Criptografía multivariable, bases de Groebner y ataques

Emilio Casanova Biscarri

Trabajo de fin de grado en Matemáticas
Universidad de Zaragoza

Directores del trabajo: José Ignacio Cogolludo Agustín y Jorge Martín Morales

julio de 2019

Prólogo

La necesidad humana de codificar mensajes data de tiempos remotos. En el Antiguo Egipto encontramos uno de los primeros métodos de encriptado de mensajes conocidos: los jeroglíficos. El arte de la criptografía se encuentra detrás de grandes acontecimientos históricos, como el caso Dreyfus, o la máquina Enigma y su relevancia en el desarrollo de la Segunda Guerra Mundial. A lo largo de la historia, las técnicas criptográficas han ido evolucionando, bien sea modificando criptosistemas existentes para evitar ataques, o bien desarrollando nuevos métodos de encriptado.

La presencia de un mayor desarrollo tecnológico, como la presencia de ordenadores cuánticos, obliga al campo de la criptografía a reinventarse constantemente. En este trabajo de fin de grado nos situamos precisamente bajo esta premisa. Estudiaremos los recursos algebraicos que sustentan el desarrollo de la criptografía algebraica mediante el uso de bases de Groebner, teoría desarrollada por Bruno Buchberger en 1965 en su tesis doctoral [1]. Desde entonces, numerosos autores han contribuido a crear criptosistemas bajo los que subyace dicha teoría, así como diversos ataques hacia los mismos.

En el primer capítulo se tratan los fundamentos algebraicos que avalan dicha teoría y las motivaciones que llevaron a Bruno Buchberger a crearla. En el segundo capítulo se presenta una familia de criptosistemas basada en la teoría de bases de Groebner, denominados criptosistemas *Barkee*, y comprobaremos su seguridad ante ciertos ataques. En el tercer y último capítulo se estudian los criptosistemas *Polly Cracker*, un caso particular de los criptosistemas *Barkee*, y estudiaremos las susceptibilidades que presentan. Por último, la memoria finaliza con una serie de conclusiones y comentarios fruto del estudio realizado.

Palabras clave: *criptografía, álgebra, bases de Groebner, Barkee, Polly Cracker, ataques.*

Summary

In the present work we provide the reader several theoretical results and practical examples related to the field of multivariate cryptography. More specifically, we will deepen our work in a family of cryptosystems based on the use of Gröbner bases.

This document is structured in 4 different chapters, focused on the following topics.

Chapter 1 - Algebraic background

First of all, we introduce the notion of ideal. In order to determine whether a polynomial lies in an ideal of $K[x]$ or not, we are forced to use the division algorithm. Subsequently, we extend the division algorithm with the aim of solving the ideal membership problem in the multivariate case. This generalization is shown to be an "imperfect" extension of the previous one, since it doesn't ensure the uniqueness of the remainder if we change the order of the divisors.

At this point, we introduce some concepts about monomial ideals. Dickson's Lemma, a key result, guarantees that every multivariate monomial ideal is finitely generated. In addition, the Hilbert Basis Theorem ensures that every multivariate ideal is finitely generated. After this, the cornerstone of our theory is presented: Gröbner bases. They are a particular kind of basis strongly related to monomial ideals that let us ensure the uniqueness of the remainder in the multivariate division algorithm when used as divisors.

The Hilbert Basis Theorem thereupon is used to show the existence of a Gröbner basis for every multivariate ideal. A very practical algorithm to create Gröbner bases, called Buchberger's algorithm in honor of his creator, can also be found at the end of the chapter.

Finally, since the uniqueness of Gröbner bases for a given ideal doesn't hold, some extra properties are required to establish a bijection between polynomial ideals and Gröbner bases. Such bases are called reduced Gröbner bases.

Chapter 2 - Barkee cryptosystems

The next chapter deals with a particular family of cryptosystems called Barkee cryptosystems, introduced by B. Barkee, which are public-key cryptosystems whose private keys are Gröbner basis. The receiver just has to divide the ciphertext by its private key, since Gröbner basis are created in order to ensure the uniqueness of the remainder, to reveal the message. Its security relies on the fact that the creation of a Gröbner basis for a given ideal should be (almost) infeasible. Two versions of these cryptosystems are shown: the first one uses dense polynomials. At this stage, we present the Moriarty attack, which can easily break these cry-

ptosystems. The second version of our cryptosystems, based on using sparse polynomials, help us overcome this situation.

Chapter 3 - Polly Cracker cryptosystems

Polly Cracker cryptosystems were developed by M. Fellows and N. Koblitz independently from the Barkee cryptosystems. Nevertheless, they are based on the same principles. The main difference between them is that Polly Cracker cryptosystems consider the private key to be a point, and decryption is archieved evaluating our ciphertext. A very suggestive example of a Polly Cracker cryptosystem linked to graph theory can be found in this section. After this, we resume our Moriarty attack to present a concrete attack to a dense Polly Cracker cryptosystem, and we verify that our sparse Polly Cracker version is better off than the dense one. In addition, we present a digital signature scheme commonly used to ensure the origin of messages in public-key cryptosystems.

Conclusion

Last but not least, this document concludes emphasising the constant effort that takes to prevent cryptographic attacks. When Gröbner basis-based cryptography started being developed, its aim was to avoid the existent attacks that broke the algebraic cryptosystems used back then. Nowadays, several attacks which compromise the reliability of this family of cryptosystems have been created. Furthermore, quantum computers present an imminent threat to their security that has to be prevented.

Keywords: *cryptography, algebra, Gröbner bases, Barkee, Polly Cracker, attacks*.

Índice general

Capítulo 1

Fundamentos algebraicos

En este primer capítulo se encuentra la base teórica sobre la que se sustenta la criptografía algebraica mediante el uso de bases de Groebner. La gran mayoría de los contenidos han sido extraídos del libro *Ideals, Varieties, and Algorithms*[2], que abarca ampliamente las cuestiones aquí abordadas.

1.1. Ideales

Definición 1. *Un subconjunto $I \subset K[x_1,...,x_n]$ se denomina un **ideal** si satisface las siguientes condiciones:*

(i) $0 \in I$,

(ii) *si* $f, g \in I$, *entonces* $f+g \in I$,

(iii) *si* $f \in I$, *y* $h \in K[x_1,...,x_n]$, *entonces,* $hf \in I$.

Definición 2. *Sean f_1,...,f_s polinomios en K[x_1,...,x_n], denotamos*

$$\langle f_1,...,f_s\rangle = \{\textstyle\sum_{i=1}^{s} h_i f_i : h_1,...,h_s \in K[x_1,...,x_n]\}.$$

*Al conjunto $\langle f_1,...,f_s\rangle$ se le denomina **ideal generado por** $f_1,...,f_s$.*

Demostración. Dicho conjunto es, en efecto, un ideal de $K[x_1,...x_n]$: $0 \in \langle f_1,...,f_s\rangle$, dado que $0 = \sum_{i=1}^{s} 0 f_i$. Además, supongamos que $f = \sum_{i=1}^{s} p_i f_i$, que $g = \sum_{i=1}^{s} q_i f_i$, y que $h \in K[x_1,...,x_n]$. Entonces, $f+g = \sum_{i=1}^{s}(p_i+q_i)f_i \in \langle f_1,...,f_s\rangle$, y $hf = \sum_{i=1}^{s}(hp_i)f_i \in \langle f_1,...,f_s\rangle$. □

Definición 3. *Diremos que un ideal I es **finitamente generado** si existen $f_1,...,f_s \in K[x_1,...,x_n]$ tales que $I = \langle f_1,...,f_s\rangle$. En tal caso, diremos que $f_1,...,f_s$ es una **base** de I.*

1.2. Polinomios en una variable

En esta subsección, abordaremos ciertas propiedades del anillo de polinomios $K[x]$.

Definición 4. *Dado $0 \neq f \in K[x]$, con $f = a_0x^m + a_1x^{m-1} + ... + a_m$, donde $a_i \in K$, y $a_0 \neq 0$, i.e. $gr(f) = m$, definimos el **término inicial** de f como $in(f) = a_0x^m$.*

Teorema 1. ***(Algoritmo de la división en** $K[x]$**)***
Sean K un cuerpo y $0 \neq g \in K[x]$, entonces, todo polinomio $f \in K[x]$ puede escribirse como

$$f = qg + r,$$

donde $q, r \in K[x]$, y o bien $r = 0$, o $gr(r) < gr(g)$. Además, dichos q y r son ***únicos****, y existe un algoritmo para calcularlos.*

La demostración se puede encontrar en el Anexo. Se sustenta en el siguiente algoritmo, presentado en formato de pseudocódigo.

Datos: g, f
Resultado: q, r
Inicialización: $q := 0, r := f$
mientras $r \neq 0$ **y** $in(g)$ *divida a* $in(r)$ **hacer**
  $q := q + in(r)/in(g)$
  $r := r - (in(r)/in(g))g$
fin

Algoritmo 1: Algoritmo de la división en $K[x]$

Ejemplo 1. *Hagamos un ejemplo para ilustrar el funcionamiento del algoritmo de la división en $\mathbb{F}_2[x]$. Vamos a dividir $f = x^7 + x^3 + 1$ entre $g = x^2 + 1$. Buscamos hallar el cociente q y el resto r. Inicializamos $q = 0$ y $r = f$.*

Paso 1: Como $r \neq 0$ y $in(g) = x^2$ divide a $in(r) = x^7$, actualizamos $q = 0 + x^7/x^2 = x^5$ y $r = (x^7 + x^3 + 1) - (x^7/x^2)(x^2 + 1) = x^5 + x^3 + 1$.

