Matemàtica Discreta II

Pràctiques

Maria Bras Amorós
Oriol Farràs Ventura

Universitat Rovira i Virgili
Novembre 2025

Aquest llibre és un resultat del Projecte Estratègic “HERMES: Ciberseguretat en computació descentralitzada” i de la Càtedra en Ciberseguretat INCIBE-URV, ambdós fruits de convenis de col·laboració subscrits entre l’Institut Nacional de Ciberseguretat (INCIBE) i la Universitat Rovira i Virgili. Aquestes iniciatives (finançades per la Unió Europea Next Generation-EU) es realitzen en el marc del Pla de Recuperació, Transformació i Resiliència, el projecte del Govern d’Espanya que traça el full de ruta per a la modernització de l’economia espanyola, la recuperació del creixement econòmic i la creació d’ocupació després de la crisi de la Covid-19, així com per preparar al país per afrontar els reptes de la pròxima dècada.





Introducció

Aquest llibre recull el conjunt de pràctiques corresponents a Matemàtica Discreta II, una assignatura impartida a la Universitat Rovira i Virgili que introdueix els fonaments matemàtics de les comunicacions digitals i de la teoria de la codificació aplicada al tractament d'errors.

Les pràctiques que s'hi presenten tenen com a objectiu familiaritzar l'estudiant amb els resultats teòrics, ampliar els continguts treballats a classe i mostrar-ne les aplicacions pràctiques. Cada capítol és autocontingut i inclou la teoria necessària per comprendre i desenvolupar les activitats proposades.

El capítol 1 introdueix l'aritmètica, l'aritmètica modular i els cossos finits, i presenta diverses aplicacions en l'àmbit de la criptografia i la detecció d'errors. S'estudien el protocol d'intercanvi de claus de Diffie-Hellman i els esquemes de xifratge AES i RSA, que permeten il·lustrar la relació entre problemes matemàtics i la criptografia.

El capítol 2 està dedicat als codis lineals, i proposa exercicis que n'exploren les propietats fonamentals i aspectes pràctics de la seva implementació. També s'hi presenten els codis de Hamming.

El capítol 3 aborda els codis cíclics, dels quals es revisen les propietats principals i les seves aplicacions. En aquest marc s'introdueixen els codis CRC, àmpliament utilitzats en la detecció d'errors en transmissions digitals. També es presenten els esquemes de compartició de secrets definits a partir de codis lineals, mostrant una connexió entre la teoria de codis i la criptografia.

El capítol 4 es centra en els codis Reed-Solomon, fonamentals en múltiples tecnologies de transmissió i emmagatzematge. La primera part del capítol presenta les matrius de Vandermonde, els codis RS primitius i diverses generalitzacions d'aquests codis. La segona part tracta aspectes pràctics de la seva implementació en hardware i la seva relació amb l'esquema de compartició de secrets de Shamir.

El capítol 5 està dedicat a dissenys avançats de codis que van més enllà del contingut teòric bàsic de l'assignatura. La primera pràctica introdueix l'entrellaçat de codis, una tècnica que redueix l'impacte dels errors en ràfega i que troba aplicació en la codificació de la informació en els discs compactes. També s'analitza l'ús dels codis lineals en l'emmagatzematge distribuït al núvol, i es presenten altres famílies de codis rellevants, com els codis de Golay, de Goppa i de Reed-Muller. El capítol conclou amb una pràctica dedicada a l'esquema criptogràfic de McEliece, basat en codis de Goppa, que exemplifica la vigència dels codis lineals en la criptografia postquàntica.

Els capítols 6 i 7 estan dedicats a pràctiques de laboratori desenvolupades amb el programari SAGE, que permeten aplicar de manera pràctica els conceptes apresos al llarg de l'assignatura. Aquestes activitats ofereixen a l'estudiant una oportunitat per consolidar els coneixements mitjançant la resolució de problemes i l'experimentació computacional, integrant teoria i pràctica en un entorn interactiu. El capítol 6 és un manual bàsic del programari, i el capítol 7 conté les pràctiques.

Índex

1 Aritmètica **7**
- 1.1 Aritmètica entera . . . 7
- 1.2 Aritmètica modular . . . 8
- 1.3 Codis de barres . . . 10
- 1.4 Criptosistema Diffie-Hellman . . . 11
- 1.5 Criptosistema RSA . . . 14
- 1.6 Aritmètica polinomial . . . 17
- 1.7 Cossos finits . . . 19
- 1.8 Criptosistema AES . . . 22
- 1.9 Algoritme de Shor . . . 24

2 Codis lineals **27**
- 2.1 Propietats . . . 27
- 2.2 Codi dual . . . 30
- 2.3 Codis de Hamming . . . 33
- 2.4 Construcció de codis lineals . . . 34
- 2.5 Detecció i correcció d'errors . . . 35
- 2.6 Implementació hardware . . . 36

3 Codis cíclics **39**
- 3.1 Propietats . . . 39
- 3.2 Identificació de codis cíclics . . . 44
- 3.3 Codis cíclics primitius . . . 46
- 3.4 Codis CRC . . . 48
- 3.5 Esquemes de compartició de secrets . . . 51

4 Codis Reed-Solomon **55**
- 4.1 Matrius de Vandermonde . . . 55
- 4.2 Codis Reed-Solomon primitius . . . 57
- 4.3 Codis Reed-Solomon no primitius . . . 60
- 4.4 Implementació de Codis Reed-Solomon . . . 62
- 4.5 Errors de transmissió . . . 65
- 4.6 Esquema de Shamir . . . 66

5 Dissenys Avançats **67**
- 5.1 Interleaving . . . 67
- 5.2 Codificació del Compact Disc . . . 69
- 5.3 Locally Repairable Codes . . . 73

5.4 Codis Golay 76
5.5 Codis de Goppa 77
5.6 Criptosistema de McEliece 78
5.7 Codis Reed-Muller 80

6 Manual de SAGE **81**
6.1 Indicacions per començar 81
6.2 Accions simples i expressions 82
6.3 Composició d'accions 83
6.4 Conjunts i seqüències 83
6.5 Funcions 85
6.6 Enters, anells $\mathbb{Z}_n$, racionals i reals 86
6.7 Anell de polinomis 88
6.8 Cossos finits 89
6.9 Àlgebra lineal 90

7 Pràctiques de SAGE **95**
7.1 Introducció a SAGE 95
7.2 Cossos finits 97
7.3 Codis lineals 99
7.4 Codis cíclics 101
7.5 Codis Reed-Solomon 102

Capítol 1
Aritmètica

1.1 Aritmètica entera

Exercici 1: Calculeu el màxim comú divisor de les següents parelles d'enters, i expresseu-lo com a combinació lineal d'aquests enters.

12, 15	18, 24	35, 255
12, 16	40, 64	144, 180

Exercici 2: Quantes solucions tenen aquestes equacions amb congruències? En el cas que en tinguin, doneu-ne les solucions.

$4x \equiv 1 \mod 16$	$4x \equiv 2 \mod 16$	$4x \equiv 8 \mod 16$
$4x \equiv 1 \mod 17$	$4x \equiv 2 \mod 17$	$4x \equiv 8 \mod 17$
$4x \equiv 1 \mod 18$	$4x \equiv 2 \mod 18$	$4x \equiv 8 \mod 18$

Exercici 3: Escriviu un algoritme que factoritza un enter, suposant que disposeu d'una llista amb tots els nombres primers inferiors al nombre en qüestió. Intenteu reduir el temps d'execució.

Exercici 4: Escriviu tres algoritmes que calculin el mcd de dos enters:

(a) Un que faci servir l'algoritme de factorització d'enters de l'exercici anterior,

(b) Un algoritme que reprodueixi el d'Euclides, i

(c) Un algoritme que sigui recursiu.

Raoneu quin dels tres és més eficient.

Exercici 5: Escriviu un algoritme que, a partir de dos enters, calculi els coeficients de la identitat de Bézout.

Exercici 6: Contesteu justificadament els següents apartats:

(a) Llisteu els enters m entre 2 i 10 (ambdós inclosos) pels que $\mathbb{Z}_m$ és un cos.

(b) Doneu tres m's diferents tals que $\mathbb{Z}_m^*$ tingui 4 elements.

1.2 Aritmètica modular

Exercici 1: Per cada un dels enters n entre 10 i 20,

(a) calculeu la phi d'Euler, i

(b) digueu explícitament quins enters són coprimers amb n.

Exercici 2: Escriviu la taula de la multiplicació de $\mathbb{Z}_7$ i de $\mathbb{Z}_8$.

Exercici 3: Sigui l'anell $\mathbb{Z}_{13}$.

(a) Quants divisors de zero té?

(b) Trobeu un element primitiu.

(c) Calculeu l'ordre de tots els elements invertibles.

Exercici 4: Sigui l'anell $\mathbb{Z}_{15}$.

(a) És un cos? Per què?

(b) Quants elements de $\mathbb{Z}_{15}$ són invertibles?

(c) Justifiqueu per què té invers el 7 a $\mathbb{Z}_{15}$.

(d) Trobeu l'invers de 7 a $\mathbb{Z}_{15}$ utilitzant la identitat de Bézout.

(e) Trobeu l'invers de 7 a $\mathbb{Z}_{15}$ utilitzant el Teorema d'Euler.

(f) Comproveu que l'element trobat és, en efecte, l'invers.

Exercici 5: Sigui l'anell $\mathbb{Z}_{16}$.

(a) És un cos?

(b) Quants elements invertibles té?

(c) Quins són tots els elements invertibles?

(d) Doneu un element invertible diferent de 1 i el seu invers.

(e) Demostreu que un element $\alpha \in \mathbb{Z}_{16}^*$ és primitiu si i només si $\alpha^4 \neq 1$.

(f) Utilitzeu l'apartat anterior per veure que $\mathbb{Z}_{16}$ no té elements primitius.

Exercici 6: Sigui l'anell $\mathbb{Z}_{19}$.

(a) És un cos?

(b) Quants elements invertibles té i quins són?

(c) Quants divisors de zero té?

(d) Busqueu un element primitiu.

(e) Busqueu un element diferent de $0, 1$ que no sigui primitiu. Quin ordre té?

Exercici 7: Sigui l'anell $\mathbb{Z}_{24}$.

(a) Quants divisors de zero té?

(b) Comproveu que tots els elements invertibles tenen ordre 2.

(c) Existeix algun element primitiu?

1.3 Codis de barres

Els codis de barres van néixer amb la intenció d'identificar productes de manera automàtica. Cada producte va associat a un codi diferent i a una representació pictòrica que es pot escanejar fàcilment.

La majoria dels productes que comprem duen un codi de barres creat d'acord amb el EAN-13. Cada codi de barres es correspon amb un número. El d'EAN-13 consta de 12 dígits que s'assignen de manera única a un producte, i un dígit de control. Aquest darrer dígit no aporta informació sobre el producte, però permet detectar algunes lectures errònies del codi.

L'últim dígit es calcula a partir dels 12 anteriors de manera que la suma de totes les xifres en posició senar més el triple de la suma de totes les xifres en posició parell és un múltiple de 10.

Exercicis:

(a) Comproveu-ho amb algun codi de barres que tingueu a mà.

(b) Definiu una funció que, donats tots els dígits d'un codi de barres fins al penúltim, en calculi l'últim.

Els codis que utilitzen redundància per tal de poder detectar possibles errors de lectura o transmissió s'anomenen codis detectors d'errors. Més endavant veurem diverses construccions.

1.4 Criptosistema Diffie-Hellman

Context històric

- Un dels usos fonamentals de la criptografia és garantir confidencialitat a les comunicacions. Una situació que es repeteix és la següent: tenim dos usuaris (que anomenem Alice i Bob) que es volen comunicar a través d'un canal insegur.
- Des de fa segles, existeixen criptosistemes de *clau simètrica*. En el cas que dos usuaris disposin d'una clau secreta k (que només ells saben), empraran aquesta clau per xifrar i desxifrar missatges. En criptografia, els usuaris s'acostumen a anomenar Alice (A) i Bob (B).
- Per emprar aquests criptosistemes que funcionen amb la clau k, abans s'ha de resoldre un altre problema: la distribució de claus. S'ha d'aconseguir que Alice i Bob es posin en contacte i puguin acordar una clau sense que ningú més la sàpiga. Aquest va ser un gran problema fins fa 50 anys.

Criptografia de clau pública

- El 1976 Diffie i Hellman van presentar un protocol d'intercanvi de claus i van introduir la noció de criptografia de clau pública. Va ser en un treball titulat *New Directions in Cryptography.* Sovint es considera que aquest treball va donar pas al que es considera la criptografia moderna.
- En els esquemes de clau pública, cada usuari té un parell de claus, que anomenem clau pública i clau privada.
- La clau pública pot ser coneguda pels altres usuaris, però la privada només l'ha de saber l'usuari en qüestió, i la ha de mantenir secreta.
- En el protocol de Diffie i Hellman, Alice i Bob generen una clau que només ells sabran. Normalment, l'objectiu d'aquest protocol és establir una clau compartida que després es farà servir en esquemes de criptografia simètrica. Aquest protocol es pot dur a terme a través d'un canal de comunicació insegur, on potser hi ha altres usuaris llegint els missatges intercanviats.
- La seguretat d'aquest protocol es basa en la dificultat de calcular el logaritme discret: Diem que x és el logaritme de h en base g si $g^x = h$. Escrivim $x = \log_g(h)$.

A continuació veurem una versió simplificada del protocol DH:

(A1) Alice tria un nombre primer p i busca un generador g de $\mathbb{Z}_p^*$.

(A2) Alice tria un enter $0 < x < p$ i calcula $h_A = g^x$

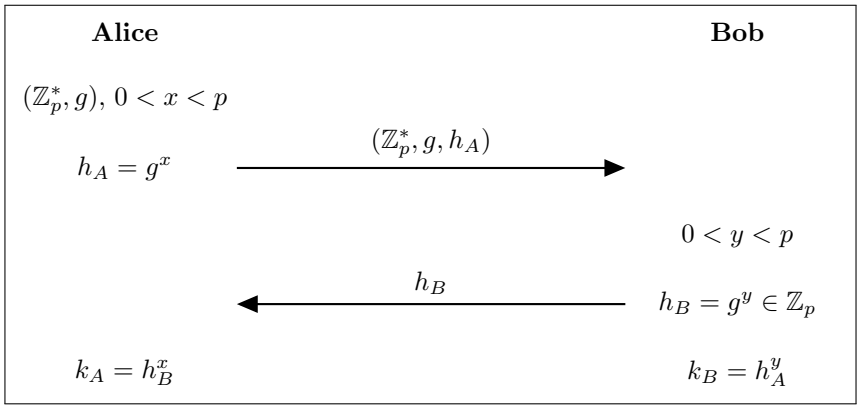
(A3) Alice envia a Bob p, g i h_A. Alice manté x secreta.

(B1) Bob genera $0 < y < p$ i calcula $h_B = g^y$.

(B2) Bob envia h_B a Alice i calcula $k_B = h_A^y$.

(A4) Alice calcula $k_A = h_B^x$.

Si una algú intercepta la comunicació i aconsegueix h_A i h_B, és molt difícil que pugui arribar a calcular k_A.



Exercicis:

(a) Comproveu que, si no hi ha errors de transmissió, $k_A = k_B$.

(b) Considereu el grup multiplicatiu $\mathbb{Z}_{11}^*$. Trobeu els generadors de $\mathbb{Z}_{11}^*$.

(c) Calculeu una instància del protocol amb $p = 11$.

(d) Calculeu una instància del protocol amb $p = 17$.

(e) Calculeu els següents logaritmes discrets, si és possible, a $\mathbb{Z}_{11}^*$: $\log_2(3)$, $\log_4(5)$, $\log_8(3)$, $\log_5(3)$, $\log_4(2)$, i $\log_5(2)$.

(f) Considereu ara una generalització del protocol amb $\mathbb{Z}_n^*$ en comptes de $\mathbb{Z}_p^*$. Pots garantir que existirà un generador de $\mathbb{Z}_n^*$? Proveu-ho amb $\mathbb{Z}_{12}^*$ i $\mathbb{Z}_{15}^*$.

Usos actuals i futurs:

- DH segueix sent una eina fonamental per garantir la seguretat de les comunicacions digitals. A la pràctica, en comptes de fer els càlculs en el grup $\mathbb{Z}_p^*$, com a l'exercici, s'empra el grup de punts de corbes el·líptiques. Per evitar atacs man-in-the-middle, es fan servir firmes digitals i certificats.
- DH es fa servir en els protocol TLS i https, així com en aplicacions de missatgeria (com whatsapp) per establir comunicacions segures entre usuaris (end-to-end encryption).
- Actualment es segueixen emprant variants d'aquest esquema. Els paràmetres, però estan molt lluny dels de l'exemple que veurem. Actualment, els paràmetres pràctics són $\mathbb{Z}_N$, on N és un enter de 2048 bits. Per tant, N és de l'ordre de 10^{600}.
- És probable que es deixi d'emprar en un futur (proper o llunyà). Des dels anys 90s, es sap que amb un ordinador quàntic suficientment potent es podria trencar RSA i DH amb poques instruccions. Concretament, es sap que es pot factoritzar N amb un algoritme que s'executa en temps polinòmic (quadràtic) en el nombre de bits de N. Ara, però, encara no hi ha ordinadors quàntics capaços de factoritzar nombres de 2048 bits.
- En els propers anys està previst substituir RSA i DH per altres esquemes que es consideren segurs envers atacants que poden disposar d'ordinadors quàntics molt potents. Aquests esquemes reben l'atribut de post-quàntics.

1.5 Criptosistema RSA

Criptosistema RSA

- El 1976 Diffie i Hellman van presentar un protocol d'intercanvi de claus i van introduir la noció de criptografia de clau pública (pràctica anterior).
- El 1977, Rivest, Shamir i Adleman van construir el primer esquema d'encriptació de clau pública, que amb el temps s'ha anomenat RSA. En aquest treball, també van presentar esquemes de firma digital, que també s'anomenen RSA.
- Suposem que Alice i Bob es volen comunicar. Tant Alice com Bob, generaran parells de claus (sk_A, pk_A) i (sk_B, pk_B) i faran públiques les claus pk_A i pk_B.
- Quan Bob vol enviar un missatge a Alice, xifrarà el missatge emprant pk_A. Quan Alice rebi el missatge xifrat, el podrà desxifrar emprant sk_A. Ara en veurem els detalls. La seguretat i la correcció de l'esquema es basa en alguns problemes d'artimètica modular. En aquest cas, les claus de Bob no es faran servir.
- Als anys 70s, l'agència d'intel·ligència britànica va descobrir independentment la noció de criptografia de clau pública, una variant de l'esquema RSA, i una variant del protocol de Diffie-Hellman. Tot i així, el descobriment es va mantenir en secret. No va ser fins a finals dels anys 90s que aquest secret es va desclassificar.

A continuació veurem una versió simplificada de l'esquema RSA:

(A1) Alice tria dos nombres primers p i q. Calcula $N = pq$.

(A2) Alice tria un enter $0 < e < N$ tal que $mcd(e, \phi(N)) = 1$. Calcula

$$d = e^{-1} \mod \phi(N).$$

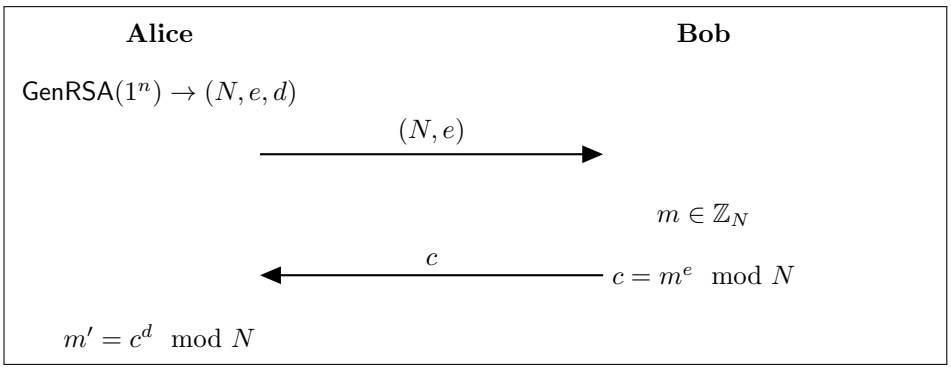
O sigui, d i e satisfan $ed = 1 \mod \phi(N)$.

