UNIVERSITAT DE BARCELONA
Facultat de Matemàtiques i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

Criptografia postquàntica LWE

Autora: Belén Martín Collado

Director: Dr. Xavier Guitart Morales
Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 8 de juny de 2024

Abstract

The aim of this study is to establish a theoretical framework for lattice-based cryptosystems, especially those based on the Learning With Errors (LWE) problem. These post-quantum cryptosystems offer a promising alternative to those reliant on the integer factorization problem or the discrete logarithm problem, as they are thought to be secure against both classical and quantum attacks.

This project explores key algorithms such as the Lenstra–Lenstra–Lovász (LLL), which compromise the security of lattice-based cryptosystems. It also presents a practical approach to attack LWE through Kannan's embedding technique and LLL.

Resum

L'objectiu d'aquest treball és establir un marc teòric per als criptosistemes basats en reticles, especialment aquells basats en el problema Learning With Errors (LWE). Aquests criptosistemes postquàntics ofereixen una alternativa prometedora a aquells que depenen del problema de factorització d'enters o el problema del logaritme discret, ja que es consideren segurs contra atacs tant clàssics com quàntics.

El treball explora algorismes clau com el Lenstra–Lenstra–Lovász (LLL), que comprometen la seguretat dels criptosistemes basats en reticles. També presenta un enfocament pràctic per atacar el LWE a través de la tècnica d'embediment de Kannan i l'LLL.

2020 Mathematics Subject Classification. 11T71, 94A60, 68P25

Agraïments

Vull expressar el meu més sincer agraïment al meu tutor, el Dr. Xavier Guitart, per la seva dedicació i consells que han donat forma a aquest treball. Gràcies a ell, he descobert l'apassionant món de la criptografia.

També vull agrair de tot cor a la meva família la seva paciència constant i suport incondicional durant aquest viatge acadèmic.

Als meus amics, vull agrair la seva companyia, que ha omplert de joia cada repte d'aquests anys.

Índex

Introducció

La criptografia és el conjunt de tècniques matemàtiques que permet protegir la informació fent-la incomprensible per qui no en sigui el destinatari. Ja a l'antiguitat, els romans feien servir el xifratge de Cèsar, on cada lletra del text era reemplaçada per una altra que es troba un nombre fix de posicions més endavant a l'alfabet. Durant la Segona Guerra Mundial, la criptografia va jugar un paper decisiu en la victòria dels Aliats, sobretot amb el trencament de la màquina Enigma de l'exèrcit nazi per part dels criptòlegs britànics de Bletchley Park. En el nostre dia a dia, la criptografia és fonamental per compartir dades en línia de manera segura, ja sigui en el comerç digital, el correu electrònic, l'emmagatzematge al núvol o les signatures digitals.

A la dècada dels 70, Whitfield Diffie i Martin Hellman van fer un pas endavant en la criptografia moderna amb la invenció dels criptosistemes de clau asimètrica. Aquests consten de dues claus diferents: una clau pública, que pot ser compartida amb tothom, i una clau privada, que es manté secreta. A diferència dels sistemes de clau simètrica, que utilitzen la mateixa clau per xifrar i desxifrar els missatges, els sistemes de clau asimètrica utilitzen una parella de claus: la clau pública per xifrar i la clau privada per desxifrar. És molt comú en criptografia l'arquetip Alice-Bob-Eve. En l'esquema de clau asimètrica, l'Alice vol enviar un missatge al Bob sense que l'Eve, una tercera persona, el pugui interceptar. El Bob genera una clau pública i una clau privada. L'Alice fa servir la clau pública per xifrar el seu missatge, i el Bob fa servir la clau privada, només coneguda per ell, per desxifrar-lo. D'aquesta manera, encara que l'Eve intercepti el missatge xifrat, no tindrà manera de desxifrar-lo perquè no té accés a la clau secreta.

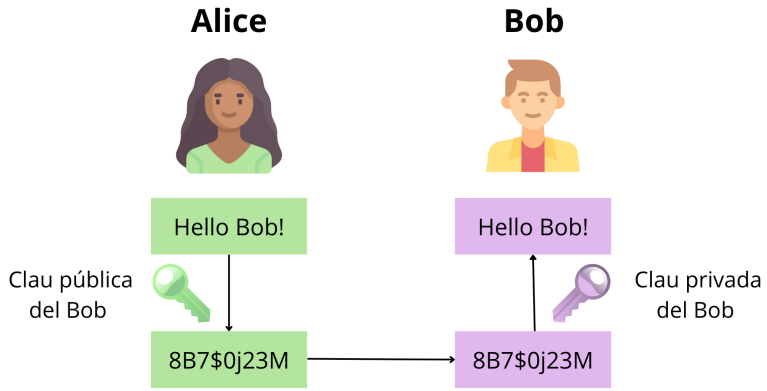


Figura 1: Diagrama Alice-Bob per il·lustrar la criptografia de clau asimètrica.

Fins ara, quasi tots els sistemes criptogràfics es basen en dos problemes matemàtics: la dificultat de factoritzar nombres enters molt grans i en el DLP (Problema del logaritme discret, o *Discrete logarithm problem* en anglès). Un dels més utilitzats és l'RSA, creat el 1979 per Rivest, Shamir i Adleman. No es coneix cap algorisme que resolgui aquests problemes en temps polinòmic en ordinadors clàssics, per la qual cosa es consideren sistemes segurs i robusts. No obstant la seva aparent robustesa, podria un ordinador quàntic trencar aquests sistemes i posar de cap per avall tot el que s'ha construït? Els ordinadors quàntics utilitzen una lògica diferent als clàssics, aprofitant propietats de la mecànica quàntica, i són més eficients que els clàssics en la resolució d'alguns problemes.

En comptes d'emprar bits (de valors 0 o 1), fan servir qubits, que són una superposició d'ambdós valors.

I, efectivament, l'any 1994 Peter Shor va presentar dos algorismes que permeten resoldre el problema de factorització d'enters i el DLP en temps polinòmic en un ordinador quàntic pràctic. De moment, els ordinadors quàntics no suposen una amenaça per a la ciberseguretat, ja que només se n'han construït prototips amb relativament poca capacitat operativa, però si s'arribessin a perfeccionar en un futur, sistemes com l'RSA quedarien obsolets, ja que es trencarien en temps polinòmic. És per això que s'han plantejat diverses alternatives que semblen resistents tant als atacs d'ordinadors clàssics com quàntics, donant lloc a l'innovador camp de la criptografia postquàntica ("postquàntica" fa referència a resistència a la computació quàntica, però se seguiria implementant en ordinadors clàssics).

L'any 1996, Ajtai i Dwork van mostrar que es pot basar la criptografia en la dificultat dels problemes de reticles. Els reticles, de manera semblant als espais vectorials, són espais n-dimensionals generats per combinacions lineals enteres de vectors de la seva base. Fins aleshores els reticles s'havien utilitzat per trencar xifratges, i no pas per xifrar. Tanmateix, es pensa que els problemes de reticles, com ara trobar vectors curts o propers, són extremadament difícils en dimensions molt grans, fins i tot per a un ordinador quàntic. Un punt del reticle representa un missatge a enviar, mentre que el mateix punt lleugerament desviat representa el missatge xifrat, que es podria desxifrar a través de la clau privada: una base "bona", és a dir, prou ortogonal.

No va ser fins el 2016 que el National Institute of Standards and Technology (NIST) va iniciar un procés per l'estandardització de criptosistemes postquàntics: dels presentats, 28 estaven basats en reticles, 24 basats en codis correctors d'errors, 13 basats en polinomis multivariats, 4 basats en les funcions de hash, i 13 en altres problemes. Nosaltres ens centrarem exclusivament en un tipus concret de sistemes basats en reticles, aquells basats en la dificultat de resoldre el problema LWE (Learning With Errors).

Ara que hem introduït el context històric, podem endinsar-nos en els continguts de la memòria. Primer, dediquem el capítol 1 a presentar les definicions i les propietats més rellevants dels reticles, que seran d'utilitat més endavant. A continuació, es descriuen alguns problemes computacionals de reticles, la dificultat dels quals és clau per a la seguretat dels criptosistemes basats en reticles: el problema del vector més curt (Shortest Vector Problem, en anglès), el problema del vector més proper (Closest Vector Problem) i algunes de les seves variants.

Al capítol 2, parlem de l'algorisme LLL, l'algorisme més important de simplificació de reticles. Aquest algorisme proporciona una base amb vectors més ortogonals entre si i més curts a partir d'una base donada d'un reticle. També permet quasi resoldre alguns problemes de reticles, ja que proporciona un vector proper a un vector més curt del reticle. Quant més gran sigui la dimensió del reticle, pitjor serà l'aproximació obtinguda. D'altra banda, parlem de l'algorisme de Babai, un algorisme molt senzill capaç de resoldre el Closest Vector Problem només si la base és suficientment ortogonal. Ambdós algorismes, sobretot l'LLL, suposen una amenaça per a la seguretat dels criptosistemes basats en reticles. Per últim, introduïm la tècnica d'encabiment de Kannan per reduir el Closest Vector Problem al Shortest Vector Problem, la qual farem servir a la part pràctica del treball per fer un atac al criptosistema basat en LWE.

Tot seguit, al capítol 3 presentem el tema clau del treball: el problema LWE (Learning With Errors). Suposem que tenim un sistema d'equacions lineals de la forma $As + e = b$,

on A és una matriu coneguda, s és un vector secret, e és un vector d'errors i b és un vector conegut. En absència d'errors, el sistema es redueix a $As = b$, que es pot resoldre fàcilment amb el mètode de Gauss-Jordan. Però introduir errors a través del vector e converteix trobar s en un repte computacionalment difícil. Els criptosistemes basats en LWE utilitzen aquesta dificultat per xifrar missatges. Sovint es treballa amb valors en l'anell d'enters mòdul q, $\mathbb{Z}/q\mathbb{Z}$. L'any 2005, Oded Regev va introduir un criptosistema basat en LWE, àmpliament estudiat per tal de garantir-ne la seguretat i l'eficiència. Nosaltres n'expliquem el funcionament i en mostrem els resultats més rellevants sobre la minimització dels errors de desxifratge i sobre com fer una bona tria de paràmetres. També desenvolupem en detall la dificultat del LWE i un resultat interessant: ambdues versions del LWE (cerca i decisió) són equivalents. Per últim, expliquem una generalització del criptosistema Regev i el criptosistema Brakerski-Vaikuntanathan (BV), tots dos basats en LWE.

Finalment, dediquem el capítol 4 a la part pràctica de la memòria. Per acabar d'enllaçar el problema LWE i els reticles, donem un enfocament pràctic en forma d'un atac al LWE. Programant un algorisme a SageMath que fa servir la tècnica d'encabiment de Kannan i l'algorisme LLL, aconseguim trobar la clau secreta s per a reticles de dimensions n diverses, arribant a resoldre'l per a $n = 28$.

1 Reticles en criptografia

1.1 Definició i propietats dels reticles

Definició 1.1. *Siguin $v_1, \ldots, v_n \in \mathbb{R}^n$ vectors linealment independents. El reticle R generat per $v_1, \ldots, v_n$ és el conjunt de combinacions lineals de $v_1, \ldots, v_n$ amb coeficients enters,*

$$R = \{a_1 v_1 + a_2 v_2 + \ldots + a_n v_n \mid a_1, a_2, \ldots, a_n \in \mathbb{Z}\}.$$

Els vectors $v_1, \ldots, v_n$ conformen la base del reticle. La dimensió de R és el nombre de vectors en una base de R, que serà el mateix per qualssevol dues bases.

Observació 1.2. Un reticle R admet més d'una base. A més, si $v_1, \ldots, v_n$ formen una base de R, és útil escriure els $v_i = (v_{i1}, \ldots, v_{in})$ com a files d'una matriu $A_{n\times n}(\mathbb{Z})$:

$$A = \begin{pmatrix} v_{11} & v_{12} & \ldots & v_{1n} \\ v_{21} & v_{22} & \ldots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{n1} & v_{n2} & \ldots & v_{nn} \end{pmatrix}.$$

Es pot obtenir una nova base fent el producte UA, on U és una matriu $n \times n$ amb coeficients enters i determinant ± 1. El conjunt de les matrius U es diu el grup lineal general sobre $\mathbb{Z}$ i es denota per $GL_n(\mathbb{Z})$.

Similarment a un espai vectorial, un reticle és generat per les combinacions lineals dels vectors de la seva base, amb la diferència que utilitza coeficients enters en comptes de reals. El reticle R es pot visualitzar com un conjunt de punts en un espai n-dimensional disposats de manera periòdica, de manera que posem un punt al final de cada vector. Per això, a l'hora de trobar un vector del reticle, podem considerar que volem trobar un punt del reticle.

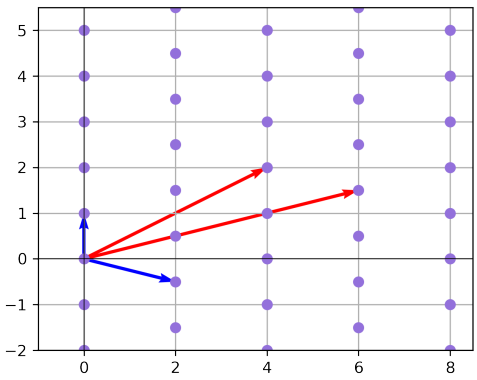


Figura 2: Un reticle $R \subset \mathbb{R}^2$ amb dues possibles bases: l'una prou ortogonal (en blau) i l'altra poc ortogonal (en vermell). A les bases prou ortogonals se les denomina bases "bones", mentre que a les poc ortogonals se les denomina bases "dolentes". Més endavant donarem criteris per quantificar l'ortogonalitat de les bases.

Definició 1.3. *Sigui R un reticle de dimensió n i $v_1, .., v_n$ una base de R. El domini fonamental de R per a aquesta base és el conjunt*

$$\mathcal{F}(v_1, ..., v_n) = \{t_1v_1 + t_2v_2 + ... + t_nv_n \mid 0 \leq t_i < 1\}.$$

Proposició 1.4. *Sigui $R \subset \mathbb{R}^n$ un reticle de dimensió n i $\mathcal{F}$ un domini fonamental de R. Per a qualsevol vector $w \in \mathbb{R}^n$ existeixen $t \in \mathcal{F}$ i $v \in R$ únics tals que*

$$w = t + v.$$

Equivalentment, la unió dels dominis fonamentals traslladats

$$\mathcal{F} + v = \{t + v \mid t \in \mathcal{F}\}$$

mentre v varia sobre els vectors del reticle R cobreix tot $\mathbb{R}^n$.