Paso 2: De nuevo, $r \neq 0$ y $in(g) = x^2$ divide a $in(r) = x^5$, actualizamos $q = x^5 + x^3$ y $r = 1$.

Paso 3: Aunque $r \neq 0$, $in(g) = x^2$ no divide a $in(r) = 1$, ergo el algoritmo finaliza, y nos aporta la siguiente información:

$$f = qg + r = (x^5 + x^3)(x^2 + 1) + 1. \square$$

Es bien conocido, aunque puede encontrarse una demostración de ello en el Anexo, que el anillo de polinomios en una variable $K[x]$ es un dominio de ideales principales (DIP). Es decir, todo ideal contenido en él es generado por un único elemento. En este punto, nos planteamos dos problemas:

(I) Problema de la descripción del ideal: ¿Existe algún método para calcular un generador de cualquier ideal en $K[x]$?

(II) Problema de la pertenencia al ideal: ¿Dados $f_1, ..., f_s \in K[x]$, existe un algoritmo para determinar si cualquier $f \in K[x]$ pertenece a $\langle f_1, ..., f_s \rangle$?

Afortunadamente, podemos responder afirmativamente a ambas cuestiones en el caso de una variable. La respuesta al primer problema (I) nos la proporciona el máximo común divisor $MCD(f_1, ..., f_s)$, dado que este es un generador del ideal $\langle f_1, ..., f_s \rangle$. La justificación y existencia de un algoritmo para su cálculo (algoritmo de Euclides) se puede encontrar en el Anexo.

Respondamos afirmativamente a nuestra segunda pregunta (II): para determinar si $f \in \langle f_1, ..., f_s \rangle$, primero calculamos un generador h, un máximo común divisor. De este modo, $f \in \langle f_1, ..., f_s \rangle$ es equivalente a que $f \in \langle h \rangle$. Posteriormente empleamos el algoritmo de la división para expresar $f = qh + r$, con $gr(r) < gr(h)$. Entonces, f pertenece al ideal si y solo si $r = 0$.

1.3. Polinomios en varias variables

En esta subsección vamos a generalizar los ingredientes vistos en el anillo de polinomios de una variable $K[x]$ al caso de varias variables, en $K[x_1,...,x_n]$, y buscamos responder de nuevo a los problemas de descripción (I) y, sobre todo, pertenencia (II) al ideal. Para empezar, de cara a generalizar el algoritmo de la división, nos vemos obligados a introducir la siguiente definición.

Definición 5. *Un **orden monomial** sobre $K[x_1,...,x_n]$ es cualquier relación $>$ en $\mathbb{Z}^n_{\geq 0}$ (o de forma equivalente, cualquier relación en el conjunto de monomios x^α, $\alpha \in \mathbb{Z}^n_{\geq 0}$) que satisfaga las tres condiciones siguientes:*

(i) $>$ *es un orden total sobre* $\mathbb{Z}^n_{\geq 0}$.

(ii) Si $\alpha > \beta$ *y* $\gamma \in \mathbb{Z}^n_{\geq 0}$, *entonces* $\alpha+\gamma > \beta+\gamma$.

(iii) $(\mathbb{Z}^n_{\geq 0}, >)$ *es un conjunto bien ordenado.*

En este trabajo utilizaremos el siguiente orden monomial.

Definición 6. ***(Orden lexicográfico)***
Sean $\alpha = (\alpha_1,...,\alpha_n)$ *y* $\beta = (\beta_1,...,\beta_n) \in \mathbb{Z}^n_{\geq 0}$. *Decimos que* $\alpha >_{lex} \beta$ *si en el vector diferencia* $\alpha - \beta$, *la componente no nula más situada a la izquierda es positiva. Igualmente, diremos que* $x^\alpha >_{lex} x^\beta$ *si* $\alpha >_{lex} \beta$.

Ejemplo 2. *Veamos un ejemplo de cómo funciona el orden lexicográfico en* $\mathbb{R}[x,y,z]$*:* $x^2yz >_{lex} xyz^3$, *dado que el vector diferencia es* $(2,1,1)-(1,1,3) = (1,0,-2)$, *cuya componente no nula más situada a la izquierda es* $1 > 0$.

Nos vemos obligados a extender la noción de término inicial de un polinomio.

Definición 7. *Sea* $0 \neq f = \sum_\alpha a_\alpha x^\alpha \in K[x_1,...,x_n]$, *y sea* $>$ *un orden monomial, llamamos*

(i) ***multigrado*** *de* f *a* $multigr(f) = max\{\beta \in \mathbb{Z}^n_{\geq 0} : a_\beta \neq 0\}$.

(ii) ***coeficiente inicial*** *de* f *a* $ci(f) = a_{multigr(f)} \in K$.

(iii) ***monomio inicial*** *de* f *a* $mi(f) = x^{multigr(f)}$.

(iv) ***término inicial*** *de* f *como* $in(f) = x^{multigr(f)} a_{multigr(f)} = ci(f)mi(f)$.

Ya poseemos todos los ingredientes algebraicos necesarios para extender el algoritmo de la división al caso de varias variables.

Teorema 2. ***Algoritmo de la división en*** $K[x_1,...,x_n]$
Fijemos un orden monomial $>$ *en* $\mathbb{Z}^n_{\geq 0}$, *y sea* $F = \{f_1,...,f_s\}$ *una s-tupla ordenada de polinomios en* $K[x_1,...,x_n]$. *Entonces, todo* $f \in K[x_1,...,x_n]$ *puede escribirse como*

$$f = a_1 f_1 + ... + a_s f_s + r,$$

donde a_i, $r \in K[x_1,...,x_n]$, *y o* $r = 0$, *o bien* r *es una combinación lineal de monomios indivisibles por* $in(f_1),...,in(f_s)$. *Además, si* $a_i f_i \neq 0$, *se tiene que* $multigr(f) \geq multigr(a_i f_i)$.

Su demostración es análoga a la del algoritmo de la división en una variable. A continuación presentamos el pseudocódigo del algoritmo que nos permite efectuar dicha división.

Datos: $f_1, ..., f_s, f$
Resultado: $a_1, ..., a_s, r$
Inicialización: $a_1 := 0, ..., a_s := 0, r := 0, p := f$
mientras $p \neq 0$ **hacer**
  $i := 1$
  $division := false$
  mientras $i \leq s$ **y** $division = false$ **hacer**
    si $in(f_i)$ *divide a* $in(p)$ **entonces**
      $a_i := a_i + in(p)/in(f_i)$
      $p := p + (in(p)/in(f_i))f_i$
      $division := true$
    en otro caso
      $i := i + 1$
    fin
  fin
  si $division = false$ **entonces**
    $r := r + in(p)$
    $p := p - in(p)$
fin

Algoritmo 2: Algoritmo de la división en $K[x_1, ..., x_n]$

Ejemplo 3. *Veamos un ejemplo de cómo funciona este algoritmo en* $\mathbb{F}_3[x,y]$ *con respecto a* $<_{lex}$. *Consideremos* $f = x^3y + x^3 + 2x^2y^2 + xy + x + 2y$, *y dividámoslo entre* $f_1 = x^2y + 1$ *y* $f_2 = xy + x$. *Buscamos calcular* a_1, a_2 *y* r. *Los inicializamos todos a 0 e igualamos* $p = f$.

Paso 1: como efectivamente $in(f_1) = x^2y$ divide a $in(p) = x^3y$, debemos actualizar los valores de a_1 y p: $a_1 = x$, y $p = x^3 + 2x^2y^2 + xy + 2y$.

Paso 2: $in(f_1) = x^2y$ no divide a $in(p) = x^3$. Posteriormente vemos que $in(f_2) = xy$ tampoco divide a $in(p) = x^3$. El algoritmo nos indica que debemos alojar el término inicial de p en el resto, debemos actualizar $r = x^3$ y $p = 2x^2y^2 + xy + 2y$.

Paso 3: ahora sí, $in(f_1) = x^2y$ divide a $in(p) = 2x^2y^2$. Actualizamos los valores para obtener $a_1 = x + 2y$ y $p = xy$.

Paso 4: $in(p) = xy$ es indivisible por $in(f_1) = x^2y$, pero sí que es divisible por $in(f_2) = xy$. Reactualizamos una vez más $a_2 = 1$ y $p = 2x$.

Paso 5: $in(p) = 2x$ es indivisible tanto por $in(f_1)$ como por $in(f_2)$. Reactualizamos $r = x^3 + 2x$ y, como $p = 0$, hemos terminado la división y el algoritmo finaliza arrojándonos la siguiente información:

$$f = a_1f_1 + a_2f_2 + r = (x + 2y)(x^2y + 1) + (1)(xy + x) + (x^3 + 2x). \square$$

Para ilustrar al lector de la sustancial diferencia entre el algoritmo de la división en una y en varias variables, retomemos el mismo ejemplo pero intercambiando el orden de f_1 y f_2 y efectuemos la división.

Ejemplo 4. *Consideremos, en* $\mathbb{F}_3[x,y]$ *con respecto a* $<_{lex}$, $f = x^3y + x^3 + 2x^2y^2 + xy + x + 2y$, *y dividámoslo entre* $f_1 = xy + x$ *y* $f_2 = x^2y + 1$.

Paso 1: $in(f_1) = xy$ divide a $in(p) = x^3y^2 \to a_1 = x^2$ y $p = 2x^2y^2 + xy + x + 2y$.

