$	$2n$	n	$2n$
	960	320	160	320
$\mathbb{P}_D$	$3l/2+n$	$l/2$	n	$2n$
	1696	512	160	320
$\mathbb{P}_R$	$4l/3$	$2l/3$	$2l/3$	l
	1364	682	682	1024

Table 12: Data–size in bits for DSA with FFC, ECC, $\mathbb{P}_D$ and $\mathbb{P}_R$, depending on the sizes l, n of q, p and for 80 bits of security strength.

As can be easily observed in the last row of the table, in particular from the values for 80 bits of security strength, the resulting sizes for public parameters and key are better than in the first row and comparable to those in the third row. However, the secret key and the signature are larger than in all the other formulations because of the larger p while, in general, ECDSA in the second row maintains the smallest sizes. Thus, with a good method for finding p of the right size, i.e., from the standard value for the security levels, it is possible to obtain a digital signature scheme with competitive data–size.

Finally, Table 13 extends Table 7 for an analysis on the performance of the DSA formulations. The shown values are the elapsed times of simple Python implementations of the algorithms for the different DSA formulations run on the cluster of the DISMA at Politecnico of Turin, on a single CPU with 46G of RAM allocated. For each level of security strength and formulation of DSA, Table 13 shows the average times of 10 randomly generated instances, whose times are taken as the minimum of 10 identical runs.

Also from this point of view, the formulation based on the Pell cubic in the last column is not comparable with the classical DSA and ECDSA or with the formulation with $\mathbb{P}_D$: despite the results in the first row are better than those for ECC, when increasing the security level ElGamal with $\mathbb{P}_R$ scales very badly, requiring generally times of one order of magnitude larger than the others.

This is again due to the choice of p since it is the order of the used cyclic group. By taking $p = (q^2 + q + 1)/3$, which is $2l$ bits long, all exponentiations required in the key generation, signature and verification algorithms are very slow because the exponent is in $\mathbb{Z}_p$. Thus, not only data are larger than in the other formulation, but also performance is highly affected and the results in Table 13 are witnesses of this behaviour.

Sec.	Alg.	FFC	ECC	$\mathbb{P}_D$	$\mathbb{P}_R$
80	Gen	0.002201	0.028173	0.003310	0.014103
	Sig	0.002349	0.028242	0.003514	0.014740
	Ver	0.004529	0.055046	0.007245	0.029093
112	Gen	0.010515	0.056601	0.013278	0.075435
	Sig	0.010660	0.056749	0.013333	0.076789
	Ver	0.021193	0.112299	0.027623	0.153284
128	Gen	0.021097	0.075419	0.029619	0.231625
	Sig	0.021220	0.075467	0.029847	0.233023
	Ver	0.042351	0.150855	0.060985	0.467125
192	Gen	0.169084	0.185556	0.216012	2.789547
	Sig	0.169643	0.186087	0.216910	2.800969
	Ver	0.338843	0.371402	0.443050	5.608597
256	Gen	0.797075	0.364633	0.983670	18.462964
	Sig	0.797973	0.366812	0.989801	18.589444
	Ver	1.593990	0.730208	2.014678	37.244228

Table 13: Average times in seconds for 10 random instances of DSA with FFC, ECC, $\mathbb{P}_D$ and $\mathbb{P}_R$, depending on the security strength.

In conclusion, there is still hope for an improvement with respect to the DSA formulation based the Pell conic but, with the current setting, adopting the cyclic group $(\mathbb{P}_R, \odot_R)$ is not a good alternative. Finding an efficient method to obtain the order p of a cyclic subgroup of the wanted size could resolve these issues and make the DSA formulation based on the Pell cubic a competitive alternative.

Part III

IS THE PELL EQUATION RELATED TO PRIMALITY TESTS?

When dealing with integers as in cryptography, it is important to know the primality of a given number. After recalling classical primality tests in Chapter 7, new powerful primality tests related to the Pell equation of degree two are introduced in Chapter 8.

7

PRIMALITY TESTS IN LITERATURE

A primality test is an algorithm that allows to determine if an odd integer has divisors different from 1 and itself.

The fact that there are infinitely many prime numbers was already proven by Euclid in its *Elements* dated around 300 BC. The problem of separating prime numbers from composite ones has always been interesting, and nowadays it is more important than ever since, as observed in Chapter 4, the confidentiality of all communications through insecure channels relies on large prime numbers. However, despite there are theoretical results that give sufficient or equivalent conditions for primality, they are not efficient.

The idea behind primality tests is to answer this question as rapidly and reliably as possible, and in order to do so different necessary conditions for primality have been exploited.

The resulting primality tests are classified depending on the reliability of their results:

- *heuristic* primality tests seems to work well in practice but there is no proof behind their reliability. A classical example is the *Fibonacci test*, which is passed by an odd number n if

  $$F_{p+1} \equiv 0 \pmod{p},$$

  where $(F_k)_k$ is the Fibonacci sequence $0, 1, 1, 2, 3, 5, 8, 13, \ldots$;

- *deterministic* primality tests give a sure result but have high computational costs. The only unconditional deterministic algorithm able to determine in polynomial time the primality of an integer number is the *AKS test* [2]: an odd number n is declared prime if for enough basis $a_i \in \mathbb{Z}$ with $\gcd(n, a_i) = 1$,

  $$(t + a_i)^n \equiv t^n + a_i \pmod{n},$$

  in the polynomial ring $\mathbb{Z}_n[t]$. This is an application of a polynomial generalization of the Fermat little theorem. However, despite the fastest version of this primality test [38] has complexity $\tilde{O}(\log^6 n)$, its practical applications are very slow;

- *probabilistic* primality tests have provable bounds on the probability of false positive results. One of the most classical and still used in combination with other primality tests is the *Fermat test*, which simply applies the Fermat little theorem: an odd integer n is declared probable prime for a base $a \in \mathbb{Z}$ if $\gcd(n, a) = 1$ and

  $$a^{n-1} \equiv 1 \pmod{n}.$$

This chapter focuses on other largely used probabilistic primality tests. Firstly, a stronger version of the Fermat test is introduced in Section 7.1. This is adopted in two of the most used probabilistic primality tests: the Rabin–Miller test and the Baillie–PSW test, which combines it with the strong Lucas test. Section 7.2 introduces the Lucas sequences that are the basis for all the Lucas probable prime tests described in Section 7.2, together with the mentioned Baillie–PSW test. Finally, Section 7.5 introduces a primality test based on the Pell conic, which then will be related to the Lucas tests.

7.1 STRONG FERMAT TEST

The Fermat test is very fast but not very reliable. In particular, there exists a composite integer n that is declared prime for any base $a \in \mathbb{Z}$ with $\gcd(n, a) = 1$. These integers are called *Carmichael numbers* and there are infinitely many of them [3].

Thus, the Fermat test should be strengthened and the principal way to do so is considering the following result.

THEOREM 7.1 Given p odd prime, if $p - 1 = 2^r s$ with s odd, then for every $a \in \mathbb{Z}_p \setminus \{0\}$

$$a^s \equiv 1 \pmod{p}, \quad \text{or} \quad a^{2^k s} \equiv -1 \pmod{p}, \quad \text{for some } 0 \leqslant k < r.$$

Proof. The square roots of unity in $\mathbb{Z}_p$ are ± 1, i.e., $x^2 \equiv 1 \pmod{p}$ if and only if $x \equiv \pm 1 \pmod{p}$. The Fermat little theorem assures that $a^{2^r s} \equiv 1 \pmod{p}$. The repeated square roots correspond to decreasing k from $r - 1$ to 0 and at each step:

- if $a^{2^k s} \equiv -1 \pmod{p}$, then the thesis is confirmed;
- if $a^{2^k s} \equiv 1 \pmod{p}$, then the next k is considered.

If $k = 0$ is reached, then $a^s \equiv 1 \pmod{p}$. □

DEFINITION 7.1 The *strong Fermat test* declares an odd integer n probable prime to base $a \in \mathbb{Z}$ if $\gcd(n, a) = 1$, $n - 1 = 2^r s$, s odd and

$$a^s \equiv 1 \pmod{n}, \quad \text{or}$$
$$a^{2^k s} \equiv -1 \pmod{n}, \quad \text{for some } 0 \leqslant k < r.$$

A composite n that satisfies this condition is called *strong pseudoprime to base* a (spsp(a)).

In [51], the author proved that a composite n is a spsp(a) to at most one quarter of all bases $a \in \mathbb{Z}$. This property is the fundamental idea behind the *Rabin–Miller test* [43, 51], which tests the integer n by applying the strong Fermat test for k different bases and declares n probably prime with a probability at most 4^{-k}. In [43], a deterministic version of this test was introduced, but its correctness relies on the unproven extended Riemann hypothesis.

7.2 LUCAS SEQUENCES

Given two parameters $P, Q \in \mathbb{Z}$, the *Lucas sequences* [40] are particular linear recurrent integer sequences $(x_k)_{k\geqslant 0}$ that, given x_0, x_1, satisfy

$$x_k = Px_{k-1} - Qx_{k-2}, \quad \text{for } k > 1.$$

François Édouard Anatole Lucas (1842–91) defined two sequences using the roots α, β of the polynomial $x^2 - Px + Q$ as

$$U_k = \frac{\alpha^k - \beta^k}{\alpha - \beta}, \quad V_k = \alpha^k + \beta^k, \quad \text{for } k \geqslant 0.$$

The discriminant of the polynomial is $D = P^2 - 4Q$, the roots are

$$\alpha = \frac{P + \sqrt{D}}{2}, \quad \beta = \frac{P - \sqrt{D}}{2},$$

and $P = \alpha + \beta$, $Q = \alpha\beta$, $\sqrt{D} = \alpha - \beta$, so that

$$\alpha^k = \frac{V_k + U_k\sqrt{D}}{2}, \quad \beta^k = \frac{V_k - U_k\sqrt{D}}{2}, \quad \text{for } k \geqslant 0.$$

Thus, the first Lucas sequence $(U_k)_{k\geqslant 0}$ and the second Lucas sequence $(V_k)_{k\geqslant 0}$ can be described as

$$\begin{cases} U_0 = 0, \quad U_1 = 1, \\ U_k = PU_{k-1} - QU_{k-2}, \end{cases} \qquad \begin{cases} V_0 = 2, \quad V_1 = P, \\ V_k = PV_{k-1} - QV_{k-2}. \end{cases} \tag{7.1}$$

Other useful formulations can be

$$U_k = \frac{PU_{k-1} + V_{k-1}}{2}, \quad V_k = \frac{DU_{k-1} + PV_{k-1}}{2}, \quad \text{for } k > 0.$$

or using matrix forms, for $k \geqslant 0$,

$$\begin{pmatrix} U_{k+1} \\ U_k \end{pmatrix} = \begin{pmatrix} P & -Q \\ 1 & 0 \end{pmatrix} \begin{pmatrix} U_k \\ U_{k-1} \end{pmatrix} = \begin{pmatrix} P & -Q \\ 1 & 0 \end{pmatrix}^k \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} V_{k+1} \\ V_k \end{pmatrix} = \begin{pmatrix} P & -Q \\ 1 & 0 \end{pmatrix} \begin{pmatrix} V_k \\ V_{k-1} \end{pmatrix} = \begin{pmatrix} P & -Q \\ 1 & 0 \end{pmatrix}^k \begin{pmatrix} P \\ 2 \end{pmatrix},$$

$$\begin{pmatrix} V_k \\ U_k \end{pmatrix} = \begin{pmatrix} P/2 & D/2 \\ 1/2 & P/2 \end{pmatrix} \begin{pmatrix} V_{k-1} \\ U_{k-1} \end{pmatrix} = \begin{pmatrix} P/2 & D/2 \\ 1/2 & P/2 \end{pmatrix}^k \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