Demostració. Considerem $v_1, \ldots, v_n$ una base de R que dona el domini fonamental $\mathcal{F}$. Aleshores, $v_1, \ldots, v_n$ són linealment independents a $\mathbb{R}^n$, per la qual cosa formen una base de $\mathbb{R}^n$. Per tant, qualsevol $w \in \mathbb{R}^n$ es pot escriure de la forma

$$w = \lambda_1v_1 + \lambda_2v_2 + \ldots + \lambda_nv_n \text{ per a certs } \lambda_1, \ldots, \lambda_n \in \mathbb{R}.$$

Ara, és possible escriure cada λ_i, $i = 1, ..., n$, com

$$\lambda_i = t_i + a_i, \text{ amb } 0 \leq t_i < 1, \ a_i \in \mathbb{Z}.$$

Llavors

$$w = (t_1+a_1)v_1+(t_2+a_2)v_2+\ldots+(t_n+a_n)v_n = (t_1v_1+t_2v_2+\ldots+t_nv_n)+(a_1v_1+a_2v_2+\ldots+a_nv_n),$$

on $t \coloneqq t_1v_1 + t_2v_2 + \ldots + t_nv_n \in \mathcal{F}$ i $v \coloneqq a_1v_1 + a_2v_2 + \ldots + a_nv_n \in R$.

Hem demostrat que w es pot escriure de la forma desitjada.

Suposem ara que w té dues representacions possibles com a suma d'un vector de $\mathcal{F}$ i d'un vector de R: $w = t + v = t' + v'$. Aleshores

$$(t_1 + a_1)v_1 + (t_2 + a_2)v_2 + \ldots + (t_n + a_n)v_n = (t'_1 + a'_1)v_1 + (t'_2 +' a_2)v_2 + \ldots + (t'_n + a'_n)v_n.$$

Com $v_1, \ldots, v_n$ són linealment independents, només pot ser

$$t_i + a_i = t'_i + a'_i \ \ \forall i = 1, \ldots, n$$

i d'aquí obtenim que

$$t_i - t'_i = a'_i - a_i \in \mathbb{Z}.$$

Però com que sabem que $0 \leq t_i < 1$ i $0 \leq t'_i < 1$, l'única manera que la quantitat $t_i - t'_i$ sigui entera és si $t_i = t'_i$. Així doncs, $t = t'$, i també $v = w - t = w - t' = v'$.

Amb això ja hem demostrat que l'escriptura de w com a suma d'un vector de $\mathcal{F}$ i d'un vector de R és única. □

Tots els dominis fonamentals de R tenen el mateix volum. Resulta que el volum d'un domini fonamental és un invariant importantíssim del reticle.

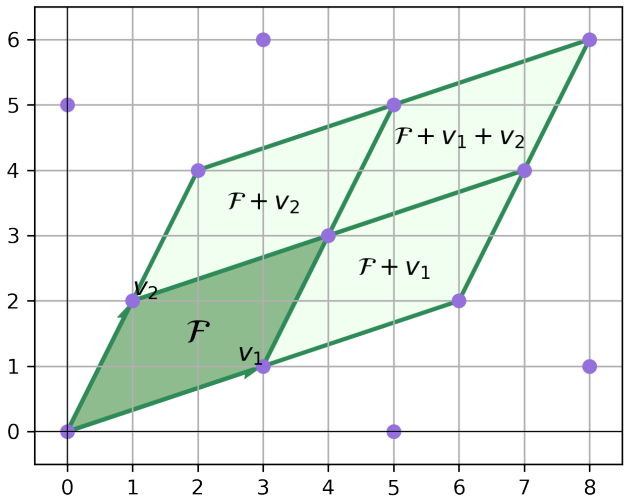


Figura 3: Les translacions de $\mathcal{F}$ per vectors de R cobreixen $\mathbb{R}^n$.

Definició 1.5. *Sigui R un reticle de dimensió n i $\mathcal{F}$ un domini fonamental de R. El volum n-dimensional de $\mathcal{F}$ es diu el determinant de R i es denota per* $\det(R)$.

Proposició 1.6. *El determinant d'un reticle R té les següents propietats:*

1. *Es pot calcular com*
$$\det(R) = |\det(v_1, \ldots, v_n)|$$
per a qualsevol base $v_1, \ldots, v_n$ de R.

2. *No depèn de la base escollida.*

Demostració. 1. Siguin R un reticle, $v_1, \ldots, v_n$ una base de R i $\mathcal{F} = \mathcal{F}(v_1, \ldots, v_n)$ el domini fonamental associat. N'hi ha prou amb demostrar que $\text{Vol}(\mathcal{F}) = |\det(v_1, \ldots, v_n)|$, ja que per definició $\det(R) = \text{Vol}(\mathcal{F})$. Primer escrivim els $v_i = (v_{i1}, \ldots, v_{in})$, $i = 1, \ldots, n$, com a files d'una matriu A associada a R,

$$A = \begin{pmatrix} v_{11} & v_{12} & \ldots & v_{1n} \\ v_{21} & v_{22} & \ldots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{n1} & v_{n2} & \ldots & v_{nn} \end{pmatrix}.$$

Podem calcular el volum de $\mathcal{F}$ com la integral de la funció constant 1 sobre la regió $\mathcal{F}$,

$$\text{Vol}(\mathcal{F}) = \int_{\mathcal{F}} dx_1 dx_2 \ldots dx_n.$$

Fixant-nos en la Definició 1.3, realitzem el canvi de variables de $x = (x_1, x_2, \ldots, x_n) \in \mathcal{F}$ a $t = (t_1, t_2, \ldots, t_n) \in [0,1]^n$ tal que

$$(x_1, x_2, \ldots, x_n) = t_1 v_1 + t_2 v_2 + \ldots + t_n v_n.$$

El canvi es pot expressar amb l'equació matricial $x = tA$, ja que efectivament

$$\begin{aligned} x = tA &= (t_1 v_{11} + \ldots + t_n v_{n1}, t_1 v_{12} + \ldots + t_n v_{n2}, \ldots, t_1 v_{1n} + \ldots + t_n v_{nn}) = \\ &= t_1 v_1 + \ldots + t_n v_n. \end{aligned}$$

Ara, la fórmula de canvi de variables per a una integral n-dimensional definida sobre una regió $\mathcal{F}$ ens diu que

$$\int_{\mathcal{F}} dx_1 dx_2 \ldots dx_n = \int_{\mathcal{G}} |\det(A)| dt_1 dt_2 \ldots dt_n,$$

on $\mathcal{G}$ és la imatge de la regió $\mathcal{F}$ sota la transformació de x a t. Per determinar la nova regió $\mathcal{G}$, observem que $\mathcal{F}$ és la imatge del cub unitari $C_n = [0,1]^n$ sota A, és a dir,

$$\begin{gathered} A : C_n \to \mathcal{F}, \\ (t_1, t_2, \ldots, t_n) \mapsto (x_1, x_2, \ldots, x_n) = t_1 v_1 + t_2 v_2 + \ldots + t_n v_n. \end{gathered}$$

Per tant, tenim $\mathcal{F} = A([0,1]^n) = AC_n$.

Quan es realitza el canvi de variables de x a t, s'està portant la regió $\mathcal{F}$ novament al cub unitari C_n. Això s'aconsegueix multiplicant per A^{-1}. Per tant, ja tenim que $\mathcal{G} = A^{-1}(AC_n) = C_n$. Finalment, ens queda el que volíem,

$$\mathrm{Vol}(\mathcal{F}) = \int_{\mathcal{F}} dx_1 dx_2 \ldots dx_n = \int_{C_n} |\det(A)| dt_1 dt_2 \ldots dt_n = |\det A| \mathrm{Vol}(C_n) = |\det A|,$$

ja que C_n té volum 1 en qualsevol dimensió.

2. Considerem dues bases qualssevol $v_1, \ldots, v_n$ i $v_1', \ldots, v_n'$ d'un mateix reticle R, representades per matrius A i A', respectivament. Sabem que ambdues bases estan relacionades per una matriu $U_{n\times n}(\mathbb{Z})$ i determinant ± 1 tal que $A' = UA$, per tant

$$|\det(A')| = |\det(UA)| = |\det(U)||\det(A)| = |\det(A)|,$$

i ja tenim que $\det(R)$ és un invariant del reticle R.

□

Podem considerar la base $v_1, \ldots, v_n$ com els vectors que delimiten els costats del paral·lepípede $\mathcal{F}$. Aleshores el volum màxim s'assoleix quan els vectors són ortogonals. Això ens condueix al següent resultat sobre la cota superior del determinant d'un reticle.

Teorema 1.7. *(Desigualtat de Hadamard). Sigui R un reticle, sigui $v_1, \ldots, v_n$ una base qualsevol de R i sigui $\mathcal{F}$ un domini fonamental de R. Aleshores*

$$\det(R) = \mathrm{Vol}(\mathcal{F}) \leq \|v_1\| \ldots \|v_n\|.$$

Quant més ortogonal és la base, més s'apropa la desigualtat de Hadamard a ser una igualtat.

Observació 1.8. Si $v_1, \ldots, v_n$ és una base ortogonal d'un reticle R, aleshores

$$\det(R) = \|v_1\| \ldots \|v_n\|.$$

Demostració. Sigui $v_1, \ldots, v_n$ una base ortogonal del reticle R. Si escrivim $v_i = (v_{i1}, \ldots, v_{in})$, $i = 1, \ldots, n$, i utilitzem aquestes coordenades de v_i com a files de la matriu A associada a R, aleshores

$$A \cdot A^T = \begin{pmatrix} v_{11} & v_{12} & \ldots & v_{1n} \\ v_{21} & v_{22} & \ldots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{n1} & v_{n2} & \ldots & v_{nn} \end{pmatrix} \cdot \begin{pmatrix} v_{11} & v_{21} & \ldots & v_{n1} \\ v_{12} & v_{22} & \ldots & v_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ v_{1n} & v_{2n} & \ldots & v_{nn} \end{pmatrix} =$$

$$= \begin{pmatrix} \|v_1\|^2 & \sum_{j=1}^n v_{1j}v_{2j} & \ldots & \sum_{j=1}^n v_{1j}v_{nj} \\ \sum_{j=1}^n v_{2j}v_{1j} & \|v_2\|^2 & \ldots & \sum_{j=1}^n v_{2j}v_{nj} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{j=1}^n v_{nj}v_{1j} & \sum_{j=1}^n v_{nj}v_{2j} & \ldots & \|v_n\|^2 \end{pmatrix} =$$

$$= \begin{pmatrix} \|v_1\|^2 & 0 & \ldots & 0 \\ 0 & \|v_2\|^2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & \|v_n\|^2 \end{pmatrix},$$

on a la darrera igualtat hem utilitzat que $v_1, \ldots, v_n$ són vectors ortogonals. Si prenem determinants a ambdues bandes, tenim que

$$\det(A \cdot A^T) = \|v_1\|^2 \|v_2\|^2 \ldots \|v_n\|^2.$$

D'altra banda,

$$\det(A \cdot A^T) = \det(A) \cdot \det(A^T) = (\det(A))^2.$$

Combinant ambdues igualtats, ja hem demostrat el que volíem.

□

Un cop hem vist el concepte de determinant d'un reticle, podem introduir un criteri per quantificar l'ortogonalitat de les bases d'un reticle: el quocient de Hadamard.

Definició 1.9. *Sigui R un reticle i sigui $v_1, \ldots, v_n$ una base de R. El quocient de Hadamard de la base $v_1, \ldots, v_n$ es defineix com la quantitat*

$$\mathcal{H}(v_1, \ldots, v_n) = \left(\frac{\det(R)}{\|v_1\| \ldots \|v_n\|} \right)^{1/n}.$$

Es té que $0 < \mathcal{H}(v_1, \ldots, v_n) \leq 1$, i quant més s'apropa el quocient a 1, més ortogonal és la base.

Exemple 1.10. Considerem el reticle tridimensional $R \subset \mathbb{R}^3$ i una base $\mathcal{B}$ de R, formada pels vectors

$$v_1 = (1, 0, 0), \quad v_2 = (0, 1, 0), \quad v_3 = (0, 0, 1).$$

Escrivim la base $\mathcal{B}$ com les files d'una matriu 3×3:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I_3.$$

Aquest reticle R té determinant

$$\det(R) = |\det(A)| = 1.$$

D'altra banda,

$$\|v_1\|\|v_2\|\|v_3\| = 1,$$

de manera que es compleix la desigualtat de Hadamard. Com era d'esperar, la base és ortogonal, ja que el quocient de Hadamard ens queda

$$\mathcal{H}(v_1, v_2, v_3) = \left(\frac{1}{1}\right)^{1/3} = 1.$$

Una altra possible base $\mathcal{B}'$ del reticle R estarà formada pels vectors

$$v_1' = (1, 0, 0), \quad v_2' = (4, 1, 1), \quad v_3' = (2, 1, 0).$$

De manera anàloga, considerem la matriu

$$A' = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 1 \\ 2 & 1 & 0 \end{pmatrix}.$$

Efectivament, $\mathcal{B}'$ és base del mateix reticle ja que $A' = UA = UI_3 = U$: la matriu de transformació és la pròpia matriu A', que té coeficients enters i determinant 1.

D'altra banda,

$$\|v_1'\|\|v_2'\|\|v_3'\| = \sqrt{90},$$

per la qual cosa tenim que

$$\mathcal{H}(v_1', v_2', v_3') = \left(\frac{1}{\sqrt{90}}\right)^{1/3} \approx 0.47,$$

i deduïm que la base $\mathcal{B}'$ és notablement menys ortogonal que la base $\mathcal{B}$.

A l'hora de representar el reticle R, diríem que $\mathcal{B}$ és una base "bona", mentre que $\mathcal{B}'$ és una base "dolenta".

1.2 Problemes de reticles

Ara ens endinsarem en alguns problemes computacionals de reticles, la dificultat dels quals ha permès crear sistemes criptogràfics. L'aplicació dels reticles a la criptografia no és gens intuïtiva, i va ser descoberta per Ajtai en [1]. Comencem explicant els dos problemes més senzills: el Shortest Vector Problem (SVP), o problema del vector més curt, en català, i el Closest Vector Problem (CVP), o problema del vector més proper. Tots els problemes es poden definir per a qualsevol norma, tot i que la més utilitzada és la L_2, és a dir, la norma euclidiana: si tenim un vector x amb coordenades $x = (x_1, \ldots, x_n)$, aleshores la seva norma euclidiana és $\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$.

1.2.1 Shortest Vector Problem

Donat un reticle R, denotem el mínim de les normes dels vectors no nuls de R com $\lambda_1(R)$.