(A3) Alice publica $pk = (N, e)$ i es guarda $sk = (N, d)$.

(B1) Si Bob té un missatge $m \in \mathbb{Z}_N^*$ per enviar a Alice i coneix pk, calcula $c = m^e \mod N$.

(B2) Bob envia c a Alice.

(A4) Al rebre c, Alice calcula $m' = c^d \mod N$. Si no hi ha hagut errors de transmissió, $m' = m$.



Comentaris sobre l'esquema RSA:

- Si una algú intercepta la comunicació i aconsegueix c és molt difícil que pugui arribar a conèixer m a partir de c i pk. En particular, és difícil calcular d a partir de N i e quan N és el producte de dos nombres primers grans.
- Aquest problema està relacionat amb el problema de factoritzar enters. Si fos fàcil factoritzar N, aleshores es podria calcular $\phi(N)$ i obtenir d.
- Amb computació convencional, factortizar nombres és molt costós. El nombre d'operacions necessàries creix subexponencialment amb la longitud en bits de N.
- En canvi, amb un ordinador quàntic suficientment potent, es podria factoritzar amb un algoritme que s'executa en temps polinòmic. Un ordinador d'aquestes característiques trencaria RSA. En els propers anys és bastant problable que RSA s'anirà substituint per altres esquemes anomenats postquàntics. Un problema semblant afecta també al protocol d'intercanvi de claus de Diffie-Hellman

Exercicis:

(a) Comproveu que, si no hi ha errors de transmissió, $m' = m$. O sigui, comproveu que

$$(m^e)^d = m \mod N$$

(b) Sigui $p = 11$, $q = 13$ i $N = pq = 143$. Trobeu un e que compleixi $mcd(e, \phi(N)) = 1$ i

(c) Amb els paràmetres anteriors, trobeu d que satisfà $ed = 1 \mod \phi(N)$. Feu-ho de dues maneres diferents: emprant la identitat de Bézout i el teorema d'Euler.

(d) Amb els paràmetres anteriors i $m = 2$, calculeu $c = m^d \mod N$ i comproveu que $c^d = 2 \mod N$.

Usos actuals i futurs d'RSA:

- Actualment es segueixen emprant variants d'aquest esquema. Els paràmetres, però estan molt lluny dels de l'exemple que veurem. Actualment, els paràmetres pràctics són $\mathbb{Z}_N$, on N és un enter de 2048 bits. Per tant, N és de l'ordre de 10^{600}.
- És probable que es deixi d'emprar en un futur (proper o llunyà). Des dels anys 90s, es sap que amb un ordinador quàntic suficientment potent es podria trencar RSA amb poques instruccions. Concretament, es sap que es pot factoritzar N amb un algoritme que s'executa en temps polinòmic (quadràtic) en el nombre de bits de N. Ara, però, encara no hi ha ordinadors quàntics capaços de factoritzar nombres de 2048 bits.
- En els propers anys és bastant problable que RSA s'anirà substituint per altres esquemes anomenats post-quàntics. Un problema semblant afecta també al protocol d'intercanvi de claus de Diffie-Hellman

1.6 Aritmètica polinomial

Exercici 1: Aquest exercici tracta de la descomposició de polinomis en producte de factors irreductibles. Sigui p un nombre primer.

(a) Algoritme A: Escriviu un algoritme que, donat un polinomi $a(x) \in \mathbb{Z}_p[x]$, trobi si el polinomi és irreductible o no.

(b) Milloreu l'algoritme A, assumint que es disposa d'una llista amb tots els polinomis irreductibles de grau inferior al grau d'$a(x)$.

(c) Modifiqueu l'algoritme anterior de manera que si el polinomi és reductible, retorni la descomposició d'$a(x)$ en el producte de polinomis irreductibles mònics per una constant. Proposeu un format per escriure el resultat de la descomposició.

(d) Algoritme B: Feu servir l'algoritme A per construir un algoritme que, donat un enter positiu n, troba tots els polinomis irreductibles de grau menor o igual que n. Podeu assumir que disposeu d'una llista on poder escriure tots els polinomis irreductibles.

En tots aquests apartats, feu una estimació del temps d'execució, i raoneu si seria possible reduir-lo.

Exercici 2: Escriviu tres algoritmes que calculen el mcd de dos polinomis a $\mathbb{Z}_p[x]$, amb p primer:

(a) Un que faci servir l'algoritme de factorització de polinomis de l'exercici anterior,

(b) Un algoritme que reprodueixi el d'Euclides, i

(c) Un algoritme que sigui recursiu.

Raoneu quin dels tres és més eficient.

Exercici 3: Escriviu un algoritme que a partir de dos polinomis a $\mathbb{Z}_p[x]$, amb p primer, calculi els coeficients de la identitat de Bézout.

Exercici 4: Calculeu el màxim comú divisor de la següent parella de polinomis, i expresseu-lo com a combinació lineal polinomial dels polinomis inicials.

(a) $x^4 + 1$ i $x^2 + 1$, definits a $\mathbb{Z}_2[x]$

(b) $x^4 + 1$ i $x^2 + 1$, definits a $\mathbb{Z}_3[x]$

(c) $x^5 + x^2 + x + 1$ i $x^3 + 1$, definits a $\mathbb{Z}_2[x]$

(d) $x^5 + x^2 + x + 1$ i $x^3 + 1$, definits a $\mathbb{Z}_3[x]$

(e) $x^5 + x^2 - x - 1$ i $x^3 - 1$, definits a $\mathbb{Z}_3[x]$

Polinomis irreductibles.

- Un polinomi és irreductible si no es pot descompondre com a producte de polinomis de grau més gran o igual que 1.
- Per comprovar que un polinomi de grau n és irreductible n'hi ha prou de comprovar que no té factors de grau més petit o igual que $\lfloor n/2 \rfloor$.
- En particular, per polinomis de grau 2 i 3, n'hi ha prou de veure que el polinomi no té arrels.
- Per polinomis de grau 4 o superior, no n'hi ha prou de comprovar que el polinomi no té arrels.

Exercici 5:

(a) Justifiqueu si són irreductibles els següents polinomis a $\mathbb{Z}_2[x]$

(a) $x^4 + 1$
(b) $x^4 + x + 1$
(c) $x^4 + x^2 + 1$
(d) $x^4 + x^3 + x + 1$
(e) $x^4 + x^3 + x^2 + x + 1$

(b) Escriviu cadascun dels polinomis de l'apartat anterior com a producte de polinomis irreductibles.

(c) Doneu el màxim comú divisor de $x^4 + 1$ i $x^4 + x^2 + 1$.

(d) Podeu expressar el màxim comú divisor de $x^4 + 1$ i $x^4 + x^2 + 1$ com a combinació lineal dels mateixos polinomis? En cas afirmatiu, doneu-ne els coeficients i feu la comprovació. En cas negatiu, doneu-ne una justificació.

Exercici 6: Justifiqueu si els polinomis (a)-(e) de l'exercici anterior són irreductibles a $\mathbb{Z}_3[x]$.

1.7 Cossos finits

Construcció d'un cos finit

- Sigui $\mathbb{Z}_p[x]$ el conjunt de polinomis amb coeficients a $\mathbb{Z}_p$ i sigui $f(x)$ un polinomi irreductible de $\mathbb{Z}_p[x]$ de grau n. Donat un polinomi qualsevol $g(x)$ de $\mathbb{Z}_p[x]$, tindrem que el residu de dividir $g(x)$ per $f(x)$ és un polinomi de grau menor que n. Per tant, els residus de dividir tots els polinomis de $\mathbb{Z}_p[x]$ per $f(x)$ seran tots els polinomis de grau menor que n.
- Considerem ara el quocient $\mathbb{Z}_p[x]/(f(x))$. Aquest quocient és el conjunt de classes de polinomis de $\mathbb{Z}_p[x]$ mòdul $f(x)$. És a dir, cada polinomi s'identifica amb el residu de dividir-lo per $f(x)$.
- Diguem α a la classe de x en el quocient $\mathbb{Z}_p[x]/(f(x))$. Tindrem que tots els elements de $\mathbb{Z}_p[x]/(f(x))$ es poden escriure de la forma $a_{n-1}\alpha^{n-1} + a_{n-2}\alpha^{n-2} + \ldots + a_0$ amb $a_i \in \mathbb{Z}_p$. És fàcil de comprovar que, d'aquests elements, n'hi ha p^n. A més com que $f(x)$ és irreductible, es té que tots aquests elements formen un cos. És el cos que anomenem $\mathbb{F}_{p^n}$ o $GF(p^n)$. Diem que p és la característica de $\mathbb{F}_{p^n}$.

Recorda:

- Un polinomi és irreductible si no es pot descompondre com a producte de polinomis de grau més gran o igual que 1.
- Per comprovar que un polinomi de grau n és irreductible n'hi ha prou de comprovar que no té factors de grau més petit o igual que $\lfloor n/2 \rfloor$.
- En particular, per polinomis de grau 2 i 3, n'hi ha prou de veure que el polinomi no té arrels.
- Per polinomis de grau 4 o superior, no n'hi ha prou de comprovar que el polinomi no té arrels.

Exercici 1: Llisteu els enters m entre 2 i 20 (ambdós inclosos) pels quals existeix un cos de m elements.

Exercici 2: Determineu per quins dels següents polinomis $m(x)$ amb coeficients a $\mathbb{Z}_3$ l'anell $\mathbb{Z}_3[x]/(m(x))$ és un cos. En el cas que sigui així, digueu quina és la cardinalitat del cos resultant.

(a) $x^2 + x + 2$

(b) $x^3 + x + 1$

(c) $x^3 + x^2 + 2$

(d) $x^4 + x^3 + 2x^2 + x + 1$

Notació potencial

- La notació polinòmica és sovint la manera natural de representar els elements del cos. Aquesta notació és útil per sumar i restar elements del cos finit però no és bona per fer multiplicacions i divisions.
- La notació potencial ens dona una manera fàcil de fer multiplicacions i divisions. En la notació potencial escrivim els elements no nuls del cos $\mathbb{F}_q$ com a potència d'un element β de manera que podem representar els elements del cos com $\{0, \beta, \beta^2, \ldots, \beta^{q-1}\}$.
- L'element β usat en la notació potencial no és un element qualsevol del cos. Necessitem que qualsevol element no nul del cos sigui potència de β. Cal que β sigui un element primitiu del cos.

Exercici 3: Considerem el cos finit $\mathbb{Z}_3[x]/(x^2 + x + 2)$.

(a) Comprova que $[x]$ és un element primitiu.

(b) Escriu la taula d'equivalències (amb els càlculs corresponents) entre les notacions potencial i polinòmica en base a l'element primitiu $[x]$.

Recorda:

- L'ordre d'un element no nul $\gamma \in \mathbb{F}_q$ és el mínim enter r tal que $\gamma^r = 1$.
- L'ordre de qualsevol element no nul de $\mathbb{F}_q$ divideix $q - 1$.
- Un element és primitiu si el seu ordre és $q - 1$.
- Per comprovar que un element és primitiu, n'hi ha prou de comprovar que per qualsevol divisor d de $q - 1$ (amb $d \neq q - 1$) tenim $\gamma^d \neq 1$.

Exercici 4:

(a) Quines de les següents estructures són un cos i, en cas de ser-ho, quants elements tenen?

  (a) $\mathbb{Z}_2[x]/(x^4 + 1)$
  (b) $\mathbb{Z}_2[x]/(x^4 + x + 1)$
  (c) $\mathbb{Z}_2[x]/(x^4 + x^2 + 1)$
  (d) $\mathbb{Z}_2[x]/(x^4 + x^3 + x^2 + x + 1)$

(b) En quins dels casos en que tenim un cos, si anomenem α a la classe de x, tenim que α és un element primitiu?

(c) Doneu una taula exponencial-polinòmica-vectorial per un cas en que α sigui primitiu. Els apartats que segueixen els referirem al mateix cas (la mateixa α i la mateixa taula).

(d) Quins són els ordres possibles dels elements del cos?

(e) Per a cadascun dels ordres possibles, doneu un element del cos amb aquell ordre.

1.8 Criptosistema AES

AES

- L'Advanced Encryption Standard (AES) és un esquema criptogràfic de clau privada que va ser estandarditzat pel National Institute of Standard and Technology (NIST) a finals del segle passat. Des d'aleshores, AES és el principal esquema de clau privada, i el fem servir a diari quan establim comunicacions segures.
- AES és un esquema de xifratge de bloc: Les dades es divideixen en diversos blocs i, durant part del procés, cada bloc es processa per separat. Les operacions que s'executen quan s'encripta un bloc són: *ByteSub*, *ShiftRow*, *MixColumn*, i *AddRoundKey*.
- Les operacions Bytesub, Mixcolumn i AddRoundKey són operacions senzilles d'aritmètica modular. Són operacions que combinen productes i sumes en $\mathbb{F}_{128}$, $\mathbb{F}_{2^8}[x]/(x^4+1)$, i $\mathbb{Z}_2^{16}$, respectivament. En aquesta pràctica ens fixarem en l'operació Bytesub.

ByteSub (I)

- Prenem el cos finit $\mathbb{F}_{2^8} = \mathbb{Z}_2[x]/(m(x))$, on

  $$m(x) = 1 + x + x^3 + x^4 + x^8 \in \mathbb{Z}_2[x]$$

  és un polinomi irreductible de grau 8. Sigui a un byte d'estat. Aquest byte es pot escriure com

  $$a = a_7a_6a_5a_4a_3a_2a_1a_0,$$

  on $a_k \in \{0, 1\}$.
- Aquest element a ara el veurem com un element del cos finit $\mathbb{F}_{2^8}$, fent servir la notació vectorial. O sigui, que a es correspon amb

  $$a_7\alpha^7 + a_6\alpha^6 + a_5\alpha^5 + a_4\alpha^4 + a_3\alpha^3 + a_2\alpha^2 + a_1\alpha + a_0,$$

  on $\alpha = [x] \in \mathbb{F}_{2^8}$ (observeu que a classe normalment escrivim el polinomi al revés, sent el primer coeficient el de grau 0).

ByteSub (II)

- L'operació Bytesub consisteix en dues transformacions. Primer calculem un valor auxiliar $c \in \mathbb{F}_{2^8}$, i finalment calculem $b \in \mathbb{F}_{2^8}$ a partir de c.

  (a) Si $a = 0$, aleshores $c = 0 \in \mathbb{F}_{2^8}$. Si $a \neq 0$, aleshores $c = a^{-1}$. Altre cop, escriurem c en notació vectorial:

  $$c = c_7\alpha^7 + c_6\alpha^6 + c_5\alpha^5 + c_4\alpha^4 + c_3\alpha^3 + c_2\alpha^2 + c_1\alpha + c_0.$$

  (b) Després apliquem la següent transformació afí definida a $\mathbb{Z}_2$:

$$b = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

i

$$b = b_7b_6b_5b_4b_3b_2b_1b_0,$$

Exercici 1: Calculeu $Bytesub(a)$ per $a = 00000001$.

Exercici 2: Sigui $a = 00000010$.

(a) Comproveu que $\alpha^{-1} = (\alpha^7 + \alpha^3 + \alpha^2 + 1)$ a $\mathbb{F}_{2^8} = \mathbb{Z}_2[x]/(m(x))$.

(b) Calculeu $Bytesub(a)$.

Exercici 3: Calculeu $Bytesub(a)$ per $a = 00000100$.

1.9 Algoritme de Shor

El 1994, Peter Shor va presentar algoritmes eficients que empren operacions quàntiques per factoritzar nombres enters i per calcular el logaritme discret. Aquest resultat és trascendental per la criptografia perquè, si s'arribessin a construir ordinadors quàntics suficientment potents, gran part de les comunicacions actuals esdevindrien insegures. En particular, els esquemes RSA i Diffie-Hellman presentats anteriorment. A continuació presentem un algoritme simplificat per factoritzar $N = pq$, on p i q són primers. Aquest algoritme es pot executar en un ordinador convencional (però llavors no és eficient).

1. Preneu un enter aleatori $1 \leqslant a \leqslant N$. Si no és coprimer amb N, ja podeu descompondre N i ja podeu parar. Si és coprimer amb N, continueu.
2. Trobeu l'ordre d'a: Calculeu potències d'a mòdul N fins que trobeu r que satisfà
$$a^r = 1 \mod N$$
3. Sabem que $a^r - 1 = 0 \mod N$, i per tant $a^r - 1 = kN$ per algun k. Si r és imparell, torneu al primer pas.
4. Si r és parell, llavors
$$a^r - 1 = (a^{\frac{r}{2}} + 1)(a^{\frac{r}{2}} - 1) = kN$$
Per tant, els factors de N divideixen $(a^{\frac{r}{2}} + 1)$ o $(a^{\frac{r}{2}} - 1)$.
5. Calculeu $x = gcd(a^{\frac{r}{2}} + 1, N)$
6. Si $x = 1$ o $x = N$, torneu al primer pas. Si $x \neq 1, N$, aleshores x és un divisor de N.
7. Dividiu N per x per obtenir la factorització completa.

El principal avantatge de l'algoritme de Shor és que el segon pas es pot fer de manera eficient amb una operació quàntica que està relacionada amb la transformada de Fourier. En canvi, amb un ordinador convencional és molt costós.

Exercici 1: Factoritzeu $N = 15$ seguint aquest algoritme

Exercici 2: Feu una estimació de la probabilitat de trobar un $1 \leqslant a \leqslant N$ que no és coprimer amb N (pas 1). Expliqueu per què, en aquest cas, ja podem descompondre N.

Exercici 3: Expliqueu per què no es pot donar el cas que els factors de N divideixin $(a^{\frac{r}{2}} - 1)$ (pas 4). Per aquest motiu, en el pas 5 assumim que els factors de N divideixen $(a^{\frac{r}{2}} + 1)$.

Exercici 4: Digueu quins són els possibles ordres dels elements coprimers amb N.

Capítol 2
Codis lineals

2.1 Propietats

Codis lineals

- Sigui $\mathbb{F}$ un cos finit. Un codi lineal C de longitud n sobre l'alfabet $\mathbb{F}$ és un subespai vectorial de $\mathbb{F}^n$.
- Definim la dimensió d'un codi com la dimensió que té com a subespai vectorial. Si un codi té dimensió k i longitud n, aleshores diem que la taxa de transmissió és k/n i la codimensió és $n-k$.

2.1.1 Matriu generadora

Matriu generadora d'un codi lineal

- Diem que una matriu G és la matriu generadora d'un codi C si les seves files són una base de C. La matriu generadora pot ser que no sigui única, perquè un subespai vectorial pot admetre més d'una base.
- Si C és un codi de longitud n i dimensió k sobre $\mathbb{F}$, aleshores la matriu generadora G és una matriu de k files i n columnes d'elements de $\mathbb{F}$.
- Per codificar una paraula $m \in \mathbb{F}^k$, multipliquem m per G.
- Diem que una matriu generadora és sistemàtica en les primeres posicions si conté la identitat en les primeres posicions. En aquest cas la descrivim com
$$G = (I|P),$$
on I és la matriu identitat, i P és una altra matriu.
- Diem que una matriu generadora és sistemàtica en les darreres posicions si conté la identitat en les darreres posicions. En aquest cas la descrivim com
$$G = (P|I),$$
on, com abans, I és la matriu identitat, i P és una altra matriu.

Distància mínima d'un codi

- La distància de Hamming entre dues paraules de la mateixa longitud és el número de posicions on difereixen. El pes d'una paraula és el número de posicions no nul·les o, equivalentment, la seva distància de Hamming al vector nul.
- La distància mínima d'un codi és la mínima distància de Hamming entre dues paraules. Coincideix amb el mínim pes de les paraules no nul·les del codi.

Exercici 1: Considereu el codi binari C definit per la següent matriu generadora:

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

(a) Escriviu totes les paraules del codi C.

(b) Digueu quina és la dimensió, la longitud, la taxa de transmissió i la codimensió del codi.

(c) Digueu si la matriu generadora és sistemàtica.

(d) Trobeu la distància mínima del codi de dues maneres diferents:

  (a) calculant la mínima distància de Hamming entre dues paraules del codi, i

  (b) calculant el mínim pes de les paraules no nul·les del codi.

Exercici 2: Considereu els codis C definits per les següents matrius generadores. Responeu els apartats de l'exercici anterior per cada un d'aquest codis.