Given two pairs $(U_m, V_m), (U_n, V_n)$ with $m, n \geqslant 0$, it is possible to obtain the values in the position $m + n$ as

$$U_{m+n} = \frac{U_mV_n + V_mU_n}{2}, \quad V_{m+n} = \frac{V_mV_n + DU_mU_n}{2}.$$

In particular, when $m = n \geqslant 0$, the resulting formulas are

$$U_{2n} = U_nV_n, \quad V_{2n} = \frac{V_n^2 + DU_n^2}{2}.$$

Example 7.1. Some Lucas sequences are known with specific names:

- $(U_k)_{k\geqslant 0}$ with $P = 1$, $Q = -1$ is the sequence

$$U_0 = 0, \quad U_1 = 1, \quad U_k = U_{k-1} + U_{k-2},$$

of the *Fibonacci numbers*

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \ldots,$$

fully described at `https://oeis.org/A000045`;

- $(V_k)_{k\geqslant 0}$ with $P = 1$, $Q = -1$ is the sequence

$$V_0 = 2, \quad V_1 = 1, \quad V_k = V_{k-1} + V_{k-2},$$

of the *Lucas numbers*

$$2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, \ldots,$$

fully described at `https://oeis.org/A000032`;

- $(U_k)_{k\geqslant 0}$ with $P = 2$, $Q = -1$ is the sequence

$$U_0 = 0, \quad U_1 = 1, \quad U_k = 2U_{k-1} + U_{k-2},$$

of the *Pell numbers*

$$0, 1, 2, 5, 12, 29, 70, 169, 408, 985, 2378, 5741, 13860, \ldots,$$

fully described at `https://oeis.org/A000129`. They are named after Pell since they are related to the solutions of the Pell equation $x^2 - 2y^2 = \pm 1$. The ratios x/y of its solutions are

$$\frac{1}{1}, \frac{3}{2}, \frac{7}{5}, \frac{17}{12}, \frac{41}{29}, \frac{99}{70}, \frac{239}{169}, \frac{577}{408}, \frac{1393}{985}, \frac{3363}{2378}, \frac{8119}{5741}, \frac{19601}{13860}, \ldots,$$

which are $(U_k + U_{k-1})/U_k$, for $k > 0$, and approximate $\sqrt{2}$;

- $(V_k)_{k\geqslant 0}$ with $P = 2$, $Q = -1$ is the sequence

$$V_0 = 2, \quad V_1 = 2, \quad V_k = 2V_{k-1} + V_{k-2},$$

of the *Pell–Lucas numbers*

$$2, 2, 6, 14, 34, 82, 198, 478, 1154, 2786, 6726, 16238, 39202, \ldots,$$

fully described at `https://oeis.org/A002203`;

- $(U_k)_{k\geqslant 0}$ with $P = 3$, $Q = 2$ is the sequence

$$U_0 = 0, \quad U_1 = 1, \quad U_k = 3U_{k-1} - 2U_{k-2} = 2^k - 1,$$

of the *Mersenne numbers*

$$0, 1, 3, 7, 15, 31, 63, 127, 255, 511, 1023, 2047, 4095, \ldots,$$

fully described at `https://oeis.org/A000225`.

7.3 PRIMALITY TESTS BASED ON LUCAS SEQUENCES

The properties of the Lucas sequences are deeply exploited to obtain probabilistic primality tests. In the following, starting from the simplest to the strongest, the main primality tests based on the Lucas sequences are described.

7.3.1 *Lucas test*

The main result exploited for testing the primality of an integer using Lucas sequences is the following generalization of the Fermat little theorem.

THEOREM 7.2 Given p odd prime, fixed the parameters $P, Q \in \mathbb{Z}$, if $D = P^2 - 4Q$ has Legendre symbol $j = \left(\frac{D}{p}\right) \neq 0$ and $\gcd(p, Q) = 1$, then

$$U_{p-j} \equiv 0 \pmod{p},$$
$$V_{p-j} \equiv 2Q^{(1-j)/2} \pmod{p}.$$

Proof. Since $\gcd(p, D) = 1$, $j = \pm 1$ so that:

- if $j = 1$, then $\sqrt{D} \in \mathbb{Z}_p$ as well as $\alpha, \beta \in \mathbb{Z}_p$ and, for the Fermat little theorem,
$$U_{p-1} = \frac{\alpha^{p-1} - \beta^{p-1}}{\alpha - \beta} \equiv 0 \pmod{p},$$
$$V_{p-1} = \alpha^{p-1} + \beta^{p-1} \equiv 2 \pmod{p};$$
- if $j = -1$, then $\sqrt{D} \notin \mathbb{Z}_p$ as well as $\alpha, \beta \notin \mathbb{Z}_p$ and, for the Frobenius morphism $\alpha^p = \beta, \beta^p = \alpha$, i.e.,
$$U_{p+1} = \frac{\alpha^{p+1} - \beta^{p+1}}{\alpha - \beta} \equiv 0 \pmod{p},$$
$$V_{p+1} = \alpha^{p+1} + \beta^{p+1} = 2\alpha\beta \equiv 2Q \pmod{p}.$$

Thus, in both cases the thesis is verified. □

DEFINITION 7.2 [4] The first congruence in Theorem 7.2 gives the *Lucas test*, which declares an odd integer n probable prime for the parameters $P, Q \in \mathbb{Z}$ with $D = P^2 - 4Q$ if $j = \left(\frac{D}{n}\right) \neq 0$, $\gcd(n, Q) = 1$ and

$$U_{n-j} \equiv 0 \pmod{n}.$$

A composite n that passes this test is called *Lucas pseudoprime with parameters* P, Q (lpsp(P, Q)).

In [4], the distribution of lpsp(P, Q) is studied but the results have been improved. In particular, the best upper bound for the number $\mathrm{LP}(x)$ of lpsp(P, Q) not exceeding a sufficiently large x is given in [30]

$$\mathrm{LP}(x) < x \exp\left(-\frac{\log x \log\log\log x}{2\log\log x}\right).$$

7.3.2 *Strong Lucas test*

Theorem 7.2 can be extended as for the Fermat little theorem with Theorem 7.1, and the result allows to strengthen the Lucas test.

THEOREM 7.3 Given p odd prime and the Lucas sequences $(U_k)_{k\geqslant 0}$, $(V_k)_{k\geqslant 0}$ with parameters $P, Q \in \mathbb{Z}$, if $D = P^2 - 4Q$ has Legendre symbol $j = \left(\frac{D}{p}\right) \neq 0$, $\gcd(p, Q) = 1$ and $p - j = 2^r s$ with s odd, then

$$\begin{aligned} &U_s \equiv 0 \pmod p, \\ &\quad \text{or} \\ &V_{2^k s} \equiv 0 \pmod p, \quad \text{for some } 0 \leqslant k < r. \end{aligned}$$

Proof. Theorem 7.2 assures that $U_{2^r s} \equiv 0 \pmod p$ and this value can be obtained through Equation 7.2 as

$$U_{2^r s} = U_{2^{r-1} s} V_{2^{r-1} s},$$

so that

$$\begin{aligned} &U_{2^{r-1} s} \equiv 0 \pmod p, \\ &\quad \text{or} \\ &V_{2^{r-1} s} \equiv 0 \pmod p. \end{aligned}$$

In the latter case, the thesis is obtained. Otherwise Equation 7.2 can be used again for halving the indices and the resulting conditions are analogous to the previous ones. Eventually, if the index $2^0 s$ is reached, then only the condition $U_s \equiv 0 \pmod p$ remains and the thesis is at last confirmed. □

DEFINITION 7.3 [4] Using Theorem 7.3, the *strong Lucas test* declares an odd integer n probable prime for the parameters $P, Q \in \mathbb{Z}$ with $D = P^2 - 4Q$ if $j = \left(\frac{D}{n}\right) \neq 0$, $\gcd(n, Q) = 1$, $n - j = 2^r s$ with s odd and

$$\begin{aligned} &U_s \equiv 0 \pmod n, \\ &\quad \text{or} \\ &V_{2^k s} \equiv 0 \pmod n, \quad \text{for some } 0 \leqslant k < r. \end{aligned}$$

A composite n that passes this test is called a *strong Lucas pseudoprime with parameters* P, Q (slpsp(P, Q)).

Clearly, any slpsp(P, Q) is a lpsp(P, Q), i.e., for all sufficiently large x, the number $\mathrm{SLP}(x)$ of slpsp(P, Q) not exceeding x is

$$\mathrm{SLP}(x) < \mathrm{LP}(x) < x \exp\left(-\frac{\log x \log\log\log x}{2 \log\log x}\right).$$

In [4] also a lower bound is obtained, but it has been improved in [25], where the authors proved that, for all sufficiently large x, there is a positive constant c such that

$$x^{(\log x)^c} < \mathrm{SLP}(x) < \mathrm{LP}(x).$$

The strong Lucas test can be combined with the strong Fermat test in order to obtain a powerful test. This was firstly introduced in [4], together with another important consideration: when $j = \left(\frac{D}{n}\right) = 1$, the strong Lucas test is equivalent to a strong Fermat test with base $a = Q$. Specifically, if $D \equiv S^2 \pmod{n}$, then the parameters could be taken as $P = S + 2$, $Q = S + 1$, so that $\alpha = S + 1$, $\beta = 1$ and $U_{n-1} = \frac{Q^{n-1}-1}{Q-1}$. Thus, in this case, the strong Lucas test and the strong Fermat test are not independent and it is not worth combining them.

A possible solution is to choose the parameters depending on the integer n to be tested, instead of fixing them for every n. The authors of [4] suggested two methods for choosing the parameters $P, Q \in \mathbb{Z}$ such that $D = P^2 - 4Q$ is not a square in $\mathbb{Z}_n$.

The most promising is the Selfridge method:

1. take $D \in \{5, -7, 9, -11, \ldots\}$ with $\left(\frac{D}{n}\right) = -1$ and $|D|$ minimum;
2. fix $P = 1$ and evaluate $Q = \frac{1-D}{4}$.

Example 7.2. If testing different odd integers with the Lucas test and the strong Lucas test, the pseudoprimes with parameters $P = 4$, $Q = 1$ up to 20000 are 38 in the first case:

$$65, 209, 629, 679, 901, 989, 1241, 1769, 1961, 1991, 2509, 2701,$$
$$2911, 3007, 3439, 3869, 5249, 5719, 5777, 6061, 6767, 6989,$$
$$9869, 11041, 12749, 13019, 13133, 13529, 13949, 14701, 14839,$$
$$15505, 15841, 16109, 18721, 18817, 19169, 19981;$$

while in the strong case they are 14:

$$209, 901, 989, 2701, 2911, 6061, 6767, 6989,$$
$$9869, 11041, 13133, 13529, 14839, 18817.$$

When adopting the Selfridge method with the Lucas test, the pseudoprimes up to 20000 are 19:

$$323, 377, 1159, 1829, 3827, 5459, 5777, 9071, 9179, 10877,$$
$$11419, 11663, 13919, 14839, 16109, 16211, 18407, 18971, 19043;$$

the full list can be found at `https://oeis.org/A217120`.

If the Selfridge method is used with the strong Lucas test, the pseudoprimes up to 20000 are just 5:

$$5459, 5777, 10877, 16109, 18971;$$

the full list can be found at `https://oeis.org/A217255`.

The combination of the strong Fermat test for base $a = 2$ and the strong Lucas test with parameters selected through the Selfridge method is called *Baillie–PSW test* [4, 50]. This is one of the mainly used tests, also because there are no known composite numbers that are declared probable primes [17]. Despite this, it is conjectured that there are infinitely many of these *Baillie–PSW pseudoprimes* [49].