Problema 1.11. (Shortest Vector Problem). Donat un reticle R de dimensió n donat per una base $v_1, \ldots, v_n$, trobar un vector no nul $v \in R$ més curt, és a dir, tal que

$$\|v\| = \lambda_1(R).$$

Observació 1.12. Pot haver-hi més d'un vector solució de l'SVP. Si u n'és una solució, llavors $-u$ també ho és, però poden haver-hi més vectors a R amb la mateixa norma que u i $-u$ que siguin linealment independents d'u. Per exemple, si $R = \mathbb{Z}^2$, els quatre vectors $(0, \pm 1)$ i $(\pm 1, 0)$ són solucions de l'SVP.

Sovint és suficient considerar la versió aproximada de l'SVP (apprSVP):

Problema 1.13. (Approximate Shortest Vector Problem). Donats un reticle R de dimensió n donat per una base $v_1, \ldots, v_n$ i un factor d'aproximació $\gamma(n)$, trobar un vector $v \in R$ de norma menor o igual que la norma del vector més curt que denotem per $\lambda_1(R)$, multiplicada per $\gamma(n)$:

$$\|v\| \leq \gamma(n)\lambda_1(R).$$

Per tant, no es troba el vector més curt, sinó un de proper que considerarem prou bo depenent de la funció $\gamma(n)$ escollida. Per exemple, un algorisme que utilitzi $\gamma(n) = 3\sqrt{n}$ (en general, $\gamma(n) = \text{poly}(n)$) per resoldre la versió aproximada de l'SVP es considera més potent que un que utilitzi $\gamma(n) = 2^{n/2}$ (exponencial).

Una altra versió de l'SVP és el Unique Shortest Vector Problem (uSVP), o problema del vector més curt únic, que es va popularitzar a partir de la seva implementació als primers criptosistemes d'Ajtai i Dwork [2]. Denotem per $\lambda_2(R)$ la norma del segon vector més curt, que ha de ser linealment independent dels primers. No es tracta realment d'un problema diferent, sinó que canvien les condicions sobre el reticle imposant que hi hagi una bretxa de separació entre les normes del primer vector més curt i el segon:

Problema 1.14. (Unique SVP Problem). Donat un reticle R de dimensió n tal que $\lambda_1(R) \ll \gamma(n)\lambda_2(R)$, on $\gamma(n) = n^{\mathcal{O}(1)}$, trobar un vector no nul $v \in R$ més curt, és a dir, amb $\|v\| = \lambda_1(R)$.

Quant més gran és γ, més fàcil és trobar el vector més curt.

Un altre problema sovint esmentat és el Shortest Independent Vectors Problem (SIVP):

Problema 1.15. (Shortest Independent Vectors Problem). Donat un reticle R de dimensió n donat per una base $\mathcal{B} \in \mathbb{Z}^{n \times n}$, trobar n vectors linealment independents de R, $w_1, \ldots, w_n$, tals que minimitzin la quantitat $\|W\| = \max_i \|w_i\|$.

1.2.2 Closest Vector Problem

Tot seguit introduïm un problema igual d'important, que es basa en trobar un vector del reticle que sigui el més proper a un vector qualsevol de l'espai. En dimensions petites sembla senzill, però en dimensions grans esdevé un problema particularment complex des del punt de vista computacional.

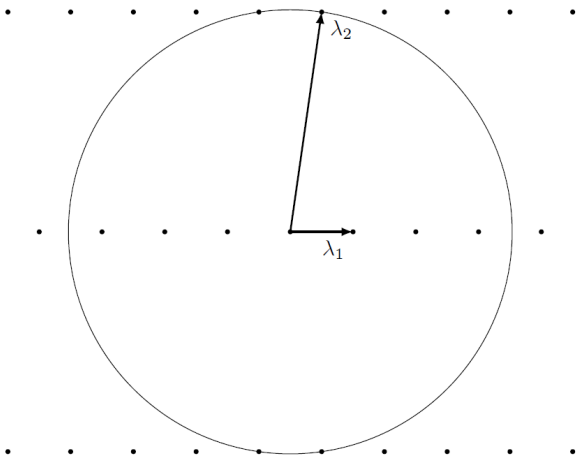


Figura 4: Un reticle R amb $\lambda_1(R) \ll \lambda_2(R)$. Imatge de [3].

Problema 1.16. (Closest Vector Problem). Donats un reticle R de dimensió n donat per una base $v_1, \ldots, v_n$ i un vector objectiu $w \in \mathbb{R}^n$ no necessàriament a R, trobar un vector $v \in R$ que sigui el més proper a w, és a dir, que minimitzi la norma euclidiana $\|w - v\|$:

$$\|w - v\| = \min\{\|w - v\| \mid v \in R\}.$$

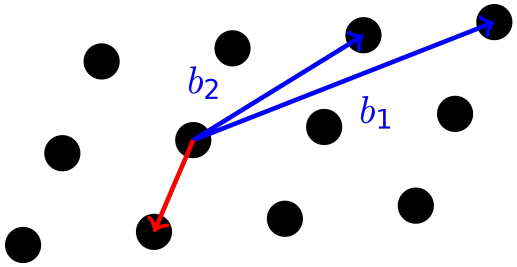
Anàlogament a l'SVP, existeix la versió aproximada del CVP (apprCVP), on busquem un vector proper a la solució del CVP:

Problema 1.17. (Approximate Closest Vector Problem). Donats un reticle R de dimensió n donat per una base $v_1, \ldots, v_n$ i un factor d'aproximació $\gamma(n)$, trobar un vector $v \in R$ tal que

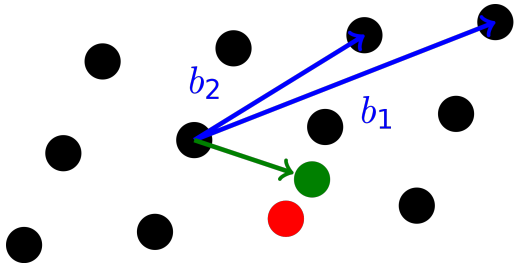
$$\|w - v\| < \gamma(n) \min\{\|w - v\| \mid v \in R\}.$$

En general, el CVP es considera un problema $\mathcal{NP}$-difícil[2] i l'SVP es considera $\mathcal{NP}$-difícil sota una certa "hipòtesi de reducció aleatòria". Ambdós problemes esdevenen computacionalment complexos quan la dimensió n del reticle creix. En la pràctica, el CVP es considera una mica més difícil que l'SVP, ja que el primer es pot reduir sovint a l'SVP en una dimensió lleugerament més gran [4]. D'altra banda, l'apprSVP es pot reduir en temps polinòmic a l'apprCVP (si obtenim solucions a l'apprCVP, aleshores n'obtenim de l'apprSVP en temps polinòmic), conservant el mateix factor d'aproximació i la dimensió del reticle. Se'n pot veure la demostració a [5]. En definitiva, l'SVP no és més complex que el CVP.

[2]La classe de complexitat $\mathcal{NP}$-difícil es defineix com el conjunt dels problemes de decisió tals que si H és un problema d'aquesta classe, tot problema de $\mathcal{NP}$ es pot transformar en H en temps polinòmic. Els problemes de decisió són aquells problemes en un sistema formal que tenen una resposta de sí o no. Per a més informació sobre la teoria de complexitat, podeu consultar la Secció 4.7 de [4].



(a) Representació del SVP: els vectors de la base en blau i el vector del reticle més curt en vermell.



(b) Representació del CVP: els vectors de la base en blau, el vector objectiu en verd i el vector del reticle més proper en vermell.

Figura 5: Comparació entre el SVP i el CVP. Imatges de [6].

2 Algorismes amb reticles

2.1 L'algorisme LLL

L'algorisme LLL és un algorisme de simplificació de reticles, és a dir, "millora"els vectors d'una base donada, fent-los més ortogonals entre si amb normes relativament més curtes, i retorna el que es coneix com a base LLL-reduïda.

Desenvolupat el 1982 per Lenstra, Lenstra i Lovász [7], és probablement l'algorisme més conegut per a problemes de reticles. Es tracta d'un algorisme de temps polinòmic (en funció del rang del reticle) que proporciona una base reduïda del reticle. Com a efecte colateral, resol les versions aproximades de l'SVP (apprSVP) i del CVP (apprCVP) amb un factor d'aproximació $\gamma(n) = 2^{\frac{n-1}{2}}$, on n és la dimensió del reticle [3, 4].

Així, en dimensions petites, l'algorisme LLL gairebé resol l'SVP i el CVP, però en dimensions grans no ho fa tan bé. Per això, la seguretat dels criptosistemes basats en reticles depèn de la incapacitat de l'LLL i d'altres algorismes per resoldre eficientment l'apprSVP i l'apprCVP amb un factor $\text{poly}(n)$.

Deixant de banda la versió aproximada, és possible obtenir-ne una solució exacta, o potser només una aproximació en factors $\text{poly}(n)$? La resposta és que l'algorisme més eficient conegut té un temps d'execució de $2^{\mathcal{O}(n)}$ [8] i requereix un espai també exponencial, cosa que el fa impracticable. De fet, l'any 2009, Daniele Micciancio i Panagiotis Voulgaris [9] van presentar un algorisme per resoldre l'SVP en temps $2^{3.199n}$ i espai $2^{1.325n}$. Existeixen altres algorismes que requereixen només espai polinòmic, però tenen un temps d'execució de $2^{\mathcal{O}(n \log n)}$ [10].

Aquest debat ens porta a la següent conjectura.

Conjectura 1. *No existeix cap algorisme de temps polinòmic que aproximi els problemes de reticles en factors polinòmics.*

De totes maneres, centrem-nos en com minimitzar la norma d'un vector. Si $v_1, \ldots, v_n$ és una base ortogonal d'un reticle R i $v = a_1 v_1 + \ldots + a_n v_n$ és un vector qualsevol de R, amb $a_1, \ldots, a_n \in \mathbb{Z}$, aleshores

$$\|v\|^2 = \|a_1 v_1 + \ldots + a_n v_n\|^2 = (a_1 v_1 + \ldots + a_n v_n) \cdot (a_1 v_1 + \ldots + a_n v_n) =$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j (v_i \cdot v_j) = \sum_{i=1}^{n} a_i^2 \|v_i\|^2, \quad \text{ja que } v_i \cdot v_j = 0 \text{ per a } i \neq j,$$

i el vector més curt de R serà simplement el vector més curt de la base. Si la base és ortonormal, aleshores l'expressió se simplifica a $\|v\|^2 = \sum a_i^2$.

Hi ha una variant de l'algorisme de Gram-Schmidt que ens permet crear una base ortogonal de $\mathbb{R}^n$ (però no del reticle, com veurem a continuació).

Suposem que $\mathcal{B} = \{v_1, v_2, \ldots, v_n\}$ és una base d'un reticle R. Volem transformar-la en una base millor, que consistiria en vectors el més curts possible, o el més ortogonals possible per tal que els productes escalars $v_i \cdot v_j$ siguin molt propers al zero. El primer pas seria construir una base ortogonal de Gram-Schmidt.

Definició 2.1. *Sigui $\mathcal{B} = \{v_1, v_2, \ldots, v_n\}$ una base d'un reticle R. La seva base ortogonal associada de Gram-Schmidt és defineix com*

$$\mathcal{B}^* = \{v_1^*, v_2^*, \ldots, v_n^*\}$$

i es construeix fent

$$v_1^* = v_1,$$
$$v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j^*, \quad \text{on} \;\; \mu_{i,j} = \frac{v_i \cdot v_j^*}{\|v_j^*\|^2} \quad \text{per a} \;\; 1 \leq j \leq i-1, \;\; i \geq 2.$$

Algorisme 1 Algorisme de Gram-Schmidt

```
Introduir una base v_1, ..., v_n
Posar v_1^* = v_1
for i = 2, 3, ..., n do
    Posar sumatori = 0
    for j = 1, ..., i - 1 do
        Posar μ_{i,j} = (v_k · v_j^*) / ||v_j^*||^2
        sumatori = sumatori + μ_{i,j} v_j^*
    end for
    v_i^* = v_i - sumatori
end for
Retornar la base de Gram-Schmidt {v_1^*, ..., v_n^*}
```

Per tant, $\mathcal{B}^* = \{v_1^*, v_2^*, \ldots, v_n^*\}$ és una base ortogonal del mateix espai vectorial generat per $\mathcal{B} = \{v_1, v_2, \ldots, v_n\}$, però no és una base del reticle R generat per $\mathcal{B}$, ja que l'algorisme de Gram-Schmidt pot generar vectors amb coeficients no enters. Tot i això, es pot demostrar que ambdues bases tenen el mateix determinant.

Proposició 2.2. *Sigui $\mathcal{B} = \{v_1, v_2, \ldots, v_n\}$ una base d'un reticle R i $\mathcal{B}^* = \{v_1^*, v_2^*, \ldots, v_n^*\}$ la base ortogonal associada de Gram-Schmidt. Aleshores*

$$\det(R) = \prod_{i=1}^{n} \|v_i^*\|.$$

Demostració. Veure l'observació de la pàgina 36 de [3]. □

D'altra banda, hi ha un algorisme gaussià per trobar una base "bona"per a un reticle de dimensió 2, proporcionant així una manera eficient de resoldre l'SVP en $\mathbb{R}^2$. Si generalitzem a reticles de $\mathbb{R}^n$, l'SVP esdevé més difícil i llavors l'algorisme LLL ens és de gran utilitat.

Definició 2.3. *(Reducció LLL). Sigui $\mathcal{B} = \{v_1, v_2, \ldots, v_n\}$ una base d'un reticle R i $\mathcal{B}^* = \{v_1^*, v_2^*, \ldots, v_n^*\}$ la base ortogonal associada de Gram-Schmidt. La base $\mathcal{B}$ es diu LLL-reduïda si satisfà les dues condicions següents:*

$$\textbf{\textit{(Condició de mida)}} \quad |\mu_{i,j}| = \frac{|v_i \cdot v_j^*|}{\|v_j^*\|^2} \leq \frac{1}{2} \quad \forall\, 1 \leq j < i \leq n.$$

$$\textbf{\textit{(Condició de Lovász)}} \quad \|v_i^*\|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \cdot \|v_{i-1}^*\|^2 \quad \forall\, 1 < i \leq n.$$

L'elecció del factor $\delta = 3/4$ a la condició de Lovász és per convenció. En realitat, si fem servir $\delta \in (1/4, 1]$, s'obtindrà una base LLL-reduïda igualment. Malauradament, per a $\delta = 1$ no es pot assegurar que l'algorisme acabi en temps polinòmic [4].

Una base d'aquest estil té propietats convenients que la converteixen en una base "bona":

Teorema 2.4. *Sigui R un reticle de dimensió n. Qualsevol base LLL-reduïda $v_1, \ldots, v_n$ de R té les següents propietats:*

1. $\prod_{i=1}^{n} \|v_i\| \leq 2^{n(n-1)/4} \det(R)$,
2. $\|v_j\| \leq 2^{(i-1)/2} \|v_i^*\| \quad \forall\, 1 \leq j \leq i \leq n.$

A més, el primer vector de la base LLL-reduïda satisfà

3. $\|v_1\| \leq 2^{(n-1)/4} |\det(R)|^{1/n}$,
4. $\|v_1\| \leq 2^{(n-1)/2} \min_{0 \neq v \in R} \|v\|$.