Paso 2: $in(f_1) = xy$ divide a $in(p) = 2x^2y^2 \to a_1 = x^2 + 2xy$ y $p = xy + x + 2y$.

Paso 3: $in(f_1) = xy$ divide a $in(p) = xy \to a_1 = x^2 + 2xy + 1$ y $p = 2y$.

Paso 4: Como $in(p) = 2y$ es indivisible tanto por $in(f_1) = xy$ como por $in(f_2) = x^2y$, actualizamos los valores de r y p: $r = 2y$ y, como $p = 0$, el algoritmo finaliza y nos indica que:

$$f = a_1f_1 + a_2f_2 + r = (x^2 + 2xy + 1)(x^2y + 1) + (0)(xy + x) + (2y).$$

Estos ejemplos muestran la importancia del orden de los divisores $f_1, ..., f_s$ a la hora de dividir, puesto que a distinto orden podemos obtener distintos cocientes $a_1, ..., a_s$ y distinto resto r. Es por este motivo que el algoritmo de la división exige que $f_1, ..., f_s$ sea una s-tupla ordenada. La unicidad del resto en $K[x]$ es un resultado clave para resolver el problema de la pertenencia al ideal (II), ya que la condición "$r = 0$" era una condición suficiente y necesaria para responder afirmativamente. Sin embargo, en $K[x_1, ..., x_n]$, la condición "$r = 0$" es solo una **condición suficiente** para determinar que $f \in \langle f_1, ..., f_s \rangle$, **no necesaria**. Antes de intentar remediar esta situación, introducimos los siguientes ingredientes y resultados para justificar la existencia de una base finita de todo ideal en $K[x_1, ..., x_n]$.

Definición 8. *Un ideal* $I \in K[x_1, ..., x_n]$ *se llama* ***ideal monomial*** *si existe un subconjunto* $A \in \mathbb{Z}^n_{\geq 0}$ *no necesariamente finito tal que todo elemento de I es una suma finita de la forma* $\sum_{\alpha \in A} h_\alpha x^\alpha$. *Lo denotamos como* $I = \langle x^\alpha | \alpha \in A \rangle$.

El siguiente ideal monomial nos será de bastante utilidad.

Definición 9. *Sea* $0 \neq I \subset K[x_1, ..., x_n]$ *un ideal, definimos:*

(i) ***IN(I)*** $= \{in(f) | f \in I\}$, *el conjunto de los términos iniciales de los elementos de I.*

(ii) Denotamos como $\langle IN(I) \rangle$ *al ideal monomial generado por los elementos de* $IN(I)$.

Un resultado fundamental sobre ideales monomiales es que son finitamente generados.

Teorema 3. ***(Lema de Dickson)*** *Todo ideal monomial en* $K[x_1, ..., x_n]$ *puede escribirse como* $I = \langle x^{\alpha(1)}, ..., x^{\alpha(s)} \rangle$, *con* $\alpha(i) \in A \ \forall i$. *En particular, I posee una base finita.*

Demostración. Procedamos por inducción. Si $n = 1$, claramente un ideal monomial $\langle x_1^\alpha | \alpha \in A \subset \mathbb{Z}_{\geq 0} \rangle$ es finitamente generado en $K[x_1]$: sea $\beta \leq \alpha$ el menor elemento de A, entonces, x_1^β divide a todo x_1^α, luego $I = \langle x_1^\beta \rangle$.

Supongamos que $n > 1$ y que el teorema es cierto hasta $n - 1$. Para simplificar la comprensión, denotaremos las variables como $x_1, ..., x_{n-1}, y$, y los exponentes como $\alpha \in \mathbb{Z}^{n-1}_{\geq 0}$ y $m \in \mathbb{Z}_{\geq 0}$. Supongamos que $I \subset K[x_1, ..., x_{n-1}, y]$ es un ideal monomial. Para encontrar generadores de I, construimos $J = \langle x^\alpha \in K[x_1, ..., x_n] | x^\alpha y^m \in I \rangle$ para algún $m \geq 0$. Entonces, J es un ideal monomial en $K[x_1, ..., x_{n-1}]$ y, por hipótesis de inducción, $J = \langle x^{\alpha(1)}, ..., x^{\alpha(s)} \rangle$. Además, la definición de J nos dice que para todo $1 \leq i \leq s$, $x^{\alpha(i)} y^{m_i} \in I$ para cierto $m_i \geq 0$. Sea

$m = max_{\{1\leq i\leq s\}}\{m_i\}$, consideremos para cada $0 \leq k \leq m-1$ el ideal $J_k \subset K[x_1,...,x_{n-1}]$ generado por los monomios x^β tales que $x^\beta y^k \in I$. Utilizando nuestra hipótesis de inducción de nuevo, J_k tiene un conjunto generador finito de monomios, $J_k = \langle x^{\alpha_k(1)},...,x^{\alpha_k(s)}\rangle$. Se tiene además que I está generado por monomios de la siguiente lista:

$$\begin{aligned}
&\text{para } J: x^{\alpha(1)}y^m,...,x^{\alpha(s)}y^m,\\
&\text{para } J_0: x^{\alpha_0(1)},...,x^{\alpha_0(s_0)},\\
&\text{para } J_1: x^{\alpha_1(1)}y,...,x^{\alpha_1(s_1)}y,\\
&\quad ...\\
&\text{para } J_{m-1}: x^{\alpha_{m-1}(1)}y^{m-1},...,x^{\alpha_{m-1}(s_{m-1})}y^{m-1}.
\end{aligned}$$

Tenemos que todo monomio de I es divisible por uno de la lista. Sea $x^\alpha y^p \in I$:

(i) si $p \geq m$, entonces $x^\alpha y^p$ es divisible entre algún $x^{\alpha(i)}y^m$ por construcción de J.

(ii) si $p \leq m-1$, entonces $x^\alpha y^p$ es divisible entre algún $x^{\alpha_p(j)}y^p$ por construcción de J_p.

Por último, falta comprobar que el conjunto finito de generadores "monomiales" puede ser extraído de un conjunto de generadores del ideal. Retomando la notación de las variables $x_1,...,x_n$, nuestro ideal monomial es $I = \langle x^\alpha | \alpha \in A\rangle$. Queremos ver que I está generado (finitamente) por los x^α, con $\alpha \in A$. Hemos visto que $I = \langle x^{\beta(1)},...,x^{\beta(s)}\rangle$ para ciertos monomios $x^{\beta(i)}$ en I. Dado que $x^{\beta(i)} \in I = \langle x^\alpha | \alpha \in A\rangle$, se sigue que $x^{\beta(i)}$ es divisible por $x^{\alpha(i)}$, para algún $\alpha(i) \in A$. Se tiene fácilmente que $I = \langle x^{\alpha(1)},...,x^{\alpha(s)}\rangle$. □

Afortunadamente, a pesar de que $K[x_1,..,x_n]$ no sea un DIP, el siguiente teorema justifica que todo ideal polinómico es finitamente generado.

Teorema 4. ***(Teorema de la base de Hilbert)***
Todo ideal en $K[x_1,...,x_n]$ posee una base finita. Es decir, $I = \langle f_1,...,f_s\rangle$, con $f_i \in I\ \forall I$.

Demostración. Trivialmente, si $I = \{0\}$, $\{0\}$ es un conjunto finito y generador de I. En caso contrario, si I contiene algún polinomio no nulo, podemos construir $g_1,...,g_s$, un conjunto generador de I, de la siguiente manera: sabemos que existen $g_1,...,g_s \in I$ tales que $\langle IN(I)\rangle = \langle in(g_1),...,in(g_s)\rangle$. Queremos comprobar que, efectivamente, $I = \langle g_1,...,g_s\rangle$.

Se tiene que $\langle g_1,...,g_s\rangle \subset I$, dado que todo $g_i \in I$. Para demostrar el otro contenido, sea $f \in I$ cualquiera. Efectuando el algoritmo de la división para dividir f entre $\{g_1,...,g_s\}$, obtendremos $f = a_1g_1 + ... + a_tg_t + r$, donde r no tendrá ningún término divisible por $in(g_1),...,in(g_t)$. Veamos que $r = 0$: como $r = f - a_1g_1 + ... - a_tg_t \in I$, si tuviésemos que $r \neq 0$, entonces, $in(r) \in \langle IN(I)\rangle = \langle in(g_1),...,in(g_s)\rangle$, de modo que $in(r)$ debe ser divisible por algún $in(g_i)$. Esto es una contradicción, ergo $r = 0$, y así, $f = a_1g_1 + ... + a_tg_t \in \langle g_1,...,g_s\rangle$, quedando demostrado el segundo contenido. □

1.4. Bases de Groebner

Tras comprobar la imperfección de la generalización del algoritmo de la división, introduciremos ingredientes algebraicos que nos ayudarán a remediar este problema.

Definición 10. *Fijado un orden monomial $>$, un conjunto finito $G = \{g_1, ..., g_t\} \subset I$ se dirá* ***base de Groebner*** *de I con respecto a $>$ si*

$$\langle in(g_1), ..., in(g_t)\rangle = \langle IN(I)\rangle.$$

Visto de otro modo, G se dirá base de Groebner si y solo si el término inicial de cualquier elemento del ideal I es divisible por el término inicial de algún elemento de G.

Proposición 1. ***(Existencia de las bases de Groebner)***
Sean $\{0\} \neq I \subset K[x_1, ..., x_n]$ un ideal y $>$ un orden monomial. Entonces, existe una base de Groebner de I con respecto de $>$. Además, toda base de Groebner de un ideal I es una base de I.