7.3.3 *Extra strong Lucas test*

A stronger primality test based on Lucas sequences can be obtained from the following result.

THEOREM 7.4 [31] Given p odd prime and the Lucas sequences $(U_k)_{k\geqslant 0}$, $(V_k)_{k\geqslant 0}$ with parameters $P \in \mathbb{Z}$, $Q = 1$, if $D = P^2 - 4$ has Legendre symbol $j = \left(\frac{D}{p}\right) \neq 0$ and $p - j = 2^r s$ with s odd, then

$$\begin{aligned} &U_s \equiv 0 \pmod p \quad \text{and} \quad V_s \equiv \pm 2 \pmod p, \\ &\quad \text{or} \\ &V_{2^k s} \equiv 0 \pmod p, \quad \text{for some } 0 \leqslant k < r. \end{aligned}$$

Proof. After Theorem 7.3, it is sufficient to prove that, if $Q = 1$ and $U_s \equiv 0 \pmod p$, then $V_s \equiv \pm 2 \pmod p$. The first hypothesis implies that $\alpha\beta \equiv 1 \pmod p$ and, since $U_s = \frac{\alpha^s - \beta^s}{\alpha - \beta}$ is null, $\alpha^s \equiv \beta^s \pmod p$. Thus, $\alpha^s \equiv \beta^s \equiv \alpha^{-s} \pmod p$. This means that $\alpha^{2s} \equiv 1 \pmod p$, i.e., $\alpha^s \equiv \pm 1 \pmod p$, so that

$$V_s = \alpha^s + \beta^s \equiv \alpha^s + \alpha^{-s} \equiv 2\alpha^s \equiv \pm 2 \pmod p,$$

and the thesis is confirmed. □

DEFINITION 7.4 [31] Using Theorem 7.4, the *extra strong Lucas test* declares an odd integer n probable prime for the parameter $P \in \mathbb{Z}$ with $D = P^2 - 4$ if $j = \left(\frac{D}{n}\right) \neq 0$, $n - j = 2^r s$ with s odd and

$$\begin{aligned} &U_s \equiv 0 \pmod n \quad \text{and} \quad V_s \equiv \pm 2 \pmod n, \\ &\quad \text{or} \\ &V_{2^k s} \equiv 0 \pmod n, \quad \text{for some } 0 \leqslant k < r. \end{aligned}$$

A composite n that passes this test is called a *extra strong Lucas pseudoprime with parameter* P (xlpsp(P)).

Clearly, any xlpsp(P) is a slpsp(P, Q) with $Q = 1$, so that, for large x, the number $\mathrm{XLP}(x)$ of xlpsp(P) not exceeding x is

$$\mathrm{XLP}(x) < \mathrm{SLP}(x) < x \exp\left(-\frac{\log x \log\log\log x}{2\log\log x}\right).$$

Since in the Selfridge method $P = 1$ and $Q = \frac{1-D}{4}$, it is not directly suitable for the extra strong Lucas test. The main adaptation for testing n takes the minimum integer $P > 2$ such that $D = P^2 - 4$ has Jacobi symbol $j = \left(\frac{D}{n}\right) = -1$.

Example 7.3. The fixed parameters adopted in Example 7.2 are suitable also for the extra strong Lucas test, and the pseudoprimes with parameter $P = 4$ (and $Q = 1$) up to 20000 are only 4:

$$989, 2701, 11041, 18817.$$

The list of pseudoprimes obtained with the adaptation of the Selfridge method is not comparable to that for the strong Lucas test. In particular, in the latter cases, the pseudoprimes up to 50000 are 9:

$$5459, 5777, 10877, 16109, 18971, 22499, 24569, 25199, 40309;$$

for the extra strong case, they are still 9 but different:

$$989, 3239, 5777, 10877, 27971, 29681, 30739, 31631, 39059;$$

this list is fully described at `https://oeis.org/A217719`.

7.4 FROBENIUS TEST

In [31], a primality test based on polynomials and the Frobenius morphism is introduced.

DEFINITION 7.5 The *Frobenius probable prime test* with respect to a monic polynomial $f(x) \in \mathbb{Z}[x]$ of degree d with discriminant D declares an odd integer $n > 1$ probable prime if $\gcd(n, f(0)D) = 1$ and it is not declared composite by the following steps:

1. *factorization step:* take $f_0(x) = f(x) \pmod n$ and, for $1 \leqslant i \leqslant d$, $F_i(x) = \gcd(x^{n^i} - x, f_{i-1}(x))$ monic and $f_i(x) = f_{i-1}(x)/F_i(x)$. If any of the monic gcd fail to exist, or $f_d(x) \neq 1$, then declare n composite;
2. *Frobenius step:* for $2 \leqslant i \leqslant d$, compute $F_i(x^n) \pmod{F_i(x)}$, if it is non–zero for some i, then declare n composite;
3. *Jacobi step:* given $S = \sum_{2|i} \deg(F_i(x))/i$, if $(-1)^S \neq \left(\frac{D}{n}\right)$, then declare n composite.

A composite n that passes this test is called *Frobenius pseudoprime with respect to* $f(x)$ (fpsp(f)).

If an odd integer n is a fpsp(f), then it is a Fermat pseudoprime to base $f(0)$. Thus, if the discriminant D of $f(x)$ is non–zero and $|f(0)| \neq 1$, then for all sufficiently large x (depending only on $|f(0)|$) the number of fpsp(f) up to x is at most

$$x \exp\left(-\frac{\log x \log\log\log x}{2\log\log x}\right).$$

This test is related to the primality test resulting from both the conditions in Theorem 7.2, which declares an odd integer n probable prime for the parameters $P, Q \in \mathbb{Z}$ with $D = P^2 - 4Q$ if $j = \left(\frac{D}{n}\right) \neq 0$, $\gcd(n, Q) = 1$ and

$$U_{n-j} \equiv 0 \pmod n,$$
$$\text{and}$$
$$V_{n-j} \equiv 2Q^{(1-j)/2} \pmod n.$$

This is exactly a Frobenius probable prime test with respect to the monic polynomial $x^2 - Px + Q$ of degree $d = 2$ with discriminant $D = P^2 - 4Q$, so that a composite n that passes this test can be called *Frobenius pseudoprime with respect to* $x^2 - Px + Q$ (fpsp(P, Q)).

Clearly, any fpsp(P, Q) is a lpsp(P, Q), i.e., for all sufficiently large x (depending only on Q), the upper bound on the number $FP(x)$ of fpsp(P, Q) not exceeding x can also be obtained as

$$FP(x) < LP(x) < x \exp\left(-\frac{\log x \log\log\log x}{2\log\log x}\right).$$

Despite there are no references in the literature, it is possible to adopt directly the Selfridge method with the Frobenius test with respect to $x^2 - Px + Q$ in order to choose good parameters $P, Q \in \mathbb{Z}$ depending on the integer n to be tested.

Example 7.4. When adopting the parameters $P = 4$, $Q = 1$ used in Example 7.2, the Frobenius pseudoprimes up to 20000 are 22:

$$209, 901, 989, 2701, 2911, 3007, 3439, 5719, 6061, 6767, 6989, 9869,$$
$$11041, 13133, 13529, 14701, 14839, 15505, 15841, 18721, 18817, 19981;$$

they are less than the Lucas pseudoprimes (38), but more than the strong and extra strong Lucas pseudoprimes (14 and 4, respectively).

However, using the Selfridge method with the Frobenius test confirms that it is not comparable with the strong and extra strong Lucas test: there are only two Frobenius pseudoprimes up to 50000, specifically 5777 and 10877. Interestingly, these pseudoprimes are the odd $n \equiv 2, 3 \pmod 5$ that are Frobenius pseudoprimes with parameters $P = 1$, $Q = -1$ (related to the Fibonacci polynomial $x^2 - x - 1$). These pseudoprimes are described at `https://oeis.org/A212423` and the list up to 1000000 is:

$$5777, 10877, 75077, 100127, 113573, 161027,$$
$$162133, 231703, 430127, 635627, 851927.$$

7.5 PELL TEST

Lucas tests can be related to the Pell conic with parameter D ($\mathcal{C}_D$) through the Pell test [21]. The idea behind this test is to exploit that, as observed in Section 2.2, if p is prime, then $\mathcal{C}_D$ over $\mathbb{Z}_p$ defined through the Pell equation as

$$\mathcal{C}_D = \left\{(x, y) \in \mathbb{Z}_p^2 \,|\, x^2 - Dy^2 \equiv 1 \pmod p\right\},$$

with the Brahmagupta product with parameter D

$$(x_1, y_1) \otimes_D (x_2, y_2) = (x_1x_2 + Dy_1y_2, x_1y_2 + y_1x_2) \pmod p,$$

is a cyclic group of order $p - j$ where $j = \left(\frac{D}{p}\right) \neq 0$ is the Legendre symbol of D over p, i.e., its quadratic character. In the tests, since n is generally non–prime, $j = \left(\frac{D}{n}\right)$ is the Jacobi symbol.

DEFINITION 7.6 [21] The *Pell test* declares an odd integer n probable prime for the parameters $D \in \mathbb{Z}$, $(x, y) \in \mathcal{C}_D$ if $j = \left(\frac{D}{n}\right) \neq 0$ and

$$y_{n-j} \equiv 0 \pmod{n}, \quad \text{where } (x_{n-j}, y_{n-j}) = (x, y)^{\otimes_D n-j}.$$

A composite n that passes this test is called *Pell pseudoprime with parameters* $D, (x, y)$ ($\mathrm{ppsp}(D, x, y)$).

Remark. The same denomination is adopted in the literature for a test that uses the Pell numbers introduced in Section 7.2 as those in the Lucas sequence $(U_k)_{k \geqslant 0}$ with $P = 2$, $Q = -1$. This test declares an odd integer n probable prime if

$$U_n \equiv \left(\frac{2}{n}\right) \pmod{n},$$

and a Pell pseudoprime (`https://oeis.org/A099011`) is a composite that passes the test.

The relation between the Pell test and the Lucas test is that, if an odd integer n passes the Pell test with parameters $D \in \mathbb{Z}$, $(x, y) \in \mathcal{C}_D$, then n passes the Lucas test with parameters $P = 2x$, $Q = 1$.

On the other hand, if n passes the Lucas test with parameters $P \in \mathbb{Z}$ and $Q = 1$, then n passes the Pell test with parameters $D = P^2 - 4$ and $(x, y) = (P/2, 1/2)$ [21]. However, if $P \in \mathbb{Z}$ is fixed for different n, this relation gives Pell tests with different parameters $(x, y) \in \mathcal{C}_D$ because of the inverse of $2 \in \mathbb{Z}_n$ that is $(n + 1)/2$. It is still possible to consider that, if P is even, then the Lucas test is related to the Pell test with fixed parameters $D = (P/2)^2 - 1$ and $(x, y) = (P/2, 1)$.

Example 7.5. Fixed the parameters $P = 4$, $Q = 1$, as observed in Example 7.2, the first two Lucas pseudoprimes are 65 and 209. In the first case, the related Pell test has

$$D = P^2 - 4 = 12, \quad (x, y) = (P/2, 1/2) \equiv (2, 33) \pmod{65},$$

while, in the second case, the parameters are $D = 12$, $(x, y) = (2, 105)$, which are different from the previous ones.

However, when considering

$$D = (P/2)^2 - 1 = 3, \quad (x, y) = (P/2, 1) = (2, 1),$$

the resulting Pell test is equivalent to the Lucas test for testing any n.