Per tant, l'algorisme LLL resol l'apprSVP amb un factor $\gamma(n) = 2^{(n-1)/2}$, on la solució trobada és el vector v_1.

Demostració. Tenint en compte la condició de Lovász i que $|\mu_{i,i-1}| \leq \frac{1}{2}$, obtenim que

$$\|v_i^*\| \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \|v_{i-1}^*\| \geq \frac{1}{2} \|v_{i-1}^*\|.$$

Si apliquem iterativament la desigualtat anterior, arribem a la següent cota superior per a la norma de v_j^* respecte de v_i^*, amb $j \leq i$:

$$\|v_j^*\|^2 \leq 2^{i-j} \|v_i^*\|^2. \tag{2.1}$$

Comencem provant les propietats d'una base LLL-reduïda $v_1, \ldots, v_n$.

1. Per demostrar la primera propietat, calculem

$$\begin{aligned}
\|v_i\|^2 &= \left\| v_i^* + \sum_{j=1}^{i-1} \mu_{i,j} v_j^* \right\|^2 \quad \text{per la Definició 2.1,} \\
&= \|v_i^*\|^2 + \sum_{j=1}^{i-1} \mu_{i,j} \|v_j^*\|^2 \quad \text{ja que } v_1^*, \ldots, v_n^* \text{ són ortogonals,} \\
&\leq \|v_i^*\|^2 + \sum_{j=1}^{i-1} \frac{1}{4} \|v_j^*\|^2 \quad \text{per la condició de mida,} \\
&\leq \|v_i^*\|^2 + \sum_{j=1}^{i-1} 2^{i-j-2} \|v_j^*\|^2 \quad \text{per (2.1)} \\
&= \frac{1+2^{i-1}}{2} \|v_i^*\|^2 \\
&\leq 2^{i-1} \|v_i^*\|^2 \quad \text{ja que } 1 \leq 2^{i-1} \text{ per a tot } i \geq 1.
\end{aligned}$$

Multiplicant aquesta desigualtat per si mateixa sobre $1 \leq i \leq n$ s'obté

$$\prod_{i=1}^{n} \|v_i\|^2 \leq \prod_{i=1}^{n} 2^{i-1} \|v_i^*\|^2 = 2^{n(n-1)/2} \prod_{i=1}^{n} \|v_i^*\|^2 = 2^{n(n-1)/2} (\det(R))^2,$$

on hem fet servir la Proposició 2.2 a l'última igualtat. Prenent l'arrel quadrada a ambdues bandes ja hem demostrat el que volíem.

2. Provem la segona propietat. Ara, per a tot $j \leq i$, fem servir la desigualtat provada prèviament amb $i = j$ i seguidament (2.1):

$$\|v_j\|^2 \leq 2^{j-1} \|v_j^*\|^2 \leq 2^{j-1} \cdot 2^{i-j} \|v_i^*\|^2 = 2^{i-1} \|v_i^*\|^2.$$

Prenent l'arrel quadrada a ambdues bandes ja hem demostrat el que volíem.

Ara demostrem les propietats que compleix el primer vector de la base LLL-reduïda.

3. Posant $j = 1$ a la propietat 2, multiplicant sobre $1 \leq i \leq n$ i aplicant la Proposició 2.2, tenim

$$\|v_i\|^n \leq \prod_{i=1}^{n} 2^{(i-1)/2} \|v_i^*\| = 2^{n(n-1)/4} \prod_{i=1}^{n} \|v_i^*\| = 2^{n(n-1)/4} \det(R).$$

Prenent l'arrel enèsima obtenim el que volíem.

4. Sigui $v \in R$ un vector no nul del reticle. Escrivim-lo com

$$v = \sum_{j=1}^{i} a_j v_j = \sum_{j=1}^{i} b_j v_j^*.$$

Observem que $a_1, \ldots, a_i \in \mathbb{Z}$ amb $a_i \neq 0$ (per tant $|a_i| \geq 1$), mentre que $b_1, \ldots, b_i \in \mathbb{R}$. Per definició, per a qualsevol k els vectors $v_1^*, \ldots, v_k^*$ són ortogonals dos a dos, i hem vist que l'espai vectorial que formen coincideix amb l'espai vectorial format pels vectors $v_1, \ldots, v_k$. Aleshores,

$$v \cdot v_i^* = a_i v_i \cdot v_i^* = b_i v_i^* \cdot v_i^* \quad \text{i} \quad v_i \cdot v_i^* = v_i^* \cdot v_i^*,$$

d'on concloem que $a_i = b_i$. Així doncs, $|b_i| = |a_i| \geq 1$. Aplicant aquesta desigualtat i la propietat 2 (amb $j = 1$), obtenim finalment

$$\|v\|^2 = \sum_{j=1}^{i} b_j^* \|v_j^*\|^2 \geq b_i^2 \|v_i^*\|^2 \geq \|v_i^*\|^2 \geq 2^{-(i-1)} \|v_1\|^2 \geq 2^{-(n-1)} \|v_1\|^2.$$

Prenent l'arrel quadrada a ambdós costats i sabent que v és qualsevol vector no nul del reticle, provem la propietat 4.

□

Finalment, podem introduir l'algorisme LLL, que busca millorar l'ortogonalitat i reduir la longitud dels vectors de la base donada.

Teorema 2.5. ***(Algorisme LLL)*** *Sigui R un reticle amb base $v_1, \ldots, v_n$. L'algorisme següent acaba en un nombre finit de passos i retorna una base LLL-reduïda per a R:*

Algorisme 2 Algorisme LLL

1: Introduir una base $v_1, \ldots, v_n$ d'un reticle R
2: Posar $k = 2$
3: Posar $v_1^* = v_1$
4: **while** $k \leq n$ **do**
5: **for** $j = 0, \ldots, k-1$ **do**
6: Posar $\mu_{k,j} = \frac{v_k \cdot v_j^*}{\|v_j^*\|^2}$
7: Posar $v_k = v_k - \lfloor \mu_{k,j} \rceil v_j^*$
8: Actualitzar la base associada de Gram-Schmidt
9: **end for**
10: **if** $\|v_k^*\| \geq \left(\frac{3}{4} - \mu_{k,k-1}^2\right) \cdot \|v_{k-1}^*\|^2$ **then**
11: Posar $k = k + 1$
12: **else**
13: Intercanviar v_{k-1} i v_k
14: Posar $k = \max(k-1, 2)$
15: Actualitzar la base associada de Gram-Schmidt
16: **end if**
17: **end while**
18: Retornar la base LLL-reduïda $\{v_1, \ldots, v_n\}$

Si denotem $M = \max \|v_i\|$, aleshores l'algorisme executa el bucle de k en un màxim de $\mathcal{O}(n^2 \log n + n^2 \log M)$ passos. En particular, s'executa en un temps polinòmic.

Observem que un cop introduïda la base del reticle, per a cada vector v_k, amb $k = 2, \ldots, n$, es calculen els coeficients de Gram-Schmidt i s'utilitzen per actualitzar el vector v_k, realitzant així una reducció de mida sobre aquest. Després es verifica si es compleix la condició de Lovász. En cas afirmatiu, l'algorisme passa al següent vector. En cas negatiu, s'intercanvia amb el vector anterior de la base i es torna a verificar la condició per al vector v_{k-1}. El procés iteratiu continua fins que tots els vectors compleixin la condició de Lovász.

Exemple 2.6. Sigui $R \subset \mathbb{R}^4$ un reticle amb base donada per les files de la matriu

$$A = \begin{pmatrix} 10 & 72 & 40 & 0 \\ 82 & 15 & 4 & 18 \\ -20 & -27 & -6 & 1 \\ 0 & 32 & 44 & 9 \end{pmatrix}.$$

El vector més curt de la base és el tercer, $v_3 = (-20, -27, -6, 1)$, amb $\|v_3\| = 34.1467$. A més, el quocient de Hadamard és $\mathcal{H}(A) = 0.3622$, per la qual cosa aquesta base és poc ortogonal.

Ara li apliquem l'algorisme LLL, programat a SageMath d'acord amb l'Algorisme 2, i obtenim la següent base donada per la matriu

$$A^{LLL} = \begin{pmatrix} 8 & -5 & 0 & 6 \\ -2 & 8 & -10 & -14 \\ -20 & -27 & -6 & 1 \\ -6 & 3 & 28 & -16 \end{pmatrix}.$$

Comprovem que ambdues matrius tenen el mateix determinant,

$$\det(A) = \det(A^{LLL}) = 229452.$$

Com era d'esperar, el vector més curt de la base LLL-reduïda és $v_1 = (8, -5, 0, 6)$, amb $\|v_1\| = 11.1803$, significativament més curt que el resultat trobat amb la base original. A més, el quocient de Hadamard és $\mathcal{H}(A^{LLL}) = 0.9889$, de manera que els vectors de la base LLL-reduïda són molt més ortogonals.

2.2 L'algorisme de Babai

El 1986, László Babai [11] va donar dos algorismes per resoldre la versió aproximada del CVP en temps polinòmic amb un factor γ exponencial. Ens centrarem en l'algorisme conegut com a algorisme d'arrodoniment. Anàlogament al que hem raonat a la Secció 2.1 per a una base ortogonal d'un reticle R, suposem que volem trobar un vector $v \in R$ que sigui el més proper a un vector donat $w \in \mathbb{R}^n$. Escrivim

$$w = t_1 v_1 + \ldots + t_n v_n \quad \text{amb } t_1, \ldots, t_n \in \mathbb{R}.$$

El nostre vector desitjat $v \in R$ es pot expressar com $v = a_1 v_1 + \ldots + a_n v_n$, amb $a_1, \ldots, a_n \in \mathbb{Z}$, ja que el vector pertany al reticle. Aleshores la distància a minimitzar és

$$\|w - v\| = (t_1 - a_1)^2 \|v_1\|^2 + \ldots + (t_n - a_n)^2 \|v_n\|^2.$$

Adonem-nos que cal prendre cada a_i com l'enter més proper al corresponent t_i.

Això funciona per a una base ortogonal, però, i en general? Si els vectors de la base són relativament ortogonals entre si (base "bona"), aleshores l'algorisme podria resoldre almenys l'apprCVP, però si no ho són (base "dolenta"), l'algorisme és altament ineficient, és a dir, el vector és massa lluny de w. A més, el resultat empitjora quan augmenta la dimensió del reticle.

En termes geomètrics, l'algorisme de Babai dona un punt v del reticle que queda dins del paral·lelepípede definit pels vectors de la base i centrat a w. Si w no és del reticle,

aleshores hi ha exactament un punt del reticle dins del paral·lelepípede, ja que el volum del paral·lelepípede coincideix amb el del reticle. A més, la forma del paral·lelepípede, que depèn de la qualitat de la base, indica si es troba o no una solució satisfactòria al CVP.

Teorema 2.7. ***(Algorisme d'arrodoniment de Babai)*** *Sigui $R \subset \mathbb{R}^n$ un reticle amb base $v_1, \ldots, v_n$ i sigui $w \in \mathbb{R}^n$ un vector objectiu. Si els vectors de la base són prou ortogonals entre si, aleshores el següent algorisme resol el CVP:*

Algorisme 3 Algorisme d'arrodoniment de Babai

1: Introduir w
2: Calcular les components $t_1, \ldots, t_n \in \mathbb{R}$ de w en la base $v_1, \ldots, v_n$
3: **for** $i = 1, \ldots, n$ **do**
4: Calcular $a_i = \lfloor t_i \rceil$
5: **end for**
6: Retornar $v = a_1 v_1 + \ldots + a_n v_n$

Així doncs, l'algorisme de Babai arrodoneix cada $t_i \in \mathbb{R}$ al seu respectiu enter més proper. Aquest procediment es pot realitzar per a qualsevol base del reticle, però clarament s'obtenen millors resultats si la base és LLL-reduïda. En aquest cas, Babai va demostrar que la norma $\|w - v\|$ està fitada per un factor exponencial del valor mínim.

Teorema 2.8. *Sigui $v_1, \ldots, v_n$ una base reduïda per LLL d'un reticle $R \subset \mathbb{R}^n$. Aleshores, donat un $w \in \mathbb{R}^n$, el resultat v de l'algorisme d'arrodoniment de Babai satisfà*

$$\|w - v\| \leq \left(1 + 2n\left(\frac{9}{2}\right)^{n/2}\right)\|w - u\|$$

per a tot $u \in R$.

Demostració. Consultar [11]. □

Exemple 2.9. Sigui $R \subset \mathbb{R}^2$ un reticle amb base donada pels vectors

$$v_1 = (-16, 102) \quad \text{i} \quad v_2 = (115, -550).$$

Observem que es tracta d'una base "dolenta", ja que el quocient de Hadamard és

$$\mathcal{H}(v_1, v_2) = \left(\frac{\det(R)}{\|v_1\|\|v_2\|}\right)^{1/2} = \left(\frac{2930}{(103.2473)(561.8941)}\right)^{1/2} = 0.2247.$$

Aplicant l'algorisme d'arrodoniment de Babai, programat a SageMath, volem trobar un vector de R que sigui proper al vector de $\mathbb{R}^2$

$$w = (4000, 87000).$$

El primer pas és calcular les components $t_1, t_2 \in \mathbb{R}$ de w en la base v_1, v_2, és a dir, tal que

$$w = t_1 v_1 + t_2 v_2.$$

Per tant, resolem el sistema d'equacions

$$4000 = -16t_1 + 115t_2 \quad \text{i} \quad 87000 = 102t_1 - 550t_2$$

i trobem que $t_1 \approx 4165.53$ i $t_2 \approx 614.33$. El següent pas és arrodonir-los a l'enter més proper, respectivament: $a_1 = 4166$ i $a_2 = 614$. Calculem el vector resultant $v \in R$:

$$v = a_1 v_1 + a_2 v_2 = 4166(-16, 102) + 614(115, -550) = (3954, 87232).$$

El vector de v queda molt lluny de w

$$\|w - v\| \approx 236.5164,$$

i per tant l'algorisme és poc pràctic, com era d'esperar.

En canvi, si apliquem l'algorisme LLL a la base v_1, v_2, obtenim la base donada pels vectors

$$v_1' = (54, 22) \quad \text{i} \quad v_2' = (35, -40),$$

que ara és una base "bona", fet que corrobora el quocient de Hadamard:

$$\mathcal{H}(v_1', v_2') = \left(\frac{\det(R)}{\|v_1'\|\|v_2'\|}\right)^{1/2} = \left(\frac{2930}{(58.3095)(53.1507)}\right)^{1/2} = 0.9723.$$

Operant de manera anàloga, s'obté el vector resultant $v' \in R$

$$v' = a_1' v_1' + a_2' v_2' = 1094(54, 22) - 1573(35, -40) = (4021, 86988).$$

La distància entre v' i w és petita

$$\|w - v'\| \approx 24.1868,$$

confirmant que l'algorisme és molt més eficient per a bases prou ortogonals i en particular LLL-reduïdes.