(a) El codi binari definit per

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

(b) El codi ternari definit per

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

(c) El codi ternari definit per

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{pmatrix}.$$

(d) El codi ternari definit per

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

(e) El codi binari definit per

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

2.2 Codi dual

Definició

- El codi dual (ortogonal) d'un codi $C \subseteq \mathbb{F}^n$ és

$$C^\perp = \{v \in \mathbb{F}^n \,:\, v \cdot c = 0 \text{ per tot } c \in C\}$$

- Com que C és un subespai vectorial, si $c_1, \ldots, c_k$ és una base de C,

$$C^\perp = \{v \in \mathbb{F}^n \,:\, v \cdot c_1 = 0, \ldots, v \cdot c_k = 0\}$$

  En particular, les files d'una matriu generadora de C són una base de C. Per tant, podem calcular $C^\perp$ resolent el sistema d'equacions lineals amb variables $x_1, \ldots, x_n$ definides per

$$\begin{aligned} x_1 c_{1,1} + \ldots + x_n c_{1,n} &= 0 \\ \vdots \qquad\qquad & \vdots \\ x_1 c_{k,1} + \ldots + x_n c_{k,n} &= 0 \end{aligned}$$

  on $c_i = (c_{i,1}, \ldots, c_{i,n})$ és la fila i-èssima d'una matriu generadora de C. Aquest és un sistema d'equacions homogeni de rang k en n variables. Per $k < n$ és un sistema indeterminat amb $n - k$ graus de llibertat.

- Per tant, el codi dual, que és el subespai vectorial de les solucions d'aquest sistema, té longitud n i dimensió $n - k$.

- Ara considerem codi dual de $C^\perp$, que per definició del codi dual és

$$(C^\perp)^\perp = \{v \in \mathbb{F}^n \,:\, v \cdot c = 0 \text{ per tot } c \in C^\perp\}$$

  Anàlogament al desenvolupament anterior, podem veure que les paraules de $(C^\perp)^\perp$ són aquells vectors $(x_1, \ldots, x_n)$ que satisfan

$$\begin{aligned} x_1 c_{1,1} + \ldots + x_n c_{1,n} &= 0 \\ x_1 c_{n-k,1} + \ldots + x_n c_{n-k,n} &= 0 \end{aligned}$$

  on $c_1, \ldots, c_{n-k}$ és una base de $C^\perp$. El rang del sistema és $n-k$, i per tant les solucions formaran un subespai vectorial de dimensió $n - (n-k) = k$. És evident que les paraules del codi en són una solució. Com que el codi té dimensió k, obtenim que

$$(C^\perp)^\perp = C$$

  per tot codi lineal.

- El darrer sistema d'equacions ens permet descriure C com els vectors solució d'un sistema determinat per una base de $C^\perp$.

Exercici 1: Considereu el codi lineal binari C definit per la matriu generadora.

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

(a) Trobeu els vectors del codi dual.

(b) Trobeu una matriu generadora del codi dual.

Matriu de control

- Les matrius generadores de $C^\perp$ s'anomenen matrius de control de C.
- Si H és una matriu de control de C, aleshores

  $$C = \{c \in \mathbb{F}^n \,:\, c \cdot h = 0 \text{ per tota fila h de } H\}$$

  Això permet descriure C a partir d'un sistema d'equacions lineals determinades per H.
- La distància mínima d'un codi coincideix amb el mínim del cardinal dels conjunts de columnes linealment dependents d'una matriu de control del codi.

Matriu de control sistemàtica

- Si una matriu generadora és de la forma

  $$G = (I|P),$$

  aleshores una matriu de control és

  $$H = (-P^T|I).$$

  I si una matriu generadora és de la forma $G = (P|I)$, aleshores una matriu de control és $H = (I| - P^T)$. Anàlogament, com que $(C^\perp)^\perp = C$, si una matriu de control és $H = (I|P)$, aleshores una matriu generadora és $G = (-P^T|I)$ i, si una matriu de control és $H = (P|I)$, aleshores una matriu generadora és $G = (I| - P^T)$.

Exercici 2:

(a) Calculeu una matriu de control H de C fent servir el fet que G és sistemàtica.

(b) Comproveu que $H \cdot G^T$ és una matriu on tots els elements són 0.

(c) El codi C es pot descriure com la solució d'un sistema d'equacions determinat per H. Trobeu un sistema d'equacions amb aquesta propietat.

(d) Calculeu la distància mínima de C a partir d'una matriu de control.

Exercici 3: Considereu els codis C definits per les següents matrius generadores. Responeu els exercicis anteriors prenent cada un dels codis següents:

(a) El codi binari definit per

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

(b) El codi ternari definit per

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

(c) El codi ternari definit per

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{pmatrix}.$$

(d) El codi ternari definit per

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

(e) El codi binari definit per

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

2.3 Codis de Hamming

Codis de Hamming

- El codi de Hamming de paràmetre t sobre $\mathbb{F}_q$ és un codi sobre $\mathbb{F}_q$ de màxima taxa de transmissió d'entre els que tenen codimensió t i capacitat correctora 1.
- Els seus paràmetres són $n = \frac{q^t-1}{q-1}$, $k = n - t$ i $d = 3$.
- Per obtenir-los, és comú determinar-lo a partir d'una matriu de control especial.
- Aquesta matriu té t files i té el màxim número possible de columnes linealment independents dues a dues. En el cas binari, això es correspon amb una matriu de t files en la que les columnes són tots els vectors no nuls de $\mathbb{F}_2^t$.
- Els codis de Hamming tenen una propietat especial: són perfectes. Això significa que qualsevol paraula de $\mathbb{F}_q^n$ està a distància 1 d'una paraula del codi.

Exercici 1:

(a) Trobeu una matriu de control pel codi de Hamming de paràmetre $t = 3$ sobre $\mathbb{F}_2$.

(b) Calculeu la dimensió d'aquest codi i la seva distància mínima a partir dels paràmetres de la matriu de control.

(c) Trobeu una matriu generadora d'aquest codi.

(d) Calculeu la seva taxa de transmissió.

Exercici 2:

(a) Escriviu les tres primeres lletres del vostre nom com una seqüència de bits, d'acord amb al taula següent.

A	00000	B	00001	C	00010	D	00011	E	00100
F	00101	G	00110	H	00111	I	01000	J	01001
K	01010	L	01011	M	01100	N	01101	O	01110
P	01111	Q	10000	R	10001	S	10010	T	10011
U	10100	V	10101	W	10110	X	10111	Y	11000
Z	11001								

(b) Codifiqueu els primers 8 bits de la teva seqüència d'acord amb el codi que heu creat.

2.4 Construcció de codis lineals

Exercici 1: Construïu un codi lineal C de longitud 4 sobre $\mathbb{F}_2$ que contingui les següents paraules:

$$(1,0,0,1),(1,1,0,0),(0,1,0,1)$$

(a) Escriviu totes les paraules de C.

(b) Quina dimensió té el codi que heu creat?

(c) Hi ha altres codis que continguin aquestes paraules, a part del codi que heu creat? Expliqueu la resposta.

Exercici 2: Ara construïu un codi lineal C' de longitud 4 sobre $\mathbb{F}_3$ que contingui les següents paraules

$$(1,0,0,1),(1,1,0,0),(0,1,0,1).$$

(a) Escriviu totes les paraules de C'.

(b) Quina dimensió té el codi que heu creat?

(c) Hi ha altres codis que continguin aquestes paraules, a part del codi que heu creat? Expliqueu la resposta.

Exercici 3:

(a) Doneu dues matrius generadores G i G' del codi lineal C sobre $\mathbb{F}_2$ de longitud 4 i dimensió 2 que conté les següents paraules:

$$(1,0,0,1),(1,1,0,0),(0,1,0,1).$$

(b) Codifiqueu el missatge $(0,1)$ amb la matriu G i G'. El resultat és el mateix? Per què?

(c) Descodifiqueu la paraula $(1,1,0,0)$ d'acord amb les dues matrius.

Exercici 4: Siguin C i C' els codis que has definit a l'Exercici 1 i a l'Exercici 2, respectivament.

(a) Doneu una matriu de control del codi C i una matriu de control del codi C'.

(b) El codi C es pot descriure com la solució d'un sistema d'equacions. Trobeu un sistema d'equacions amb aquesta propietat. Feu el mateix pel codi C'.

2.5 Detecció i correcció d'errors

Exercici 1: Sigui C el codi sobre $\mathbb{F}_5$ amb matriu generadora

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 & 1 \end{pmatrix}.$$

Una matriu de control de C és

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 4 \\ 0 & 1 & 0 & 2 & 2 \\ 0 & 0 & 1 & 1 & 3 \end{pmatrix}.$$

(a) calculeu la dimensió i la distància mínima de C. Quina capacitat té C per detectar i corregir errors?.

(b) De les següents paraules, digueu quines són del codi. Corregeix els esborralls i les paraules que no siguin del codi, explicant com ho has fet.

$$(2,2,0,4,2), (0,4,2,3,0), (0,*,2,3,0), (1,4,2,3,1)$$

Exercici 2: Trobeu una matriu generadora d'un codi de Hamming de paràmetre 3 sobre $\mathbb{F}_2$. Considereu les seqüències de bits 0101110, 0101111 i 0010111. Per cada una d'aquestes sequències, comproveu si és una paraula del codi. Si ho és, digueu quin ha estat el missatge enviat. Si no ho és, corregiu-la.

2.6 Implementació hardware

- Els codis lineals es poden implementar fàcilment en maquinari digital. Un codificador lineal de taxa k/n pren k bits d'informació i genera n bits de sortida mitjançant una matriu generadora G. Cada bit de sortida és una combinació lineal (en $\mathbb{F}_2$) dels bits d'entrada. En maquinari, aquestes combinacions es realitzen amb portes XOR.
- En un circuit digital, la codificació $\mathbf{c} = \mathbf{u}G$ pot implementar-se de dues maneres principals:
  - **Codificador combinacional:** tots els bits d'informació $\mathbf{u}$ estan disponibles simultàniament, i les sortides es calculen mitjançant un conjunt de portes XOR connectades segons les columnes de G. La lectura del resultat també es fa simultàniament.
  - **Codificador seqüencial:** els bits d'informació arriben un a un, i s'emmagatzemen parcialment en registres (retards) per generar les combinacions adequades. Aquesta arquitectura és necessària quan G implica dependències temporals o quan la transmissió és contínua. Els retards serveixen per:
    - memoritzar bits d'entrada anteriors mentre s'introdueixen de nous;
    - permetre càlculs síncrons en seqüències llargues;
    - reduir la latència total mitjançant paral·lelisme o pipelining.

Exercici 1 Considerem un codi binari lineal amb $k = 4$ bits d'informació i $n = 7$ bits de sortida, definit per la matriu generadora

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Suposeu que els bits d'informació són $\mathbf{u} = (u_1, u_2, u_3, u_4)$.

Exercici 2

(a) Dibuixeu o descriviu textualment un circuit digital que implementi aquestes operacions utilitzant únicament portes XOR. Indiqueu com es connecten els senyals $u_1, \ldots, u_4$ per generar cada sortida c_j.

(b) Si cada porta XOR té un retard τ, calculeu la latència total del codificador combinacional.

Exercici 3

(a) Trobeu una matriu de control del codi generat per G.

(b) Determineu les expressions de la síndrome $u_1, \ldots, u_4$ també es pot descriure en funció dels bits rebuts $c_1, \ldots, c_7$.

(c) Dibuixeu o descriviu textualment un circuit digital que implementi aquestes operacions utilitzant únicament portes XOR.

Exercici 4 Suposeu ara que els bits u_i arriben de manera seqüencial (un per cicle de rellotge). Expliqueu com s'haurien d'introduir *retards* (memòries o registres) per garantir que les combinacions XOR puguin calcular-se correctament. Quin és el criteri general per decidir on col·locar aquests retards?

Capítol 3
Codis cíclics

3.1 Propietats

3.1.1 Definició

Codis cíclics

Un codi lineal $\mathcal{C}$ és cíclic si tots els desplaçaments cíclics de les paraules de $\mathcal{C}$ pertanyen a $\mathcal{C}$.
Equivalentment, podem dir que un codi lineal $\mathcal{C}$ és cíclic si per tot $(a_1, a_2, \ldots, a_n) \in \mathcal{C}$ es compleix que $(a_n, a_1, a_2, \ldots, a_{n-1})$ pertany a $\mathcal{C}$

Exercici 1: Determina quins dels codis següents són codis lineals, i quins són codis lineals cíclics. Explica el per què.

(a) Codi lineal de longitud 4 sobre $\mathbb{F}_2$ amb matriu generadora $\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$.

(b) Codi lineal de longitud 4 sobre $\mathbb{F}_2$ amb matriu generadora $\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$.

(c) Codi de repetició de longitud k a $\mathbb{F}$.
Aquest codi es pot definir com el conjunt de vectors

$$\{(a, a, \ldots, a) \in \mathbb{F}^k \ : \ a \in \mathbb{F}\}.$$

(d) Codi de paritat parell sobre $\mathbb{F}_2$ de longitud 4. Aquest codi es pot definir com el conjunt de vectors

$$\{(a_1, a_2, a_3, a_4) \in \mathbb{F}_2^4, : \ a_1 + a_2 + a_3 + a_4 = 0\}.$$

Representació polinomial de les paraules d'un codi

Sigui $\mathbb{F}$ un cos finit i $n > 0$. En el context de codis lineals cíclics, ens convé identificar vectors amb polinomis. Per tot vector $v = (v_0, v_1, \ldots, v_{n-1}) \in \mathbb{F}^n$, identifiquem v amb un polinomi

$$v(x) = v_0 + v_1 x + \ldots + v_{n-1} x^{n-1}.$$

Aquest és un polinomi de grau $n-1$ amb coeficients a $\mathbb{F}$.

Aleshores, tot codi lineal $C \subseteq \mathbb{F}^n$ determina un conjunt de polinomis de grau menor que n amb coeficients a $\mathbb{F}$.

3.1.2 Polinomi generador

Polinomi generador

Sigui $C \subseteq \mathbb{F}_q^n$ un codi lineal i cíclic. Aleshores existeix un polinomi mònic $g(x)$ amb coeficients a $\mathbb{F}_q$ que satisfà la següent propietat:
Per tot $v = (v_0, \ldots, v_{n-1}) \in \mathbb{F}_q^n$,

$$v \in C \iff v(x) \text{ és divisible per } g(x)$$

Aquest polinomi és únic, i s'anomena polinomi generador de C.

Propietats del polinomi generador:

(a) El grau del polinomi generador és $n-k$, on n és la longitud del codi C i k la seva dimensió.

(b) El polinomi generador $g(x)$ d'un codi cíclic de longitud n verifica

$$g(x)|x^n-1.$$

(c) Per altra banda, si un polinomi $p(x) \in \mathbb{F}[x]$ és mònic i divideix x^n-1, aleshores $p(x)$ és el polinomi generador d'un codi cíclic de longitud n sobre $\mathbb{F}$.

(d) Tota paraula del codi es pot descriure a partir de $g(x)$:

A partir de la definició, obtenim que totes les paraules del codi es poden escriure com $m(x)\cdot g(x)$, on $m(x)$ és un polinomi de grau $k-1$. Per tant, C admet una matriu generadora del tipus

$$G=\begin{pmatrix} g_0 & g_1 & \dots & g_{n-k} & 0 & 0 & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_{n-k} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & g_0 & g_1 & \dots & g_{n-k} & 0 \\ 0 & 0 & \dots & 0 & g_0 & g_1 & \dots & g_{n-k} \end{pmatrix}$$

Observeu que el polinomi associat al vector mG, on $m=(m_0,\dots,m_{k-1})$, és el polinomi $m(x)\cdot g(x)$

(e) En particular, el polinomi generador d'un codi cíclic $\mathcal{C}$ es correspon amb una paraula del codi, $(g_0,\dots,g_{n-k},0,\dots,0)$.

(f) D'entre tots els polinomis associats a paraules del codi, $g(x) \in \mathcal{C}$ és el polinomi mònic que té grau mínim.

Exercici 2: L'objectiu d'aquest exercici és trobar codis lineals cíclics de longitud 3 sobre $\mathbb{F}_2$

(a) Trobeu tots els polinomis de $\mathbb{F}_2[x]$ de grau $\leqslant 2$ que dividixen x^3-1.

(b) Per les propietats anteriors, aquests polinomis són polinomis generadors de codis cíclics. Trobeu una matriu generadora per cada un d'aquests codis.

Exercici 3: Sigui $\mathcal{C}$ el codi cíclic de longitud 7 sobre $\mathbb{F}_2$ amb polinomi generador $g(x)=1+x+x^2+x^4$.

(a) Trobeu una matriu generadora de $\mathcal{C}$.

(b) Quina és la dimensió de $\mathcal{C}$?

(c) Trieu un missatge compatible amb aquest codi i codifiqueu-lo amb la matriu que has trobat al primer apartat.

Exercici 4:

(a) Doneu la taula exponencial-vectorial de $\mathbb{Z}_2[x]/x^3+x+1$, utilitzant $\alpha = [x]$.

(b) Comproveu que el polinomi $g(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^3)$ és el polinomi generador d'un codi cíclic de longitud 7. Digueu quina és la seva dimensió.

(c) Doneu una matriu generadora del codi generat per $g(x)$.

(d) Preneu els 12 primers bits de la seqüència de bits generada a partir del vostre nom, d'acord amb la taula següent.

A	00000	B	00001	C	00010	D	00011	E	00100
F	00101	G	00110	H	00111	I	01000	J	01001
K	01010	L	01011	M	01100	N	01101	O	01110
P	01111	Q	10000	R	10001	S	10010	T	10011
U	10100	V	10101	W	10110	X	10111	Y	11000
Z	11001								

Escriviu el missatge corresponent a $\mathbb{F}_8$ i codifiqueu-lo de dues maneres diferents

(a) Amb la matriu generadora de l'apartat anterior

(b) Amb codificació sistemàtica.

3.1.3 Polinomi de control

- El polinomi de control d'un codi cíclic $\mathcal{C}$ de longitud n amb polinomi generador $g(x)$ és
$$h(x) = \frac{x^n - 1}{g(x)}.$$
- Una paraula $w(x)$ és del codi si $w(x)h(x) = 0 \,(\mod x^n - 1)$.

Exercici 5

(a) Trobeu el polinomi de control del codi C de l'exercici 3.

(b) Doneu una matriu de control de C.

3.1.4 Distància mínima prevista

- La fita de Singleton estableix que en un codi lineal de longitud n i distància mínima d, la dimensió k satisfà
$$k \leqslant n - d + 1.$$
- La distància mínima prevista d'un codi cíclic C és el màxim enter δ tal que hi ha $\delta - 1$ arrels de g que són potències consecutives diferents d'un element primitiu α
$$\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+\delta-2}.$$
- Es compleix que, si d és la distància mínima real del codi, aleshores $d \geqslant \delta$

Exercici 6 Trobeu la distància mínima prevista dels codis dels exercicis 3 i 4.

3.2 Identificació de codis cíclics

Exercici 1 A continuació donem diverses matrius generadores de codis lineals de longitud 4. Digueu quins dels codis corresponents són cíclics, i explica el **per què**.

(a) Codi sobre $\mathbb{F}_2$ amb $G = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$.

(b) Codi sobre $\mathbb{F}_3$ amb $G = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$.

(c) Codi sobre $\mathbb{F}_2$ amb $G = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$.

(d) Codi sobre $\mathbb{F}_2$ amb $G = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

(e) Codi sobre $\mathbb{F}_2$ amb $G = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$.

(f) Codi sobre $\mathbb{F}_3$ amb $G = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$.

(g) Codi sobre $\mathbb{F}_3$ amb $G = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 2 \end{pmatrix}$.

Exercici 2 Digueu quins dels següents codis són codis lineals cíclics.

(a) Codi de paritat senar sobre $\mathbb{F}_2$ de longitud 4.
Aquest codi es pot es pot definir com el conjunt de vectors

$$\{(a_1, a_2, a_3, a_4) \in \mathbb{F}_2^4, : \ a_1 + a_2 + a_3 + a_4 = 1\}.$$

(b) Codi de paritat parell sobre $\mathbb{F}_2$ de longitud n. Aquest codi es pot definir com el conjunt de vectors

$$\{(a_1, \ldots, a_n) \in \mathbb{F}_2^n, : \ \sum_{i=1}^{n} a_i = 0\}.$$

(c) Codi de sobre $\mathbb{F}_q$ de longitud n definit com el conjunt de vectors

$$\{(a_1, \ldots, a_n) \in \mathbb{F}_q^n, : \ \sum_{i=1}^{n} a_i = 0\}.$$

Exercici 3 Pels codis lineals cíclics de l'apartat anterior, doneu el polinomi generador i una matriu generadora.