In general, it is not possible to fix the parameters $D \in \mathbb{Z}$ and $(x, y) \in \mathcal{C}_D$ for testing different integers with the Pell test, because $x^2 - Dy^2 \equiv 1 \pmod{n}$ and this can not be true for any integer n. For overcoming this issue, the use of a parametrization of $\mathcal{C}_D$ can be helpful. As introduced in Theorem 2.1, a possible parametrization is

$$\phi_D(m) = \begin{cases} \left(\frac{m^2+D}{m^2-D}, \frac{2m}{m^2-D}\right), & \text{if } m \in \mathbb{Z}_n \setminus \{\pm\sqrt{D}\}, \\ (1, 0), & \text{if } m = \alpha \text{ point at infinity.} \end{cases}$$

In this way, different integers can be tested with fixed parameters $D, m \in \mathbb{Z}$ by applying the Pell test with D and $(x, y) = \left(\frac{m^2+D}{m^2-D}, \frac{2m}{m^2-D}\right)$.

With this formulation, if n passes the Lucas test with parameters $P \in \mathbb{Z}$, $Q = 1$, then n passes the Pell test with parameters $D = P^2 - 4$ and $m = P + 2$, which can be fixed for different n. Clearly not every couple $D, m \in \mathbb{Z}$ can be found with this relation.

Example 7.6. The Lucas test with parameters $P = 4$ and $Q = 1$ introduced in Example 7.2 is related to the Pell test with parameters $D = P^2 - 4 = 12$ and $m = P + 2 = 6$, so that they generate the same list of pseudoprimes.

Conversely, if n passes the Pell test with parameters $D, m \in \mathbb{Z}$, then n passes the Lucas test with parameters $P = 2\frac{m^2+D}{m^2-D}$, $Q = 1$. However, the parameter P depends on n, so that it is not always possible to obtain an equivalent Lucas test with fixed parameters for testing different integers.

Example 7.7. Fixed the parameters $D = m = 3$ for a Pell test, the related Lucas test has parameters $P = 2\frac{m^2+D}{m^2-D} = 4$, $Q = 1$, so that the Pell pseudoprimes are the same of Example 7.2.

It is important to notice that, despite the same Lucas test is obtained, the values of D and m are different from those in the previous example. This means that there are Pell tests with different parameters that are equivalent to each others.

A different case is the Pell test with parameters $D = 2$, $m = 3$, whose pseudoprimes up to 20000 are:

$$
\begin{gathered}
33, 55, 145, 319, 561, 579, 589, 779, 899, 989, 1079, 1595, 1649, \\
1685, 1711, 1807, 1829, 2123, 2507, 2915, 3013, 3201, 3281, 3707, \\
3827, 4687, 5339, 5447, 5633, 5671, 5885, 6369, 6441, 6901, 7061, \\
8711, 9179, 9379, 9773, 9869, 9899, 10403, 10585, 11001, 11521, \\
11537, 11659, 13201, 13299, 14023, 14065, 14111, 14257, 14279, \\
14795, 15189, 15707, 18241, 18299, 18535, 18721, 19561, 19951.
\end{gathered}
$$

This test is related to a Lucas test with parameters $P = 2\frac{m^2+D}{m^2-D} = \frac{22}{7}$ and $Q = 1$, so that P can not be fixed and for all $n \equiv 0 \pmod 7$ it does not exist.

In conclusion, when the parameters $D, m \in \mathbb{Z}$ of a Pell test can be related to the fixed parameters $P \in \mathbb{Z}$, $Q = 1$ of a Lucas test, the pseudoprimes have equivalent distributions, i.e., for all sufficiently large x, the number $PP(x)$ of Pell pseudoprimes with parameters $D, m \in \mathbb{Z}$ not exceeding x is

$$
PP(x) = LP(x) < x \exp\left(-\frac{\log x \log\log\log x}{2\log\log x}\right).
$$

8

PRIMALITY TESTS BASED ON SEQUENCES

The Pell test is based on the points of the Pell conic that, with $\otimes_D$, is a cyclic group with identity $(1,0)$ and order $n-j$, where $j=\left(\frac{D}{n}\right)\neq 0$ is the Jacobi symbol. However, it considers only the y coordinate of $(x,y)^{\otimes_D n-j}\in\mathcal{C}_D$. A possible improvement can be obtained by considering both the coordinates of the obtained point.

DEFINITION 8.1 [9] The *strong Pell test* declares an odd integer n probable prime for the parameters $D\in\mathbb{Z}$, $(x,y)\in\mathcal{C}_D$ if $j=\left(\frac{D}{n}\right)\neq 0$ and

$$(x_{n-j},y_{n-j})=(x,y)^{\otimes_D n-j}=(1,0).$$

A composite n that passes this test is called *strong Pell pseudoprime with parameters* $D,(x,y)$ (sppsp(D,x,y)).

Clearly, any sppsp(D,x,y) is a ppsp(D,x,y), i.e., for all sufficiently large x, the number $\mathrm{SPP}(x)$ of sppsp(D,x,y) not exceeding x is

$$\mathrm{SPP}(x)<\mathrm{PP}(x)<x\exp\left(-\frac{\log x\log\log\log x}{2\log\log x}\right).$$

The check in the strong Pell test can be written in a matrix form as

$$\begin{pmatrix}x_{n-j}\\ y_{n-j}\end{pmatrix}=\begin{pmatrix}x & Dy\\ y & x\end{pmatrix}^{n-j}\begin{pmatrix}1\\ 0\end{pmatrix}\equiv\begin{pmatrix}1\\ 0\end{pmatrix}\pmod n. \tag{8.1}$$

A similar approach can be adopted for the Lucas test using the matrix forms in Section 7.2, so that the check on the first Lucas sequence $(U_k)_{k\geqslant 0}$ is the second row of

$$\begin{pmatrix}U_{n-j+1}\\ U_{n-j}\end{pmatrix}=\begin{pmatrix}P & -Q\\ 1 & 0\end{pmatrix}^{n-j}\begin{pmatrix}1\\ 0\end{pmatrix}\equiv\begin{pmatrix}1\\ 0\end{pmatrix}\pmod n. \tag{8.2}$$

In this chapter, the generalization of these matrix constructions is exploited to obtain new primality tests inspired by those seen in Section 7.3 and Section 7.5. In Section 8.1, the main result on linear recurrent sequences of order two using a matrix approach is introduced. The Lucas test and the Pell test, as well as their connection, arise as particular cases. Moreover, in this way, the strong Pell test is related to a new stronger version of the Lucas test in Section 8.2. Section 8.3 and Section 8.4, describe two further generalizations of the Pell test and the Lucas test, respectively. For both the tests, a deep study on the choice of their parameters is conducted. In addition, a method for choosing their parameters similar to the one proposed by Selfridge in [4] is obtained in Section 8.5, which shows some empirical results on these tests when their parameters are fixed and when the adaptations of the Selfridge method for choosing the parameters are adopted.

8.1 LINEAR RECURRENT SEQUENCES FOR PRIMALITY TESTS

This matrix structure observed for the Pell and Lucas test can be generalized by considering that any matrix $M \in \mathbb{Z}^{2\times 2}$ generates the linear recurrent sequences

$$\begin{pmatrix} \tilde{v}_k \\ \tilde{u}_k \end{pmatrix} = M^k \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{for } k \geqslant 0,$$

and the following result holds.

LEMMA 8.1 [9] Let $D \in \mathbb{Z}$ be the discriminant of the characteristic polynomial of $M \in \mathbb{Z}^{2\times 2}$. If p is prime and $\det(M), D \not\equiv 0 \pmod p$, then

1. $(\tilde{v}_{p-1}, \tilde{u}_{p-1}) \equiv (1, 0) \pmod p$, when $\sqrt{D} \in \mathbb{Z}_p^\times$;
2. $(\tilde{v}_{p+1}, \tilde{u}_{p+1}) \equiv (\det(M), 0) \pmod p$, when $\sqrt{D} \notin \mathbb{Z}_p^\times$.

Proof. Let α, β be the roots of the characteristic polynomial of M, so that M is similar to the diagonal matrix

$$\begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix},$$

and there are two possible scenarios:

1. if $\sqrt{D} \in \mathbb{Z}_p^\times$, then $\alpha, \beta \in \mathbb{Z}_p^\times$ and, for the Fermat little theorem, $\alpha^{p-1} \equiv \beta^{p-1} \equiv 1 \pmod p$. Thus, $M^{p-1} = \mathrm{Id}$ and
$$\begin{pmatrix} \tilde{v}_{p-1} \\ \tilde{u}_{p-1} \end{pmatrix} = M^{p-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \pmod p;$$
2. if $\sqrt{D} \notin \mathbb{Z}_p^\times$, then $\alpha, \beta \notin \mathbb{Z}_p$ and, for the Frobenius morphism, $\alpha^p = \beta, \beta^p = \alpha$, i.e.,
$$\begin{pmatrix} \tilde{v}_{p+1} \\ \tilde{u}_{p+1} \end{pmatrix} = M^p \cdot M \begin{pmatrix} 1 \\ 0 \end{pmatrix} \equiv \det(M) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \pmod p.$$

Thus, in both cases the thesis is verified. □

This result allows to define new primality tests by declaring an odd integer n probable prime given the sequences $(\tilde{u}_k)_{k\geqslant 0}, (\tilde{v}_k)_{k\geqslant 0}$ generated through $M \in \mathbb{Z}^{2\times 2}$ with $\gcd(n, \det(M)) = 1$ and D discriminant of the characteristic polynomial, if $j = \left(\frac{D}{n}\right) \neq 0$ and

$$(\tilde{v}_{n-j}, \tilde{u}_{n-j}) \equiv \begin{cases} (1, 0) \pmod n, & \text{if } j = 1, \\ (\det(M), 0) \pmod n, & \text{if } j = -1. \end{cases}$$

8.2 STRONG PELL TEST AND DOUBLE LUCAS TEST

Thanks to Equation 8.1, the strong Pell test introduced in Definition 8.1 can be seen as an example of this construction.

For the Lucas test, Equation 8.2 shows how it could be seen as part of an enhanced test that, with the matrix notation used in Lemma 8.1, has $(\tilde{u}_k)_{k\geqslant 0} = (\tilde{v}_{k-1})_{k\geqslant 0} = (U_k)_{k\geqslant 0}$.

DEFINITION 8.2 [9] The *double Lucas test* declares an odd integer n probable prime for the parameters $P, Q \in \mathbb{Z}$ with $D = P^2 - 4Q$ if $j = \left(\frac{D}{n}\right) \neq 0$, $\gcd(n, Q) = 1$ and

$$(U_{n-j+1}, U_{n-j}) \equiv \begin{cases} (1,0) \pmod n, & \text{if } j = 1, \\ (Q,0) \pmod n, & \text{if } j = -1. \end{cases}$$

A composite n that passes this test is called *double Lucas pseudoprime with parameters* P, Q (dlpsp(P, Q)).

Clearly each dlpsp(P, Q) is a lpsp(P, Q), i.e., for all sufficiently large x, the number $\mathrm{DLP}(x)$ of dlpsp(P, Q) not exceeding x is

$$\mathrm{DLP}(x) < \mathrm{LP}(x) < x \exp\left(-\frac{\log x \log\log\log x}{2\log\log x}\right).$$

PROPOSITION 8.1 An odd integer n passes the double Lucas test with parameters $P, Q \in \mathbb{Z}$ if and only if it passes the Frobenius probable prime test with respect to $x^2 - Px + Q$.