2.3 Tècnica d'encabiment de Kannan

La tècnica d'encabiment de Kannan (*Kannan's embedding technique*) [12] permet reduir el CVP al SVP. Sigui R un reticle donat per una base $v_1, \ldots, v_n$ i considerem el vector objectiu $w \in \mathbb{R}^n$. Com hem vist a l'algorisme de Babai, una solució al CVP correspon al vector $v = a_1 v_1 + \ldots + a_n v_n \in R$, per a $a_1, \ldots, a_n \in \mathbb{Z}$, tal que la norma $\|w - v\|$ està fitada superiorment. Notem que el vector $e = w - v$ té una norma $\|e\|$ petita (és un vector curt). La nostra meta és trobar v a partir d'e. Volem definir un nou reticle R' que contingui el vector e. Sigui $M \in \mathbb{R}_{>0}$. El reticle R' està definit pels vectors de $\mathbb{R}^{n+1}$

$$(v_1, 0), \ldots, (v_n, 0), (w, M),$$

els quals formen una base de $\mathbb{R}^{n+1}$. Observem que (e, M) pertany a R', ja que fent la combinació lineal de les files amb coeficients enters $(-a_1, \ldots, -a_n, 1)$ tenim

$$-a_1(v_1, 0) - \ldots - a_n(v_n, 0) + 1 \cdot (w, M) = (-a_1 v_1 - \ldots - a_n v_n + w, M) = (e, M).$$

Així doncs, resolent l'SVP en el reticle R' trobarem e, i fent $w - e$ podrem resoldre el CVP.

2.3.1 Elecció de M

El resultat de la tècnica d'encabiment és el vector (e, M), el qual s'espera que sigui solució de l'SVP o almenys suficientment proper a un vector més curt del reticle R'. El grau d'èxit depèn de la mida del vector e comparada amb les mides dels vectors curts de R.

D'altra banda, la tria de M també hi influeix. Diversos autors suggereixen la tria de $M = \|e\|$ [12, 13]. Si M es tria més gran, llavors la bretxa de separació al problema uSVP esdevé més petita. En canvi, si M és massa petit, amb probabilitat no nul·la existeix un vector $x \in R'$ tal que $\|x + c \cdot (e, M)\| < \|(e, M)\|$ per a $c \in \mathbb{Z}$, segons [14]. Malauradament, a la pràctica no coneixem el valor de $\|e\|$, ja que precisament pretenem trobar el vector e. Experimentalment, s'ha comprovat que fer servir $M = 1$ és més eficient que no pas $M = \|e\|$ [14, 15].

El següent lema il·lustra que sota determinades condicions sobre M, i mentre el vector objectiu w sigui proper a un vector del reticle, aleshores el CVP es pot reduir a l'SVP.

Lema 2.10. *Sigui un reticle $R \subset \mathbb{Z}^n$ donat per la base $v_1, \ldots, v_n$ i denotem per λ_1 la norma d'un vector no nul més curt de R. Sigui $w \in \mathbb{R}^n$ i sigui $v \in R$ un vector més proper a w. Definim $e = w - v$. Suposem que $\|e\| < \lambda_1/2$ i sigui $M = \|e\|$. Aleshores (e, M) és un vector no nul més curt del R' de la tècnica d'encabiment.*

Demostració. Els vectors de R' es poden escriure de la forma

$$l_{n+1}(e, M) + \sum_{i=1}^{n} l_i(v_i, 0) = \left(\sum_{i=1}^{n} l_i v_i + l_{n+1} e, l_{n+1} M \right)$$

per a $l_1, \ldots, l_{n+1} \in \mathbb{Z}$. Comprovem que tot vector no nul de R' té norma com a mínim la norma del vector (e, M). Primerament, observem que

$$\|(e, M)\|^2 = \|e\|^2 + M^2 = 2\|e\|^2 < 2{\lambda_1}^2/4 = {\lambda_1}^2/2,$$

de manera que $\|(e, \pm M)\| < \lambda_1/\sqrt{2}$.

- **Cas** $l_{n+1} = 0$. Els vectors de R' seran de la forma $(u, 0)$, amb $u \in R$. Aleshores veiem fàcilment que

$$\|(u, 0)\| = \|u\| \geq \lambda_1 > \sqrt{2}\|(e, M)\| > \|(e, M)\|.$$

- **Cas** $l_{n+1} = 1$. Els vectors de R' seran de la forma (u, M), amb $u \in R$ de la forma $u = x + e$ per a un $x \in R$. Com que v és una solució al CVP respecte de w, aleshores $\|e\| \leq \|e + x\|$ per a tot $x \in R$. Aleshores

$$\|(u, M)\|^2 = \|u\|^2 + M^2 \geq \|e\|^2 + M^2 \geq \|e\|^2 = \|(e, M)\|^2,$$

  i per tant $\|(u, M)\| \geq \|(e, M)\|$.

- **Cas** $l_{n+1} \geq 2$. Els vectors de R' seran de la forma $(u, l_{n+1}M)$, amb $u \in R$. Aleshores

$$\|(u, l_{n+1}M)\|^2 \geq \|(0, l_{n+1}M\|^2 = (l_{n+1}M)^2 \geq 4M^2 = 4\|e\|^2 = 2\|(e, M)\|^2,$$

  i per tant $\|(u, l_{n+1}M)\| \geq \sqrt{2}\|(e, M)\| > \|(e, M)\|$.

□

3 Problema LWE

Introduïm una secció clau en aquest treball: el problema LWE (Learning With Errors). Es basa en la dificultat de resoldre sistemes d'equacions lineals quan s'introdueix un cert nivell d'errors en les observacions o "soroll".

Imaginem que tenim un sistema d'equacions lineals de la forma $As + e = b$, on A és una matriu coneguda, s és un vector secret que volem amagar, e és un vector d'errors i b és un altre vector, ambdós coneguts. En un escenari ideal sense errors ($e = 0$), el sistema es reduiria a $As = b$, el qual seria relativament senzill de resoldre aplicant el mètode de Gauss-Jordan. Tanmateix, en el context del LWE, introduïm errors aleatoris en les observacions, convertint així la resolució del sistema, és a dir, trobar s, en un repte computacionalment difícil.

Els sistemes de criptografia basats en LWE aprofiten la dificultat de resoldre el problema LWE. El vector secret s i els errors introduïts s'utilitzen per xifrar els missatges, de manera que un possible atacant té extremadament difícil recuperar el missatge original sense conèixer s. Normalment es treballa amb valors a l'anell d'enters mòdul q, $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$.

Problema 3.1. (Problema LWE de cerca). Sigui $\mathbb{Z}_q$ l'anell d'enters mòdul q, on q és un primer positiu. Siguin n i m enters positius. Sigui una distribució de probabilitat χ, de la qual extraurem valors d'error $e_1, \ldots, e_m \in \mathbb{Z}_q$. Sigui $s = (s_1, \ldots, s_n) \in \mathbb{Z}_q^n$ un vector fixat i $a_1, \ldots, a_m \in \mathbb{Z}_q^n$ vectors de longitud n, de manera que $a_i = (a_{i1}, \ldots, a_{in})$. Es calculen els valors $b_i = a_i \cdot s + e_i \in \mathbb{Z}_q$, $i = 1, \ldots, m$. El problema consisteix en:

Donat el conjunt de m parells $\{(a_i, b_i)\}_{i=1}^m$, hem de trobar $s \in \mathbb{Z}_q^n$.

3.1 Descripció del criptosistema Regev

En el seu article del 2005, Regev introdueix un criptosistema basat en LWE [16]. Està parametritzat pels enters positius n (paràmetre de seguretat) i m (nombre d'equacions), un nombre primer positiu q (mòdul de l'anell) i un real positiu α (paràmetre de soroll). Totes les operacions es realitzen mòdul q. La distribució de probabilitat χ acostuma a ser una distribució discreta gaussiana sobre $\mathbb{Z}_q$, centrada al zero i amb desviació estàndard $\alpha q/\sqrt{2\pi}$, on α és un real positiu.

Posteriorment, Regev i Micciancio [17] profunditzen en el mateix criptosistema, l'anàlisi d'errors de desxifratge i l'elecció dels paràmetres per tal de minimitzar-los i garantir-ne la seguretat. Nosaltres ens basarem en aquesta descripció.

Tenim l'escenari usual en criptografia: l'Alice vol enviar un missatge al Bob. Prèviament, el Bob escull una clau pública amb la qual l'Alice xifra el missatge per tal que l'Eve no els l'intercepti. L'Alice envia el missatge encriptat al Bob. Finalment, el Bob el desencripta utilitzant la seva clau secreta.

Idea

- El Bob escull n, m, q i una distribució de probabilitat χ, de la qual extraurà errors "petits".
- El Bob genera valors d'error enters $e_1, \ldots, e_m$ a partir de la distribució de probabilitat χ.

- El Bob genera un vector secret aleatori $s = (s_1, ..., s_n) \in \mathbb{Z}_q^n$.
- El Bob genera vectors $a_1, ..., a_m \in \mathbb{Z}_q^n$, de manera que $a_i = (a_{i1}, ..., a_{in})$, $i = 1, ..., m$.
- El Bob calcula els m valors

$$b_i = a_i \cdot s + e_i \in \mathbb{Z}_q,$$

on $\cdot$ denota el producte escalar usual entre dos vectors.

Així, s'obté un sistema d'm equacions

$$\begin{cases} a_{11}s_1 + a_{12}s_2 + ... + a_{1n}s_n + e_1 = b_1 \\ a_{21}s_1 + a_{22}s_2 + ... + a_{2n}s_n + e_2 = b_2 \\ \qquad\qquad\qquad\qquad\qquad\qquad\quad \vdots \\ a_{m1}s_1 + a_{m2}s_2 + ... + a_{mn}s_n + e_m = b_m \end{cases},$$

que es pot expressar a través de la igualtat matricial $As+e = b$. La matriu $A \in \mathbb{Z}_q^{m \times n}$ és aquella que té per files els vectors a_i,

$$A = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & ... & a_{mn} \end{pmatrix},$$

s és el vector columna corresponent al vector secret, e és el vector columna format pels errors e_i i b és el vector columna format pels valors b_i o vector solució del sistema.

Així doncs, la clau pública decidida pel Bob està formada per q i els parells (a_i, b_i):

$$\boxed{K_{\text{publica}} = (q, \{(a_i, b_i)\}_{i=1}^m) = (q, A, b).}$$

La clau privada del Bob correspon al vector s:

$$\boxed{K_{\text{privada}} = s.}$$

Xifratge

El sistema permet d'enviar un missatge d'un únic bit, ja sigui el 0 o l'1. Considerem $\{-1, 0, 1\}^m$, el conjunt de les m-tuples de components -1, 0 o 1. Per exemple, per a $m = 2$ tindríem el conjunt $\{(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)\}$.

Com veurem tot seguit, a l'Alice li interessa fer una combinació lineal amb coeficients aleatoris de les m files de la matriu A. Essencialment, es fa una combinació lineal de les m equacions del sistema, de manera que s'obté una nova equació. Una tupla aleatòria del conjunt $\{-1, 0, 1\}^m$ contindrà els coeficients d'aquestes combinacions lineals. Han de ser coeficients petits respecte als de la matriu A per tal de no fer massa gran els errors e_i.

a) Enviament del bit 0

- L'Alice tria una tupla aleatòria $T \in \{-1, 0, 1\}^m$, on pensem T com un vector fila.
- L'Alice calcula el vector fila $TA \in \mathbb{Z}_q^n$. Observem que T conté els coeficients de combinacions lineals de les m equacions.
- De manera similar, calcula el producte escalar $T \cdot b \in \mathbb{Z}_q$.
- Finalment, envia el parell $(TA, T \cdot b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ al Bob.

b) Enviament del bit 1

- De nou, l'Alice tria una tupla aleatòria $T \in \{-1, 0, 1\}^m$.
- L'Alice calcula TA de la mateixa manera que abans.
- No obstant això, aquest cop calcula $T \cdot b + \left\lfloor \frac{q}{2} \right\rfloor$, on $\lfloor \cdot \rfloor$ denota la funció part entera inferior, que es defineix per a un $x \in \mathbb{R}$ com:

  $$\lfloor x \rfloor = \max\{k \in \mathbb{Z} \mid k \leq x\}.$$

  En altres paraules, la funció part entera inferior arrodoneix un real a l'enter més proper per sota.
- Finalment, envia el parell $(TA, T \cdot b + \left\lfloor \frac{q}{2} \right\rfloor) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ al Bob.

Per generalitzar, si diem $x \in \{0, 1\}$ al bit a enviar, l'Alice envia al Bob el parell

$$(u, v) := (TA, T \cdot b + x \left\lfloor \frac{q}{2} \right\rfloor) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

Desxifratge

- El Bob rep el parell (u, v) i calcula la quantitat $y := v - u \cdot s$ mòdul q.

  Com sabem que desxifra correctament? Es té que

  $$y = T \cdot b + x \left\lfloor \frac{q}{2} \right\rfloor - TA \cdot s = T \cdot (As + e) + x \left\lfloor \frac{q}{2} \right\rfloor - TA \cdot s = T \cdot e + x \left\lfloor \frac{q}{2} \right\rfloor = \begin{cases} T \cdot e, & \text{si } x = 0; \\ T \cdot e + \left\lfloor \frac{q}{2} \right\rfloor, & \text{si } x = 1. \end{cases}$$

- Si $y \in [0, \frac{q}{4}] \cup (\frac{3q}{4}, q - 1]$, aleshores desxifra el missatge com un 0.
- Si $y \in (\frac{q}{4}, \frac{3q}{4})$, aleshores el desxifra com un 1.

En conclusió, el missatge s'interpreta com un 0 si y és més a prop del 0 que de $\left\lfloor \frac{q}{2} \right\rfloor$, i com un 1 altrament.

3.1.1 Anàlisi d'errors

Diem que hi haurà un error de desxifratge si l'algorisme interpreta un 0 com un 1 o viceversa. Si no hi hagués errors a les mostres, aleshores r seria 0 o $\left\lfloor \frac{q}{2} \right\rfloor$, i no hi hauria errors de desxifratge. Per tant, hi haurà un error de desxifratge si i només si

$$|T \cdot e| > \frac{q}{4}.$$

Es pot minimitzar la probabilitat que aquesta quantitat sigui gran escollint amb cura els diversos paràmetres.