Exercici 4 En aquest exercici estudiem la relació entre codis cíclics MDS sobre $\mathbb{F}_2$.

(a) Sigui $n = 3$

  (a) Doneu dos codis $\mathbb{F}_2$ que siguin MDS. Un de distància mínima 2, i un altre de distància mínima n.

  (b) Comproveu que un és el dual de l'altre.

  (c) Comproveu que aquests codis són cíclics, i dona el polinomi generador de cada un d'ells.

(b) Sigui ara n un valor genèric

  (a) Doneu dos codis $\mathbb{F}_2$ que siguin MDS. Un de distància mínima 2, i un altre de distància mínima n.

  (b) Comproveu que aquests codis són únics, i que un és el dual de l'altre.

  (c) Comproveu que aquests codis són cíclics, i doneu el polinomi generador de cada un d'ells.

3.3 Codis cíclics primitius

Codis cíclics primitius:

Es diu que un codi lineal cíclic $C \subseteq \mathbb{F}_q^n$ és primitiu si $n = q - 1$.

Recordatori:

Sigui $\mathbb{F}_q$ un cos finit.

- L'ordre d'un element $\gamma \in \mathbb{F}_q^*$ és el menor enter positiu r tal que $\gamma^r = 1$.
- Per qualsevol element $\gamma \in \mathbb{F}_q^*$ es compleix $\gamma^{q-1} = 1$.
- Per tant qualsevol element $\gamma \in \mathbb{F}_q^*$ és una arrel del polinomi $x^{q-1} - 1$.
- Com que $(x - \gamma)$ divideix $x^{q-1} - 1$ per tot $\gamma \in \mathbb{F}_q^*$, i el cardinal de $\mathbb{F}_q^*$ és $q - 1$, deduïm que
$$x^{q-1} - 1 = \prod_{\gamma \in \mathbb{F}_q^*} (x - \gamma)$$

Propietats dels codis cíclics primitius:

(a) Si $g(x)$ és un polinomi generador d'un codi cíclic, aleshores $g(x)$ divideix $x^n - 1$. Si, a més, el codi és cíclic i primitiu, aleshores $g(x)$ divideix
$$x^n - 1 = x^{q-1} - 1 = \prod_{\gamma \in \mathbb{F}_q^*} (x - \gamma)$$
Com que tots els factors de $x^n - 1$ són del tipus $(x - \gamma)$ i tenen multiplicitat 1, aleshores $g(x)$ es pot factoritzar com
$$g(x) = (x - \beta_1)(x - \beta_2) \cdot \ldots \cdot (x - \beta_{n-k}),$$
on $\beta_1, \beta_2, \ldots, \beta_{n-k}$ són elements diferents de $\mathbb{F}_q^*$.

(b) Recíprocament, per tot $\beta_1, \beta_2, \ldots, \beta_{n-k} \in \mathbb{F}_q^*$, diferents dos a dos, el polinomi
$$g(x) = (x - \beta_1)(x - \beta_2) \cdot \ldots \cdot (x - \beta_{n-k}),$$
és el polinomi generador d'un codi cíclic primitiu de longitud $n = q - 1$.

Exercici 1 Volem construir un codi lineal cíclic primitiu de dimensió 2 i llargada 4.

(a) Sobre quin cos estan definits, aquests codis?

(b) Doneu el polinomi generador d'un codi d'aquest tipus.

(c) Quants codis lineals cíclics primitius de dimensió 2 i llargada 4 hi ha?

Exercici 2 Volem construir un codi lineal cíclic primitiu de dimensió 2 i llargada 6.

(a) Sobre quin cos estan definits, aquests codis?

(b) Doneu el polinomi generador d'un codi d'aquest tipus.

Exercici 3 En aquest exercici reprenem el codi d'una pràctica anterior. Sigui C el codi cíclic primitiu contingut a $\mathbb{F}_8^7$ generat per $g(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^3)$.

(a) Trobeu el polinomi de control.

(b) Trobeu la distància mínima del codi, a partir de la distància mínima prevista del codi i de la fita de Singleton.

(c) Compareu la capacitat de detecció d'errors i la taxa de transmissió d'aquest codi i la del codi de Hamming binari de paràmetre $t = 3$.

3.4 Codis CRC

Codis CRC

- L'ús de codis binaris cíclics amb algoritmes de codificació i detecció d'errors eficients és fonamental en les comunicacions digitals i en la protecció de dades.
- A l'apartat 3.1.4 n'hem vist dos casos extrems, amb redundància 1 i $n-1$, respectivament. El primer és molt comú perquè permet detectar un error.
- Aquests codis tenen per matrius generadores les matrius G_n i G'_n, definides com:
$$G_n = (11\ldots1) \in \mathbb{Z}_2^{1\times n}$$
$$G'_n = (I|P) \in \mathbb{Z}_2^{(n-1)\times n} \text{ on } P = (G'_{n-1})^T$$
- A la pràctica, els codis binaris cíclics més utilitzats són codis amb redundància $1 \leqslant n-k \leqslant 64$ i $n-k \ll n$ (que vol dir que n és molt més gran que $n-k$). Es fan servir, sobretot, per detectar errors.

Codis CRC

- Alguns d'aquests codis estan inclosos a la família de Cyclic Redundancy Checking (CRC) codes.
- Els codis CRC són codis binaris orientats a la detecció eficient d'errors, i es codifiquen de manera sistemàtica fent servir l'algoritme de la divisió que vam veure a classe. No tots els codis CRC són cíclics, però tots ells tenen un polinomi generador. La paraula associada a un missatge $i(x)$ és
$$i(x)x^{n-k} - R(x),$$
on $R(X)$ és el residu de dividir $i(x)x^{n-k}$ per un polinomi $g(x)$.
- En general, els codis CRC s'etiqueten com CRC-X-Y, on X és el nombre de bits de redundància ($X = n-k$) i Y és un identificador (opcional). El codi de distància mínima 2 de l'exercici anterior s'anomena CRC-1.
- L'ús de codis CRC permet la detecció d'errors de memòria que ens avisen de la degradació del suport en el que les dades estan emmagatzemades.
- Els codis CRC estan inclosos en la majoria de protocols de comunicació digitals, ja sigui de comunicació entre dispositius (p.e. ordinador-USB) o de la xarxa (wifi, ethernet, xarxes mòbils). És especialment útil quan l'ús que es fa de l'aplicació no sigui streaming exigent, i es pugui demanar el reenviament de paquets de dades en els que s'han detectat errors.

Codificació i descodificació CRC

- El caràcter recursiu de la divisió fa que aquesta operació admeti optimitzacions hardware. En particular, es pot implementar amb uns circuits molt senzills, com els de les figures 3.1 i 3.2.
- Els bits del missatge s'ordenen com el polinomi associat, de grau més alt a grau més baix. Si $n \gg n-k$ (n és significativament més gran que $n-k$), aquesta codificació és especialment eficient, perquè la divisió es pot començar encara que no s'hagin rebut tots els n bits del dividend. El residu final de la divisió és la redundància que s'afegeix al missatge. En el cas de codificar amb matriu generadora sistemàtica, necessitem conèixer tots els bits del missatge per poder generar la redundància corresponent. A més, si la longitud del missatge és gran, emmagatzemar la matriu de control pot generar requisits de memòria no desitjats.
- En rebre una paraula, la detecció d'errors es fa de manera anàloga. Es divideix el polinomi associat a la paraula pel polinomi generador, i si el residu és 0 aleshores es conclou que no hi ha hagut errors. Des del punt de vista de hardware, aquesta comprovació té l'avantatge mencionat a la codificació: es pot començar l'algoritme de detecció d'errors sense haver acabat de rebre tota la paraula. A més, el circuit és el mateix.
- El fet que la codificació sigui sistemàtica facilita la lectura del missatge enviat. En el moment que es rep la paraula, si es considera que no hi ha errors, n'hi ha prou amb recuperar els primers k bits.

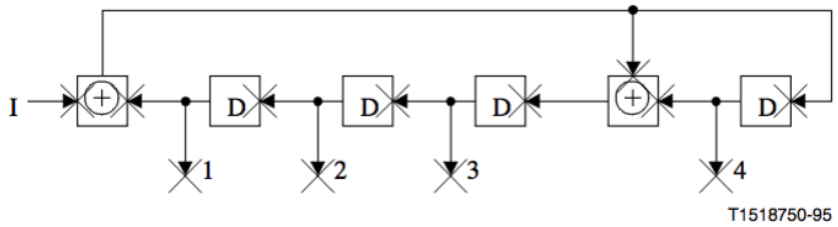


Figura 3.1: Circuit associat a CRC-4 de l'estàndard UIT-T G.704 per sistemes de transmissió digital. Aquest circuit es correspon al codi amb polinomi generador $g(x) = x^4 + x + 1$. A la figura, I és l'input, D són registres de desplaçament, i els operadors són XORs (sumes a $\mathbb{F}_2$). El circuit s'inicia amb tots els registres a zero, i es va iterant a mesura que avança la divisió. A l'últim pas de la divisió, els registres 1, 2, 3, 4 seran els 4 bits de redundància, que corresponen als coeficients de grau 3, 2, 1 i 0 del polinomi, respectivament. S'executa a 2048 kbits/s.

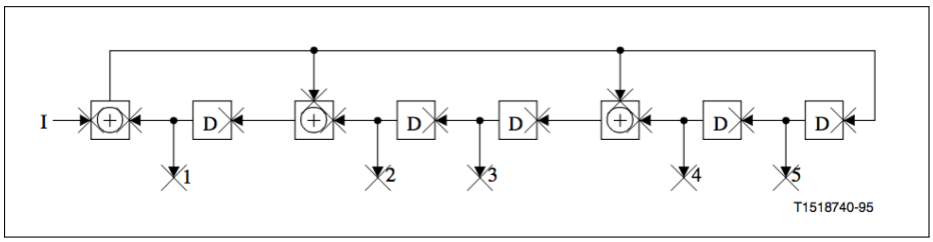


Figura 3.2: Circuït associat a CRC-5 de l'estàndard UIT-T G.704 per sistemes de transmissió digital. Aquest circuit es correspon al codi amb polinomi generador $g(x) = x^5 + x^4 + x^2 + 1$. S'executa a 6312 kbits/s.

Exercici 1 De codis CRC n'hi ha centenars (`https://users.ece.cmu.edu/~koopman/crc/`). La gran majoria de codis no són MDS, i la distància mínima queda lluny de $n - k + 1$, però estan triats per tal de poder detectar molts errors (més que d-1). En general, es busquen polinomis amb pocs coeficients no nuls, per reduir els costos computacionals. Però a la vegada, això pot ser contraproduent. Si el codi és lineal i cíclic, quina relació hi ha entre el nombre de coeficients de g diferents de zero i la distància mínima del codi?

Exercici 2 A l'exercici anterior mencionem un fenomen que es dona sovint als codis CRC: que es poden detectar errors de més pes que la capacitat detectora d'errors. Per il·lustrar aquest fenomen, considera el codi binari amb la següent matriu de control.

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

(a) Quina és la distància mínima? Quina és la capacitat detectora d'errors?

(b) Digueu quins dels següents vectors d'error es poden detectar:

$$1000, 1100, 1110, 0011, 0001$$

Exercici 3 Comproveu que els circuits amb delays descrits a les figures calculen la codificació d'acord amb els polinomis corresponents. Justifiqueu l'avantatge d'aquest disseny.

3.5 Esquemes de compartició de secrets

Motivació
Els esquemes de compartició de secrets van ser introduïts per Shamir i Blakley el 1979. L'exemple que es mostra com a motivació a l'article de Shamir és el següent:

Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present.

I a l'article de Blakley és el següent:

Certain cryptographic keys, such as a number which makes it possible to compute the secret decoding exponent in an RSA public key cryptosystem, or the system master key and certain other keys in a DES cryptosystem, are so important that they present a dilemma. If too many copies are distributed one might go astray. If too few copies are made they might all be destroyed.

A la pràctica següent veurem esquemes que poden resoldre aquests dos problemes plantejats.

Definició
Un esquema de compartició de secrets és un mètode per generar fragments (shares) d'un secret:

- Input: s (el secret) i valors aleatoris
- Output: $s_1, \ldots, s_n$ (fragments).

De manera que alguns conjunts puguin recuperar el secret i d'altres no puguin obtenir cap informació de s. Així, per cada conjunt $A \subseteq P = \{1, \ldots, n\}$,

- Si s es pot recuperar a partir de $\{s_i : i \in A\}$, diem que A és autoritzat.
- Si $\{s_i : i \in A\}$ no aporta cap informació sobre s, diem que A és prohibit.

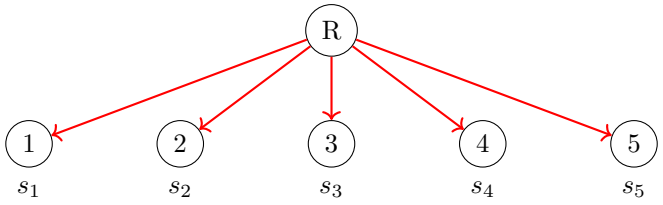
Els esquemes de compartició de secrets són una peça fonamental en la construcció d'esquemes i aplicacions criptogràfiques com son els protocols de computació multipart segura, control d'accés, protocols de votació electrònica, o esquemes de firma digital distribuïda.

Un esquema senzill

- Sigui $G = \mathbb{Z}_n$ el conjunt dels possibles secrets.
- El secret $s \in G$ es comparteix amb els participants $P = \{1, 2, 3, 4, 5\}$
- El repartidor tria $s_1, \ldots, s_4 \in G$ uniformement a l'atzar, i calcula

$$s_5 = s - s_1 - \ldots - s_4.$$

- El repartidor envia s_i al participant i



Exercici 1 Considereu l'esquema senzill presentat a l'apartat anterior

(a) Quins conjunts de participants poden recuperar el secret a partir dels seus fragments?

(b) I la resta de conjunts, quina informació tenen de s?

Exercici 2 Considereu l'esquema anterior per $G = \mathbb{Z}_2$ i $n = 5$.

$$C = \{(s, s_1, \ldots, s_n) : \text{ on } s_1, \ldots, s_n \text{ són possibles fragments de } s\}.$$

(a) El conjunt C és un codi lineal? Quin?

(b) Trobeu la matriu generadora d'aquest codi

Exercici 3 Ara analitzarem la generalització de l'esquema anterior per $G = \mathbb{Z}_2$ i n arbitrari. L'esquema és el següent:

- Sigui $G = \mathbb{Z}_2$ el conjunt dels possibles secrets.
- El secret $s \in G$ es comparteix amb els participants $P = \{1, \ldots, n\}$
- El repartidor tria $s_1, \ldots, s_{n-1} \in G$ uniformement a l'atzar, i calcula

$$s_n = s - s_1 - \ldots - s_{n-1}.$$

- El repartidor envia s_i al participant i.

Per aquest esquema,

(a) Quins conjunts de participants poden recuperar el secret a partir dels seus fragments?

(b) I la resta de conjunts, quina informació tenen de s?

(c) Partint d'aquest esquema, definiu

$$C = \{(s, s_1, \ldots, s_n) : \text{ on } s_1, \ldots, s_n \text{ són possibles fragments de } s\}.$$

El conjunt C és un codi lineal? Quin?

(d) Trobeu la matriu generadora del codi C.

Exercici 4 En aquest exercici, construirem un esquema de compartició de secrets a partir d'un codi lineal. Sigui

$$C = \{(c_0, c_1, \ldots, c_n)\}.$$

un codi lineal sobre $G = \mathbb{F}_q$ de dimensió k i distància mínima d. Per compartir un secret s, el repartidor considereu la família de paraules de C en les que $c_0 = s$. Després tria uniformement a l'atzar una paraula d'aquesta família. Sigui $c = (c_0, c_1, \ldots, c_n)$ la paraula triada. Aleshores el repartidor envia c_i al participant i.

(a) El conjunt $\{1, \ldots, n\}$ pot recuperar el secret?

(b) Un conjunt de mida $1 < r < n$ pot determinar el secret? A partir de la distància mínima, digueu per quins r es pot garantir que el conjunt pot recuperar el secret.

(c) Trobar la família de paraules de C en les que $c_0 = s$ pot ser molt costós, i la llista pot ser molt llarga. Hi ha un mètode alternatiu més eficient que fa servir una matriu generadora del codi. Sigui G és la matriu generadora del codi que és sistemàtica en les primeres posicions. Descriviu una manera de generar els fragments del secret fent servir la matriu G.

Exercici 5 Continuant amb l'exercici anterior, considereu el cas que tots els participants comparteixen els seus fragments, però algun d'ells menteix, aportant un valor que no és correcte. Això els dona avantatge, perquè ells disposen de més informació que la resta, i per tant és una situació que s'ha d'evitar. L'esquema proposat a l'exercici anterior, però garanteix certa resiliència contra aquests atacs.

(a) Suposeu que $d = 5$, i només un participant ha mentit. Quan intentin trobar el secret, què passarà?

(b) Relacioneu la capacitat d'evitar participants fraudulents amb la capacitat correctora d'errors.

Capítol 4
Codis Reed-Solomon

4.1 Matrius de Vandermonde

Introducció

- Les matrius de Vandermonde són un tipus de matrius amb una estructura determinada que aporta propietats molt interessants.
- Tenen aplicacions en moltes àrees com les comunicacions digitals, el porcessat d'imatge, antenes (MIMO) i s'empren en el càlcul de la transformada discreta de Fourier (DFT), interpolació de funcions, ...
- En aquest exercici veiem matrius de Vandermonde definides sobre cossos finits, però les propietats que descrivim també es compleixen per cossos no finits.

Definició
Donats $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{F}_q$, la matriu de Vandermonde de $\alpha_1, \ldots, \alpha_n$ d'ordre r es defineix com

$$V_r(\alpha_1, \alpha_2, \ldots, \alpha_n) = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \ldots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{r-1} & \alpha_2^{r-1} & \ldots & \alpha_n^{r-1} \end{pmatrix}$$

El determinant de $V_n(\alpha_1, \alpha_2, \ldots, \alpha_n)$ satisfà

$$|V_n(\alpha_1, \alpha_2, \ldots, \alpha_n)| = \prod_{1 \leqslant i < j \leqslant n} (\alpha_j - \alpha_i)$$

Exercici 1

(a) Construïu les matrius següents:

(a) La matriu de Vandermonde de $0, 1, 2, 3, 4 \in \mathbb{F}_5$ d'ordre 3.

(b) La matriu de Vandermonde de $0, 1, 3, 5 \in \mathbb{F}_7$ d'ordre 3.

(c) La matriu de Vandermonde de $0, 1, \alpha, \alpha^2 \in \mathbb{F}_4$ d'ordre 3

(b) Calculeu el determinant de la matriu de Vandermonde de $1, 2, 3 \in \mathbb{F}_5$ d'ordre 3 de dues maneres diferents: fent servir el procediment habitual i fent servir la fórmula del requadre superior.

Avaluació de polinomis
Sigui f el poliniomi definit com $f(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}$, amb $a_0, \ldots, a_{n-1} \in \mathbb{F}_q$.

Llavors es compleix la igualtat següent. Per tot $\beta \in \mathbb{F}_q$,

$$\begin{pmatrix} 1 & \beta & \beta^2 & \ldots & \beta^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{pmatrix} = f(\beta)$$

Per tant, prenent α un element primitiu,

$$\begin{pmatrix} 1 & 1 & 1 & 1 & \ldots & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \ldots & \alpha^{(n-1)} \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \ldots & \alpha^{2(n-1)} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \ldots & \alpha^{3(n-1)} \\ \vdots & & & & & \vdots \\ 1 & \alpha^{n-1} & \alpha^{2(n-1)} & \alpha^{3(n-1)} & \ldots & \alpha^{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} f(1) \\ f(\alpha) \\ \vdots \\ f(\alpha^{n-1}) \end{pmatrix}.$$

Exercici 2

(a) Sigui $f(x) = 1 + 2x + 3x^3 \in \mathbb{F}_5[x]$. Calculeu $f(2)$ multiplicant un vector per una matriu de Vandermonde.