Proof. Both the tests require $U_{n-j} \equiv 0 \pmod n$, while the second check differs. From the properties of Lucas sequences described in Section 7.2, $V_k = 2U_{k+1} - PU_k$ for $k \geqslant 0$, so that:

- when $j = 1$, n passes the double Lucas test if

  $$(U_n, U_{n-1}) \equiv (1,0) \pmod n,$$

  which is true if and only if $V_{n-1} \equiv 2 \pmod n$;
- when $j = -1$, n passes the double Lucas test if

  $$(U_{n+2}, U_{n+1}) \equiv (Q,0) \pmod n,$$

  which is true if and only if $V_{n+1} \equiv 2Q \pmod n$.

Thus, the conditions in the double Lucas test are equivalent to

$$(U_{n-j}, V_{n-j}) \equiv \left(0, 2Q^{(1-j)/2}\right) \pmod n,$$

as required by the Frobenius test. □

Thus, as for the Frobenius test, the double Lucas test is not comparable with strong and extra strong Lucas tests.

In addition, $\mathrm{DLP}(x) = \mathrm{FP}(x)$ for all sufficiently large x (depending only on Q), which also confirms the upper bound deduced previously from $\mathrm{LP}(x)$.

As for all Lucas tests, it is possible to use the Selfridge method with the double Lucas test in order to find good parameters depending on n, and the resulting pseudoprimes are those listed in Example 7.4.

As observed for Lucas and Pell tests in Section 7.5, there is an equivalence between double Lucas and strong Pell tests that can be easily proved using the matrix approach introduced in Section 8.1.

PROPOSITION 8.2 [9] If n passes the double Lucas test with parameters $P \in \mathbb{Z}$, $Q = 1$, then n passes the strong Pell test with parameters $D = P^2 - 4$ and $(X, Y) = (P/2, 1/2)$.

On the other hand, if n passes the strong Pell test with parameters $D \in \mathbb{Z}$, $(X, Y) \in \mathcal{C}_D$, then n passes the double Lucas test with parameters $P = 2X$ and $Q = 1$.

Proof. Equation 8.1 and Equation 8.2 introduced the matrices

$$C = \begin{pmatrix} X & DY \\ Y & X \end{pmatrix} \quad \text{and} \quad L = \begin{pmatrix} P & -Q \\ 1 & 0 \end{pmatrix},$$

related to the strong Pell test and the double Lucas test, respectively. Since $\det(C) = 1$ and $\det(L) = Q$, they can be similar only if $Q = 1$.

If n is passes the double Lucas test with parameters $P \in \mathbb{Z}$, $Q = 1$, then the matrix

$$R_1 = \begin{pmatrix} 1 & P \\ 0 & 2 \end{pmatrix},$$

gives the relation

$$R_1^{-1} \cdot L \cdot R_1 = \begin{pmatrix} P/2 & (P^2-4)/2 \\ 1/2 & P/2 \end{pmatrix} = \begin{pmatrix} X & DY \\ Y & X \end{pmatrix},$$

so that n passes the strong Pell test with parameters $D = P^2 - 4$ and $(X, Y) = (P/2, 1/2)$.

On the other hand, if n passes the strong Pell test with parameters $D \in \mathbb{Z}$ and $(X, Y) \in \mathcal{C}_D$, then

$$R_2 = \begin{pmatrix} 1 & -X \\ 0 & Y \end{pmatrix},$$

gives the relation

$$R_2^{-1} \cdot C \cdot R_2 = \begin{pmatrix} 2X & -X^2 + DY^2 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} P & -1 \\ 1 & 0 \end{pmatrix},$$

so that n passes the double Lucas test with parameters $P = 2X$ and $Q = 1$. □

This relation presents the same problems of that between Lucas and the Pell test described in Section 7.5: when fixing the parameters $P \in \mathbb{Z}$, $Q = 1$ of a double Lucas test, the corresponding strong Pell test has parameters $D = P^2 - 4$ and $(X, Y) = (P/2, 1/2)$, which change with n because of the inverse of $2 \in \mathbb{Z}_n$. It is still possible to consider that, if P is even, then the double Lucas test is related to the strong Pell test with fixed parameters $D = (P/2)^2 - 1$ and $(X, Y) = (P/2, 1)$.

Example 8.1. The double Lucas pseudoprimes with parameters $P = 4$, $Q = 1$ up to 20000 are the Frobenius pseudoprimes from Example 7.4:

$$209, 901, 989, 2701, 2911, 3007, 3439, 5719, 6061, 6767, 6989, 9869,$$
$$11041, 13133, 13529, 14701, 14839, 15505, 15841, 18721, 18817, 19981.$$

While the strong Pell test on the first pseudoprime 209 has

$$D = P^2 - 4 = 12, \quad (X, Y) = (P/2, 1/2) \equiv (2, 105) \pmod{209},$$

the parameters for the second pseudoprime 901 are

$$D = P^2 - 4 = 12, \quad (X, Y) = (P/2, 1/2) \equiv (2, 451) \pmod{901}.$$

However, when considering

$$D = (P/2)^2 - 1 = 3, \quad (X, Y) = (P/2, 1) = (2, 1),$$

the resulting strong Pell test is equivalent to the double Lucas test for testing any n.

In general, it is not possible to fix $D \in \mathbb{Z}$, $(X, Y) \in \mathcal{C}_D$ for testing different integers with the strong Pell test, because it is necessary that $X^2 - DY^2 \equiv 1 \pmod{n}$ and this can not be true for any integer n.

The use of the parametrization for $\mathcal{C}_D$ introduced in Theorem 2.1 allows to obtain that if n passes the double Lucas test with parameters $P \in \mathbb{Z}$, $Q = 1$, then n passes the strong Pell test with parameters $D = P^2 - 4$ and $m = P + 2$, which can be fixed for different n, but not every couple $D, m \in \mathbb{Z}$ can be found.

Example 8.2. The double Lucas test with $P = 4$, $Q = 1$ is related to the strong Pell test with parameters $D = P^2 - 4 = 12$, $m = P + 2 = 6$, so that they generate the same list of pseudoprimes in Example 8.1.

On the other hand, if n passes the strong Pell test with parameters $D, m \in \mathbb{Z}$, then n passes the Lucas test with parameters $P = 2\frac{m^2+D}{m^2-D}$, $Q = 1$, where P depends on n, so that it is not always possible to obtain an equivalent double Lucas test with fixed parameters.

Example 8.3. Fixed the parameters $D = m = 3$ for a strong Pell test, the related double Lucas test has parameters $P = 2\frac{m^2+D}{m^2-D} = 4$, $Q = 1$, so that the strong Pell pseudoprimes are again the same of Example 8.1.

Since the values of D and m are different from those in the previous example, there are strong Pell tests with different parameters that are equivalent to each other.

The pseudoprimes up to 20000 given by the strong Pell test with parameters $D = 2$, $m = 3$ are less than those in Example 7.5:

$$33, 145, 561, 589, 899, 1079, 1595, 1711, 1807, 1829, 2507, 2915,$$
$$3013, 3201, 3281, 3707, 5339, 5447, 5633, 6369, 6441, 7061, 8711,$$
$$9179, 9869, 10403, 10585, 11001, 11521, 11537, 13201, 13299,$$
$$14257, 14279, 14795, 15189, 18241, 18299, 18721, 19561, 19951.$$

The related double Lucas test has $P = 2\frac{m^2+D}{m^2-D} = \frac{22}{7}$ and $Q = 1$, so that P can not be fixed and for all $n \equiv 0 \pmod 7$ it does not exist.

As for all the Lucas tests, the Selfridge method can be directly used with the double Lucas test in order to find good parameters $P, Q \in \mathbb{Z}$ depending on the integer n to be tested. This is the original idea behind the usage of the method with the Frobenius test with respect to $x^2 - Px + Q$, all the results have been described in Section 7.4.

Example 8.4. The double Lucas pseudoprimes obtained using the Selfridge method are described at `https://oeis.org/A212423` and the list up to 1000000 is:

$$5777, 10877, 75077, 100127, 113573, 161027,$$
$$162133, 231703, 430127, 635627, 851927.$$

8.3 GENERALIZED PELL PRIMALITY TEST

In order to obtain a relation between strong Pell tests and double Lucas tests with any $Q \in \mathbb{Z}$, it is possible to consider the generalized Pell conic with parameter D and norm Q ($\mathcal{C}_{D,Q}$) introduced in Section 2.3

$$\mathcal{C}_{D,Q} = \left\{(x, y) \in \mathbb{Z}_n^2 \,|\, x^2 - Dy^2 \equiv Q \pmod n\right\}.$$

Despite the product $\otimes_D$ is no more well defined over $\mathcal{C}_{D,Q}$, taking a point $(x, y) \in \mathcal{C}_{D,Q}$ defines the linear recurrent sequences

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} x & Dy \\ y & x \end{pmatrix}^k \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{for } k \geqslant 0,$$

where the determinant of the matrix is $x^2 - Dy^2 \equiv Q \pmod n$.

Using Lemma 8.1 with these sequences gives the following test.

DEFINITION 8.3 [9] The *generalized Pell test* declares an odd integer n probable prime for the parameters $D \in \mathbb{Z}$, $(x, y) \in \mathcal{C}_{D,Q}$, if $j = \left(\frac{D}{n}\right) \neq 0$, $\gcd(n, Q) = 1$ and

$$(x_{n-j}, y_{n-j}) \equiv \begin{cases} (1, 0) \pmod n, & \text{if } j = 1, \\ (Q, 0) \pmod n, & \text{if } j = -1. \end{cases}$$

A composite n that passes this test is called *generalized Pell pseudoprime with parameters* D, (x, y) ($\mathrm{gppsp}(D, x, y)$).

Considering the matrices C and L related to a generalized Pell and a double Lucas test, respectively and R_1, R_2 introduced in Proposition 8.2, it is possible to prove the following result.

PROPOSITION 8.3 If n passes the double Lucas test with parameters $P, Q \in \mathbb{Z}$ then n passes the generalized Pell test with $D = P^2 - 4Q$ and $(X, Y) = (P/2, 1/2)$.

On the other hand, if n passes the generalized Pell test with parameters $D \in \mathbb{Z}$, $(X, Y) \in \mathcal{C}_D$, then n passes the double Lucas test with $P = 2X$ and $Q = X^2 - DY^2$.

This result relates a double Lucas test with fixed parameters to a generalized Pell test with parameters that change with n because of the inverse of $2 \in \mathbb{Z}_n$, analogously to what observed for the Pell test in Example 7.5 and for the strong Pell test in Example 8.1.

The important difference with respect to those cases is that the generalized Pell test can have fixed parameters $D \in \mathbb{Z}$, $(X, Y) \in \mathbb{Z}^2$ for testing different n, as long as $X^2 - DY^2 = Q \not\equiv 0 \pmod{n}$, and this test is related to a double Lucas test with fixed parameters.

In addition, it is noteworthy that a double Lucas test with fixed parameters P even, $Q \in \mathbb{Z}$ is always equivalent to a generalized Pell test with fixed parameters $D = (P/2)^2 - Q$ and $(X, Y) = (P/2, 1)$.

Example 8.5. A generalized Pell test with fixed parameters $D = 3$, $(X, Y) = (2, 1)$ is equivalent to a double Lucas test with fixed parameters $P = 2X = 4$, $Q = X^2 - DY^2 = 1$, so that the resulting generalized Pell pseudoprimes up to 20000 are those in Example 8.1. Conversely, Proposition 8.3 relates a double Lucas test with parameters $P = 4$, $Q = 1$ with a generalized Pell test with parameters $D = P^2 - 4Q = 12$, $(X, Y) = (P/2, 1/2) = (2, 1/2)$, where Y depends on n. It is still possible to relate this double Lucas test with the initial generalized Pell test, whose fixed parameters can be retrieved as $D = (P/2)^2 - Q = 3$ and $(X, Y) = (P/2, 1) = (2, 1)$.