Demostración. Sea I un ideal no trivial, entonces, el conjunto $\{g_1, ..., g_s\}$ construido en la demostración del teorema de la base de Hilbert 4 es una base de Groebner por definición. Además, como $\langle IN(I)\rangle = \langle in(g_1), ..., in(g_s)\rangle$, el mismo argumento empleado para demostrar dicho teorema nos dice que $I = \langle g_1, ..., g_s\rangle$, de modo que G es una base de I. □

Veamos que las bases de Groebner nos permiten al fin dar respuesta al problema de la pertenencia al ideal (II).

Proposición 2. *Sea $G = \{g_1, ..., g_t\}$ una base de Groebner del ideal $I \subset K[x_1, ..., x_n]$ con respecto del orden monomial $>$, y sea $f \in K[x_1, ..., x_n]$, entonces:*
(i) el resto de f entre G es único y no depende del orden de los elementos de G.
(ii) $f \in I \iff$ el resto de f entre G es 0.

Demostración. (i) Procedamos por reducción al absurdo. Supongamos que empleando el algoritmo de la división obtenemos $f = a_1g_1 + ... + a_tg_t + r_1 = b_1g_1 + ... + b_tg_t + r_2$, con $r_1 \neq r_2$. Entonces, $(r_2 - r_1) = (b_1 - a_1)g_1 + ... + (b_t - a_t)g_t \in I$. Por lo tanto, existirá algún $g_i \in G$ tal que $in(g_i)$ divida a $in(r_2 - r_1)$. De modo que $in(g_i)$ divide a algun término de alguno de los restos r_1 o r_2, lo cual nos conduce a una contradicción, dado que no serían restos, ergo el resto r es único. Por lo tanto, considerando una permutación cualquiera de G, el algoritmo de la división nos conduciría a $f = a'_1g_{\sigma(1)} + ... + a'_tg_{\sigma(t)} + r$ (notar que lo único que aseguramos es la unicidad del resto, no de los cocientes $a_1, ..., a_t$). A partir de ahora **denotaremos el resto de f entre G como f^G**.

(ii) Hemos visto que si $f^G = 0$, entonces $f \in I$. Para demostrar la otra implicación, supongamos que $r \neq 0$ y lleguemos a contradicción. Si $f \in I$, entonces, aplicando el algoritmo de la división, $f = a_1g_1 + ... + a_tg_t + f^G$, por tanto, $f^G = f - (a_1g_1 + ... + a_tg_t) \in I$. Como el resto pertenece al ideal, existirá algún $g_i \in G$ tal que $in(g_i)$ divida a $in(f^G)$, lo cual nos conduce de nuevo a contradicción, ya que un resto no puede satisfacer esta propiedad, luego $f^G = 0$. Hemos resuelto el problema de la pertenencia al ideal (II) en $K[x_1, ..., x_n]$.

□

Definición 11. *Sean $0 \neq f, g \in K[x_1, ..., x_n]$ con $\alpha = multigr(f)$ y $\beta = multigr(g)$, llamamos* ***S-polinomio*** *de f y g a la siguiente expresión*

$$S(f,g) = \frac{x^\gamma}{in(f)}f - \frac{x^\gamma}{in(g)}g, \text{ donde } \gamma_i = max\{\alpha_i, \beta_i\}.$$

Ejemplo 5. *Calculemos el S-polinomio de $f_1 = x^2 - y$ y $f_2 = x^3 - z$. Como $\alpha = multideg(f_1) = (2,0,0)$ y $\beta = multideg(f_2) = (3,0,0)$, entonces, $\gamma = (3,0,0)$. De modo que*

$$S(f_1,f_2)=\frac{x^3}{x^2}(x^2-y)-\frac{x^3}{x^3}(x^3-z)=-xy+z.$$

La definición previa busca producir la cancelación de los términos iniciales de f y g. Esto se hace más evidente al constatar que $x^\gamma = MCM(mi(f),mi(g))$. Nos será de gran utilidad para caracterizar las bases de Groebner.

Proposición 3. ***(Criterio de Buchberger con S-polinomios)***
$G=\{g_1,...,g_t\}$ es una base de Groebner de $\{0\}\neq I\subset K[x_1,...,x_n]$ con respecto del orden monomial $>$ si y solo si para todo $i\neq j$, el resto de $S(g_i,g_j)$ entre G es 0.

La anterior proposición no solo sirve para caracterizar las bases de Groebner, sino que también otorga una herramienta para crearlas. Si partimos de un ideal I del que conocemos una base $G=\{f_1,...,f_s\}$ que puede ser o no de Groebner, tenemos que, por construcción, $S(f_i,f_j)\in I\ \forall i\neq j$. De este modo, $in(S(f_i,f_j))\in IN(I)$. Aplicando la proposición anterior, podemos distinguir dos casos:

(i) si el resto de $S(f_i,f_j)$ entre G es 0, entonces, $in(S(f_i,f_j))\in\langle in(f_1),...,in(f_s)\rangle$.

(ii) si el resto es distinto de cero, entonces $in(S(f_i,f_j))\in IN(I)$, pero $in(S(f_i,f_j))\notin\langle in(f_1),...,in(f_s)\rangle$. De modo que podemos añadir $S(f_i,f_j)$ a la base (o equivalentemente, el resto $S(f_i,f_j)^G$), y repetir el proceso para comprobar de nuevo si se trata de una base de Groebner.

Esta es la idea que subyace bajo el siguiente teorema.

Teorema 5. ***(Algoritmo de Buchberger)***
Sea $\{0\}\neq I=\langle f_1,...,f_s\rangle$ un ideal, entonces, el siguiente algoritmo nos permite construir una base de Groebner de I en un número finito de pasos:

Datos: $F=\{f_1,...,f_s\}$
Resultado: una base de Groebner $G=\{g_1,...,g_t\}\subset F$
Inicialización: $G:=F$
repetir
  $G':=G$
  para *cada par* $\{p,q\}$ *con* $p\neq q$ *en* G' **hacer**
    $S:=S(p,q)^{G'}$
    si $S\neq 0$ **entonces**
      $G:=G\cup\{S\}$
    fin
  fin
hasta que $G=G'$;

Algoritmo 3: Algoritmo de Buchberger

Demostración. Si $S(f_j,f_k)^G=0$ para todo $j\neq k$, el criterio de Buchberger termina la demostración. En caso contrario, si tenemos que $S(f_j,f_k)^G\neq 0$ para ciertos $j\neq k$, entonces, como $S(f_j,f_k)\in I$, $S(f_j,f_k)^G\in I$, luego $G\subset I$ en todo paso del algoritmo. Vemos que I es también generado por $\{f_1,...,f_s,S(f_j,f_k)^G\}$. Además, $in(S(f_j,f_k)^G)$ es indivisible por cualquier $in(f_i)$, de modo que $\langle in(f_1),...,in(f_s)\rangle\subset\langle in(f_1),...,in(f_s),in(S(f_j,f_k)^G)\rangle$. Si repetimos este proceso un número finito de pasos, en caso de ser necesario, obtendremos $G=\{f_1,...,f_s,f_{s+1},...,f_{s+t}\}$, y se tendría la siguiente cadena de ideales:

$$\langle in(f_1),...,in(f_s)\rangle\subset\langle in(f_1),...,in(f_{s+1})\rangle\subset...\subset\langle in(f_1),...,in(f_{s+t})\rangle.$$

Por último, la condición de la cadena ascendente (ACC) de ideales en $K[x_1,...,x_n]$ nos garantiza que el algoritmo finaliza en un número finito de pasos. □

Veamos un ejemplo de cómo funciona el algoritmo.

Ejemplo 6. *Sean $f_1 = x^2 - y$ y $f_2 = x^3 - z$. Vamos a construir una base de Groebner de $I = \langle f_1, f_2 \rangle$ con respecto a $>_{lex}$. Partimos de $F = \{x^2 - y, x^3 - z\}$. En el ejemplo anterior hemos visto que $S(f_1, f_2) = -xy + z$. Como claramente $in(S(f_1, f_2)) = -xy \notin \langle in(f_1), in(f_2) \rangle = \langle x^2 \rangle$ (es decir, $S(f_1, f_2)^F \neq 0$), el criterio de Buchberger nos garantiza que F no es base de Groebner.*

Utilicemos el algoritmo de Buchberger para construir una base de Groebner a partir de F. Sea $G' = F$, el algoritmo nos dice que debemos añadir $S(f_1, f_2)$ a la base, luego ahora $G' = \{x^2 - y, x^3 - z, -xy + z\}$. Calculamos de nuevo S-polinomios: $S(x^2 - y, -xy + z) = xz - y^2$. Por el mismo argumento que antes, este S-polinomio no reduce a 0 módulo G', luego lo añadimos a la base, $G' = \{x^2 - y, x^3 - z, -xy + z, xz - y^2\}$. Este proceso continúa sucesivamente hasta obtener $G = \{x^2 - y, x^3 - z, -xy + z, xz - y^2, y^3 - z^2\}$, una base de Groebner de I con respecto a $>_{lex}$.

El problema que presenta el algoritmo presentado previamente es que, habitualmente, produce bases de Groebner con más elementos de los necesarios. Esto se debe a que un ideal puede poseer distintas bases de Groebner.

Lema 1. *Sea G una base de Groebner de un ideal I. Si $p \in G$ es un polinomio tal que $in(p) \in \langle IN(G - \{p\}) \rangle$, entonces $G - \{p\}$ es también una base de Groebner de I.*

Demostración. Se tiene que $\langle IN(G) \rangle = \langle IN(I) \rangle$. Si tenemos que $in(p) \in \langle IN(G - \{p\}) \rangle$, por la propia definición de base de Groebner se sigue que $G - \{p\}$ es también una base de Groebner de I. □

Apoyémonos en el lema anterior para definir una base de Groebner en la que no exista ningún elemento redundante.