(b) Trobeu el polinomi $g(x) \in \mathbb{F}_5[x]$ de grau com a molt 2 que satisfà $g(2) = 2$, $g(3) = 3$, $g(4) = 1$

4.2 Codis Reed-Solomon primitius

Codis Reed-Solomon

Sigui $\mathbb{F}_q$ el cos de q elements i sigui w un element primitiu de $\mathbb{F}_q$. Aleshores $\mathbb{F}_q = \{0, 1, w, w^2, \ldots, w^{q-2}\}$. Sigui $n = q - 1$.
Diem que el codi de Reed-Solomon primitiu sobre $\mathbb{F}_q$ de dimensió k determinat per w és el codi lineal de $\mathbb{F}_q^n$ que té com a matriu generadora

$$G = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & w & w^2 & \ldots & w^{n-1} \\ 1 & w^2 & w^4 & \ldots & w^{2(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & w^{k-1} & w^{(k-1)2} & \ldots & w^{(k-1)(n-1)} \end{pmatrix}$$

Els codis d'aquest tipus els denotem $RS_q(k)$.

Exercici 1

(a) Podeu descriure les matrius generadores de codis RS com a matrius de Vandermonde? Com?

(b) Construïu una matriu generadora d'un codi RS sobre $\mathbb{F}_5$ de dimensió 3.

Matriu de control de codis RS

Sigui w un element primitiu de $\mathbb{F}_q$. La matriu

$$H = \begin{pmatrix} 1 & w & w^2 & \ldots & w^{n-1} \\ 1 & w^2 & w^4 & \ldots & w^{2(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & w^{n-k} & w^{(n-k)2} & \ldots & w^{(n-k)(n-1)} \end{pmatrix}$$

és una matriu de control del codi Reed-Solomon primitiu de dimensió k i llargada $n = q - 1$ determinat per w.

Exercici 2

(a) Construïu una matriu de control H d'un codi RS primitiu sobre $\mathbb{F}_5$ de dimensió 3. Anomenem C aquest codi.

(b) Per trobar la distància mínima del codi, calculeu el mínim número de columnes linealment dependents de H. Quina és la distància mínima de C?

Exercici 3 En aquest exercici treballem les quatre definicions dels codis RS primitius.

(a) Trieu un element primitiu de $\mathbb{F}_7$ i doneu una matriu generadora d'un codi $RS_7(3)$. Sigui C el codi determinat per aquesta matriu generadors. Aquesta matriu l'anomenem G_1

(b) Doneu una matriu de control de C que sigui la trasposada d'una matriu de Vandermonde. Aquesta matriu l'anomenem H_1

(c) Descriviu C com un codi d'avaluació.

(d) Trobeu el polinomi generador del codi C.

(e) Doneu una matriu generadora de C diferent a G_1 a partir del polinomi generador.

(f) Calculeu el polinomi de control de C.

(g) Doneu una matriu de control de C diferent a H_1. Construïu aquesta matriu a partir del polinomi de control.

Exercici 4

(a) Preneu un element primitiu w i dona una matriu de control H del codi $RS_5(2)$ determinat per w.

(b) Calculeu el mínim número de columnes linealment dependents de H.

(c) Comproveu si les paraules (0132), (1111), (1211), (1241), (1243), (2413) pertanyen al codi $RS_5(2)$ que has triat.

(d) Per cada una de les paraules c del codi $RS_5(2)$ trobades a l'apartat anterior, trobeu el polinomi de $f_c \in \mathbb{F}_q[x]^{<2}$ que satisfà

$$c = (f_c(1), f_c(w), f_c(w^2), f_c(w^3)).$$

Codis MDS

- La fita de Singleton estableix que per tot codi de longitud n i dimensió k la distància mínima és com a molt $n - k + 1$. Els codis que assoleixen aquesta fita s'anomenen Maximum Distance Separable (MDS) codes. En aquests codis,
$$d = n - k + 1,$$
- on d és la distància mínima del codi. Els codis MDS tenen un paper molt rellevant en la teoria de codis perquè, al tenir la màxima distància mínima possible, són els que tenen més capacitat per detectar i corregir errors.

- Els codis Reed-Solomon primitiussón codis MDS.
- En el cas en què $n < q-1$, també és possible construir codis MDS que són molt semblants als RS primitius (els veurem a la pràctica següent). En algunes referències, aquests codis també s'inclouen a la família dels codis RS.

Exercici 5 En aquest exercici analitzem la relació entre codis Reed Solomon determinats per elements primitius diferents.

(a) Primerament, considereu dos codis $RS_5(3)$ determinats pels elements primitius $w = 2$ i $w = 3$.

   (a) Trobeu matrius generadores del tipus Vandermonde per aquests dos codis.

   (b) Quina relació hi ha entre les dues matrius generadores?

   Ara analitzem el cas general. Considereu dos codis $RS_q(k)$ determinats per elements primitius diferents. Per aquests dos codis, considereu matrius generadores del tipus Vandermonde descrites al principi de la pràctica. Les anomenem G_1 i G_2.

   (a) Quina relació hi ha entre G_1 i G_2?

   (b) Considereu els polinomis generadors dels dos codis. Argumenteu per què són diferents, en general.

4.3 Codis Reed-Solomon no primitius

Exercici 1 Siguin $\alpha_1, \ldots, \alpha_n$ elements diferents d'un cos finit $\mathbb{F}_q$. Considereu la matriu de control següent.

$$H = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \ldots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \ldots & \alpha_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_{n-k} & \alpha_{n-k}^2 & \ldots & \alpha_{n-k}^{n-1} \end{pmatrix}$$

(a) Quina és la distància mínima del codi determinat per una matriu H d'aquest tipus?

(b) Sigui α un element primitiu de $\mathbb{F}_q$. Quins valors de n i $\alpha_1, \ldots, \alpha_{n-k}$ prendries per tal d'obtenir la matriu de control d'un codi Reed Solomon primitiu?

Exercici 2 Sigui H una matriu de control com la de l'exercici anterior, on $\alpha_1, \ldots, \alpha_n$ són elements diferents d'un cos finit $\mathbb{F}_q$ amb $n \leqslant q-1$. Sigui C el codi associat a H.

(a) Sigui $(c_0, \ldots, c_{n-1})$ una paraula del codi determinat per C. Considereu ara el polinomi

$$c(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{n-1} x^{n-1}$$

Quant val $c(\alpha_1)$? Quant val $c(\alpha_i)$ per $1 \leqslant i \leqslant n-k$?

(b) Trobeu un polinomi mònic que divideix tots els polinomis associats a paraules del codi.

(c) Sigui $g(x)$ el polinomi definit a l'apartat anterior. Justifiqueu que tota paraula del codi està associada a un polinomi $p(x)g(x)$, on $p(x)$ és un polinomi de grau inferior a k.

Exercici 3 Sigui $n < q-1$ i $k \leqslant n$. Considereu els elements $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$, tots ells diferents, i el conjunt següent:

$$C = \left\{ (p(\alpha_1), p(\alpha_2), \ldots, p(\alpha_n)) : p \in \mathbb{F}_q[x]^{<k} \right\}.$$

(a) C és un codi lineal. Per què?

(b) Quin és el mínim pes de les paraules no nul·les d'aquest codi?
(Ajuda: Quantes arrels pot tenir un polinomi de grau $< k$?).

(c) Justifiqueu per què és un codi MDS.

(d) Doneu una matriu generadora d'aquest codi.

Els codis considerats als exercicis anteriors també se'ls anomena codis de Reed-Solomon, en algunes referències. Nosaltres, a teoria i als problemes, ens restringim al cas de codis RS primitius.

Exercici 4 Construcció de codis MDS

(a) Digueu quins codis MDS coneixes de dimensió $k = 1$. Per aquests codis, digueu si hi ha alguna restricció relacionada amb n i q

(b) Digueu quins codis MDS coneixes de dimensió $k = n - 1$. Per aquests codis, digueu si hi ha alguna restricció relacionada amb n i q

(c) Digueu quins codis MDS coneixes de dimensió $k > 1$. Per aquests codis, digueu quines restriccions hi ha relacionades amb n i q.

4.4 Implementació de Codis Reed-Solomon

Aplicacions

- L'interès pels codis RS està fonamentat en el fet que la relació entre la distància mínima i la redundància és l'òptima (són codis MDS), i que hi ha algoritmes eficients per la correcció d'errors.
- Durant les darreres dècades s'han millorat els algoritmes de codificació i descodificació, i també s'han desenvolupat circuits específics.
- Els codis Reed-Solomon són de gran interès. Son emprats en aplicacions que dedicades a s'emmagatzematge de dades, i transmissió de dades en aplicacions que no admeten demores ni errors (àudio, connexió vídeo-HDMI,...). Des dels anys 60, s'han utilitzat en nombroses aplicacions, com l'escriptura de dades en CDs, DVDs, Blu-Ray, codis QR, transmissió de dades amb WiMAX i DVB, i moltes altres teconologies.
- La codificació és sistemàtica, en general.

Codificació sistemàtica

- Per codificar sistemàticament, es pot buscar una matriu generadora del tipus $(Id|P)$ o $(P|Id)$ i codificar amb aquesta matriu. Ara bé, a nivell hardware hi ha solucions més eficients que eviten emmagatzemmar la matriu.
- En el cas dels codis Reed Solomon primitius, hem vist que admeten codificació sistemàtica a partir d'una interpretació polinomial del codi. O sigui, que el poden veure com el conjunt de paraules que es corresponen amb els polinomis
$$p(x)g(x),$$
on $g(x)$ és el polinomi generador i $p(x)$ és un polinomi de grau inferior a $n-k$. En aquest cas, la codificació és com la que vam veure amb els codis cíclics.
- En aquest cas, però, la divisió és entre polinomis de $\mathbb{F}_q[x]$, on normalment $q > 2$. Això requereix mètodes eficients per executar operacions aritmètiques a $\mathbb{F}_q$. Sovint, a les solucions hardware, els elements de $\mathbb{F}_{2^r}$ es guarden en notació vectorial i els productes es realitzen amb l'ajuda d'una look-up-table.
- La descodificació és més complicada que la codificació, perquè inclou correcció d'errors.
- A les figures següents es mostra un cas concret de circuïts embedded per la codificació i descodificació de codis RS. Els diagrames següents mostren els diferents operacions principals i les principals comunicacions entre els blocs operatius.

Exercici 1 Descriviu un mètode de codificació sistemàtica per C que considereu eficient. Discutiu la seva eficiència.

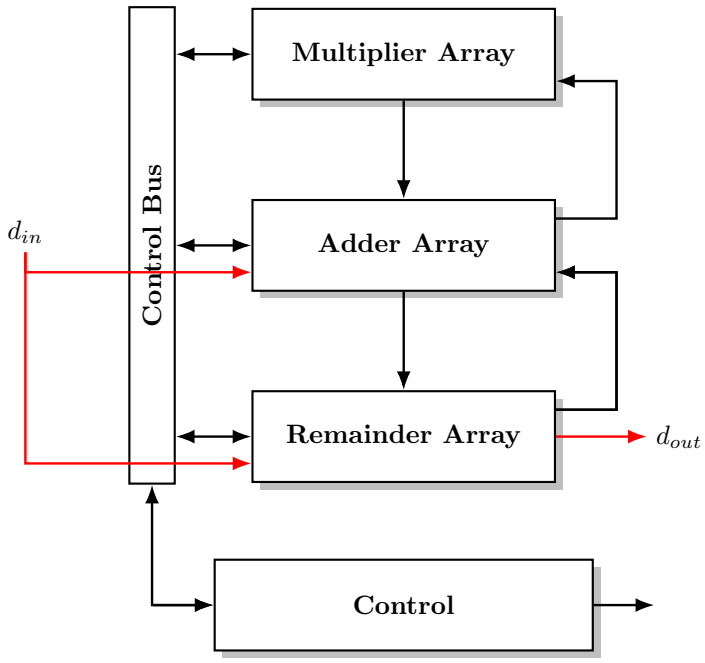


Figura 4.1: Esquema de codificació sistemàtica en hardware.

Exercici 2 Interpreteu les figures corresponents a la codificació sistemàtica i la descodificació dels codis RS, descrivint les operacions que es realitzen.

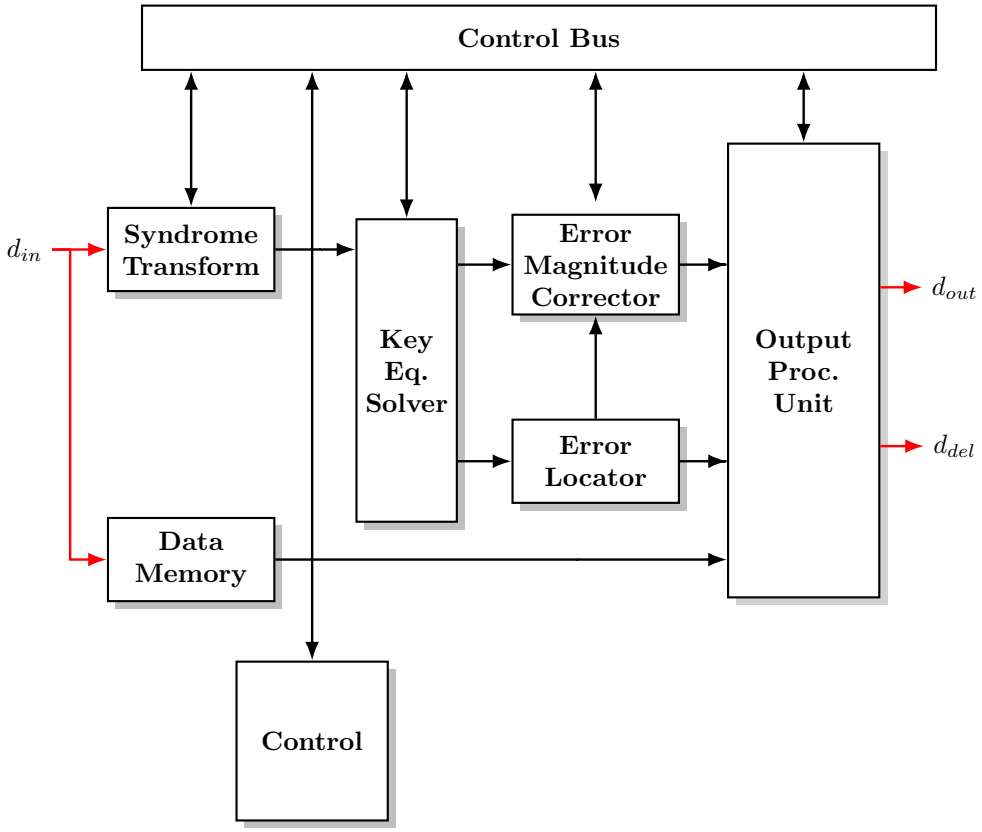


Figura 4.2: Esquema de descodificació sistemàtica en hardware.

4.5 Errors de transmissió

Exercici 2 Volem construir dos codis lineals:

- Un codi MDS C_1 sobre $\mathbb{F}_{2^4}$ amb $k = 5$ i $d = 3$.
- Un codi MDS C_2 sobre $\mathbb{F}_{2^5}$ amb $k = 4$ i $d = 3$.

Observeu que, en ambdós casos, codificarem missatges que es poden escriure amb 20 bits, que tractarem com 5 elements de $\mathbb{F}_{2^4}$ i 4 elements de $\mathbb{F}_{2^5}$, respectivament.

(a) Quina és la capacitat correctora d'aquests codis C_1 i C_2? Quina és la longitud de cada codi?

(b) Quina és la taxa de transmissió dels codis C_1 i C_2? Si les paraules del codi s'envien per un canal de comunicació binari, quants bits s'han d'enviar per cada missatge?

(c) Ara suposem que en la transmissió dels bits pel canal, hi ha una ràfega d'errors al principi de la transmissió, i que després la transmissió és perfecta. Suposeu que l'error és el següent:

   (a) 100...00
   (b) 111100...00
   (c) 1111100...00
   (d) 1000100...00
   (e) 1000010...00
   (f) 0000110...00

   Per cada un d'aquests casos, digueu si els codis C_1 i C_2 garanteixen la correcció de la paraula transmesa.

(d) Indiqueu com obtindríeu codis del tipus C_1 i C_2.

4.6 Esquema de Shamir

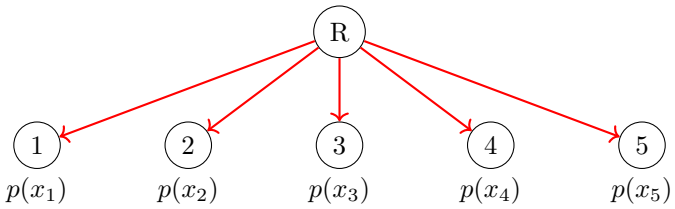
L'esquema de Shamir és un esquema de compartició de secrets (vegeu l'apartat dedicat a aquests esquemes al capítol anterior). Aquest esquema es va presentar el 1979 i és un dels més coneguts i emprats. Aquest esquema està definit per cossos finits $\mathbb{F}$ que tinguin més elements que el nombre de participants, $|\mathbb{F}| > n$, i depèn d'un paràmetre $t < n$

- Input: $s \in \mathbb{F}$
- Per compartir el secret s, el repartidor pren $t-1$ elements $a_1, \ldots, a_t \in \mathbb{F}$ uniformement a l'atzar.
- El repartidor defineix el polinomi

$$p(x) = s + a_1 x + \ldots + a_t x^t$$

- A cada participant, el repartidor li envia una evaluació polinomi. O sigui, el repartidor tria $x_1, \ldots x_n \in \mathbb{F}$ diferents i diferents de zero, i envia $p(x_i)$ al participant i.

El dibuix següent és pel cas $n = 5$.



Exercici 1 Com vam fer al capítol anterior, ara analitzarem el codi generat per aquest esquema.

$$C = \{(s, s_1, \ldots, s_n) : \text{ on } s_1, \ldots, s_n \text{ són possibles fragments de } s\}.$$

Observeu que a l'esquema de Shamir, $p(0) = s$. Per tant, un cop fixats $x_1, \ldots, x_n$, el codi definit abans és equivalent a

$$C = \{(p(0), p(x_1), \ldots, p(x_n)) : \text{ on } p \text{ és un polinomi de grau } t\}.$$

(a) El conjunt C és un codi lineal? Quin?

(b) Descriviu l'esquema de Shamir matricialment: trobeu una matriu M per la qual els fragments es poden obtenir com $(s, a_1, \ldots, a_n)M$.

Exercici 2 Responeu a les preguntes següents tenint en compte les propietats del codi C.

(a) Per què els conjunts de mida $t+1$ poden recuperar el secret? Què en podem dir dels de mida t?

(b) I si hi ha participants que menteixen? Quina és la resiliència d'aquest esquema?

Capítol 5
Dissenys Avançats

5.1 Interleaving

Interleaving

- L'*interleaving* (o entrellaçat) és una tècnica que consisteix a reordenar els símbols de diverses paraules de codi per reduir l'impacte d'errors agrupats.
- Donades t paraules del codi $\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(t)} \in \mathbb{F}_q^n$ a enviar, la tècnica d'interleaving de bloc consisteix en escriure les paraules per files en una matriu

$$M = \begin{pmatrix} c_1^{(1)} & c_2^{(1)} & \ldots & c_n^{(1)} \\ c_1^{(2)} & c_2^{(2)} & \ldots & c_n^{(2)} \\ \vdots & \vdots & & \vdots \\ c_1^{(t)} & c_2^{(t)} & \ldots & c_n^{(t)} \end{pmatrix},$$

  i enviar la informació en blocs d'informació determinat per les columnes. O sigui, enviaríem les paraules $(c_1^1, c_1^2, \ldots, c_1^t)$, després $(c_2^1, c_2^2, \ldots, c_2^t)$, i així successivament.

Exercici 1: Suposem que hem codificat tres missatges amb un codi de distància mínima 3, obtenint tres paraules binàries de longitud 7:

$$\mathbf{c}^{(1)} = (1,1,1,0,0,0,0), \quad \mathbf{c}^{(2)} = (1,0,0,1,1,0,0), \quad \mathbf{c}^{(3)} = (0,1,0,1,0,1,0).$$

Mostreu com queden les paraules transmeses després d'aplicar interleaving de bloc.

Exercici 2: Ara compararem la capacitat per corregir errors i esborralls amb les dues estratègies d'enviament de missatges: enviant una paraula després de l'altra o aplicar l'interleaving de bloc. Pels casos següents, discutiu en quin cas es pot recuperar el missatge.