Changing the fixed parameters of the generalized Pell test to $D = 3$, $(X, Y) = (4, 2)$ results in having a relation with the double Lucas test with fixed parameters $P = 2X = 8$, $Q = X^2 - DY^2 = 4$. Conversely, Proposition 8.3 relates this double Lucas test with a generalized Pell test with parameters $D = P^2 - 4Q = 48$, $(X, Y) = (P/2, 1/2) = (4, 1/2)$, which change depending on n, while the other obtained relation is with the generalized Pell test with parameters $D = (P/2)^2 - Q = 12$ and $(X, Y) = (P/2, 1) = (4, 1)$. Since the values of D and (X, Y) are different from those in the initial test, there are generalized Pell tests with different parameters that are equivalent to each others.

The pseudoprimes up to 20000 given by the generalized Pell test with parameters $D = 3$, $(X, Y) = (4, 2)$ or $D = 12$, $(X, Y) = (4, 1)$ are:

$$2701, 13019, 15841, 18721.$$

They correspond to the double Lucas pseudoprimes for $P = 8$, $Q = 4$.

In conclusion, an integer is a gppsp(D, X, Y) if and only if it is a dlpsp(P, Q) with P even. Thus, for all sufficiently large x (depending only on $Q = X^2 - DY^2$), the number $\mathrm{GPP}(x)$ of gppsp(D, X, Y) not exceeding x is

$$\mathrm{GPP}(x) = \mathrm{DLP}(x) < x \exp\left(-\frac{\log x \log\log\log x}{2\log\log x}\right),$$

or analogously with the upper bound for the number $\mathrm{FP}(x)$ of Frobenius pseudoprimes with respect to $x^2 - Px + Q$.

In the following, some results on the parameters which determine equivalent generalized Pell tests are shown.

PROPOSITION 8.4 The generalized Pell test is independent of the sign of the parameters $X, Y \in \mathbb{Z}$.

Proof. As observed in Section 2.4, exponentiation with respect to $\otimes_D$ can be obtained by exploiting the generalized Rédei polynomials introduced in Definition 2.4 so that, considering the point (X, Y) as a binomial in $\mathbb{Z}[t]/(t^2 - D)$, for any $k \geqslant 0$,

$$(x_k, y_k) = (X, Y)^{\otimes_D k} \cong (X + tY)^k = a_k(D, X, Y) + t b_k(D, X, Y),$$

where

$$x_k = a_k(D, X, Y) = \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{2i} D^i X^{k-2i} Y^{2i},$$
$$y_k = b_k(D, X, Y) = \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{2i+1} D^i X^{k-2i-1} Y^{2i+1}.$$

The integers n to be tested are odd and $j = \left(\frac{D}{n}\right) = \pm 1$, so that $n - j$ is even and changing the sign of X or Y results in obtaining

$$\begin{cases} a_{n-j}(D, X, -Y) = a_{n-j}(D, X, Y), \\ b_{n-j}(D, X, -Y) = -b_{n-j}(D, X, Y), \end{cases}$$

or analogously

$$\begin{cases} a_{n-j}(D, -X, Y) = a_{n-j}(D, X, Y), \\ b_{n-j}(D, -X, Y) = -b_{n-j}(D, X, Y). \end{cases}$$

Thus, in the generalized Pell test, the check on x_{n-j} is the same in both cases, while the check $y_{n-j} = a_{n\pm 1}(D, X, Y) \equiv 0 \pmod{n}$ is satisfied if and only if $-b_{n\pm 1}(D, X, Y) \equiv 0 \pmod{n}$.

In conclusion, an integer n that passes the generalized Pell test for the parameters D and (X, Y) still passes it if the sign of X or Y (or both) is changed, and vice versa. □

PROPOSITION 8.5 The generalized Pell test is independent of the choice of $D, Y \in \mathbb{Z} \setminus \{0\}$ as long as DY^2 remains unchanged.

Proof. The previous formulation can be written for $k \geqslant 0$ as

$$x_k = a_k(D, X, Y) = \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{2i} X^{k-2i} (DY^2)^i,$$

$$y_k = b_k(D, X, Y) = \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{2i+1} X^{k-2i-1} (Dy^2)^i y.$$

Thus, if the parameters $D, Y \in \mathbb{Z}$ and $D', Y' \in \mathbb{Z}$ of a generalized Pell test have $DY^2 = D'Y'^2$, then $j = \left(\frac{D}{n}\right) = \left(\frac{D'}{n}\right)$, $a_k(D, X, Y) = a_k(D', X, Y')$ for any k and the check on x_{n-j} is equivalent.

In the check on y_{n-j}, $b_k(D, X, Y) = Y\, b_k(DY^2, X, 1)$ with $Y \neq 0$, so that

$$b_{n-j}(D, X, Y) \equiv 0 \pmod{n} \Leftrightarrow b_{n-j}(DY^2, X, 1) \equiv 0 \pmod{n}.$$

Analogously, $b_k(D', X, Y') = Y'\, b_k(D'Y'^2, X, 1)$ with $Y' \neq 0$, so that

$$b_{n-j}(D', X, Y') \equiv 0 \pmod{n} \Leftrightarrow b_{n-j}(D'Y'^2, X, 1) \equiv 0 \pmod{n},$$

and the thesis is confirmed because $DY^2 = D'Y'^2$. □

8.4 GENERALIZED LUCAS PRIMALITY TEST

Lemma 8.1 allows also to generalize the double Lucas test by adding a third parameter $R \in \mathbb{Z}$. The resulting linear recurrent sequences are

$$\begin{pmatrix} \tilde{V}_k \\ \tilde{U}_k \end{pmatrix} = \begin{pmatrix} P & -Q \\ R & 0 \end{pmatrix}^k \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{for } k \geqslant 0, \tag{8.3}$$

with discriminant of the matrix $D = P^2 - 4QR$ and the following test can be defined.

DEFINITION 8.4 [9] The *generalized Lucas test* declares an odd integer n probable prime for the parameters $P, Q, R \in \mathbb{Z}$ with $D = P^2 - 4QR$ if $j = \left(\frac{D}{n}\right) \neq 0$, $\gcd(n, QR) = 1$ and

$$(\tilde{V}_{n-j}, \tilde{U}_{n-j}) \equiv \begin{cases} (1, 0) \pmod{n}, & \text{if } j = 1, \\ (QR, 0) \pmod{n}, & \text{if } j = -1. \end{cases}$$

A composite n that passes this test is called *generalized Lucas pseudoprime with parameters* P, Q, R (glpsp(P, Q, R)).

As for the generalized Pell test, some results on the equivalence among generalized Lucas tests with different parameters are shown. In particular, the following one holds also for all the other tests based on Lucas sequences.

PROPOSITION 8.6 The generalized Lucas test is independent of the sign of the parameter $P \in \mathbb{Z}$.

Proof. The linear recurrent sequences defined in Equation 8.3 are a generalization of the Lucas sequences $(U_k)_{k\geqslant 0}$ and $(V_k)_{k\geqslant 0}$ introduced in Equation 7.1, and can be described as

$$\begin{cases} \tilde{U}_0 = 0, \quad \tilde{U}_1 = R, \\ \tilde{U}_k = P\tilde{U}_{k-1} - QR\tilde{U}_{k-2}, \end{cases} \qquad \begin{cases} \tilde{V}_0 = 1, \quad \tilde{V}_1 = P, \\ \tilde{V}_k = P\tilde{V}_{k-1} - QR\tilde{V}_{k-2}. \end{cases} \tag{8.4}$$

If the sign of $P \in \mathbb{Z}$ is changed, then the obtained sequences are

$$\begin{cases} \tilde{U}'_0 = 0, \quad \tilde{U}'_1 = R, \\ \tilde{U}'_k = -P\tilde{U}'_{k-1} - QR\tilde{U}'_{k-2}, \end{cases} \qquad \begin{cases} \tilde{V}'_0 = 1, \quad \tilde{V}'_1 = -P, \\ \tilde{V}'_k = -P\tilde{V}'_{k-1} - QR\tilde{V}'_{k-2}, \end{cases}$$

so that, for any $k \geqslant 0$,

$$\begin{cases} \tilde{U}'_k = (-1)^{k+1}\tilde{U}_k, \\ \tilde{V}'_k = (-1)^k\tilde{V}_k. \end{cases}$$

This can be verified by induction on the index k:

- if $k = 0$, then $\tilde{U}'_0 = -\tilde{U}_0 = 0$ and $\tilde{V}'_0 = \tilde{V}_0 = 1$;
- if $k = 1$, then $\tilde{U}'_1 = \tilde{U}_1 = R$ and $\tilde{V}'_1 = -\tilde{V}_1 = -P$;
- if $k > 1$ is even, then assuming the thesis true for $k-1$ and $k-2$ results in having

$$\begin{cases} \tilde{U}'_k = -P\tilde{U}'_{k-1} - QR\tilde{U}'_{k-2} = -P\tilde{U}_{k-1} + QR\tilde{U}_{k-2} = -\tilde{U}_k, \\ \tilde{V}'_k = -P\tilde{V}'_{k-1} - QR\tilde{V}'_{k-2} = P\tilde{V}_{k-1} - QR\tilde{V}_{k-2} = \tilde{V}_k; \end{cases}$$

- if $k > 1$ is odd, then assuming the thesis true for $k-1$ and $k-2$ results in having

$$\begin{cases} \tilde{U}'_k = -P\tilde{U}'_{k-1} - QR\tilde{U}'_{k-2} = P\tilde{U}_{k-1} - QR\tilde{U}_{k-2} = \tilde{U}_k, \\ \tilde{V}'_k = -P\tilde{V}'_{k-1} - QR\tilde{V}'_{k-2} = -P\tilde{V}_{k-1} + QR\tilde{V}_{k-2} = -\tilde{V}_k. \end{cases}$$

In the generalized Lucas test, $k = n \pm 1$ with n odd, so that the interesting case is k even. Thus, when changing the sign of $P \in \mathbb{Z}$, the check on $\tilde{V}_k$ remains unchanged, while $\tilde{U}_{n\pm 1} \equiv 0 \pmod{n}$ if and only if $-\tilde{U}_{n\pm 1} \equiv 0 \pmod{n}$. □

This result is true for all tests based on Lucas sequences, so that when studying these tests with fixed parameters for testing different integers, it is sufficient to focus only on the instances with $P \geqslant 0$.

PROPOSITION 8.7 The generalized Lucas test is independent of the choice of $Q, R \in \mathbb{Z}$, as long as the value of QR remains unchanged.

Proof. The generalized Lucas sequences with parameters $P, Q, R \in \mathbb{Z}$ can be compared with the classical Lucas sequences with parameters P and $Q' = QR$. In particular, by induction on the index $k \geqslant 0$, it is possible to prove that $\tilde{U}_k = RU_k$ and $\tilde{V}_k = V_k$:

- if $k = 0$, then $\tilde{U}_0 = RU_0 = 0$ and $\tilde{V}_0 = V_0 = 1$;
- if $k = 1$, then $\tilde{U}_1 = RU_1 = R$ and $\tilde{V}_1 = V_1 = P$;
- if $k > 1$, then assuming the thesis true for $k-1$ and $k-2$ results in having

$$\begin{aligned}\tilde{U}_k &= P\tilde{U}_{k-1} - QR\tilde{U}_{k-2} = PRU_{k-1} - QR^2U_{k-2} \\ &= R(PU_{k-1} - Q'U_{k-2}) = RU_k,\end{aligned}$$

as well as

$$\begin{aligned}\tilde{V}_k &= P\tilde{V}_{k-1} - QR\tilde{V}_{k-2} = PV_{k-1} - QRV_{k-2} \\ &= PV_{k-1} - Q'V_{k-2} = V_k.\end{aligned}$$

Thus, the generalized tests with parameters $P, Q, R \in \mathbb{Z}$ is equivalent to the double Lucas test with parameters $P, Q' = QR$.