Observació 3.2. Si escollim representants de les classes en l'interval $[-\left\lfloor \frac{q}{2} \right\rfloor, \left\lfloor \frac{q}{2} \right\rfloor]$ si q és senar i en l'interval $[-\frac{q}{2}+1, \frac{q}{2}]$ si q és parell, les condicions se simplifiquen a:

- Si $y \in (-\frac{q}{4}, \frac{q}{4})$, aleshores el Bob desxifra un 0.
- Altrament, desxifra un 1.

El nostre objectiu és minimitzar la probabilitat que el valor absolut de $T \cdot e$ sigui superior a $q/4$ per a un $T \in \{-1, 0, 1\}^m$ fixat.

Recordem que, per indicar que una variable aleatòria X segueix una distribució gaussiana (també coneguda com a normal) de mitjana μ i desviació estàndard σ, sovint s'escriu $X \sim N(\mu, \sigma)$. En particular, es diu que X segueix una normal estàndard si $X \sim N(0, 1)$. A més, recordem les següents propietats:

1. Si $X \sim N(\mu, \sigma)$ i $a \in \mathbb{R}$, aleshores $aX \sim N(\mu, a\sigma)$.
2. Si $X_1, \ldots, X_n$ són variables aleatòries independents tals que $X_i \sim N(\mu_i, \sigma_i)$, $i = 1, \ldots, n$, aleshores $X_1 + \ldots + X_n \sim N(\mu_1 + \ldots, \mu_n, \sqrt{{\sigma_1}^2 + \ldots + {\sigma_n}^2})$.

En el nostre cas, cada error e_i, $i = 1, \ldots, m$, segueix una distribució $N(0, \alpha q/\sqrt{2\pi})$. Si escrivim $T = (t_1, \ldots, t_m)$ i $e = (e_1, \ldots, e_m)^T$, aleshores

$$T \cdot e = t_1 e_1 + \ldots + t_m e_m.$$

Observem que per la primera propietat es té $t_i e_i \sim N(0, t_i \alpha q/\sqrt{2\pi})$. I per la segona propietat tenim que $T \cdot e \sim N(0, \alpha q/\sqrt{2\pi}\sqrt{t_1^2 + \ldots + t_m^2}) = N(0, \alpha q \|T\|/\sqrt{2\pi})$, és a dir, $T \cdot e$ segueix una distribució normal de mitjana zero i desviació estàndard $\alpha q \|T\|/\sqrt{2\pi}$. D'altra banda, l'esperança del quadrat de cada coordenada t_i de T, o moment d'ordre 2, és

$$\frac{1}{3} \sum_{t_i \in \{-1,0,1\}} t_i^2 = \frac{2}{3},$$

per la qual cosa la norma de T és aproximadament, amb alta probabilitat,

$$\|T\| \approx \sqrt{\frac{2m}{3}}.$$

Així doncs, es pot estimar que la quantitat $T \cdot e$ segueix una distribució normal de mitjana zero i desviació estàndard $\alpha q \sqrt{m/3\pi}$. En conclusió, la probabilitat d'error d'un missatge consistent en un bit es pot estimar com la probabilitat que el valor absolut d'una variable amb aquesta distribució sigui superior a $q/4$, és a dir,

$$\text{probabilitat d'error d'un bit} \approx 2\left(1 - \Phi\left(\frac{1}{4\alpha}\sqrt{\frac{3\pi}{m}}\right)\right),$$

on Φ denota la funció de distribució acumulativa de la distribució normal estàndard.

3.1.2 Paràmetres

A [17], Regev i Micciancio suggereixen com escollir els diferents paràmetres n, m, q, α. És essencial triar els paràmetres òptims per tal de garantir la màxima eficiència i seguretat

i la mínima probabilitat d'errors de desxifratge. Tots els logaritmes són en base 2 i totes les operacions es realitzen mòdul q.

En primer lloc, volem que quan algú faci servir per encriptar la clau pública (A, b) escollida aleatòriament, aleshores el resultat no contingui informació sobre el missatge xifrat. Dit d'una altra manera, volem que els 0 o 1 xifrats estiguin uniformment distribuïts i que l'algorisme no pugui endevinar el bit més enllà d'endevinar-lo aleatòriament. Això s'aconsegueix si el nombre de possibilitats per a T és molt més gran que el nombre d'elements al nostre rang, és a dir, si

$$3^m >> q^{n+1}.$$

En segon lloc, volem que amb certs paràmetres el problema LWE sigui difícil. La teoria suggereix que q sigui un nombre primer, que l'elecció de m és irrellevant, que es compleixi $\alpha q > n$ i que α hauria de ser el més gran possible. Malauradament, això no ens dona cap pista sobre la seguretat del criptosistema donat un conjunt específic de paràmetres. Per aquest motiu, s'han de fer servir resultats experimentals, dels quals obtenim una cota inferior d'α,

$$\alpha \geq 1.5\sqrt{2\pi}\max\left\{\frac{1}{q}, 2^{-2\sqrt{n\log q\log\delta}}\right\}.$$

I ara, tenint en compte les dues equacions anteriors, finalment fem una tria concreta de paràmetres segurs:

$$\begin{cases} m = \frac{(n+1)\log q + 200}{\log 3}, \\ \alpha = 4 \cdot \max\left\{\frac{1}{q}, 2^{-2\sqrt{n\log q\log(1.01)}}\right\}, \end{cases} \tag{3.1}$$

on hem triat $\delta = 1.01$.

Pel que fa als paràmetres restants n i q, no se'n dona una tria directa, sinó que s'haurien d'estudiar experimentalment juntament amb m i α.

n	m	q	α
136	2008	2003	0.0065

Taula 1: Exemple concret de paràmetres realistes que segueix (3.1), extret de [17].

Al seu article [18], però, Regev dona una altra possible tria de paràmetres:

$$\begin{cases} m = 1.1 \cdot n\log q, \\ \alpha = \frac{1}{\sqrt{n}\log^2 n}, \end{cases} \quad , \quad \text{essent } q \text{ un primer tal que } n^2 < q < 2n^2. \tag{3.2}$$

Per últim, és interessant mencionar que el software SageMath permet triar els paràmetres més adients per a un n donat seguint (3.2). Vegem un exemple per a $n = 20$:

```
from sage.crypto.lwe import Regev
Regev(20)
```

S'introdueix com a argument n, i opcionalment també m. La sortida de Sage és:

```
LWE(20, 401, Discrete Gaussian sampler over the Integers with
    sigma = 1.915069 and c = 401, 'uniform', None)
```

El primer paràmetre és $n = 20$. El segon és q, així que veiem que, per a $n = 20$, Sage recomana $q = 401$. Efectivament, és un nombre primer entre n^2 i $2n^2$, i Sage escull concretament el primer immediatament següent a n^2. Tot seguit apareix la distribució discreta gaussiana, amb $\sigma = 1.915069$, d'acord amb (3.2) si tenim en compte que $\sigma = \alpha/\sqrt{2\pi}$. La mitjana és $c = 401$, però com que les operacions es realitzen mòdul q, que la mitjana sigui q en comptes de zero no afecta al resultat final. El paràmetre **"uniform"** indica que el secret es triarà uniformement i aleatòriament sobre $\mathbb{Z}_q$. Per últim, **"none"** significa que no fitem m superiorment.

3.2 Dificultat del LWE

És natural que arribats a aquest punt ens preguntem si el problema LWE es considera difícil i per què. Hi ha diversos motius per creure que ho és. En primer lloc, els algorismes més coneguts per al LWE tenen un temps d'execució exponencial en n, i això no millora ni tan sols amb algorismes quàntics. En segon lloc, ens basem en la dificultat *worst-case* de problemes de reticles, com ara l'apprSVP i l'SIVP, comentats a la Secció 1.2.

Pel que fa a la dificultat de l'apprSVP, se sap que és $\mathcal{NP}$-difícil per a γ petites, senzill per a factors d'aproximació exponencials $\gamma = 2^{\mathcal{O}(n)}$ [7] i difícil (però no $\mathcal{NP}$-difícil) per a factors d'aproximació polinòmics de la forma n^k, on $k \geq 1$ és una constant.

Són aquests últims els que ens interessen, ja que donarien una solució més exacta al problema, és a dir, donarien un vector més proper al vector més curt. Diem que segueix sent difícil ja que els algorismes més eficients que es coneixen s'executen en temps exponencial $2^{\mathcal{O}(n)}$, com ja comentàvem a la Secció 2.1. Podem aplicar les mateixes propietats de l'apprSVP a l'SIVP.

Introduïm primer una definició bàsica sobre reticles:

Definició 3.3. *La distribució gaussiana discreta sobre el reticle R amb paràmetre r, denotada per $\chi_{R,r}$, és la distribució que assigna massa proporcional a $e^{-\pi\|x/r\|^2}$ a cada $x \in R$. Suposant que r no és molt petit, les mostres extretes de $\chi_{R,r}$ són vectors del reticle R de norma aproximadament $\sqrt{n}r$.*

Definició 3.4. *El dual d'un reticle R en $\mathbb{R}^n$, denotat per R^*, és el reticle donat pel conjunt de vectors $y \in \mathbb{R}^n$ tal que $x \cdot y \in \mathbb{Z}$ per tots els vectors $x \in R$.*

Oded Regev va enunciar la següent proposició que resumeix la complexitat del LWE [18]:

Proposició 3.5. *Sigui $q \geq 2$ un enter i $\alpha \in (0, 1)$ un nombre real. Suposem que existeix un oracle que resol el problema LWE amb mòdul q i paràmetre d'error α. Aleshores, donat un reticle R qualsevol i un nombre polinòmic suficientment gran de mostres de la distribució gaussiana discreta $\chi_{R^*,r}$ per a un r no molt petit, i un punt x a distància com a molt $\alpha q/(\sqrt{2}r)$ de R, podem trobar el punt (únic) del reticle més proper a x en temps polinòmic.*

Això significa que la resolució del problema LWE implicaria la resolució d'un problema de reticles *worst-case* que es creu que no es pot resoldre d'una altra manera. Com que aquests són difícils fins i tot amb una base "bona", d'aquí deduïm la dificultat del LWE.

Oded Regev [16] va establir una reducció quàntica, és a dir, realitzada amb un algorisme quàntic, de problemes de reticles *worst-case* com ara del l'apprSVP i l'apprSIVP al

LWE. Això vol dir que trencar el criptosistema basat en el LWE implica que existeix un algorisme quàntic eficient per resoldre l'apprSVP, la qual cosa és molt difícil. Si suposem que no existeix aquest algorisme quàntic eficient, aleshores no serà possible trencar el criptosistema. Observem que aquesta reducció és només quàntica, i per tant una mica més feble que si impliqués un algorisme no-quàntic per als problemes de reticles. Malgrat aquests raonaments quàntics, destaquem que el problema LWE i els criptosistemes associats són completament clàssics i no tenen res a veure amb raonaments quàntics.

La reducció es descriu en detall al següent teorema.

Teorema 3.6. *Sigui $\alpha \in \mathbb{R}_{>0}$ i sigui χ_α la distribució gaussiana discreta sobre $\mathbb{Z}_q$ de mitjana zero i desviació estàndard $\alpha q/\sqrt{2\pi}$. Suposem que existeix un oracle que resol el problema LWE amb paràmetres n, m, q, χ_α amb $\alpha q > \sqrt{n}$, $q \leq \text{poly}(n)$ és primer i $m \leq \text{poly}(n)$. Aleshores, donat un reticle qualsevol de dimensió n, existeix un algorisme quàntic amb temps d'execució* $\text{poly}(n)$ *que resol els problemes de reticles worst-case SIVP i SVP (versió decisional).*

Observem que m no juga un paper important. De fet, es pot prendre m tan gran com es vulgui (quant més gran, més senzill esdevé el problema). Si un dia es troba una reducció no-quàntica, es tindria una garantia de seguretat més forta.

3.2.1 Decisió implica cerca

A partir d'ara direm que un conjunt de parells $\{(a_i, b_i)\}$ és un conjunt de parells LWE si existeix un vector s tal que

$$b_i = a_i \cdot s + e_i,$$

per a e_i extretes d'alguna distribució d'error χ que doni valors d'error "petits".

A l'inici de la Secció 3 hem definit el problema LWE de cerca, consistent en trobar el vector secret s a partir d'un conjunt de parells LWE donat $\{(a_i, b_i)\}_{i=1}^m$. Però hi ha un altre problema a destacar relacionat amb LWE:

Problema 3.7. (Problema LWE de decisió). Donat el conjunt de parells $\{(a_i, b_i)\}_{i=1}^m$, hem de decidir si els parells són LWE, o si per a cada i b_i ha estat triat aleatòriament i uniformement del conjunt $\mathbb{Z}_q$.

Resulta que ambdós problemes són equivalents. Primer veiem la implicació més intuïtiva:

Proposició 3.8. ***(Cerca implica decisió).*** *Si es pot resoldre el problema LWE de cerca, aleshores es pot resoldre el problema LWE de decisió.*

Demostració. Suposem que tenim un oracle que resol el problema LWE de cerca. Donat un conjunt de mostres $\{(a_i, b_i)\}_{i=1}^m$ per al problema de LWE de decisió, podem utilitzar l'oracle de LWE-cerca per obtenir un vector s. Per a cada i, considerem la quantitat $b_i - a_i \cdot s$. Si s és la solució correcta, aquesta diferència hauria d'estar distribuïda segons la distribució d'errors χ. □

Però l'altra implicació també és certa! Si podem distingir els parells LWE dels que no ho són, aleshores podem trobar s, sempre i quan q sigui polinòmic en n. Es tracta d'un important resultat de reducció:

Proposició 3.9. ***(Decisió implica cerca).*** *Si es pot resoldre el problema LWE de decisió, aleshores es pot resoldre el problema LWE de cerca, sempre i quan $q = poly(n)$.*

Demostració. Suposem que tenim un oracle que resol el problema LWE de decisió. Donat un conjunt de mostres $\{(a_i, b_i)\}_{i=1}^m$ per al problema de LWE de cerca, podem utilitzar l'oracle de LWE-decisió per obtenir un vector s.

Sigui $k \in \mathbb{Z}_q$. Volem determinar si s_j, la coordenada j-èsima del vector s, és igual a k. Generem enters aleatoris $r_1, \ldots, r_m$ i substituïm els parells $\{(a_i, b_i)\}$ pels parells $\{(a'_i, b'_i)\}$, on

$$a'_i = a_i + (0, \ldots, 0, r_i, 0, \ldots, 0) \quad \text{i} \quad b'_i = b_i + r_i k.$$

Si $s_j = k$, aleshores els nous parells seran parells LWE segons l'oracle de LWE-decisió, ja que, per a cada i, tenim:

$$\begin{aligned} b'_i - a'_i \cdot s &= (b_i + r_i k) - (a_i + (0, \ldots, 0, r_i, 0, \ldots, 0)) \cdot s = \\ &= b_i + r_i k - a_i \cdot s - (0, \ldots, 0, r_i, 0, \ldots, 0) \cdot s = \\ &= (b_i - a_i \cdot s) + r_i k - r_i s_j = \\ &= e_i + r_i (k - s_j). \end{aligned}$$

Observem que si $s_j = k$, aleshores el resultat és e_i, i aleshores els parells són LWE. En canvi, si $s_j \neq k$, aleshores el nombre $r_i(k - s_j)$ és aleatori ja que r_i és aleatori, i aleshores l'oracle de LWE-decisió descartaria els parells.