- Hi ha hagut un problema de transmissió i el receptor ha deixat de rebre 2 bits consecutius.
- Hi ha hagut un error de transmissió i el receptor ha rebut la informació amb 3 errors consecutius.

- Hi ha hagut un problema de transmissió i el receptor ha deixat de rebre 6 bits consecutius.
- Hi ha hagut un problema de transmissió i el receptor ha deixat de rebre bits consecutius.

Justifiqueu per què l'interleaving és una pràctica útil quan és probable que els errors es produeixin en ràfegues. Discutiu si aporta algun avantatge quan els errors són uniformement distribuïts.

Exercici 3:

(a) Proposeu un esquema d'interleaving senzill que pugui implementar-se fàcilment en maquinari fent servir un buffer o una memòria reduïda.

(b) Proposeu una implementació hardware que només faci servir portes delay.

5.2 Codificació del Compact Disc

Cas d'ús de codis Reed Solomon

- En aquesta pràctica ens aproximarem a l'ús de codis RS en l'emmagatzematge de música en CDs.
- L'ús de CDs ja no és gaire popular, però hem triat aquest cas d'ús perquè els codis RS que s'utilitzen tenen paràmetres petits.
- El DVD, per exemple, els codis que es fan servir són codis RS amb $n > 180$, i amb Blue-Ray es fan servir codes amb $n > 200$.
- En el CD, igual que vam veure a la pràctica anterior, s'entrellacen elements de diferents paraules (interleaving) per tal d'aconseguir millors prestacions.

Compact Disc Digital Audio (CDDA)

- El 1980 Sony i Philips van arribar a un acord per definir un model d'escriptura en CDs, que es va ratificar el 1987 com a estàndard al IEC 60908.
- En el *Red Book Standard*, Sony i Philips van descriure els detalls tècnics de l'escriptura de música en CDs. Implícitament, aquest estàndard va definir la qualitat de so de la música emmagatzemada en aquests dispositius. Més endavant s'anomenaria *qualitat CD*.
- La principal millora respecte formats i suports d'àudio anteriors (disc de vinil, cintes magnètiques, k7, 8 pistes, cilindre de fonògraf,...) és que la música s'emmagatzema en format digital, creant un format compatible amb els ordinadors. És un format que permet l'emmagatzemament de música i la transferència entre diferents sense alteració del contingut. Un dels reptes que es van haver d'afrontar va ser garantir la reproducció correcta del contingut en situacions en les que hi ha errors de lectura deguts a imperfeccions del suport (pols, ditades, ratllades...).

Especificacions

- La informació que s'enregistra en un CD és binària, i es grava en espiral en un disc. A la superfície del disc hi ha *valls* i *muntanyes* que s'interpreten com 1s i 0s. Un emissor làser envia un feix al CD, i el làser torna reflectit. Si hi ha una vall, el feix ha recorregut més distància que si hi ha una muntanya. Això es pot detectar analitzant la fase de la senyal enviada. El làser recorre el disc a 1,3 m/s. Detalls sobre aquesta part del sistema CD es poden trobar a `http://hyperphysics.phy-astr.gsu.edu/hbase/Audio/cd.html`
- La música es sampleja a 44.1 kHz, i per cada mostra es dediquen 16 bits per quantificar la mostra de cada canal (so estéreo). Això vol dir que hi ha $2^{16} = 65536$ valors possibles per cada mostra, i que es fa una mostra cada $2.268 \cdot 10^{-5}$ segons. Aleshores per cada segon d'àudio es graven 1411200 bits.
- Les dades s'agrupen en 6 peces de 16 bits pel canal esquerre (L) i 6 peces de 16 bits pel canal dret (R). Això es transforma en 24 peces de 8 bits (L) i 24 peces de 8 bits (R). En aquest moment, tenim la informació en 48 bytes.

Exercici 1 Construïm un codi lineal d'acord amb les especificacions següents:

(a) Volem que els símbols del codi siguin bytes.

(b) Volem codificar missatges de 24 bytes.

Tenint en compte aquestes propietats, sobre quin cos convindria definir el codi? Quina és la dimensió del codi?

D'entre els polinomis irreductibles sobre $\mathbb{F}_2$ següents, tria'n un que pugui ser útil per construir el cos sobre el qual definirem el codi.

(a) $x^7 + x + 1$

(b) $x^8 + x^4 + x^3 + x + 1$

(c) $x^9 + x + 1$

(d) $x^{10} + x^3 + 1$

Exercici 2 Cada bit ocupa uns 6 μm en un CD, i per tant cada byte ocupa 48 μm. Volem codificar missatges de 24 bytes. Considerem que hi pot haver errors de lectura de 1152 μm en el CD. Quina hauria de ser la longitud del codi per tal de corregir els errors provocats? En aquest cas, quina és la taxa de transmissió?

CIRC

- Com hem vist a l'exercici anterior, els codis correctors d'errors ens poden ajudar a la transmissió informació sense errors, però la solució considerada no és gens eficient.
- Per poder emmagatzemar més informació, es va dissenyar una tècnica que combina dos codis RS. Es diu Cross-interleaved Reed-Solomon Codes (CIRC). L'ús d'aquesta tecnologia es basa en l'assumpció que quan hi ha errors de lectura, normalment venen seguits. Això és molt comú en CDs, degut a les ratllades i a la pols.
- El primer codi, anomenat C2, és un codi MDS de dimensió 24 i llargada 28, i s'utilitza codificació sistemàtica. El segon codi, anomenat C1, és un codi MDS de dimensió 28 i llargada 32, i s'utilitza codificació sistemàtica. En ambdos codis, els 4 bytes finals es consideren bytes de control. Els símbols són bytes.

Exercici 3

(a) Per què creus que es trien codis sistemàtics? Quins són els seus avantatges?

(b) Com construiries un codi sistemàtic del tipus C1 i un codi del tipus C2.

Exercici 4

(a) Quants errors pot corregir C1? Quants errors pot corregir C2? Quina és la taxa de transmissió de cada un dels codis?

(b) Ara suposem que un missatge de 24 bytes primer es codifica amb C2, i després la paraula resultant es codifica amb C1. Sigui c la paraula resultant de codificar un missatge amb amb C2 i després amb C1. Si en comptes de c rebéssim una paraula c' amb errors, quants errors es podrien corregir? Explica-ho.

CIRC (II)

- Com hem vist a l'exercici anterior, la composició dels dos codis té una bona taxa de transmissió, però corregeix pocs errors. Una partícula de pols introdueix errors que no es podrien corregir.
- A CIRC, els codis C1 i C2 no es composen directament, no es fa de la manera que s'ha dit al Problema 2. Els bytes de les paraules del codi C2 es barregen amb les d'altres paraules (*cross-interleaved*). Aquest és un tipus de codis dels anomenats *burst error correcting codes*, i estan dissenyats per paliar l'efecte d'errors que venen seguits `https://en.wikipedia.org/wiki/Burst_error-correcting_code`

Interleaving

- L'*interleaving* es fa entrellaçant elements de diverses paraules del codi. Per fer-ho en hardware, és comú utilitzar *delays* en els circuits de transmissió. És una opció molt econòmica i que en el cas dels CDs ha donat molt bons resultats.
- Considerem un exemple senzill: suposem que tenim paraules d'un codi binari de llargada 3, i tenim 3 cables on viatgen els 3 bits en paral·lel. Suposem que posem un retard d'un clock de rellotge al segon cable i un retard de 2 clocks al tercer cable. El que tenim a cada instant al final dels tres cables són tres bits de paraules diferents: és el primer element és d'una paraula, el segon element és de la paraula anterior, i el tercer element és de la de fa dos clocks. D'aquesta manera, podem entrellaçar tres paraules.
- En el CIRC, fent servir retards, s'entrellacen paraules dels codis C1 i C2, de diverses maneres. En el cas de C1, l'input són 28 bytes, i cada un té un retard diferent, de 0 a 27 clocks de rellotge. D'aquesta manera, es barregen 28 missatges diferents.

Exercici 5

(a) Quin és l'avantatge d'entrellaçar paraules d'un codi?

(b) Sigui c una paraula del codi CIRC. Si hi hagués errors de lectura en els tres primers bytes de c, els podria corregir? Raona la resposta.

Resultat

- Després d'entrellaçar les paraules del codi C2, es codifica amb el codi C1, i es tornen a entrellaçar les paraules a la sortida. Amb aquesta tècnica, és possible corregir errors seguits de fins a 4000 bits (2.5mm), aproximadament.
- Al descodificar, primer es descodifiquen les paraules del codi C1, i després les del codi C2. Si al descodificar una paraula de 32 bytes d'acord amb el codi C1 es detecten més errors dels que es poden corregir, aleshores es descodifica com un missatge de 28 esborralls.

Exercici 6

(a) Al requadre anterior expliquem una estratègia que s'utilitza per descodificar els missatges. Explica per què és millor descodificar-lo com un missatge de 28 esborralls en comptes d'un missatge de 28 errors.

(b) Compara el codi que has proposat a l'Exercici 1 amb el CIRC, comparant la taxa d'informació i els errors que es poden corregir.

5.3 Locally Repairable Codes

Aquest apartat el dediquem a una aplicació important dels codis MDS: l'emmagatzematge distribuït. La nostra activitat digital remota creix continuament (podeu consultar `https://www.domo.com/data-never-sleeps`). Això crea un problema creixent: emmagatzemar aquesta informació.

L'usuari dels serveis remots confia en el proveïdor, i assumeix que les seves dades no desapareixieran (correus electrònics a Gmail, dades de compres a Amazon, videos pujats a Youtube, fotos pujades a Instagram, tweets, llistes de Spotify, perfil de Linkedin ...). Això fa que els proveïdors de serveis hagin de tenir molta cura de l'emmagatzematge de les dades, per tal que no es perdi informació.

Però, amb el temps, amb major o menor mesura, els dispositius dedicats a l'emmagatxamatge de dades es degraden, i acaben perdent informació. Per veure la freqüència error dels discs durs per fabricant, podeu consultar estudis com aquest:
`https://www.backblaze.com/blog/backblaze-drive-stats-for-q3-2023/`

Tradicionalment, les grans companyies optaven per l'emmagatzematge distribuït, mantenint diverses rèpliques de les mateixes dades en diversos llocs del món. Típicament, eren tres copies de cada fitxer. Així s'evitava perdre dades si en un d'aquests llocs hi havia un problema.

Darrerament, els codis lineals s'utilitzen per reduir la quantitat d'informació a emmagatzemar i per reduir el cost computacional de la regeneració de les dades, en el cas que es detectin errors. Microsoft Azure, Facebook i Blackblaze, per exemple, en fan servir.

Per il·lustrar aquesta funcionalitat, veurem un exemple de *locally repairable codes*. L'exemple està extret de l'article *Locally Repairable Codes* de Dimitris S. Papailiopoulos i Alexandros G. Dimakis. Una solució simplificada està mostrada a les figures següents.

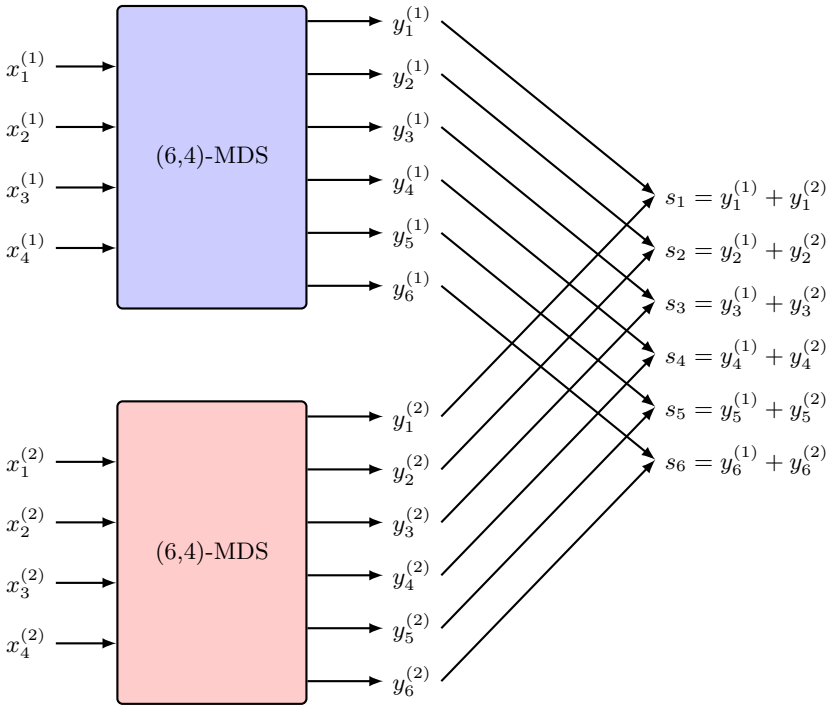
- Sigui $m = (x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, x_4^{(1)}, x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, x_4^{(2)})$ el missatge que volem emmagatzemar.

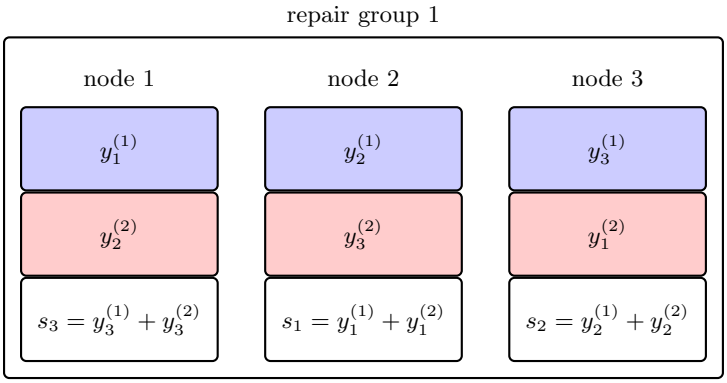
- Aquest vector de 8 elements el divideixen en dos vectors de quatre elements, i cada vector es codifica amb un codi MDS, obtenint els elements $y_1^{(1)}, \ldots, y_6^{(1)}$, $y_1^{(2)}, \ldots, y_6^{(2)}$.

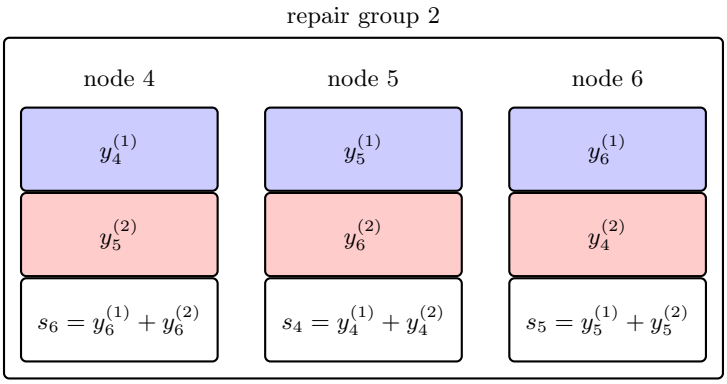
- Després es generen els elements $s_1, \ldots, s_6$ a partir de $y_1^{(1)}, \ldots, y_6^{(1)}, y_1^{(2)}, \ldots, y_6^{(2)}$ seguint l'equació
$$s_i = y_i^{(1)} + y_i^{(2)}.$$
S'obté doncs un total de 18 elements que es guarden en 6 nodes tal com es mostra a la imatge (Block Placement).

- Els nodes 1,2,3 s'emmagatzemen en llocs propers, i els nodes 4,5,6 també.

- L'avantatge d'aquest disseny és que quan apareix un error en un node, es pot corregir fent servir només els nodes en el mateix repair grup (veure l'exercici).


$x_1^{(1)}$
$x_2^{(1)}$
$x_3^{(1)}$
$x_4^{(1)}$
(6,4)-MDS
$y_1^{(1)}$
$y_2^{(1)}$
$y_3^{(1)}$
$y_4^{(1)}$
$y_5^{(1)}$
$y_6^{(1)}$
$x_1^{(2)}$
$x_2^{(2)}$
$x_3^{(2)}$
$x_4^{(2)}$
(6,4)-MDS
$y_1^{(2)}$
$y_2^{(2)}$
$y_3^{(2)}$
$y_4^{(2)}$
$y_5^{(2)}$
$y_6^{(2)}$
$s_1 = y_1^{(1)} + y_1^{(2)}$
$s_2 = y_2^{(1)} + y_2^{(2)}$
$s_3 = y_3^{(1)} + y_3^{(2)}$
$s_4 = y_4^{(1)} + y_4^{(2)}$
$s_5 = y_5^{(1)} + y_5^{(2)}$
$s_6 = y_6^{(1)} + y_6^{(2)}$


repair group 1
node 1
node 2
node 3
$y_1^{(1)}$
$y_2^{(2)}$
$s_3 = y_3^{(1)} + y_3^{(2)}$
$y_2^{(1)}$
$y_3^{(2)}$
$s_1 = y_1^{(1)} + y_1^{(2)}$
$y_3^{(1)}$
$y_1^{(2)}$
$s_2 = y_2^{(1)} + y_2^{(2)}$


repair group 2
node 4
node 5
node 6
$y_4^{(1)}$
$y_5^{(2)}$
$s_6 = y_6^{(1)} + y_6^{(2)}$
$y_5^{(1)}$
$y_6^{(2)}$
$s_4 = y_4^{(1)} + y_4^{(2)}$
$y_6^{(1)}$
$y_4^{(2)}$
$s_5 = y_5^{(1)} + y_5^{(2)}$

Exercici 1

(a) En l'exemple descrit a la figura es fa servir un (6,4)-MDS code, que és un codi MDS de longitud n=6 i dimensió k=4. Construïu un codi d'aquest tipus, donant-ne una matriu generadora.

(b) Estatègia de "reparació": Si ens adonem que el node 1 de la figura no és correcta (a través d'un codi CRC, per exemple), descartarem la informació del node i el "repararem" amb la informació del nodes en el mateix repair group. Amb la informació dels nodes 2 i 3 n'hi ha prou. Com es pot fer?

(c) Pels casos següents, digueu si es poden recuperar els nodes, en el cas que es detectin els errors següents:

   (a) node 4
   (b) nodes 1 i 2
   (c) nodes 1 i 4
   (d) nodes 1, 2 i 3
   (e) nodes 1, 2 i 5
   (f) nodes 1, 2, 5 i 6

5.4 Codis Golay

Definició El *codi de Golay binari* $\mathcal{G}_{23}$ és un codi lineal binari de longitud 23, dimensió 12 i distància mínima 7. És un codi *perfecte*, ja que les esferes de radi 3 al voltant dels vectors del codi particionen $\mathbb{F}_2^{23}$. Aquest codi va ser presentat per Marcel Golay el 1949 i ha tingut molta repercussió en el món de les telecomunicacions. És cíclic i el seu polinomi generador, que és

$$g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1.$$

Exercici 1

(a) Comproveu que aquest polinomi divideix $x^{23} + 1$. Comproveu també que el polinomi $g^*(x)$ també divideix $x^{23} + 1$.

(b) Doneu una matriu generadora i una matriu de control d'aquest codi.

Extensions de codis Sigui C és un codi binari de longitud n. Definim el codi estès C' com:

$$C' = \{(c_1, c_2, ..., c_n, c_{n+1}) : (c_1, c_2, ..., c_n) \in C, \sum_{i=1}^{n+1} c_i = 0\}$$

Exercici 2 Sigui C és un codi lineal de matriu generadora G i matriu de control H.

(a) Trobeu, en funció d'aquestes, la matriu generadora i de control del codi C'.

(b) Si C té distància mínima d, trobeu la distància mínima de C'.

Exercici 2 El *codi de Golay binari estès* és el codi binari que s'obté extenent el codi $\mathcal{G}_{23}$, i es denomina $\mathcal{G}_{24}$.

(a) Trobeu els paràmetres d'aquest codi.

(b) Escriviu una matriu generadora de $\mathcal{G}_{24}$ extenent la de $\mathcal{G}_{23}$ amb una columna adicional.

(c) Comproveu que $\mathcal{G}_{24} = \mathcal{G}_{24}^{\perp}$.