Definición 12. *Sea I un ideal polinómico, denominamos una* ***base de Groebner minimal*** *de dicho ideal a una base de Groebner G tal que:*

(i) $ci(p) = 1$ para todo $p \in G$.

(ii) Para todo $p \in G$, $in(p) \notin \langle IN(G - \{p\}) \rangle$.

Si deseamos generar una base de Groebner minimal de un ideal, basta emplear el algoritmo de Buchberger para generar una base de Groebner y aplicar el Lema 1 para eliminar los términos redundantes que pudieran aparecer. Sin embargo, un ideal puede admitir bases de Groebner minimales distintas, aunque siempre de la misma cardinalidad.

Proposición 4. *Sea I un ideal polinómico. Supongamos que $G_1 = \{g_1^{(1)}, ..., g_s^{(1)}\}$ y $G_2 = \{g_1^{(2)}, ..., g_t^{(2)}\}$ son dos bases de Groebner minimales de I con respecto de un orden monomial fijo. Entonces $s = t$ y, reordenando debidamente, $in(g_i^{(1)}) = in(g_i^{(2)})$ para todo $1 \leq i \leq s$.*

Demostración. Procedamos por reducción al absurdo. Supongamos que $s \geq t$. Notemos que $g_1^{(1)} \in I$, entonces, como G_2 es una base de Groebner, el término inicial de algún $g_i^{(2)}$ divide a $in(g_1^{(1)})$. Pongamos que, por ejemplo, $in(g_1^{(2)})$ divide a $in(g_i^{(1)})$.

Por otro lado, $g_1^{(2)} \in I$, y G_1 es también una base de Groebner. Por lo tanto, el término inicial de algún $g_j^{(1)}$ divide a $in(g_1^{(2)})$. De este modo se sigue que $in(g_j^{(1)})$ divide a $in(g_1^{(1)})$. Como G_1 es una base de Groebner minimal, entonces forzosamente $j = 1$, y así $in(g_1^{(1)}) = in(g_1^{(2)})$.

Se procede análogamente con $g_2^{(1)}$. En este proceso, ni $g_1^{(1)}$ ni $g_1^{(2)}$ vuelven a jugar ningún rol. Se prosigue así sucesivamente y, si tuviéramos que $s > t$, encontraríamos un $g_i^{(2)}$ previamente emparejado con $g_i^{(1)}$ que compartiría término inicial con $g_{t+1}^{(1)}$. Esto entra en contradicción con el hecho de que G_1 es una base de Groebner minimal, luego $s = t$. □

La siguiente definición nos permitirá establecer una biyección entre un ideal polinómico y una base de Groebner minimal que cumpla una propiedad añadida.

Definición 13. *Sea I un ideal polinómico, llamamos **base de Groebner reducida** de I a una base de Groebner G tal que:*

(i) $ci(p) = 1$ *para todo* $p \in G$.

(ii) Para todo $p \in G$, *ningún monomio de* p *pertenece a* $\langle IN(G - \{p\})\rangle$.

Proposición 5. *Sea I un ideal polinómico no trivial. Entonces, fijado un orden monomial, I posee una única base de Groebner reducida.*

Demostración. Diremos que un elemento $g \in G$ está **reducido** respecto de G si ningún monomio de g pertenece a $\langle IN(G - \{p\})\rangle$. Procederemos a demostrar la proposición partiendo de una base de Groebner minimal, y reduciremos todos sus elementos.

Primero de todo, si g está reducido respecto de G, entonces g estará reducido respecto de cualquier otra base de Groebner minimal que lo contenga, y que posea el mismo conjunto de términos iniciales.

Dado $g \in G$, llamemos $g' = g^{G-\{g\}}$ y $G' = (G - \{g\}) \cup \{g'\}$. Veamos que G' es una base de Groebner minimal de I. Se tiene que $in(g) = in(g')$, dado que al dividir g entre $G - \{g\}$, forzosamente $in(g)$ se traslada al resto, puesto que no es divisible por ningún elemento de $IN(G - \{g\})$. Queda así demostrado que $\langle IN(G')\rangle = \langle IN(G)\rangle$. Además, como $G' \subset I$, G' es una base de Groebner minimal. Asimismo, g' está reducido respecto de G' por construcción.

Si repetimos el proceso previo a todos los elementos de G obtendremos una base de Groebner reducida, pues todo elemento acabará reducido. Notemos que una vez que reducimos un elemento en el proceso, nunca deja de estarlo, dado que nunca se altera su término inicial.

Falta demostrar la unicidad de dichas bases. Supongamos que poseemos dos bases de Groebner reducidas G_1 y G_2. Empleando la proposición 4, tenemos que $IN(G_1) = IN(G_2)$. De este modo, dado $g^{(1)} \in G_1$, existirá un $g^{(2)} \in G_2$ tal que $in(g_1) = in(g_2)$. Para probar la unicidad, nos falta probar que $g^{(1)} = g^{(2)}$. Para ello, tomemos la diferencia $g^{(1)} - g^{(2)} \in I$. Dado que G_1 es una base de Groebner del ideal, entonces, $\left(g^{(1)} - g^{(2)}\right)^G = 0$. Como los términos

iniciales de ambos son iguales, ambos se cancelan en la resta, y el resto de términos no son divibles por ninguno de los $IN(G_1) = IN(G_2)$, pues ambas bases son reducidas. Esto implica que $\left(g^{(1)} - g^{(2)}\right)^G = g^{(1)} - g^{(2)}$, luego que $g^{(1)} - g^{(2)} = 0$. De aquí se sigue que $G_1 = G_2$. □

Hagamos por último un pequeño paréntesis para introducir dos definiciones que nos serán de gran utilidad en el siguiente capítulo.

Definición 14. *Sea $f = \sum_\alpha a_\alpha x^\alpha \in K[x_1,...,x_n]$ un polinomio, llamamos **soporte** del polinomio f al conjunto de vectores $Sop(f) = \{\alpha \in \mathbb{N}^n | a_\alpha \neq 0\}$. Diremos que f es **β-sparse** si $\#Sop(f) \leq \beta$ para algún $\beta \in \mathbb{N}$.*

Ejemplo 7. *Por ejemplo, el polinomio $f = 3x^4z + x^2y + 5xy - 1 \in \mathbb{R}[x,y,z]$ tiene por soporte $Sop(f) = \{(4,0,1),(2,1,0),(1,1,0)\}$. Vemos así que f es $3-$sparse.*

Definición 15. *Sean $F \subset K[x_1,...,x_n]$, $f \in F$ y $>$ un orden monomial, llamamos **forma normal** de f respecto de F y $>$ a un polinomio mónico $\overline{f}$ tal que se pueda escribir como $\overline{f} = f + h$, con $h \in \langle F \rangle$, y tal que $Sop(f) \cap (IN(F) + \mathbb{N}^n) = \emptyset$.*

Capítulo 2

Criptografía con bases de Groebner

En este capítulo veremos cómo pueden ser utilizados en criptografía los ingredientes algebraicos previamente descritos. Antes de avanzar, tenemos que familiarizarnos con el concepto de *criptografía*, que es la ciencia o arte de crear procedimientos que nos permitan encriptar y desencriptar informaciones mediante el uso de claves. De este modo, se busca transmitir informaciones encriptadas e ilegibles que solo ciertos receptores sepan desencriptar.

Llamamos criptosistema a un algoritmo criptográfico concreto que realice esta labor. Distinguimos dos grandes grupos de criptosistemas: los de *clave privada* o *simétricos*, caracterizados por emplear la misma clave para encriptar y para desencriptar, y los de *clave pública* o *asimétricos*, que poseen dos claves diferentes. En este segundo tipo de criptosistemas, una de las claves es pública y se emplea para encriptar mensajes, y la otra, que es privada, sirve para desencriptar la información. El ejemplo más conocido de criptosistema de clave pública es el RSA. Asimismo, los criptosistemas que emplean la teoría de bases de Groebner son de clave pública. Un ejemplo de este tipo de sistemas son los criptosistemas de *Barkee* y los criptosistemas *Polly Cracker*. Comenzaremos tratando los criptosistemas Barkee en abstracto, para pasar en el siguiente capítulo al ámbito concreto.

2.1. Criptosistema Barkee

El primer criptosistema que vamos a analizar fue introducido por B. Barkee, y es también conocido como criptosistema Polly Cracker con bases de Groebner.

Hemos comprobado que el algoritmo de la división en varias variables no garantiza la existencia de un único resto, dado que depende del orden de los divisores. No obstante, hemos visto que la división entre bases de Groebner sí que nos permite superar esta adversidad, garantizando la existencia de un único resto independiente del orden de los divisores (proposición 2 del capítulo previo).

La argucia sobre la que se sustenta el criptosistema Barkee es la siguiente: Bob quiere enviar un mensaje m, un polinomio. Si el receptor, Alice, posee una base de Groebner G de un ideal I donde m no se ve reducido a 0 módulo G, entonces, sumando a m un polinomio aleatorio en I, el receptor podrá desencriptar el mensaje simplemente aplicando el algoritmo de la división empleando G. El objetivo es además, construir una clave pública que permita encriptar el mensaje, pero que sea lo suficientemente compleja como para que un tercer individuo, Eve, no pueda calcular una base de Groebner fácilmente.