Given two generalized Lucas tests with parameters $P, Q, R \in \mathbb{Z}$ and $P, Q', R' \in \mathbb{Z}$, respectively, if $QR = Q'R'$, then they are both equivalent to the same double Lucas test and the thesis is verified. □

The first consequence of this result is that each glpsp(P, Q, R) is a dlpsp(P, QR), and vice versa, so that for all sufficiently large x, the number $\mathrm{GLP}(x)$ of glpsp(P, Q, R) not exceeding x is

$$\mathrm{GLP}(x) = \mathrm{DLP}(x) < x \exp\left(-\frac{\log x \log\log\log x}{2\log\log x}\right),$$

or analogously with the upper bound for the number $\mathrm{FP}(x)$ of Frobenius pseudoprimes with respect to $x^2 - Px + QR$.

Despite this equivalence makes the generalized Lucas test less important, adapting the Selfridge method gives very interesting results. As for the other tests based on Lucas sequences, the idea is to test the integer n with parameters that have discriminant D such that $\left(\frac{D}{n}\right) = -1$, so that the test is not equivalent to a strong Fermat test. The resulting method for choosing the parameters involves to:

1. fix $P, R > 0$;
2. take $D \in \left\{P^2 - 4RQ \mid Q \in \mathbb{Z} \setminus \{0\}\right\}$ with $\left(\frac{D}{n}\right) = -1$ and $|D|$ minimum;
3. evaluate $Q = \frac{P^2 - D}{4R}$.

In the following section an analysis on the resulting pseudoprimes with different fixed parameters is conducted in order to find their best values.

x	0			1			2			3			
$D \setminus y$	1	2	3	1	2	3	1	2	3	1	2	3	μ_D
5	108	144	126	17	11	14	178	11	14	19	7	17	56
-7	102	104	137	11	3	21	3	7	10	9	5	6	35
-11	92	156	157	4	6	12	4	7	8	2	8	7	39
13	106	120	132	7	2	8	4	8	10	6	4	4	34
-15	81	118	156	8	15	10	5	24	10	7	9	15	38
17	108	187	109	12	8	5	6	12	5	8	5	6	39
$\mu_{(x,y)}$	100	138	136	10	8	12	33	12	10	9	6	9	

Table 14: Number of gppsp(D, x, y) up to 2^{20} for different values of the parameters D, (x, y) and their arithmetic means with fixed D or (x, y).

8.5 NUMERICAL EXPERIMENTS

Table 14 collects the number of pseudoprimes up to $2^{20} = 1.048.576$ for the generalized Pell test with different choices of the parameters:

- D is taken among the first six non–square values used in the Selfridge method (the average number of D to be tried is less than 2 [4]), which are the interesting cases because of the relation between generalized Pell test and double Lucas test;
- (x, y) has integer coordinates between 0 and 3, since negative values behave as positive ones because of Proposition 8.4. Points with coordinate $y = 0$ can be excluded because Proposition 8.5 assures that they are equivalent to cases with $D = 0$.

The collected data strongly depend on the values of the parameters. However, their arithmetic means for fixed D or (x, y), shown in the last column and row, respectively, allow to understand which values can be considered more reliable, for example in an adaptation of the Selfridge method introduced in Section 7.3.2.

In particular, if the parameters for the generalized Pell test are not fixed for each integer n to be tested, but instead are taken as:

- $D \in \{5, -7, 9, -11, \ldots\}$ with $j = \left(\frac{D}{n}\right) = -1$ and $|D|$ minimum;
- $(x, y) = (3, 2)$, the case with lowest arithmetic mean in Table 14;

then an alternative version of the Selfridge method for selecting the parameters is obtained and, as for the classical version, it works empirically better than fixing the parameters independently of n.

In particular, when adopting this method for testing all the odd integers smaller than $2^{44} = 17.592.186.044.416$, each number was correctly declared prime or composite, i.e., no generalized Pell pseudoprimes were found.

P	0			1			2			3			
Q \ R	1	2	3	1	2	3	1	2	3	1	2	3	μ_Q
1	–	118	128	–	2	6	–	118	9	165	251	128	103
−1	–	118	128	60	223	4	121	18	250	74	4	5	91
2	118	251	140	2	20	4	118	223	4	251	13	7	96
−2	118	251	140	223	8	60	18	17	6	4	409	9	105
3	128	140	250	6	4	6	9	4	25	128	7	250	80
−3	128	140	250	4	60	11	250	6	9	5	9	9	73
$\mu_{P,R}$	123	170	173	59	53	15	103	64	51	105	116	68	

Table 15: Number of glpsp(P, Q, R) up to 2^{20} for different values of the parameters P, Q, R and their arithmetic means with fixed Q or P, R.

Analogously, Table 15 shows the number of pseudoprimes up to 2^{20} for the generalized Lucas test with different parameters:

- $0 \leqslant P \leqslant 3$, since negative values of P can be excluded thanks to Proposition 8.6;
- Q can be positive or negative and, in particular, is taken among the first six values obtained by the Selfridge method;
- $1 \leqslant R \leqslant 3$, since Proposition 8.7 assures that it is possible to consider only positive values of R.

Since cases with same value of QR are equivalent, they return the same number of pseudoprimes, but they are collected in order to study the behavior of the test with fixed P, R for the adaptation of the Selfridge method.

Some trivial cases are excluded because they generate sequences for which Equation 8.3 is satisfied by many odd integers, namely:

- $P = 0$, $Q = R = 1$, related to $\left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ that has period 4;
- $P = 0$, $Q = -1$, $R = 1$, related to $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ that has period 2;
- $P = Q = R = 1$, related to $\left(\begin{smallmatrix} 1 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ that has period 6.

The case with $P = 2$, $Q = R = 1$ is also excluded since its discriminant D is zero. When $R = 1$ the test is simply the double Lucas tests, but these cases are included for the sake of completeness, as well as the cases with $P = 0$ in which $\tilde{U}_{2k} = 0 \ \forall k \geqslant 0$, i.e., in the test only the check on $\tilde{V}_{n-j} = (-QR)^{\frac{n-j}{2}}$ is significant.

The collected quantities strongly depend on the chosen parameters. Table 15 contains also the arithmetic means of the values for fixed Q in the last column and for fixed P, R in the last row, which allow to understand what are the most reliable choices.

When adapting the Selfridge method to the generalized Lucas test, the best choices for the fixed parameters and the consequent set of possible values of D are, in order of best statistical results:

1. $P = 1$, $R = 3$ and $D \in \{-11, 13, -23, 25, \ldots\}$;
2. $P = 2$, $R = 3$ and $D \in \{-8, 16, -20, 28, \ldots\}$;
3. $P = 1$, $R = 2$ and $D \in \{-7, 9, -15, 17, \ldots\}$;
4. $P = 1$, $R = 1$ and $D \in \{5, -7, 9, -11, \ldots\}$ gives the double Lucas test with the Selfridge method, the first pseudoprime is 5777.

In all these cases, D is taken such that $\left(\frac{D}{n}\right) = -1$ and $|D|$ is minimum, while $Q = \frac{P^2 - D}{4R}$.

In particular, when testing the integers up to 2^{44} with the parameters obtained through method 1, only primes were declared primes, i.e., no generalized Lucas pseudoprime with parameters $P = 1$, $R = 3$ and Q from Selfridge was found. Thus, as for the classical Selfridge method, this adaptation seems to work empirically better than fixing the same parameters for different n.

CONCLUSIONS

In this work, the Pell equation and different classical and new generalizations have been studied and applied to different branches of cryptography.

The first part was a theoretical study with the target to obtain as much information as possible, while introducing a versatile notations and focusing on the requirements for the wanted applications.

Starting from the famous Diophantine equation that took the interest of Archimedes, its properties in generic and finite fields, and as a group structure have been collected all under the same notation. In particular, the solutions of the Pell equation have been associated to a conic with an efficient and useful parametrization to a projectivization that allows to reduce the size of each point. The obtained results concern the solutions over generic fields and also finite fields, for all the possible choices of the parameter D. The sets have also been equipped with an operation inspired by Brahmagupta, resulting in a cyclic group. Moreover, the generalization of the Pell equation with a constant term $Q \neq 1$ have been studied together with a generalization of the operation between the points. Among the obtained results, it has been proven that all the Pell conics with the same quadratic character of D are isomorphic and the map has been explicitly obtained. Finally, by exploiting a relation with Rédei polynomials and rational functions, an efficient algorithm for the exponentiation on the projectivization is recalled using the introduced notation.

In the same way, the properties of the generalization of the Pell equation in cubic case over generic and finite fields have been deepened. Its solutions over a field have been associated to a cubic with an operation obtained analogously to the quadratic case. When dealing with finite fields, a parametrization was found for any choice of the parameter R, but the structure is generally a group, but it is not cyclic for any R. In some cases, the explicit inverse of the parametrization is still unknown. Moreover, finding the same results for a generic field is still an open issue. As for the quadratic case, the study moved to the generalization of the cubic Pell equation with a constant term $Q \neq 1$. After defining a generalization of the operation between the points, the same parametrization for the simple case is adapted to the generalized case and, when the inverse is explicitly known, a group isomorphism between any Pell cubic with same cubic character of R can be obtained. The study is concluded by generalizing the Rédei polynomials and rational functions in order to obtain a new efficient algorithm for the exponentiation on the projectivization related to the Pell cubic.

The second part of this work focused on the public–key cryptosystems that can be obtained by exploiting the results from the first part.

Firstly, after recalling the state of the art with particular attention to the schemes based on the Pell equation or similar constructions, new enhancements and formulations with security based on the classical IFP and DLP are introduced. For all the proposals, a practical implementations written in Python is developed and, when possible, the efficient exponentiation algorithm is adopted. Specifically, the problem of an RSA–like cryptosystem from the literature has been solved thanks to the obtained results. However, despite the promising expectations, in a comparison with the classical RSA with same security strength and maximum message length, the new cryptosystem needs still to be optimized. On the other hand, the classical ElGamal cryptosystem is formulated using the cyclic groups related to the Pell conic. Besides two standard and scholastic formulations that are not comparable to the classical version in FFC and ECC, an alternative formulation that exploits the found explicit group isomorphisms seem to be a good competitor. In particular, it works better than both the classical versions in all the standard security levels, except for the highest, while maintaining the smallest data–size when the maximum message length is fixed. Moreover, also DSA is tackled: the proposed formulation uses the projectivization with D non–square. For lower security levels, in a trade–off between performance and size of the public data, the classical DSA is best for the former, while the proposal for the latter. When higher security is required, ECDSA is still the best option for both data–size and performance.

Following the structure of the first part, the same work has been conducted using the Pell cubic. Considering the different cases depending on the choice of R, two new RSA–like cryptosystem were introduced following a suggestion from the literature. However, the obtained schemes are not even comparable to the formulation with the Pell conic since the available choices for R are not optimal. From the point of view of DLP–based cryptosystems, ElGamal and DSA have again been addressed. Also in this case, the formulations with the cyclic groups related to the Pell cubic are not as efficient as the classical versions or the best one for the quadratic case. Generally, the problem is due to the missing inverses of the parametrizations with the optimal choices for R, but this can be a starting point since finding them could still lead to competitive cryptosystems.

The third and final part focused on probabilistic primality tests, with particular attention to the tests based on the Lucas sequences. This interest arises from a link between those sequences and the one obtained from the powers of a point on a Pell conic. Firstly, the state of the art for probabilistic primality tests is described, together with some specific tests that exploit sequences of integers modulo the n to be tested.