5.5 Codis de Goppa

Definició

- Sigui $\mathbb{F}_{2^m}$ un cos finit i $L = (\alpha_1, \ldots, \alpha_n)$ una llista d'elements distints de $\mathbb{F}_{2^m}$.
- Siguin $g(x) \in \mathbb{F}_{2^m}[x]$ un polinomi tal que $g(\alpha_i) \neq 0$ per a tot i. El *codi de Goppa binari* definit per L i $g(x)$ és:

$$\Gamma(L, g) = \left\{ \mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{F}_2^n : \sum_{i=1}^{n} \frac{c_i}{x - \alpha_i} \equiv 0 \pmod{g(x)} \right\}.$$

Exercici 1: Mostreu que el conjunt $\Gamma(L, g)$ és un subespai vectorial de $\mathbb{F}_2^n$.

Exercici 2: Mostreu que si $\deg(g) = t$, aleshores la distància mínima del codi satisfà $d \geq 2t + 1$.

Exercici 3: Doneu una matriu de control genèrica del codi $\Gamma(L, g)$.

5.6 Criptosistema de McEliece

Criptografia basada en codis

- El criptosistema de McEliece és un esquema de clau pública, i compleix la mateixa funció que l'esquema RSA, vist anteriorment.
- Va ser proposat per Robert McEliece el 1978, i la construcció es basa en codis de Goppa. Es pot considerar que va ser el punt de partida de la criptografia basada en codis.
- Durant molts anys no s'ha tingut en compte a nivell pràctic perquè les claus públiques són molt grans, comparades amb les de RSA. Però recentment s'ha reconsiderat el seu ús pràctic perquè RSA és vulnerable a atacs quàntics
- La seguretat de l'esquema de McEliece s'ha estudiat intensivament, i no es coneixen atacs quàntics que siguin eficients,
- El 2025 el NIST ha estandaritzat l'esquema HQC. És un esquema és basat en codis i la seva seguretat es basa en els mateixos problemes que l'esquema de McEliece.

Principis de seguretat
La seguretat de l'esquema de McEliece, i de la majoria d'esquemes de clau pública basats en codis, es basa en el paradigma següent.

- Sigui A una matriu, m un vector, e un vector de pes petit, i

$$c = mA + e$$

- Un problema clàssic de la teoria de codis és el següent:

Es pot recuperar m a partir de c i A eficientment?

- En general sabem que si A és la matriu generadora s'un codi *bones* capacitats correctores, aleshores és eficient. Per exemple, els codis Reed-Solomon i els codis de Goppa.
- Tanmateix, si A és una matriu aleatòria, no hi ha cap mètode eficient, en general.

Exercici 1: Hem dit que si A és una matriu aleatòria, és difícil trobar m. Suposem que un adversari sap que e té pes t, coneix B i c.

Suposeu que intentem trobar m fent servir correcció d'errors a partir de la síndrome. O sigui, calculem una matriu de control corresponent a A, calculem la síndrome de c, i busquem el vector e' que té la mateixa síndrome que c. Feu una estimació del nombre de càlculs necessaris per trobar e.

Esbós de l'esquema
A continuació presentem una versió simplificada de l'esquema de McEliece. Suposem que Alice vol rebre missatges de manera segura. Aleshores generarà una clau pública, i una de privada (com a RSA).

(a) Alice tria

- Una matriu aletòria S que és invertible.
- La matriu generadors d'un codi de Goppa A.

(b) Alice publica $B = SA$, una matriu que sembla aleatòria. Tanmateix, manté A i S com a claus privades.

(c) Si Bob vol enviar un missatge m a Alice, envia

$$c = mB + e$$

Per algú que escolti la conversa, en principi serà difícil obtenir m a partir de c i B.

(d) Ara bé, com que

$$c = mB + e = (mS)A + e,$$

Alice pot calcular $m' = mS$ fent servir les propietats de correcció d'errors dels codis de Goppa, i llavors calcular $m = m'S^{-1}$. Això es pot fer perquè S és invertible.

Exercici 2: Discutiu l'eficiència dels algoritmes de xifrat i desxifrat.

Exercici 3: En l'esquema real, en comptes de fer servir codis de Goppa directament, es fan servir una versió binària dels codis. A més es consideren aquestes optimitzacions.

(a) En comptes de guardar A com a clau privada, es guarda la matriu de control sistemàtica.

(b) En comptes de prendre una matriu S aleatòria, es pren una funció pseudoaleatòria que genera S a partir d'una *llavor* aleatòria r, molt més petita que S.

Discutiu els avantatges computacionals i/o d'emmagatzemament que suposen aquests canvis.

5.7 Codis Reed-Muller

Definició

- Sigui $\mathbb{F}_2^m$ l'espai de tots els vectors binaris de longitud m. Denotem per $\mathcal{P}_{r,m}$ el conjunt de tots els polinomis en m variables $x_1, \ldots, x_m$ sobre $\mathbb{F}_2$ de grau total com a màxim r. Aquests són polinomis en les variables $x_1, \ldots, x_m$ en què els monomis són del tipus $x_{i_1} \cdots x_{i_\ell}$ amb $i_1 < \ldots < i_\ell$ i $\ell \leqslant r$. Les constants 0 i 1 també pertanyen a $\mathcal{P}_{r,m}$.
- Observeu que els monomis de $\mathcal{P}_{r,m}$ es poden ordenar per grau i per ordre lexicogràfic de les variables.
- El *codi de Reed–Muller* $\mathrm{RM}(r, m)$ és el conjunt de totes les avaluacions d'aquests polinomis en tots els punts de $\mathbb{F}_2^m$:

$$\mathrm{RM}(r, m) = \{(f(v))_{v \in \mathbb{F}_2^m} : f \in \mathcal{P}_{r,m}\}.$$

- Cada codi té longitud $n = 2^m$ i és un subespai de $\mathbb{F}_2^n$. (La dimensió k la calculem en un dels exercicis). Per codificar un missatge $x = (x_1, \ldots, x_k) \in \{0, 1\}^k$, es pren el polinomi $f \in \mathcal{P}_{r,m}$ en què el coeficient del monomi i-èssim és x_i. Després, s'evalua f en tots els vectors de $\mathbb{F}_2^m$, obtenint un vector de longitud 2^m.

Exercici 1

(a) Demostreu que la dimensió de $\mathrm{RM}(r, m)$ és igual a $\sum_{i=0}^{r} \binom{m}{i}$.

(b) Proveu que el dual d'un codi Reed–Muller també és d'aquesta família, i que $\mathrm{RM}(r, m)^\perp = \mathrm{RM}(m - r - 1, m)$.

Exercici 2

(a) Proveu que $\mathrm{RM}(0, m)$ és el codi de repetició i que $\mathrm{RM}(m, m) = \mathbb{F}_2^{2^m}$.

(b) Determineu els paràmetres (longitud, dimensió i distància mínima) de $\mathrm{RM}(1, 3)$.

Capítol 6
Manual de SAGE

6.1 Indicacions per començar

- La pàgina oficial de SAGE és `http://sagemath.org`. Des d'allà podeu descarregar-vos el programa.
- Es comença amb `./sage` i s'acaba amb `quit`
- Tots els comandaments han d'acabar en salt de línia
- Per interrompre un càlcul: `Ctrl + C`
- Es pot fer servir la tecla ↑ per recuperar codi escrit anteriorment
- Per escriure comandes de sistema: `!` *comandes*
- Per llegir d'un fitxer *fin.sage*: `load fin.sage`
- Per llegir d'un fitxer *fin.sage* i escriure els resultats en un fitxer *fout*: `./sage fin.sage > fout`
- Per comentar una línia: `# ...`
  Tot el que hi hagi després de # no serà llegit per SAGE.
- Per comentar tot un paràgraf o part d'un fitxer escrivim tres vegades cometes al principi i al final i el que quedi entremig no serà llegit per SAGE.

```
"""
bla
bla
bla
"""
```

- Per buscat comandes: `?`primeres lletres

6.2 Accions simples i expressions

Assignació de variables i escriptura

- Per assignar variables: `nom_variable=valor_variable`
- Per mostrar per pantalla:
  - El valor de les variables $var1, var2, \ldots$: `print var1, var2, ..`
  - Text: `print` "`text`"

Operadors relacionals

- $<$, $>$
- $=$, $!=$
- $>=$, $<=$
- `in`, `not in`

Operadors booleans

- `x and y`
- `x or y`
- `not x`

Funcions definides pel SAGE

SAGE té definides les seves pròpies funcions que depenen d'una o vàries variables (o cap variable). Per cridar-les escrivim el nom de la funció seguit de les variables entre parèntesis. Per exemple,

```
sage: floor(3.141592)
3
sage: gcd(12,8)
4
sage: sum([3,2,5])
10
```

Hi ha funcions que retornen més d'un valor. Per exemple la funció `divmod(a,b)` ens retorna el quocient i el residu de dividir a entre b.

```
sage: divmod(19,7)
2 5
```

Per assignar aquests valors a variables ho fem de la manera següent:

```
sage: q,r=divmod(19,7)
sage: q
2
sage: r
5
```

També podríem assignar els dos valors de cop:

```
sage: D=divmod(19,7)
sage: D
(2, 5)
sage: D[0]
2
sage: D[1]
5
```

6.3 Composició d'accions

Composició condicional

- ```
  if ():
      ..
  elif ():
      ..
  else:
      ..
  ```

Composició iterativa

- `for i in [i_inicial..i_final]:`
- `for i in llista:`
- `while condicio:`

6.4 Conjunts i seqüències

Conjunts i seqüències

Tant els conjunts com les seqüències són col.leccions d'objectes. Un conjunt no és ordenat, per tant, un element pot ser en un conjunt com a molt una vegada. Una seqüència, en canvi, és ordenada i, per tant, la repetició és possible. Les seqüències s'escriuen entre [ ]. Els conjunts es construeixen a partir de seqüències $set([..])$.

Per exemple,

```
sage: t = [ (-11)^2, (-7)^2, (-5)^2, (-3)^2, 3^2, 5^2, 7^2, 11^2 ]
sage: q = set(t)
sage: t
[121, 49, 25, 9, 9, 25, 49, 121]
sage: q
{121, 25, 9, 49}
```

La i-èssima entrada d'una seqüència (o d'un conjunt) `t` és `t[i]`. Però compte perquè SAGE enumera les posicions des de 0.
```

```
sage: t[1] = 1000
sage: t
[ 121, 1000, 25, 9, 9, 25, 49, 121]
sage: t[3]
9
```

SAGE té constructors especials per a seqüències. Les expressions

`[a..b]` i `[a,a+k..b]`

designen respectivament les progressions $[a, a+1, a+2, \ldots, b]$ i $[a, a+k, a+2k, \ldots, \tilde{b}]$ on $\tilde{b}$ és el màxim enter de la forma $a+ik$ menor o igual que b.

D'altra banda,

```
[ expressio(x) for x in D ]
[ expressio(x,y) for x in D for y in E ]
```

denota la seqüència de valors "expressio(x)" (resp. "expressió(x,y)") avaluada per tot $x \in D$ (resp. $x \in D, y \in E$).

Així mateix,

```
[ expressio(x) for x in D if condicio ]
```

denota la seqüència de valors "expressio(x)" avaluada per tot $x \in D$ tals que el booleà condició és cert. Per exemple, hauríem pogut crear `t` de la manera següent:

```
sage: t = [ n^2 for n in [-11,-9..11] if is_prime(abs(n)) ]
```

Podem aplicar una funció a tots els elements d'una llista de la manera següent:

```
sage: map(cos, [0,pi..6*pi])
[1, -1, 1, -1, 1, -1, 1]
```

Operacions

- `len(S)` (cardinal de S) per conjunts i seqüències.
- `sum(S)` (suma dels elements de S) per conjunts i seqüències.
- `prod(S)` (producte dels elements de S) per conjunts i seqüència
- `A.union(B)`, `A.intersection(B)`, `A.difference(B)` ($A \cup B$, $A \cap B$, $A \backslash B$) per conjunts.
- `S + T` (`S` concatenat amb `T`) per conjunts i seqüències.
- `min(S)`, `max(S)` per conjunts i seqüències.
- `S.append(x)` (afegir x al final de S) per seqüències.
- `S.remove(x)` (esborrar x de S) per seqüències.
- `S.index(x)` (posició de l'element x dins de S) per seqüències.

- `S.insert(i,x)` (moure una posició els elements amb índex igual o superior a i i afegir x a la posició i) per seqüències.
- `S.reverse()` (invertir) per seqüències.
- `S.sort()` (ordenar) per seqüències.
- `S[randint(0,len(S))]` (element aleatori de S) per seqüències.

Booleans

- `x in C; x not in C` per conjunts i seqüències.
- `A.issubset(B); A.issuperset(B)` per conjunts.
- `D == C`, `D != C` per conjunts i seqüències.

6.5 Funcions

La declaració general d'una funció de n arguments per la que s'hagi de fer una sèrie d'operacions és:

```
def f(x1, x2, ..):
  operacions necessaries
  return (val1,val2,..)
```

Per exemple,

```
def solucions_reals_equacio_segon_grau(a,b,c):
  discriminant=b^2-4.0*a*c
  arrel_discr=sqrt(discriminant)
  if (discriminant > 0):
    print "hi ha dues solucions reals"
    sol1=(-b+arrel_discr)/(2*a)
    sol2=(-b-arrel_discr)/(2*a)
    return (sol1,sol2)
  elif (discriminant == 0):
    print "hi ha una solucio real"
    sol=-b/(2.0*a)
    return (sol)
  else:
    print "no hi ha solucions reals"
    return ()
```

Observem que els espais són rellevants. Ara podem cridar-la:

```
sage: solucions_reals_equacio_segon_grau(1,0,-1)
hi ha dues solucions reals
(1.00000000000000, -1.00000000000000)
```

Com que en l'exemple donat la funció retorna dos valors, podem assignar-los a dues variables:

```
sage: a,b=solucions_reals_equacio_segon_grau(1,0,-1)
hi ha dues solucions reals
sage: a
1.00000000000000
sage: b
-1.00000000000000
```

Aquesta crida, però, ens donaria un error si la solució no existís o fos única.

Observem que totes les variables utilitzades són per defecte locals i que no ha calgut declarar-les. Per tant una variable externa a la funció amb nom igual a una de les variables locals no es modificarà a causa de la funció.

6.6 Enters, anells $\mathbb{Z}_n$, racionals i reals

Enters

L'anell dels enters el cridem `Integers()` A continuació llistem algunes funcions pels enters:

- Operacions bàsiques: `+, -, *, /, ^`
- `n // m` (quocient de dividir n entre m)
- `n % m` (n mòdul m)
- `divmod(a,b)` (quocient i residu de dividir a entre b)
- `abs(n)` (valor absolut)
- `randint(a,b)` (un enter aleatori entre a i b, incloent a i b)
- `random_prime(a)` (un primer aleatori menor o igual que a)
- `divisors(n), prime_divisors(n), number_of_divisors(n)`
- `gcd(m,n), gcd(S), lcm(m,n), lcm(S)` (on `S` és una seqüència d'enters)
- `xgcd(m,n)` (retorna tres valors d, a i b on d és el màxim comú divisor de m i n i on $am + bn = d$)
- `euler_phi(n)`
- `factorial(n)`

Booleans

- `is_odd(i), is_even(i)`
- `is_prime(i), is_prime_power(i)`
- `is_square(n)`

Anells $\mathbb{Z}_n$

L'anell de residus mòdul n el cridem `Integers(n)`. Si posem $R = Integers(n)$ per algun n aleshores els elements de R els cridem utilitzant $R(1)$, $R(2)$, etc. Algunes funcions que ens poden ser útils són

- Operacions bàsiques: `+, -, *, ^`
- `inverse_mod(x,m)` (invers de x mod m)
- `solve_mod(expr1 == expr2,m)` (resol equacions amb congruències; és important que les incògnites s'hagin alliberat abans, per exemple amb `var('x')`)
- `primitive_root(n)`
- `multiplicative_order(x), additive_order(x)`

Booleans

- `is_field()` per saber si un conjunt és un cos.
- `is_unit()` per saber si un element és invertible.

L'exemple següent us pot ser il·lustratiu.

```
sage: is_prime(19)
True
sage: Z19=Integers(19)
sage: [x for x in Z19]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]
sage: primitive_root(19)
2
sage: additive_order(Z19(2))
19
sage: multiplicative_order(Z19(2))
18
sage: multiplicative_order(Z19(5))
9
sage: set([2^i % 19 for i in [1..18]])
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
sage: set([5^i % 19 for i in [1..18]])
{1, 4, 5, 6, 7, 9, 11, 16, 17}
```

Racionals

El cos de les fraccions a/b es crida `RationalField()`. Algunes operacions que podem fer amb els racionals són

- Operacions bàsiques: `+, -, *, /, ^`
- `numerator(q)`
- `denominator(q);`

Reals

El cos dels reals el cridem `RealField()`. Algunes funcions pels reals són:

- Operacions bàsiques: `+, -, *, /, ^`
- `ceil(r), floor(r)`
- `abs(r)`
- `sqrt(r)`

Equacions

Es poden resoldre equacions utilitzant `solve`. Escrivint ?`solve` el sage ens dóna una explicació molt extensa. Aquí repetim els primers exemples:

```
sage: x, y = var('x, y')
sage: solve([x+y==6, x-y==4], x, y)
[[x == 5, y == 1]]
sage: solve([x^2+y^2 == 1, y^2 == x^3 + x + 1], x, y)
[[x == -1/2*I*sqrt(3) - 1/2, y == -1/2*sqrt(-I*sqrt(3) + 3)*sqrt(2)],
 [x == -1/2*I*sqrt(3) - 1/2, y == 1/2*sqrt(-I*sqrt(3) + 3)*sqrt(2)],
 [x == 1/2*I*sqrt(3) - 1/2, y == -1/2*sqrt(I*sqrt(3) + 3)*sqrt(2)],
 [x == 1/2*I*sqrt(3) - 1/2, y == 1/2*sqrt(I*sqrt(3) + 3)*sqrt(2)],
 [x == 0, y == -1],
 [x == 0, y == 1]]
sage: solve([sqrt(x) + sqrt(y) == 5, x + y == 10], x, y)
[[x == -5/2*I*sqrt(5) + 5, y == 5/2*I*sqrt(5) + 5],
[x == 5/2*I*sqrt(5) + 5, y == -5/2*I*sqrt(5) + 5]]
[x^2+y^2 == 1, y^2 == x^3 + x + 1], x, y, soluti
sage: for solution in solutions: print solution[x].n(digits=3),
 ",", solution[y].n(digits=3)
-0.500 - 0.866*I , -1.27 + 0.341*I
-0.500 - 0.866*I , 1.27 - 0.341*I
-0.500 + 0.866*I , -1.27 - 0.341*I
-0.500 + 0.866*I , 1.27 + 0.341*I
0.000 , -1.00
0.000 , 1.00
```

6.7 Anell de polinomis

Definició de polinomis

Per generar l'anell de polinomis sobre un anell `R` ho fem així:

```
sage: P.<x>=PolynomialRing(R)
```

A més, així queda definida `x` com la indeterminada dels polinomis de P. Per escriure un polinomi de `P` ho podem fer de dues maneres:

```
sage: x^3-7*x^2+5
x^3 - 7*x^2 + 5
sage: P([5,0,-7,1])
x^3 - 7*x^2 + 5
```

Operacions

- Operacions bàsiques: `+, -, *, /, ^`
- `f(a)` per avaluar un polinomi f en a
- `f.degree()`
- `f.coeffs()` retorna tots els coeficients de f mentres que `f.coefficients()` només retorna els coeficients no-nuls.
- `f.leading_coefficient(f)`
- `parent(f)` (ens diu a on pertany el polinomi f)
- `divmod(f,g)` (quocient i residu de dividir f entre g)
- `f // g, f % g`
- `gcd(f,g), xgcd(f,g), lcm(f,g)` (vegeu l'apartat d'enters)
- `factor(f)`
- `f.roots(f)`

Booleans

- `f.is_irreducible()`
- `f.is_primitive()`

Vegem un exemple:

```
sage: P.<x>=PolynomialRing(GF(49,'a'))
sage: f=4*x^5+5*x+2
sage: g=x^8+6*x^7+x^2
sage: d,a,b=XGCD(f,g)
sage: d
1
sage: d == a*f+b*g
True
```

6.8 Cossos finits

Construcció de cossos finits i extensions

Per crear els cossos finits de q o p^n elements escrivim

```
sage: K:=GF(q,'a')
```

o bé

```
sage: K.<a>=GF(q)
```

Queda així definida també a com la classe de la indeterminada dels polinomis sobre $\mathbb{Z}/(p\mathbb{Z})$. Només podem obviar la variable a quan $n = 1$.

També podem definir cossos finits forçant un determinat polinomi generador f definit dins l'anell de polinomis sobre $\mathbb{Z}/(p\mathbb{Z})$ i de grau n utilitzant

```
sage: F=GF(q, modulus=f)
```

Operacions

- Operacions bàsiques: `+, -, *, /, ^`
- `K.polynomial()`
- `minimal_polynomial(b)` equivalent a `minpoly(b)`
- `multiplicative_order(b)`
- `len(K)` equivalent a `K.cardinality()`
- `K.random_element()`

Booleans

- `f.is_primitive()`

6.9 Àlgebra lineal

Construcció de vectors

Donat un cos `K`, denotem el conjunt de vectors de K^n per `VectorSpace(K,n)`. Per crear vectors podem fer una coerció dins l'espai dels vectors corresponent o bé els podem definir directament. Per exemple,

```
sage: K=GF(9,'a')
sage: V3=VectorSpace(K,3)
sage: v=V3([1,0,1])
sage: v
(1, 0, 1)

w=vector(K,[1,2,3])
sage: v
(1, 2, 0)

sage: v[1]
0
sage: w[1]
2
```

Operacions

- Operacions bàsiques: `+, -, *`
- `v.inner_product(w)` (producte escalar)
- `v.pairwise_product(w)` (producte vectorial)

Construcció de matrius

Donat un anell `R`, denotem el conjunt de matrius de m files i n columnes `MatrixSpace(R,m,n)`. Per crear matrius podem fer una coerció dins l'espai de les matrius corresponent o bé les podem definir directament.