El criptosistema tiene la siguiente estructura: sean K un cuerpo, I un ideal en $K[x_1,...,x_n]$, y $G=\{g_1,...,g_t\}$ una base de Groebner de I (respecto a un orden monomial).

- **Clave privada:** una base de Groebner $G=\{g_1,...,g_t\}$ de I.
- **Clave pública:** $F=\{f_1,...,f_s\}$, base de I o de un ideal contenido en I y un conjunto de formas normales $N=\{v_1,...,v_t\}$ con respecto de I.
- **Mensajes:** el espacio de los mensajes que podemos enviar, M, será el conjunto de polinomios cuyos términos no estén contenidos en $IN(I)$. Esta condición nos permite garantizar que un mensaje $m=\sum_{i=1}^{t} m_i v_i \in M$ no se ve reducido a 0 módulo G.
- **Encriptado:** el proceso de encriptado de un mensaje $m \in M$ consiste en escoger aleatoriamente polinomios $h_1,...,h_s \in K[x_1,...,x_n]$ y considerar $p=\sum_{i=1}^{s} h_i f_i$. Al efectuar la suma $c=p+m$ obtenemos nuestro mensaje encriptado.
- **Desencriptado:** el receptor del mensaje cifrado reduce c a m módulo G usando el algoritmo de Buchberger.

Ejemplo 8. *Bob desea enviar empleando un criptosistema Barkee en $\mathbb{R}[x,y,z]$. Para ello, empleará en la clave pública $F=\{x^2-y, x^3-z, -xy+z\}$, cuyos elementos sabemos que conforman parte del ideal generado por la base de Groebner del ejemplo 6 del capítulo anterior. Además, tomará como polinomios normales de la clave pública $N=\{y,z\}$, que claramente son formas normales (su soporte tiene intersección vacía con $IN(F)+\mathbb{N}^3$). El mensaje a enviar será $m=2y+5z$.*

Antes de enviar el mensaje a Alice, escoge aleatoriamente $h_1=y+1$, $h_2=y-1$ y $h_3=xy+z-3$, y encripta el mensaje:

$$c=p+m=(y+1)(x^2-y)+(y-1)(x^3-z)+(xy+z-3)(-xy+z)+2y+5z= \\ x^3y-x^3-x^2y^2+x^2y+x^2+3xy-y^2-yz+y+z^2+3z.$$

Alice, conocedora de la base de Groebner, solo debe aplicar el algoritmo de la división, para obtener $m=c^G$.

El ataque más básico a este criptosistema sería que Eve produjera, mediante el algoritmo de Buchberger, una base de Groebner a partir de la clave pública. En el ejemplo anterior, este ataque es muy eficaz, pues hemos comprobado que en dos iteraciones obtendríamos una base de Groebner. No obstante, en la siguiente sección veremos que hay casos concretos en los que el cálculo de una base de Groebner ni siquiera es necesario, dado que existen métodos de ataque menos costosos.

2.2. Representación de polinomios

Antes de introducir nuestro primer criptosistema, dado un orden monomial $>$ fijo, introduciremos dos maneras habituales de representar los polinomios en álgebra computacional:

- **Representación densa:** podemos ordenar el conjunto exponentes cuyo grado es menor o igual que d, denotado por $T(n,d):=\{\alpha \in \mathbb{N}^n : \alpha=\alpha_1+...+\alpha_n \leq d\}$. De este modo, podemos representar un polinomio $f=\sum_\alpha a_\alpha x^\alpha \in K[x_1,...,x_n]$ con grado d mediante un

vector $(a_\alpha)_{\alpha \in T(n,d)} \in K^{|T(n,d)|}$. Esto nos motiva a decir que un polinomio $f \in K[x_1,...,x_n]$ tiene **grado d en el sentido de la notación densa de polinomios** si todos los exponentes que figuran en f están en $T(n,d)$, pero no en $T(n,d-1)$. Dado que generalmente trabajaremos sobre cuerpos finitos $K = \mathbb{F}_q$, la relación $\{x_i^q = x_1 | 1 \leq i \leq n\}$ nos permite simplificar aún más la representación densa como $T_q(n,d) := \{\alpha \in \mathbb{N}^n : \alpha \leq d, \alpha_i \leq q-1 \ \forall i\}$.

- **Representación *sparse*:** podemos representar los polinomios mediante sus coeficientes no nulos y sus correspondientes monomios. Sea $f = \sum_\alpha a_\alpha x^\alpha \in K[x_1,...,x_n]$, con grado d, el conjunto $\{(a_\alpha, \alpha) \in K \times \mathbb{N}^n\}$ representa a f en álgebra computacional.

El uso de polinomios densos con grado bajo y polinomios *sparse* de grado elevado pueden sernos de utilidad para mejorar la seguridad de nuestros criptosistemas.

2.3. Criptosistema Barkee: versión densa

Recurriremos primero a la versión densa de dichos criptosistemas, es decir, aquella que emplea polinomios densos y de grado no muy elevado para encriptar el mensaje. El esquema es el mismo que el presentado en la sección 2.1, pero exigiendo que los polinomios h_i sean densos de, como máximo, grado d_1 (en el sentido de la notación densa de polinomios), y que los polinomios de F sean densos de hasta grado d_2.

El coste de encriptado y desencriptado oscila entre $\mathcal{O}\binom{d_1+d_2+n}{n}$, el tiempo que precisamos para analizar un mensaje denso, y $\mathcal{O}\left(\binom{d_1+d_2+n}{n}^2\right)$, el coste del algoritmo de Buchberger en el caso genérico.

2.4. Ataque lineal al criptosistema Barkee

El criptosistema que vamos a presentar es también conocido como ataque de Moriarty, debido al apellido de su ideólogo.

Se fundamenta en tratar los coeficientes de los h_i como incógnitas, para así obtener ecuaciones lineales identificando los coeficientes de c con los coeficientes de p. Llamemos $M(n)$ al conjunto de monomios de, como máximo, grado n respecto a la notación densa de polinomios. Concretamente consideramos los polinomios h_i como polinomios de grado $d = gr(c) - d_2$, y resolvemos la ecuación $h_i = \sum_{\rho \in M(d)} b_\rho \rho$, con con respecto de las incógnitas b_ρ y v_i. Si el sistema no tiene solución, aumentamos el grado d en una unidad y repetimos el proceso. De esta manera, cuando alcancemos $d = d_1$ (el grado de los polinomios aleatorios empleados para encriptar el mensaje), el sistema tendrá solución y podremos desencriptar el mensaje.

Dicho ataque no solo es correcto, sino que Eve es capaz de romper la variante densa del criptosistema Polly Cracker en el mismo tiempo computacional que le cuesta a Bob encriptar el mensaje. La demostración de ello se puede encontrar en [5], pp. 22-27. En el siguiente capítulo veremos cómo efectuar un ataque de Moriarty a un criptosistema concreto.

2.5. Criptosistema Barkee: versión *sparse*

Hemos comentado que recurrir a polinomios densos como clave pública no soluciona la susceptibilidad de nuestro criptosistema Barkee. Otra estrategia es trabajar con polinomios β-sparse (es decir, polinomios con un soporte pequeño) pero de grado elevado. El esquema del criptosistema es el mismo que en el caso denso, pero requiriendo que los f_i y los h_i sean β-sparse de grado elevado. De esta manera, la ruptura del criptosistema es mucho más ardua mediante el ataque lineal que efectúa Eve, dado que el número de filas y columnas en la matriz de ataque crece exponencialmente con n y $gr(c)$, mientras que el trabajo de Bob sólo se ve aumentado por los factores s, $Sop(f_i)$ y $Sop(h_i)$.

Capítulo 3

Criptosistemas Polly Cracker

En este capítulo introducimos los criptosistemas Polly Cracker, ideados por M. Fellows y N. Koblitz de manera independiente al criptosistema Barkee. A pesar de sustentarse también sobre la teoría de bases de Groebner, la gran diferencia es que ahora clave pública pasa de ser una base de Groebner para ser una raíz del ideal, es decir, un cero del sistema de polinomios.

3.1. Polly Cracker abstracto y concreto

Veamos **en abstracto** las características de estos criptosistemas: sea K un cuerpo finito.

- **Clave privada:** un cero ξ_0 del ideal. Es decir, un $\xi_0 \in K$ tal que $f(\xi_0) = 0 \; \forall f \in I$.
- **Clave pública:** un conjunto de polinomios $F = \{f_1, ..., f_s\}$ que genere I, un ideal.
- **Mensajes:** ahora el espacio de mensajes es el cuerpo, $M = K$.
- **Encriptado:** el proceso de encriptado de un mensaje $m \in M$ consiste en un polinomio aleatorio $h \in K[x_1, ..., x_n]$. Al efectuar la suma $c = h + m$ obtenemos nuestro mensaje encriptado.
- **Desencriptado:** simplemente se evalua c en el cero, y así, $c(\xi_0) = h(\xi_0) + m = m$.

Si deseamos generar un criptosistema Polly Cracker **concreto**, necesitamos saber generar las claves. Para ello, basta considerar aleatoriamente un $\xi_0 \in K$ (será la clave privada) y unos polinomios $f_1', ..., f_s'$ cualesquiera. De este modo, tomando $f_i := f_i' - f_i'(\xi_0)$, podemos tomar $\{f_1, ..., f_s\}$ como la clave pública.