Then, after generalizing the Lucas test through the definition of primality tests based on linear recurrent sequences, an enhanced version of the Lucas test is defined by considering an additional congruence. This can be related to a primality test based on the Pell conic, which leads to a generalization that exploits the generalized Pell equation with constant term $Q \neq 1$. In particular, thanks to an analysis of the test with fixed parameters, the Selfridge method used for the Lucas tests can be adapted in order to choose good parameters depending on the integer to be tested. Analogously, a further generalization of the Lucas test has been defined and the Selfridge method was adapted. Both the resulting primality tests are very powerful since no pseudoprimes smaller than 2^{44} were found.

Also in this context there are still open issues, since further study on the pseudoprimes for these tests could be useful. In particular, finding the first pseudoprime, if any exists, is of great importance, especially in a comparison with the standard Baillie–PSW test, for which it is conjectured that there are infinitely many pseudoprimes, but no one was found yet.

BIBLIOGRAPHY

[1] A. Aabrandt and V. L. Hansen. "A Note on Powers in Finite Fields." In: *International Journal of Mathematical Education in Science and Technology* 47.6 (2016), pp. 987–991.

[2] M. Agrawal, N. Kayal, and N. Saxena. "PRIMES is in P." In: *Annals of Mathematics* 160.2 (2004), pp. 781–793.

[3] W. R. Alford, A. Granville, and C. Pomerance. "There are infinitely many Carmichael numbers." In: *Annals of Mathematics* 140 (1994), pp. 703–722.

[4] R. Baillie and S. S. Jr. Wagstaff. "Lucas Pseudoprimes." In: *Mathematics of Computation* 35.152 (1980), pp. 1391–1417.

[5] C. Ballot. "Strong arithmetic properties of the integral solutions of $X^3 + DY^3 + D^2Z^3 - 3DXYZ = 1$, where $D = M^3 \pm 1$, $M \in \mathbb{Z}^*$." In: *Acta Arithmetica* 89 (1999), pp. 259–277.

[6] E. J. Barbeau. *Pell's Equation*. New York-Berlin: Springer-Verlag, 2003.

[7] S. Barbero, U. Cerruti, and N. Murru. "Generalized Rédei rational functions and rational approximations over conics." In: *International Journal of Pure and Applied Mathematics* 64 (2010), pp. 305–317.

[8] E. Barker. *SP 800-57 Part 1: Recommendation for Key Management*. Tech. rep. NIST, 2020.

[9] D. Bazzanella, A. J. Di Scala, S. Dutto, and N. Murru. "Primality tests, linear recurrent sequences and the Pell equation." In: *The Ramanujan Journal* 57.1 (2022), pp. 755–768.

[10] M. Bellare and P. Rogaway. "Optimal asymmetric encryption." In: *Advances in Cryptology – EUROCRYPT '94*. Springer, 1995, pp. 92–111.

[11] E. Bellini and N. Murru. "An efficient and secure RSA–like cryptosystem exploiting Rédei rational functions over conics." In: *Finite Fields and their Applications* 39 (2016), pp. 179–194.

[12] E. Bellini, N. Murru, A. J. Di Scala, and M. Elia. "Group law on affine conics and applications to cryptography." In: *Applied Mathematics and Computation* 409.125537 (2021).

[13] L. Bernstein. "Fundamental units from the preperiod of a generalized Jacobi–Perron algorithm." In: *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1974.268–269 (1974), pp. 391–409.

[14] L. Bernstein. "Units and Periodic Jacobi–Perron Algorithms in Real Algebraic Number Fields of Degree 3." In: *Transactions of the American Mathematical Society* 212 (1975).

[15] J. Bourgain. "A Remark on Solutions of the Pell Equation." In: *International Mathematics Research Notices* 2015.10 (2015), pp. 2841–2855.

[16] D. R. L. Brown. *SEC 2: Recommended Elliptic Curve Domain Parameters (Ver. 2.0)*. Tech. rep. Certicom Research, 2010.

[17] Z. Chen and Greene J. "Some comments on Baillie–PSW pseudoprimes." In: *Fibonacci Quarterly* 41.4 (2003), pp. 334–344.

[18] M. Cipu. "Explicit formula for the solution of simultaneous Pell equations $x^2 - (a^2 - 1)y^2 = 1$, $y^2 - bz^2 = v_1^2$." In: *Proceedings of the American Mathematical Society* 146.3 (2018), pp. 983–992.

[19] B. Cohen. "Chebyshev polynomials and Pell equations over finite fields." In: *Czechoslovak Mathematical Journal* 71 (2021), pp. 491–510.

[20] P. H. Daus. "Normal Ternary Continued Fraction Expansions for the Cube Roots of Integers." In: *American Journal of Mathematics* 51.1 (1929), pp. 67–98.

[21] A. J. Di Scala, N. Murru, and C. Sanna. "Lucas pseudoprimes and the Pell conic." In: *arXiv:2001.00353* (2020).

[22] L. E. Dickson. *History of the Theory of Numbers, Vol. II, Diophantine analysis*. Carnegie Institution of Washington, 1920.

[23] W. Diffie and M. Hellman. "New directions in cryptography." In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654.

[24] T. ElGamal. "A Public–Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms." In: *IEEE Transactions on Information Theory* 31 (1985), pp. 469–472.

[25] P. Erdös, P. Kiss, and Sárközy. "A Lower Bound for the Counting Function of Lucas Pseudoprimes." In: *Mathematics of Computation* 51.183 (1988), pp. 315–323.

[26] *FIPS 186-4: Digital Signature Standard (DSS)*. Tech. rep. National Institute of Standards and Technology, 2013.

[27] *FIPS 186: Digital Signature Standard (DSS)*. Tech. rep. National Institute of Standards and Technology, 1994.

[28] E. Fouvry. "On the size of the fundamental solution of the Pell equation." In: *Journal für die reine und angewandte Mathematik (Crelles Journal)* 2016.717 (2016), pp. 1–33.

[29] R. Fu and H. Yang. "On the solvability of the simultaneous Pell equations $x^2 - ay^2 = 1$ and $y^2 - bz^2 = v_1^2$." In: *International Journal of Number Theory* 17.9 (2021), pp. 1997–2008.

[30] D. M. Gordon and C. Pomerance. "The Distribution of Lucas and Elliptic Pseudoprimes." In: *Mathematics of Computation* 57.196 (1991), pp. 825–838.

[31] J. Grantham. "Frobenius Pseudoprimes." In: *Mathematics of Computation* 70.234 (2000), pp. 873–891.

[32] M. Gysin and J. Sebery. "How to use Pell's equation in cryptography." In: *preprint* (1999).

[33] B. He, A. Pinter, and A. Togbé. "On simultaneous Pell equations and related Thue equations." In: *Proceedings of the American Mathematical Society* 143.11 (2015), pp. 4685–4693.

[34] C. G. J. Jacobi. *Gesammelte Werke*. Vol. VI. Berlin: Reimer, 1891, pp. 385–426.

[35] M. J. Jacobson and H. C. Williams. *Solving the Pell Equation*. CMS Books in Mathematics. New York: Springer, 2009.

[36] K. Koyama. "Fast RSA–type schemes based on singular cubic curves $y^2 + axy \equiv x^3 \pmod{n}$." In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1995, pp. 329–340.

[37] F. Lemmermeyer. *Introduction to Cryptography*. Citeseerx, 2006.

[38] H. W. Lenstra Jr. and C. Pomerance. "Primality testing with Gaussian periods." In: *Journal of the European Mathematical Society* 21 (2019), 1229–1269.

[39] R. Lidl, G. L. Mullen, and Turnwald G. *Dickson polynomials*. Pitman Monographs and Surveys in Pure and Applied Mathematics. New York: Longman, 1993.

[40] E. Lucas. "Sur les rapports qui existent entre Ia theorie des nombres et le calcul integral." In: *Comptes Rendus Paris* 82 (1876), pp. 1303–1305.

[41] G. B. Mathews. "On the arithmetic theory of the form $x^3 + ny^3 + n^2z^3 - 3nxyz$." In: *Proceedings of the London Mathematical Society* S1-21.1 (1889), pp. 280–287.

[42] A. J. Menezes and S. A. Vanstone. "A note on cyclic groups, finite fields, and the discrete logarithm problem." In: *Applied Algebra in Engineering, Communication and Computing* 3 (1992), pp. 67–74.

[43] G. L. Miller. "Riemann's Hypothesis and Tests for Primality." In: *Journal of Computer and System Sciences* 13.3 (1976), pp. 300–317.

[44] A. M. Mishra. "A Digital Signature Scheme Based on Pell Equation." In: *International Journal of Innovative Research in Science, Engineering and Technology* 3.1 (2014), pp. 8596–8600.

[45] W. More. "Fast Evaluation of Rédei Functions." In: *Applicable Algebra in Engineering, Communication and Computing* 6.3 (1995), pp. 171–173.

[46] N. Murru and F. M. Saettone. "A Novel RSA–Like Cryptosystem Based on a Generalization of the Rédei Rational Functions." In: 10737 LNCS (2018), pp. 91–103.

[47] N. R. Murthy and M. N. S. Swamy. "Cryptographic applications of Brahmagupta–Bhaskara equation." In: *IEEE Transactions on Circuits and Systems I* 53 (2006), pp. 1565–1571.

[48] Sahadeo Padhye. "A Public Key Cryptosystem Based on Pell Equation." In: *IACR Cryptology ePrint Archive* 2006/191 (2006).

[49] C. Pomerance. "Are There Counterexamples to the Baillie–PSW Primality Test?" In: *unpublished* (1984).

[50] C. Pomerance, J. L. Selfridge, and S. S. Wagstaff. "The pseudoprimes to $25 \cdot 10^9$." In: *Mathematics of Computation* 35.151 (1980), pp. 1003–1026.

[51] M. O. Rabin. "Probabilistic algorithm for testing primality." In: *Journal of Number Theory* 12.1 (1980), pp. 128–138.

[52] R. L. Rivest, A. Shamir, and L. Adleman. *On Digital Signatures and Public–Key Cryptosystems*. Tech. rep. Massachusetts Inst of Tech Cambridge Lab for Computer Science, 1977.

[53] L. Rédei. "Uber eindeuting umkehrbare polynome in endlichen korpen." In: *Acta Scientiarum Mathematicarum (Szeged)* 11 (1946), pp. 85–92.

[54] A. Teckan. "The number of solutions of Pell equations $x^2 - ky^2 = N$ and $x^2 + xy - ky^2 = N$ over $\mathbb{F}_p$." In: *Ars Combinatorica* 102 (2011), pp. 225–236.

[55] A. Tekcan, A. Ozkoc, C. Kocapinar, and H. Alkan. "The Pell equation $x^2 - Py^2 = Q$." In: *International Journal of Physical and Mathematical Sciences* 4.7 (2010), pp. 795–798.

[56] S. Vanstone. "Responses to NIST's proposal." In: *Communications of the ACM* 35.7 (1992), pp. 50–52.

[57] A. Weil. *Number theory: an approach through history*. Boston: Birkhauser, 1984.

[58] M. J. Wiener. "Cryptanalysis of short RSA secret exponents." In: *IEEE Transactions on Information Theory* 36.3 (1990), pp. 553–558.

[59] C. L. E. Wolfe. "On the indeterminate cubic equation $x^3 + Dy^3 + D^2z^3 - 3Dxyz = 1$." In: *University of California Publications in Mathematics* 1.16 (1923), pp. 359–369.

[60] P. Xi. "Counting fundamental solutions to the Pell equation with prescribed size." In: *Compositio Mathematica* 154 (2018), pp. 2379–2402.