Repetim iterativament aquest procés per a diferents valors de k per a cada $j = 1, \ldots, n$. Com que $q = \text{poly}(n)$, podem obtenir el vector s en un nombre de passos polinòmic en n. □

3.3 Altres criptosistemes amb LWE

3.3.1 Generalització del criptosistema Regev

Tot i que prèviament ens hem basat en el criptosistema descrit per Regev i Micciancio, ara explicarem la generalització que proposen d'aquest criptosistema [17]. En primer lloc, la introducció d'un paràmetre enter $r \geq 1$ amplia el conjunt de les m-tuples de $\{-1, 0, 1\}^m$ a $\{-r, -r+1, \ldots, r\}^m$, de manera que ara tenim $2r + 1$ possibles coeficients per crear la nostra combinació lineal d'equacions del sistema i xifrar el missatge.

En segon lloc, s'introdueixen els paràmetres enters t i l, de manera que ara l'espai de missatges és $\mathbb{Z}_t^l$. Abans teníem $l = 1$ perquè havíem fixat l'enviament d'un únic bit, i $t = 2$ ja que podíem enviar un 0 o un 1. Es considera una funció f, que mapeja l'espai de missatges $\mathbb{Z}_t^l$ a $\mathbb{Z}_q^l$ multiplicant cada coordenada per q/t i arrodonint-la a l'enter més proper. També es considera la funció inversa f^{-1}, que pren un element de $\mathbb{Z}_q^l$, divideix cada coordenada per q/t, l'arrodoneix a l'enter més proper i retorna un element de $\mathbb{Z}_t^l$. Igual que abans, χ denota la distribució discreta gaussiana sobre $\mathbb{Z}_q$ centrada al zero i amb desviació estàndard $\alpha q/\sqrt{2\pi}$, essent α un real positiu.

Vegem l'esquema complet a continuació:

- **Paràmetres:** $n, m, q, l, t, r \in \mathbb{Z}$ i $\alpha \in \mathbb{R}_{>0}$.
- **Clau privada:** el Bob escull $s \in \mathbb{Z}_q^{n \times l}$ a partir de la distribució de probabilitat χ. La clau privada és s.
- **Clau pública:** el Bob escull $A \in \mathbb{Z}_q^{m \times n}$ i $e \in \mathbb{Z}_q^{m \times l}$. La clau pública és $(A, b = As + e)$.
- **Xifratge:** donat un element de l'espai de missatge $v \in \mathbb{Z}_t^l$ i una clau pública (A, b), l'Alice tria un vector aleatori $T \in \{-r, -r+1, \ldots, r\}^m$ i envia el parell $(u, v) := (A^T T, b^T T + f(v)) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^l$ al Bob.
- **Desxifratge:** el Bob rep el parell (u, v) i calcula $f^{-1}(v - s^T u)$.

3.3.2 Brakerski-Vaikuntanathan (BV)

El criptosistema Brakerski-Vaikuntanathan (BV) [19] és un altre esquema de xifratge basat en la dificultat de resoldre el problema LWE presentat el 2011, posterior a l'esquema presentat per Regev. Nosaltres en presentem la versió adaptada de [20].

El plantejament és molt similar al criptosistema clàssic LWE: el Bob escull paràmetres n, m, q, una distribució de probabilitat χ i genera $s \in \mathbb{Z}_q^n$ i $a_1, \ldots, a_m \in \mathbb{Z}_q^n$, amb la diferència que en aquest cas calcula els b_i com

$$b_i = a_i \cdot s + 2e_i \in \mathbb{Z}_q.$$

Per enviar un bit $x \in \{0, 1\}$, l'Alice tria un vector aleatori $T \in \{0, 1\}^m$, però aquest cop envia al Bob el parell

$$(u, v) := (TA, T \cdot b + x) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

En el procés de desxifratge, el Bob rep el parell (u, v) i calcula la quantitat $y := v - u \cdot s$ mòdul 2. Això funciona ja que y justament correspon al bit x:

$$y = T \cdot b + x - TA \cdot s = T(As + 2e) + x - TA \cdot s = 2Te + x \equiv x \pmod 2.$$

4 Resolució del LWE amb la tècnica d'encabiment

Ja hem vist relacions entre el problema LWE i els reticles. Ara en proporcionem una altra de caire pràctic en forma d'un atac al criptosistema basat en LWE. En el seu article del 2018, Yuntao Wang et al. [15] apliquen l'anomenada tècnica d'encabiment per reduir el LWE a l'uSVP, utilitzant l'algorisme de reducció de reticles BKZ. Nosaltres, però, farem servir l'algorisme LLL, tal i com es fa a [12, 14].

Primerament, cal introduir el concepte de reticle q-ari, on q és un enter positiu. Intuïtivament, és un reticle al qual afegim els vectors de coordenades totes enteres i múltiples de q (tot i que no tots els vectors del reticle q-ari són d'aquesta forma). Per exemple, en un reticle 2-ari de $\mathbb{R}^2$, podríem obtenir punts com el (0,0), (2,4), (-6,8), etc.

Definició 4.1. *Sigui R un reticle de dimensió n, i m i q enters. Un reticle $R \subset \mathbb{Z}^m$ és un reticle q-ari si $q\mathbb{Z}^m \subset R$.*

Definició 4.2. *Donada una matriu $A \in \mathbb{Z}_q^{m\times n}$ $(m > n)$, el reticle q-ari associat a A és*

$$R_{(A,q)} = \{v \in \mathbb{Z}_q^m \mid v \equiv Ax \ (mod\ q) \text{ per algun } x \in \mathbb{Z}^n\}.$$

Podem construir un sistema de generadors $\mathcal{B}$ del reticle $R_{(A,q)}$ com les files de la matriu

$$\mathcal{B} = \begin{pmatrix} A^T \\ \hline qI_m \end{pmatrix} \in \mathbb{Z}^{(m+n)\times m},$$

és a dir, afegint qI_m, on I_m és la matriu identitat de dimensió m, sota la matriu transposada d'A, A^T. Observem que els generadors són les columnes de A juntament amb els vectors $q(1, 0, \ldots, 0), q(0, 1, 0, \ldots, 0), \ldots, q(0, \ldots, 0, 1)$.

Finalment, per tal de descartar les files linealment dependents, triangulem superiorment $\mathcal{B}$, obtenint una matriu B, les files de la qual seran una base del reticle q-ari.

L'algorisme implementat es detalla a l'Algorisme 4.

4.1 Elecció dels paràmetres

4.1.1 Com escollir M

Com ja comentàvem a la Secció 2.3.1, als experiments de [14, 15] observen una major eficiència de la tècnica d'encabiment per a $M = 1$, tot i que també proven factors més grans.

En aquesta anàlisi, estudiem $M = 1$ i també $M = 5$, ja que en alguns casos hem observat una major eficiència en la resolució del LWE per a $M = 5$. Aquesta reducció del temps d'execució pot deure's al fet que, per a certes σ, $\|e\|$ és més propera a 5 que a 1, d'acord amb alguns autors [12, 13], que suggereixen prendre $M = \|e\|$.

4.1.2 Com escollir m

Sabem que cada error e_i, $i = 1, \ldots, m$, segueix una distribució gaussiana discreta de mitjana zero i de desviació estàndard σ. Aleshores $\|e\|^2$ segueix una distribució $\sigma^2 \cdot \chi_m^2$ (χ_m^2 denota la distribució khi quadrat amb m graus de llibertat), de manera que podem estimar $\|e\|^2 \approx m\sigma^2$ i per tant $\|e\| \approx \sqrt{m}\sigma$ [15].

Algorisme 4 Tècnica d'encabiment de Kannan per resoldre el problema LWE

1: Introduir un parell LWE $(A, b) \in (\mathbb{Z}_q^{m\times n}, \mathbb{Z}_q^m)$, on $b \equiv As + e \pmod q$.
2: Construir un conjunt de generadors del reticle q-ari representat per la matriu $\mathcal{B}$.
3: Triangular superiorment $\mathcal{B}$ per obtenir una matriu B, les files de la qual són una base del reticle q-ari.
4: Aplicar la tècnica d'encabiment a B amb b i el factor M, augmentant la matriu una dimensió:

$$B' = \begin{pmatrix} B & 0 \\ b & M \end{pmatrix} \in \mathbb{Z}^{(m+1)\times(m+1)}.$$

5: Aplicar l'algorisme LLL a B' per obtenir una base LLL-reduïda B'_{LLL}.
6: **for** $v \in B'_{LLL}$ **do**
7: **if** v acaba en M **then**
8: Guardar v, ja que volem trobar (e, M) (vector curt).
9: **end if**
10: **end for**
11: Seleccionar el(s) vector(s) amb norma mínima. Els candidats a vector d'error e, e_{trobat}, seran aquests sense la darrera component.
12: **for** e_{trobat} **do**
13: Calcular el candidat a vector secret s, s_{trobat}, fent eliminació gaussiana al sistema $(b - e_{trobat}) = As_{trobat}$.
14: **if** $s = s_{trobat}$ **then**
15: Retornar $s \in \mathbb{Z}_q^n$ i $e \in \mathbb{Z}_q^m$.
16: **end if**
17: **end for**

Sigui R un reticle de dimensió m i R' un de dimensió $m + 1$ (el reticle ampliat). Denotem per $\lambda_1(R)$ la norma del vector més curt del reticle R, per $\lambda_1(R')$ la norma del vector més curt de R' i per $\lambda_2(R')$ la del segon vector més curt de R'. Tot seguit seguirem l'anàlisi de [17]. Suposant que σ és prou petita, es pot assumir que

$$\|e\| << \lambda_2(R') \approx \lambda_1(R).$$

D'una banda, es pot estimar $\lambda_1(R) \approx \sqrt{\frac{m}{2\pi e}} q^{1-\frac{n}{m}}$. D'altra banda, per a $M = 1$ podem aproximar $\lambda_1(R') \approx \|(e, M)\| \approx \sqrt{m}\sigma$. El nostre objectiu és maximitzar la bretxa de separació γ entre el segon i el primer vector més curt al problema uSVP del reticle R'

$$\gamma(m) = \frac{\lambda_2(R')}{\lambda_1(R')} \approx \frac{\lambda_1(R)}{\lambda_1(R')} \approx \frac{\sqrt{\frac{m}{2\pi e}} q^{1-\frac{n}{m}}}{\sqrt{m}\sigma},$$

per la qual cosa obtenim la condició

$$\sigma \ll q^{1-\frac{n}{m}} \tag{4.1}$$

o, equivalentment, $\sigma/q \ll q^{-n/m}$. Observem que, si $m \gg n$, aleshores $q^{1-\frac{n}{m}}$ tendeix a q i tindrem un marge més ampli per escollir σ. En canvi, si $m \gtrsim n$, aleshores $q^{1-\frac{n}{m}}$ s'apropa a 1 i aquest marge es redueix.

Així doncs, Wang et al. [15] suggereixen que, per a una n donada, la m òptima vingui donada per l'expressió

$$m = \lfloor \sqrt{n \log_2(q)/\log_2(\delta)} \rceil, \tag{4.2}$$

on es tria $\delta \in [1.010, 1.011, \ldots, 1.025]$. Nosaltres escollim $\delta = 1.025$ per tal d'obtenir una m més petita i així minimitzar el temps d'execució de l'algorisme.

No obstant això, experimentalment troben l'ajust lineal de m per a $n \geq 45$

$$m = \lfloor 4.82n - 98.7 \rceil, \tag{4.3}$$

tot i que la funció està ben definida per a $n \geq 26$.

4.2 Resultats experimentals

Dediquem aquesta part a comentar els resultats experimentals obtinguts en la implementació de l'Algorisme 4, programat a SageMath. Establim els paràmetres:

- n: dimensió del vector secret s. Estudiem $n \in \{3, 5, 10, 15, 20, 27, 28\}$.
- m: dimensió del vector d'error e. Estudiem valors tals que $m > n$, d'acord amb [15]. Per a $n \in \{3, 5, 10\}$ determinem m segons (4.2). Per a $n \in \{15, 20\}$ prenem $m > n$ aleatòries no massa grans: $m = 30$ i $m = 25$, respectivament. Per a $n \in \{27, 28\}$ determinem m segons (4.3), ja que altrament els valors serien massa grans.
- q: nombre primer immediatament posterior a n^2.
- $M \in \{1, 5\}$.
- $\sigma/q \approx 0.10 \cdot q^{-n/m}$ per satisfer (4.1).

Recollim els resultats en dues taules: la Taula 2 mostra els casos en què es troben les claus del criptosistema i la Taula 3 mostra alguns casos en què no es troben.

n	3	5	10	15	20	27	28
m	17	26	43	30	25	31	36
q	11	29	101	227	401	733	787
σ/q	0.07	0.05	0.03	0.007	0.0008	0.0005	0.0006
M	5	5	5	1	1	1	5
Troba s?	SÍ	SÍ	SÍ	SÍ	SÍ	SÍ	SÍ
Temps LLL (s)	12.731	230.109	7690.652	3007.487	318.800	535.374	3556.372

Taula 2: Casos en què es troba la clau secreta s.

n	3	20	27	28
m	17	25	31	36
q	11	401	733	787
σ/q	0.2	0.002	0.0008	0.001
M	5	1	1	5
Troba s?	NO	NO	NO	NO
Temps LLL (s)	14.371	307.891	624.992	5197.087

Taula 3: Casos en què no es troba la clau secreta s.

Observem a la Taula 2 que la separació entre n i m és directament proporcional a la mida relativa de l'error σ/q. Quan aquesta separació és petita, com al parell ($n = 20, m =$

25), aleshores la mida relativa de l'error σ/q esdevé petita. Això ens condueix a σ molt petites i per tant errors molt petits. Per a una n donada, podríem augmentar la separació entre m i n ($m \gg n$) per obtenir σ més grans, però ja no respectaríem (4.2) ni (4.3), i a més el temps d'execució seria impracticable.