Esta familia de criptosistemas no es más que un caso concreto de los criptosistemas Barkee, dado que si I tiene un cero, todo elemento de K se encuentra como forma normal respecto de I. La diferencia es la manera de desencriptar: antes empleábamos división mediante una base de Groebner, mientras que ahora evaluamos en un punto. La seguridad de estos criptosistemas radica en la complejidad de resolver un sistema de ecuaciones que nos proporcione los ceros del ideal. Uno de los resultados claves sobre los que se sustentaron Fellows y Koblitz es que podemos escoger los polinomios de tal manera que nuestro criptosistema esté en correspondencia con un problema NP-completo (ver [4], capítulo 5, teorema 3.1.). Veamos un ejemplo, proporcionado por ambos autores, que pone en evidencia la relación entre los criptosistemas Polly Cracker y la teoría de grafos.

Ejemplo 9. *Sea $\Gamma(V,E)$ un grafo con nodos $V=\{1,...,n\}$ y aristas $E\subset\{\{i,j\},1\leq i<j\leq n\}$. Llamaremos* ***3-coloración*** *del grafo a una aplicación $\phi: V\to\{1,2,3\}$ tal que $\{i,j\}\in E \implies \phi(i)\neq\phi(j)$. Podemos generar una 3-coloración de la siguiente manera: asignamos los valores $\{0,1\}$ a un conjunto de $3n$ variables X_{ij}, $1\leq i\leq n$, $1\leq j\leq 3$, imponiendo que $X_{ij}=1 \iff \phi(i)=j$. Transformemos nuestro problema en un problema polinómico. Sea $G:=G_0\cup G_1\cup G_2\subset\mathbb{F}_2[X_{ij},1\leq i\leq n,1\leq j\leq 3]$, donde*

(i) $G_0=\{X_{i1}X_{i2},X_{i1}X_{i3},X_{i2}X_{i3}:1\leq i\leq n\}$. *Esto garantiza que no podamos colorear cada vértice de dos maneras distintas, dado que la ecuación $X_{ij}X_{ik}=0$ con $j\neq k$ implica que una de las dos variables ha de ser 0, pues nos encontramos en $\mathbb{Z}_2$.*

(ii) $G_1=\{X_{i1}+X_{i2}+X_{i3}-1:1\leq i\leq n\}\}$. *De este modo, cada vértice tendrá al menos un color. Esto se debe a que cada ecuación $X_{i1}+X_{i2}+X_{i3}-1=0$ garantiza que, o bien las tres variables valen 1, o solo una de ellas vale 1. Al juntar esta condición con la condición (i), obtenemos que cada vértice podrá estar coloreado únicamente por un solo un color.*

(iii) $G_2=\{X_{i1}X_{j1},X_{i2}X_{j2},X_{i3}X_{j3},\{i,j\}\in E\}$, *es decir, dos vértices adyacentes han de tener distinto color, ya que las ecuaciones $X_{ik}X_{jk}=0$ garantizan que una de las variables ha de ser 0.*

En este criptosistema, la clave pública es G, y la clave privada es una 3-coloración de Γ. Para encriptar un mensaje $m\in\mathbb{F}_2$, escogemos un polinomio aleatorio p en el ideal generado por los polinomios de G. Encriptamos $c=p+m$, y el receptor, que conoce una 3-coloración del grafo (que no deja de ser un cero α del sistema de ecuaciones), evalúa $c(\alpha)=p(\alpha)+m=m$.

3.2. Ataque lineal a un criptosistema Polly Cracker

Dado que los criptosistemas Polly Cracker son un caso particular de los criptosistemas Barkee, el ataque de Moriarty a la versión densa visto en 2.4. también será capaz de romper un criptosistema Polly Cracker denso. Veamos cómo efectuar dicho ataque. El mensaje cifrado tiene la siguiente estructura

$$c=p+m=\textstyle\sum_{i=1}^{s}h_if_i+m, \text{ con } m\in K.$$

Debemos identificar de qué grado son los polinomios que buscamos. Los h_i y los f_i polinomios densos, supongamos que el máximo grado (en términos de densidad) de dichos polinomios es d_1 y d_2 respectivamente. Precisamente por ser densos, lo más probable es que los polinomios h_i sean de grado $d=gr(c)-d_1$.

Llamamos $K'=\mathbb{F}[m',\{\beta_\mu\}^{(1)}_{\mu\in T_q(n,d)},...,\{\beta_\mu\}^{(s)}_{\mu\in T_q(n,d)}]$. Generamos

$$c'=\textstyle\sum_{i=1}^{s}f_ih_i', \text{ con } h_i'=\sum_{\alpha\in T_q(n,d)}\beta_\alpha^{(i)}x^\alpha\in K'[x_1,...,x_n].$$

Reorganizamos los términos de c' para escribir $c'=\sum_{\alpha\in T_q(n,gr(c))}g_\alpha x^\alpha$, donde $g_\alpha\in K'$. Dichos g_α son polinomios lineales en $K'[x_1,...,x_n]$, dado que solo multiplicamos los coeficientes de los h_i' con los coeficienetes de los f_i, que son conocidos.

Podemos, por tanto, comparar los coeficientes de los $g_\alpha x^\alpha$ y c mediante un sistema de ecuaciones lineales, que representaremos en forma matricial. Resolvemos dicho sistema mediante eliminación gaussiana. Entonces:

(i) Si el sistema posee soluciones, la coordenada m' de dicha solución será el mensaje m que buscábamos.

(ii) Si no posee solución, incrementamos d en una unidad y repetimos todo el proceso.

Corolario. *El ataque previamente descrito es correcto, y nos permite desencriptar el mensaje original m. Además, dicho ataque termina en, como máximo, en* $d_2+d_1-gr(c)+1$ *iteraciones.*

Demostración. Si el sistema lineal obtenido tiene soluciones, el sistema genera polinomios $h'_i \in K[x_1,...,x_n]$, y un $m' \in K$ tales que $c=\sum_{i=1}^{s} h'_i f_i + m'$. Claramente se tiene que $m = c(\xi_0) = \sum_{i=1}^{s} h'_i(\xi_0) f_i(\xi_0) + m' = m'$, de modo que $m' = m$ para toda solución que encontremos.

Por otro lado, en la iteración número $d_2+d_1-gr(c)+1$, habremos aumentado el valor de d hasta d_2, por lo tanto el sistema tendrá al menos una solución: los coeficientes de los polinomios originales h_i utilizados en el proceso de encriptado, y el mensaje m. □

Ejemplo 10. *Veamos un ejemplo en* $\mathbb{F}_{17}[x,y]$. *Consideraremos que en la clave pública hay solo dos polinomios (*$s=2$*)*

$$f_1 = x+3y+2 \text{ y } f_2 = 3x+2y+1,$$

y el mensaje cifrado (construido de tal manera que encripte el mensaje $m=2$*) será*

$$c = 7x^2+8xy+14x+3y^2+11y+7.$$

Notemos que los f_i *tienen una estructura de polinomio denso de grado* 1. *Dado que estamos atacando un criptosistema Polly Cracker denso, lo más probable es que el grado de los* h_i *sea* $d = gr(c) - maxgr(f_i) = 2-1 = 1$. *De modo que construimos nuestros polinomios* h'_i *de la siguiente manera:*

$$\begin{aligned} h'_1 &= \beta_0^{(1)} + \beta_{(1,0)}^{(1)} x + \beta_{(0,1)}^{(1)} y, \\ h'_2 &= \beta_0^{(2)} + \beta_{(1,0)}^{(2)} x + \beta_{(0,1)}^{(2)} y. \end{aligned}$$

Se tiene que $T_q(n,gr(c)) = T_{17}(2,2) = \{(0,0),(1,0),(0,1),(2,0),(1,1),(0,2)\}$ *será el conjunto de los exponentes de los polinomios densos que usaremos para desencriptar el mensaje. Construimos nuestros* g_α *observando los coeficientes en* f_1 *y* f_2*:*

$$\begin{aligned} g_0 &= 2\beta_0^{(1)} + \beta_0^{(2)} + m', \\ g_{(1,0)} &= \beta_0^{(1)} + 2\beta_{(1,0)}^{(1)} + 3\beta_0^{(2)} + \beta_{(1,0)}^{(2)}, \\ g_{(0,1)} &= 3\beta_0^{(1)} + 2\beta_{(0,1)}^{(1)} + 2\beta_0^{(2)} + \beta_{(0,1)}^{(2)}, \\ g_{(2,0)} &= \beta_{(1,0)}^{(1)} + 3\beta_{(1,0)}^{(2)}, \\ g_{(1,1)} &= 3\beta_{(1,0)}^{(1)} + \beta_{(0,1)}^{(1)} + 2\beta_{(1,0)}^{(2)} + 3\beta_{(0,1)}^{(2)}, \\ g_{(0,2)} &= 3\beta_{(0,1)}^{(1)} + 2\beta_{(0,1)}^{(2)}. \end{aligned}$$

Para terminar, debemos resolver el siguiente sistema lineal

$$\begin{bmatrix} 2 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 2 & 0 & 3 & 1 & 0 & 0 \\ 3 & 0 & 2 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 3 & 0 & 0 \\ 0 & 3 & 1 & 0 & 2 & 3 & 0 \\ 0 & 0 & 3 & 0 & 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} \beta_0^{(1)} \\ \beta_{(1,0)}^{(1)} \\ \beta_{(0,1)}^{(1)} \\ \beta_0^{(2)} \\ \beta_{(1,0)}^{(2)} \\ \beta_{(0,1)}^{(2)} \\ m' \end{bmatrix} = \begin{bmatrix} 7 \\ 14 \\ 11 \\ 7 \\ 8 \\ 3 \end{bmatrix},$$

sistema que nos devuelve el valor $m' = 2$ tras aplicar eliminación gaussiana. Este valor es, en efecto, el mensaje inicial que se quería enviar, puesto que c lo habíamos construido de la siguiente manera

$$c = (x+y+1)f_1 + (2x+3)f_2 + 2.$$

Si dicho sistema no admitiera solución alguna, aumentaríamos el grado de los h_i' en una unidad, y repetiríamos el proceso. Ya sabíamos que no sería necesario, dado que el corolario anterior nos garantiza que el ataque termina en $d_1 + d_2 - gr(c) + 1 = 1 + 1 - 2 + 1 = 1$ iteración.