```
sage: F9.<alpha>=GF(9)
sage: M=MatrixSpace(F9,2,3)
sage: M([alpha,2*alpha,3*alpha,alpha,alpha^2,alpha^3])
[      alpha     2*alpha           0]
[      alpha   alpha + 1 2*alpha + 1]
sage: matrix(2,3,[alpha,2*alpha,3*alpha,alpha,alpha^2,alpha^3])
[      alpha     2*alpha           0]
[      alpha   alpha + 1 2*alpha + 1]
sage: matrix(3,2,[alpha,2*alpha,3*alpha,alpha,alpha^2,alpha^3])
[      alpha     2*alpha]
[          0       alpha]
[  alpha + 1 2*alpha + 1]
```

Si volem podem especificar l'anell sobre el que està definida una matriu. Així les dues comandes següents ens donarien matrius diferents.

```
sage: matrix(Integers(3),[[1,2],[3,4]])
[1 2]
[0 1]
sage: matrix(Integers(4),[[1,2],[3,4]])
[1 2]
[3 0]
```

Per cridar els elements d'una matriu utilitzarem `[,]` i per cridar les seves files utilitzarem `[]`. Seguint l'exemple anterior,

```
sage: m=matrix(F9,[[alpha,2*alpha,3*alpha],[alpha,alpha^2,alpha^3]])
sage: m[0,1]
2*alpha
sage: m[1]
(alpha, alpha + 1, 2*alpha + 1)
```

Per sumar, restar i multiplicar per escalars es fa amb la notació usual. Per trobar la matriu inversa d'una matriu invertible `m` escriurem

```
m^-1
```

Per trobar les solucions d'un sistema lineal $xm = v$ on m és una matriu i v és un vector tenim `m.solve_left()` mentre que per resoldre el sistema $xm = v$ tenim `m.solve_right()`.

```
sage: m=matrix(Integers(),[[0,1],[2,0]])
sage: m
[0 1]
[2 0]
sage: v=vector(Integers(),[2,2])
sage: v
```

```
(2, 2)
sage: m.solve_left(v)
(2, 1)
sage: m.solve_right(v)
(1, 2)
```

Per calcular submatrius vegem l'exemple de la guia de SAGE:

```
Take the 3 x 3 submatrix starting from entry (1,1)
in a 4 x 4 matrix:

   sage: m = matrix(4, [1..16])
   sage: m.submatrix(1, 1)
   [ 6  7  8]
   [10 11 12]
   [14 15 16]

Same thing, except take only two rows:

   sage: m.submatrix(1, 1, 2)
   [ 6  7  8]
   [10 11 12]

And now take only one column:

   sage: m.submatrix(1, 1, 2, 1)
   [ 6]
   [10]

You can take zero rows or columns if you want:

   sage: m.submatrix(1, 1, 0)
   []
```

Operacions amb matrius en general

- Operacions bàsiques: `+, -, *`
- `A.nrows(); A.ncols()`
- `m.transpose()`
- `rank(m), kernel(m), image(m)`

Booleans

- `m.is_square()`

Operacions amb matrius quadrades

- Operacions bàsiques: `+, -, *, ^`

- `m.determinant()`
- `m.trace()`
- `m^-1`

Booleans

- `m.is_symmetric()`
- `m.is_invertible()`, `m.is_singular()`
- `m.is_zero()`, `m.is_one()`

Espais vectorials

Subespais vectorials

La manera de generar un subespai vectorial d'un espai `V` és per mitjà d'un conjunt de generadors `v1, v2, ... , vn` de la manera següent:

```
sage: sub < V | v1, .. , vn >;
```

Operacions amb espais vectorials

- `BaseField(V);`
- `Dimension(V);`
- `Generators(V);`
- `V.i;` (i-èssim generador de `V`)
- `Basis(V); BasisMatrix(V);`
- `Coordinates(V,v);` (coordenades de `v` en `V`)
- `Complement(V,U);` (complement de `U` en `V`)
- `U+V;` (suma dels espais `U` i `V`)
- `U meet V;` (intersecció dels espais `U` i `V`)

Sobre la base d'un espai vectorial

En generar un subespai utilitzant `sub < V | v1, .. , vn >`, SAGE en determina una base que no és necessàriament la base `v1, .. , vn`. Per forçar una determinada base hem d'utilitzar `VectorSpaceWithBasis();` Vegem-ho en l'exemple següent:

```
sage: V3:=VectorSpace(RealField(),3);
sage: M:=KMatrixSpace(RealField(),2,3);
sage: B:=M![1,1,1,
sage:       1,0,0];
sage: VGB:=sub < V3 | B[1],B[2] >;
sage: Generators(VGB);
```

```
{
    (1 1 1),
    (1 0 0)
}
sage: Basis(VGB);
[
    (1 0 0),
    (0 1 1)
]
sage: VBB:=VectorSpaceWithBasis(B);
sage: Generators(VBB);
{
    (1 1 1),
    (1 0 0)
}
sage: Basis(VBB);
[
    (1 1 1),
    (1 0 0)
]
sage: A:=M![4,0,0,
sage:       0,1,1];
sage: VBA:=VectorSpaceWithBasis(A);
sage: Coordinates(VBA,V![2,3,3]);
[ 1/2, 3 ]
sage: Coordinates(VBB,V![2,3,3]);
[ 3, -1 ]
```

Observem que podem definir una funcio per canviar de base:

```
sage: CanviBase:=func<A,B,
      w|Coordinates(VectorSpaceWithBasis(B),w*A)>;
sage: V2:=VectorSpace(RealField(),2);
sage: CanviBase(A,B,V2![1/2,3]);
[ 3/1, -1 ]
```

Espais quocients

La manera per generar l'espai quocient d'un espai `V` respecte el subespai `U` és

```
quo < V | U >;
```

Si el subespai pel que volem fer quocient ens ve donat per un conjunt de generadors `v1, .. , vn`, aleshores

```
quo < V | v1, .. , vn >;
```

Capítol 7
Pràctiques de SAGE

7.1 Introducció a SAGE

L'objectiu d'aquesta pràctica és la familiarització amb el programa SAGE. Necessitareu construir de llistes i definir una funció.

Exercici 1: Utilitzeu comandes de SAGE per resoldre de la llista de problemes del llibre els problemes següents:

- Per les parelles de nombres següents, calculeu el seu màxim comú divisor i escriviu-lo com a combinació lineal entera dels nombres donats: $(15, 23)$, $(111, 58)$, $(220, 116)$, $(522, 371)$.
- Quantes solucions tenen aquestes equacions amb congruències? Doneu-ne les solucions.
  (a) $26x \equiv 3 \mod 13$
  (b) $13x \equiv 3 \mod 26$
  (c) $26x \equiv 0 \mod 13$
  (d) $9x \equiv -3 \mod 15$
  (e) $5x \equiv 4 \mod 7$
  (f) $5x \equiv 7 \mod 7$
- Calculeu l'invers de 13 a $\mathbb{Z}_{21}$, a $\mathbb{Z}_{22}$, i a $\mathbb{Z}_{23}$.

Exercici 2: Utilitzeu el constructor de llistes de SAGE per resoldre el següent problema:

- Trobeu els elements invertibles i els divisors de zero de $\mathbb{Z}_{14}$ i $\mathbb{Z}_{15}$.

Exercicis 3: Construïu conjunts per comprovar els resultats dels problemes següents per a un nombre gran de casos:

- Si x i y són dos enters senars, aleshores $x^2 + y^2$ no és quadrat de cap enter.
- Si p és primer, aleshores el p i $(p-1)!$ són coprimers

Exercici 4: Per a tots els codis de barres, la suma de totes les xifres en posició senar més el triple de la suma de totes les xifres en posició parell és un múltiple de 10.

(a) Comproveu-ho utilitzant anells finits per algun codi de barres que tingueu a mà.

(b) Definiu una funció que, donats tots els dígits d'un codi de barres fins al penúltim, en calculi l'últim.

7.2 Cossos finits

L'objectiu d'aquesta pràctica és la construcció i manipulació de cossos finits i polinomis.

Recordeu que per definir l'anell $P = R(x)$ de polinomis sobre un anell R utilitzem `P.<x>=PolynomialRing(R)`. Al mateix temps queda definida x com la variable indeterminada dels polinomis.

Anàlogament podem definir un cos finit $F = \mathbb{F}_{p^n}$ com `F.<alpha>=GF(`p^n`)` o bé `F=GF(`p^n`,'alpha')`. Al mateix temps queda definida *alpha* com la classe de la variable indeterminada dels polinomis sobre $\mathbb{Z}/(p\mathbb{Z})$. També podem definir cossos finits forçant un determinat polinomi generador f definit dins l'anell de polinomis sobre $\mathbb{Z}/(p\mathbb{Z})$ i de grau n utilitzant `F=GF(`q^n`, 'a', modulus=f)`.

Exercici 1: Cos finit de 27 elements

(a) Construïu el cos `F27` de 27 elements. Quin polinomi ha utilitzat SAGE per construir-lo?

(b) Genereu una taula amb la representació exponencial i polinomial de tots els elements de `F27`.

(c) Genereu una taula amb la representació exponencial i vectorial de tots els elements de `F27`.

(d) Podríem haver construït el cos de 27 elements a partir del polinomi $x^3 + x^2 + x + 2$? I amb el polinomi $x^3 + x^2 + x + 1$? Per què? ✍

(e) Si heu contestat afirmativament en algun dels casos de la pregunta anterior, construïu de nou el cos de 27 elements utilitzant ara aquest polinomi.

(f) Es pot construir una taula amb la representació exponencial i vectorial a partir de la classe de la variable indeterminada del polinomi generador? Per què? ✍

(g) Definiu una funció que permeti trobar els elements primitius d'un cos. Trobeu tots els elements primitius de `F27`.

Exercici 2: Cos finit de 32 elements

(a) Construïu l'anell $\mathbb{Z}_{32}$. Construïu una llista amb tots els seus elements invertibles.

(b) En general, quants elements invertibles té $\mathbb{Z}_m$? ✍ Comproveu-ho per l'anell de l'apartat anterior.

(c) Construïu el cos finit `F32` de 32 elements. Construïu una llista amb tots els seus elements invertibles.

(d) En general, quants elements invertibles té un cos de p^n elements, on p és primer? ✍ Comproveu-ho utilitzant l'apartat anterior.

(e) Trobeu l'ordre de tots els elements del cos `F32` diferents de zero.

(f) Segons la teoria, quin número ha de ser divisible per tots els ordres? ✍ Comproveu-ho per l'exemple de l'apartat anterior.

(g) Construïu una llista amb tots els elements primitius d'aquest cos.

(h) En general, quants elements primitius té un cos de p^n elements, on p és primer? ✍ Comproveu-ho per l'exemple de l'apartat anterior.

Exercici 3: Fixeu un primer p i un exponent m.

(a) Definiu una funció que passi una cadena de dígits de $\mathbb{Z}_p$ a una cadena de símbols de $\mathbb{F}_{p^n}$.

(b) Definiu una funció que passi una cadena de símbols de $\mathbb{F}_{p^n}$ a una cadena de dígits de $\mathbb{Z}_p$.

(c) Comproveu les funcions amb l'exemple del faisà.

7.3 Codis lineals

L'objectiu d'aquesta sessió és familiaritzar-se amb la construcció i manipulació de codis amb les funcions específiques de SAGE. Per realitzar-la cal saber els continguts de teoria. Per a les comandes pròpies dels codis podeu utilitzar el *SDSU Sage Tutorial* que trobareu a
`http://www-rohan.sdsu.edu/~mosulliv/sagetutorial/`.

Exercici 1: Un codi lineal ternari, C_t, ve donat per la matriu generadora següent:

$$G = \begin{pmatrix} 0 & 0 & 0 & 2 & 0 & 1 & 1 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 1 \\ 0 & 2 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

(a) Construïu el codi C_t.

(b) Trobeu la matriu de control de C_t.

(c) Comproveu que la paraula (1122022) pertany al codi de dues maneres, utilitzant un boolea de SAGE i utilitzant la matriu de control.

(d) Trobeu una matriu generadora sistemàtica per C_t.

(e) Codifiqueu la informació (2101) utilitzant primer la matriu generadora sistemàtica i després la matriu G. Comproveu que obtenim dues paraules diferents i comproveu que totes dues pertanyen al codi.

(f) Quina informació hem de codificar utilitzant la matriu sistemàtica perquè resulti la segona paraula? Per què?

(g) Calculeu la longitud i la dimensió d'aquest codi.

(h) Trobeu la mínima distància entre les paraules del codi. Quants errors pot corregir i quants en pot detectar? Quants esborralls pot corregir?

(i) Corregiu la paraula següent amb un error

$$(0210011).$$

Justifiqueu tots els passos.

(j) Doneu un exemple de paraula amb dos errors que es puguin detectar i un exemple de paraula amb dos errors que no es puguin detectar. Raoneu la resposta

(k) Obteniu el codi ternari les paraules $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ del qual satisfan

$$\begin{cases} 2x_4 + x_6 + x_7 = 0 \\ x_1 + 2x_2 = 0 \\ x_1 + x_2 + 2x_3 + x_6 + x_7 = 0 \\ 2x_2 + x_3 + x_5 + x_6 = 0 \end{cases}$$

Exercici 2:

(a) A $\mathbb{F}_9$ considerem el codi C amb matriu generadora

$$G = \begin{pmatrix} 1 & \alpha^3 & \alpha & \alpha^6 \\ 0 & 1 & 0 & \alpha^2 \end{pmatrix}.$$

Codifiqueu la cadena de trits 01200210. Doneu el resultat també com a cadena de trits.

(b) Doneu una matriu de control del codi.

(c) Es tracta d'un codi cíclic?

(d) Quina és la distància mínima de C?

(e) Quants errors es poden corregir en cada paraula rebuda? I quants esborralls? Quants errors es poden detectar?

(f) Corregiu els esborralls de la cadena de trits 20??0211. Doneu el resultat en trits.

Exercici 3:

(a) Definiu una funció que donada una cadena de dígits i una matriu de símbols doni la cadena de dígits resultant de codificar utilitzant la matriu.

(b) Comproveu la funció amb el problema anterior.

7.4 Codis cíclics

Exercici 1: Sigui C_c un codi cíclic de longitud 8 amb polinomi generador $g(x) = 1 + x + x^3 + 2x^4 \in \mathbb{Z}_3[x]$.

(a) Construïu el codi C_c utilitzant `CyclicCode(n,g)`. Quina és la matriu generadora utilitzada per SAGE?

(b) Codifiqueu de forma sistemàtica la informació $(2, 0, 2, 1)$ amb el codi C_c utilitzant el polinomi generador d'aquest codi.

(c) Codifiqueu de forma sistemàtica la informació $(2, 0, 2, 1)$ amb el codi C_c utilitzant una matriu generadora sistemàtica. Trobeu alguna diferència? 

(d) Calculeu el polinomi de control de C_c, $h(x)$.

(e) Trobeu la matriu de control de C_c. Quina relació té amb el polinomi de control? 

(f) Construïu el codi dual de C_c que denotem $C_c^\perp$.

(g) Doneu els paràmetres $[n, k, d]$ del codi $C_c^\perp$.

(h) Doneu el polinomi generador de $C_c^\perp$ per observació de la seva matriu generadora.

(i) Calculeu el polinomi de control de $C_c^\perp$ a partir del seu polinomi generador.

(j) Comproveu que els polinomis generadors i de control de C_c i $C_c^\perp$ divideixen el polinomi $x^8 - 1$ a $\mathbb{Z}_3[x]$.

(k) Definiu una funció amb SAGE que, donat un polinomi qualsevol, ens retorni el seu polinomi recíproc.

(l) Podeu reconèixer qui és el recíproc de $g(x)$? I el de $h(x)$?

(m) Comproveu que el vector $c = (2, 1, 2, 2, 2, 2, 0, 1)$ pertany al codi $C_c^\perp$. Diguem w a la suma de c amb l'error $e = (0, 0, 2, 0, 1, 0, 0, 0)$. Calculeu la síndrome de w. Es pot assegurar a partir de la síndrome que w conté algun error? 

(n) Descodifiqueu el vector w emprant una comanda del SAGE. Obtenim c de nou? Quin problema hi ha hagut?

Exercici 2: Fixeu un primer p, un exponent m i el polinomi generador $g(x)$ d'un codi cíclic sobre $\mathbb{F}_{p^n}$.

(a) Definiu una funció que passi una cadena de dígits de $\mathbb{Z}_p$ a una cadena de dígits de $\mathbb{Z}_p$ codificada de manera sistemàtica emprant el polinomi generador.

(b) Comproveu les funcions amb l'exemple del faisà.

7.5 Codis Reed-Solomon

L'objectiu d'aquesta sessió és construir i manipular codis Reed Solomon. Per realitzar-la cal saber els continguts de teoria. Us serà útil el constructor de codis `ReedSolomonCode(n,k,Fq)`. Perquè ens quadri amb les definicions de teoria haurem d'especificar `ReedSolomonCode(n,k,Fq,[f for f in Fq if f != Fq(0)])`.

Exercici 1: Sigui C_1 el codi Reed Solomon primitiu i en sentit estricte sobre $\mathbb{F}_{2^4}$ amb distància mínima prevista $d = 5$.

(a) Trobeu la longitud i la dimensió del codi C_1.

(b) Quina és la capacitat correctora del codi? Quants errors binaris pot corregir com a màxim aquest codi? Cada quants bits d'informació? ✍

(c) Calculeu el polinomi generador a partir de les seves arrels, sabent que la distància mínima prevista és 5. Sobre quin cos està definit el polinomi generador?

(d) Comproveu que el polinomi anterior genera C_1 com a codi cíclic.

(e) Codifiqueu de forma sistemàtica en les primeres components, utilitzant el polinomi generador, el primer missatge de la seqüència $10\alpha^7000\alpha0\alpha^701$ de símbols de $\mathbb{F}_{2^4}$, on α és un element primitiu de $\mathbb{F}_{2^4}$.

(f) Trobeu la posició i el valor dels errors del vector que té per síndrome $0\alpha^6\alpha^{14}\alpha^2$.

(g) Suposant que s'ha utilitzat una codificació sistemàtica en les primeres components, descodifiqueu la seqüència $\alpha^80\alpha^{10}\alpha^{14}\alpha^{14}00\alpha^700\alpha0\alpha^70\alpha^5$ (no podeu fer servir la comanda `decode`).

Exercici 2: Fixeu un primer p, un exponent m i una dimensió k (menor que p^m). Considereu el codi RS primitiu sobre $\mathbb{F}_{p^m}$ de dimensió k.

(a) Definiu una funció que donada una cadena de dígits codificats de $\mathbb{Z}_p$ que eventualment contingui alguns errors (cadena rebuda), retorni la cadena de dígits de $\mathbb{Z}_p$ corresponents a la paraula del codi RS sense errors (paraula enviada). Haureu d'utilitzar l'algorisme de les divisions successives.

(b) Comproveu les funcions amb l'exemple del problema 7.8.