Una manera de solucionar esta susceptibilidad es, de nuevo, trabajar con la versión *sparse*. Veamos un ejemplo de ello.

Ejemplo 11. *Trabajemos sobre $\mathbb{F}_{13}[x,y]$. Nuestra clave pública está conformada, de nuevo, por solo dos polinomios*

$$f_1 = 3x^{11}y^9 + 2 \text{ y } f_2 = 7x^9y^7 + 5.$$

Notemos que ambos polinomios son sparse, dado que su grado es elevado y su soporte es reducido. El mensaje cifrado (que esconde $m = 7$) es

$$c = 8x^{11}y^{11} + 12x^{11}y^9 + 6x^9y^7 + 2x^2y^2 + 6.$$

Si efectuásemos el ataque de Moriarty previamente descrito, comenzaríamos construyendo los polinomios h_i' con grado $d = 2$. Sin embargo, la construcción de c la hemos realizado de la siguiente manera

$$c = (4x^{11}y^{11})f_1 + (3x^2y^2 + 1)f_2 + 7.$$

Por lo tanto, deberemos realizar 21 iteraciones hasta deducir correctamente el grado de los h_i iniciales. Además, en cada iteración el sistema lineal que debemos resolver crece.

Como anunciábamos en el capítulo previo, hemos comprobado de forma concreta por qué el ataque de Moriarty es poco eficaz para romper criptosistemas Polly Cracker *sparse*.

3.3. Firma digital de criptosistemas de clave pública

En un mundo digitalizado, donde los mensajes por correo electrónico y otras vías de comunicación digitales son cada vez más habituales, no solo es necesario encriptar de forma segura un mensaje, sino también garantizar su procedencia. Un procedimiento de ataque habitual es que Eve (nuestra atacante) genere falsos mensajes para que Alice (nuestra receptora) los desencripte y así obtener pistas que le indiquen cómo opera el cripsosistema empleado. Este peligro puede ser evitado empleando esquemas de firma digital, y la familia de criptosistemas Polly Cracker previamente estudiada es indicada para ello. Veamos cómo se procede de manera genérica, tal y como apuntan R.L. Rivest, A. Shamir, y L. Adleman [7], a la firma de mensajes en criptosistemas de clave pública.

Supongamos de nuevo que Alice y Bob son dos usuarios de un criptosistema de clave pública, y que ambos poseen un método de encriptado y otro de desencriptado. Llamemos a los métodos de encriptado y desencriptado de Alice E_A y D_A respectivamente, y E_B y D_B a los métodos de Bob. Para enviar un mensaje m, Bob genera su firma, que será $S = D_B(m)$. Una vez hecho esto, emplea el proceso de encriptado de Alice para encriptar su firma y se lo envía a Alice. Ella recibe $E_A(S)$, que puede desencriptar usando su procedimiento de desencriptado D_A para obtener S y emplea el método de encriptado de Bob para obtener $E_B(S) = m$. De forma esquemática sería

$$m \xrightarrow{\text{Bob firma}} D_B(m) \xrightarrow{\text{Bob encripta}} E_A(D_B(m)) \xrightarrow{\text{Alice desencripta}} D_B(m) \xrightarrow{\text{Alice encripta}} m.$$

Notar que este procedimiento es posible únicamente en criptosistemas de clave pública, puesto que tanto Bob como Alice emplean el método de encriptado del otro durante el proceso. Alice ya no solo recibe un mensaje, sino que recibe un mensaje y una firma (m, S). Este procedimiento supone que la firma depende tanto del mensaje enviado como del emisor del mensaje. Además, Alice tiene la garantía de que el mensaje está enviado por Bob, dado que solo Bob puede generar dicha firma, y Bob tiene la certeza de que Alice no puede modificar el mensaje, pues si lo hiciera, la firma cambiaría.

Capítulo 4

Conclusión

A lo largo de este documento hemos analizado la teoría de bases de Groebner y ciertos criptosistemas profundamente ligados a ella. En el último capítulo, hemos comprobado cómo realizar un ataque a un criptosistema Polly Cracker concreto, y cómo evitar dicho ataque. Sin embargo, a pesar de que hayamos dado una solución concreta para evitar el ataque, existen otros tantos ataques antes los cuales la versión *sparse* se vuelve susceptible. Por citar un ejemplo, el ataque inteligente mediante álgebra lineal es uno de aquellos capaces de romperlo. El trasfondo de dicho ataque es, en esencia, parejo al ataque de Moriarty, pero escogiendo sabiamente aquellos coeficientes de los polinomios h_i' de forma que se reduzca considerablemente el tamaño del sistema lineal a resolver. Dicho ataque es tremendamente efectivo para romper la variante *sparse*. No obstante, el uso simultáneo de polinomios densos de grado reducido y polinomios *sparse* de grado elevado en la clave pública genera un sistema eficaz para evitar el ataque inteligente mediante álgebra lineal. Sirva este párrafo como metáfora de la necesidad de renovación en el campo de la criptografía. Cada vez que un criptosistema se ve vulnerado, aparece un nuevo desafío.

La riqueza teórica que subyace bajo los criptosistemas de clave pública basados en la teoría de bases de Groebner los convierte en un interesante objeto de estudio, a pesar de que numerosos autores como B. Barkee, D. C. Can, J. Ecks, T. Moriarty y R. F. Ree [3] coincidan en que serán siempre vulnerables. No obstante, las bases de Groebner se encuentran detrás de grandes logros en otros muchos campos: robótica, procesamiento de la imagen, estadística e ingeniería entre otros.

Un muy interesante estudio realizado por T. S. Rai se puede encontrar en [6], donde se propone como alternativa trabajar no sobre un anillo de polinomios, sino sobre un álgebra libre (es decir, eliminando la propiedad conmutativa de nuestro anillo de polinomios). Esto lleva a construir una teoría análoga sustentada en ideales biláteros. En dicho documento se establece un claro paralelismo entre todos los resultados aquí desarrollados: el algoritmo de la división en el caso no conmutativo, bases de Groebner no conmutativas, algoritmo de Buchberger no conmutativo, etc. La principal motivación de este estudio es el hecho de que, sobre un álgebra libre, existen ideales que no son finitamente generados. Esto, en teoría, garantiza que el cálculo de una base de Groebner para dichos ideales es completamente inviable. Sin embargo, autores como M. Sala, T. Mora, L. Perret, S. Sakata y C. Traverso [8] señalan que este hecho dificulta tanto el trabajo del desencriptador como el del encriptador, de modo que descartan su uso.

El constante avance en el terreno de la computación es uno de los principales retos para la

criptografía. Cálculos que antaño resultaban costosos se pueden realizar en cuestión de segundos gracias a los avances en rapidez de computación. La certeza de un futuro de ordenadores cuánticos vulneraría completamente la seguridad del sistema criptográfico por excelencia: el criptosistema RSA. Todo ello genera la necesidad de, o bien mejorar los métodos de encriptado existentes, o bien crear procedimientos alternativos. El uso de criptosistemas cuyo trasfondo es la teoría de bases de Groebner es uno de los ingeniosos intentos que seguirán estudiándose para solventar esta situación.

Bibliografía

[1] Bruno Buchberger. "Bruno Buchberger's PhD thesis 1965: An Algorithm for Finding the Basis Elements of the Residue Class Ring of a Zero Dimensional Polynomial Ideal". En: *J. Symb. Comput.* 41 (mar. de 2006), págs. 475-511. DOI: 10.1016/j.jsc.2005.09.007.

[2] John Little David Cox & Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 1992, págs. 1-111. ISBN: 978-0-387-35651-8.

[3] Julia Ecks y col. "Why You Cannot Even Hope to Use Göbner Bases in Public-Key Cryptography? An Open Letter to a Scientist Who Failed and a Challenge to Those Who Have Not Yet Failed". En: *Journal of Symbolic Computation* 18 (jun. de 2004). DOI: 10.1006/jsco.1994.1061.

[4] Neal Koblitz. *Algebraic aspects of cryptography. Algorithms and Computation in Mathematics, 3.* Springer-Verlag, 1998, págs. 22-27. ISBN: 978-3540634461.

[5] Le Van Ly. "Polly Two - a public-key cryptosystem based on Polly Cracker". En: *http://www-brs.ub.ruhr-uni-bochum.de/netahtml/HSS/Diss/LyLeVan/diss.pdf* (ene. de 2002).

[6] Tapan S. Rai. *Infinite Grobner Bases And Noncommutative Polly Cracker Cryptosystems*. Springer, 2004. ISBN: 978-0-387-35651-8.

[7] Ronald Rivest, Adi Shamir y Leonard M. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". En: *Commun. ACM* 26 (ene. de 1983), págs. 96-99. DOI: 10.1145/359340.359342.

[8] Massimiliano Sala y col. *Gröbner Bases, Coding, and Cryptography*. Ene. de 2009, págs. 300-301. DOI: 10.1007/978-3-540-93806-4.