L'equació (4.2) és útil per a n més petites (fins $n = 10$), ja que hi ha certa separació entre n i m, permetent que σ/q sigui raonablement gran i pugui trobar-se la clau. En canvi, per a n més grans ($n \geq 26$), fem servir (4.3) per calcular m. Tot i que aquesta equació ajusta m perquè sigui viable computacionalment almenys fins a $n = 28$, la separació entre n i m segueix sent petita, proporcionant valors molt petits de σ/q. Per tant, la disminució de σ/q a valors molt reduïts dificulta el trencament del criptosistema. A la Taula 3 trobem quatre instàncies on prendre un valor de σ/q més gran de l'establert segons (4.1) impedeix trobar la clau secreta. A la primera el valor de m s'ha trobat segons (4.2), a la segona aleatòriament tal que $m > n$, i a la tercera i quarta segons (4.3). En tots els casos els resultats mostren que si σ/q no és suficientment petita, no es troba la clau, confirmant així que la relació entre σ/q i la separació entre n i m influeix sobre l'eficàcia de l'atac.

Paral·lelament, hem comprovat quin factor d'encabiment M, si 1 o 5, és més eficient computacionalment per als diversos valors de n. Com ja hem comentat, per a n més petites es permeten σ més grans, donant lloc a errors més grans. Llavors és més probable que tinguem $M = 5 \approx \|e\|$, i per això l'algorisme és més eficient. Per a n més elevades, però, σ només pot ser molt petita i els errors molt petits, essent més probable que $M = 1 \approx \|e\|$. Concloem, doncs, que per a ambdós valors és possible trobar s, i que la diferència radica en el temps d'execució de l'algorisme.

L'Algorisme 4 resulta extremadament eficaç per resoldre el problema LWE en dimensions n no gaire elevades, ja que la probabilitat d'èxit observada és d'1 quan es respecta la relació $\sigma/q \approx 0.10 \cdot q^{-n/m}$. Això significa que a la línia 13 del bucle **for** només es troba un vector s_{trobat}, que acaba sent precisament la clau secreta s. Si no es mantenen les condicions òptimes per a n, m i σ/q, aleshores la probabilitat d'èxit és zero, ja que la norma del vector d'error e resulta ser més gran que la del vector e_{trobat}, obtingut a partir dels vectors de la base LLL-reduïda amb norma mínima. Això no és un problema de quedar-nos amb els vectors amb norma mínima: encara que no es faci, l'algorisme LLL és incapaç de retornar (e, M) i per tant de retornar un $e_{trobat} = e$.

5 Conclusions

Hem cobert diverses àrees importants en aquest treball. Primerament, hem presentat les nocions fonamentals de reticles, on destaca el quocient de Hadamard com a eina per avaluar l'ortogonalitat de la base d'un reticle. Els reticles es poden aplicar a la criptografia aprofitant la complexitat de problemes com el Shortest Vector Problem (SVP) i el Closest Vector Problem (CVP), especialment en dimensions n elevades del reticle.

Hem vist com dos algorismes, l'LLL i el d'arrodoniment de Babai, poden comprometre la seguretat dels criptosistemes basats en reticles, amb exemples concrets. Aquests algorismes resolen l'apprSVP i l'apprCVP, respectivament, amb un factor exponencial i temps polinòmic. Així doncs, el seu rendiment empitjora considerablement amb la dimensió n i, en el cas del de Babai, és altament ineficient per a bases "dolentes".

Tot seguit, ens hem centrat en el criptosistema Regev, basat en la dificultat del problema Learning With Errors (LWE). L'elecció adequada de paràmetres minimitza la probabilitat d'errors de desxifratge i maximitza tant l'eficiència com la seguretat. Per què es pensa que el problema LWE és difícil? Resulta que resoldre el LWE implicaria resoldre certs problemes de reticles que es consideren molt difícils fins i tot amb una base "bona". A més, resoldre el LWE implicaria l'existència d'un algorisme quàntic eficient per resoldre l'apprSVP, un repte ara per ara insuperable. D'altra banda, provem que ambdues versions de LWE, la de cerca i la de decisió, són equivalents.

Finalment, hem desenvolupat un algorisme que aplica la tècnica d'encabiment de Kannan i l'algorisme LLL per resoldre el LWE, ja que aquesta tècnica redueix el LWE a l'uSVP. Malgrat comptar amb recursos computacionals limitats, hem aconseguit atacar amb èxit el LWE per a dimensions $n \in \{3, 5, 10, 15, 20, 27, 28\}$ sota determinades condicions per a l'error relatiu σ/q i el nombre d'equacions m, que garanteixen que es trobi e com a solució de l'uSVP i per tant la clau secreta s. Per a n més elevades, però, l'algorisme és impracticable. La pregunta sobre la seguretat completa d'aquests criptosistemes segueix oberta, deixant espai a futures investigacions.

Apèndix

Algorisme 4

In [ ]:
```
from sage.stats.distributions.discrete_gaussian_integer import DiscreteGaus
sianDistributionIntegerSampler
import pandas as pd
import timeit
```

In [ ]:
```
def GramSchmidt(n, B, V):
    V[0] = B[0]
    for i in range(1, n): # i de 1 a n-1
        sum = zero_vector(QQ, n)
        for j in range(0, i): # j de 0 a i-1
            mu = B[i].dot_product(V[j])/(V[j].dot_product(V[j]))
            sum += mu * V[j]
        V[i] = B[i] - sum

    return V
```

In [ ]:
```
def LLL(n, LLL, V):
    k = 1
    while k <= n-1:
        for j in range(0, k): # j de 0 a k-1

            # Reducció de mida
            mu = LLL[k].dot_product(V[k-1-j])/(V[k-1-j].dot_product(V[k-1-
j]))
            LLL[k] -= round(mu) * LLL[k-1-j]

            # Actualitzem la base GS associada a la base LLL
            V = GramSchmidt (n, LLL, V)

        # Condició de Lovász
        mu2 = LLL[k].dot_product(V[k-1])/(V[k-1].dot_product(V[k-1]))
        lovasz = (0.75 - pow(mu2, 2)) * (V[k-1].dot_product(V[k-1]))
        if  V[k].dot_product(V[k]) >= lovasz:
            k = k+1


        else: # Intercanviar v_(k-1) i v_k
            aux = LLL[k]
            LLL[k] = LLL[k-1]
            LLL[k-1] = aux
            k = max(k-1, 1) # evitem que k sigui menor que 1

            # Actualitzem la base GS
            V = GramSchmidt (n, LLL, V)

    return LLL
```

In [ ]:
```
def canvi(p, q):
    # Calculem un representant de p mod q en el rang (-q/2 , q/2]
    r = p.lift()
    if r > q/2:
        r -= q

    return r
```

INPUT: una instància LWE. Generem un enter n, un primer q immediatament posterior a n^2 i un enter m. Generem un vector secret aleatori $s \in \mathbb{Z}_q^n$ i una matriu $A \in \mathbb{Z}_q^{m \times n}$. Generem un vector $e \in \mathbb{Z}_q^m$ d'errors extrets d'una distribució discreta gaussiana centrada al zero i de desviació estàndard σ. Si e és nul, salta una advertència i en creem un altre. Canviem els representants d'e a l'interval $(-q/2, q/2]$. Per últim, generem el vector $b = As + e \in \mathbb{Z}_q^m$.

In [ ]:
```
n = 3

q = n^2
while not q.is_prime():
    q += 1

# Generem l'anell Z/qZ
Z = IntegerModRing(q)

m = 17

s = vector(Z, [Z.random_element() for _ in range(n)])

# Generem m vectors de longitud n i els guardem en una matriu A
a = [vector(Z, [Z.random_element() for _ in range(n)]) for _ in range(m)]
A = matrix(a)

b = A * s

# ERRORS
sigma = 0.07*q
D = DiscreteGaussianDistributionIntegerSampler(sigma=sigma, c=0)
e = vector(Z, [D() for _ in range(m)])

# Si el vector e és nul, generem advertència i en creem un altre
while e.is_zero():
    print("Advertència: el vector d'errors és zero.")
    e = vector(Z, [D() for _ in range(m)])
```

```
# Canviem els representants a [-q/2, q/2]
e = vector([canvi(i, q) for i in e])

b += e
```

PAS 1: construir la base del reticle q-ari. Construïm la matriu qI_m. Transposem la matriu A, canviant-la als enters, i concatenem la matriu transposada A^T amb qI_m a sota: serà una matriu $B \in \mathbb{Z}^{(m+n)\times m}$. Eliminem els vectors linealment dependents de B triangulant-la superiorment amb el mètode B.echelon_form(transformation=True). Així, ens quedem amb una matriu $B \in \mathbb{Z}^{m \times m}$ de vectors linealment independents.

In [ ]:
```
qidentitat = q*identity_matrix(m,ZZ)

A_enters = A.change_ring(ZZ) # canviem als enters (si no, operarà en Z/qZ i
a B donarà la matriu nul·la)
A_t = A_enters.transpose()
B = A_t.stack(qidentitat)

# Eliminem els vectors linealment dependents de B
B = B.echelon_form(transformation=True)[0]

# Ens quedem amb la submatriu m x m
valors = list(range(m))
B = B.matrix_from_rows_and_columns(valors, valors)
```

PAS 2: reescalar B expandint en una dimensió. Afegim un 0 al final de cada fila de B i una nova fila amb el vector (b, M). Ens queda una matriu $B_{ext} \in \mathbb{Z}^{(m+1)\times(m+1)}$.

In [ ]:
```
noves_files = [B[i].list() + [0] for i in range(m)]
B_ext = matrix(noves_files)

M = 5
b_amb_M = b.list() + [M]
B_ext = B_ext.stack(vector(ZZ, b_amb_M))
```

PAS 3: apliquem l'algoritme LLL. Calculem la base V de Gram-Schmidt associada a la base B_{ext}. Mesurem el temps d'execució de la funció LLL. Cridem l'algoritme LLL i obtenim una base LLL-reduïda, LLL_{base}.

In [ ]:
```
# PAS 3: apliquem l'algoritme LLL
B_llista = B_ext.rows()

dim = m + 1

LLL_base = B_llista.copy()

# Calculem la base V de Gram-Schmidt associada a la base B_ext
V = [None] * dim
V = GramSchmidt(dim, B_llista, V)

# Mesurem el temps d'execució de la funció LLL
temps_execucio = timeit.timeit(lambda: LLL(dim, LLL_base, V), number=1)

# Cridem l'algoritme LLL
LLL_base = LLL(dim, LLL_base, V)
```

Trobem vectors curts i.e vectors de LLL_{base} acabats en M, ja que els candidats a vector e seran aquests menys M. Els guardem a una llista *resultats_error*.

In [ ]:
```
resultats_error = []
for vec in LLL_base:
     if vec[-1] == M:
        resultats_error.append(vec[:-1])

print("Els candidats a vectors error són:", resultats_error)
```

Calculem les normes dels vectors a *resultats_error*, trobem la norma mínima i seleccionem el(s) vector(s) amb norma igual a la norma mínima.

In [ ]:
```
# Calcular les normes dels vectors en resultats_error
normes = []
for vec in resultats_error:
    normes.append((float(vec.norm())))

# Trobar la norma mínima
norma_minima = min(normes)
print(norma_minima)

# Seleccionar els vectors amb norma igual a la norma mínima
resultats_error = [vec for vec, norma in zip(resultats_error, normes) if no
rma == norma_minima]
print("Els candidats a vectors error (norma mínima) són:", resultats_error)
```

PAS 4: obtenir el vector s aplicant eliminació gaussiana a $(b - e_{trobat}) = As$ amb cada vector d'error candidat e_{trobat}. Intentem resoldre el sistema. Si aquest té solució, guardem els possibles vectors secrets com a s_{trobat} a una llista *resultats_s*. Si no en té, aleshores salta un avís.

In [ ]:
```
resultats_s = []

for e_trobat in resultats_error:

    b_sense_e = b-e_trobat

    # Formem la matriu ampliada d'A
    A_ampliada = A.augment(b_sense_e, subdivide=True)

    # Apliquem eliminació gaussiana per triangular A ampliada
    Uc = A_ampliada.echelon_form()
```

```
    # Extraiem la matriu triangular superior U i el vector ampliat c
    U = Uc[:,:-1]
    c = Uc[:,-1]

    # Intentem resoldre el sistema As=(b-e) i.e Us = c
    try:
        s_trobat = U.solve_right(c)
        vector_s = vector([s_trobat[i][0] for i in range(n)]) # passem-lo a
vector
        resultats_s.append(vector_s)

    except ValueError as e:
        if "matrix equation has no solutions" in str(e):
            print("El sistema no té solució")
        else:
            raise

# Després de provar tots els candidats a error, indicar els candidats a s t
robats
print("Els possibles vectors s són:", resultats_s)
```

Comprovem per a cada s_{trobat} si coincideix amb s. Si La probabilitat de trencar el criptosistema es defineix com:

$$\text{probabilitat} = \begin{cases} \frac{1}{\text{len(resultats\_s)}}, & \text{si } s_{trobat} = s; \\ 0, & \text{si } s_{trobat} \neq s. \end{cases}$$

In [ ]:
```
for s_trobat in resultats_s:
    if s_trobat == s:
        print("El vector secret s està entre els candidats")
        probabilitat = float(1/len(resultats_s))
        print("Probabilitat de trencar el criptosistema:", probabilitat)
    else:
        print("El vector secret s NO està entre els candidats")
        probabilitat = 0
```

Creem un DataFrame de *pandas* per guardar els diversos paràmetres, vectors, resultats i temps d'execució de l'LLL. Generem un nom d'arxiu basat en els valors de n, m, q, M i σ i el guardem en un arxiu CSV.

In [ ]:
```
data = {
    'Variable': ['n', 'm', 'q', 'A', 's', 'b', 'e', 'M', 'sigma', 'Resultat
s Error', 'Resultats s', 'Probabilitat','Temps execució'],
    'Valor': [n, m, q, A, s, b, e, M, sigma, resultats_error, resultats_s,
probabilitat, temps_execucio]
}

df = pd.DataFrame(data)

# Generem un nom d'arxiu
filename = f'dades_taula_{n}-{m}-{q}-{M}-{sigma:.4f}.csv'

# Guardar el DataFrame en un arxiu CSV amb el nom generat
df.to_csv(filename, index=False)

# Mostrar el DataFrame
print(df)
```

Referències

[1] M. Ajtai, "Generating hard instances of lattice problems," *Complexity of computations and proofs*, vol. 13, pp. 1-32, 2004.

[2] M. Ajtai and C. Dwork, "A public-key cryptosystem with worst-case/average-case equivalence," *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pp. 284-293, 1997.

[3] M. Hartmann, "The Ajtai-Dwork cryptosystem and other cryptosystems based on lattices,"Master's thesis, Institute of Mathematics, University of Zurich, 2015.

[4] J. Hoffstein, J. Pipher, and J. H. Silverman, *An Introduction to Mathematical Cryptography*, 2nd ed. New York: Springer, 2014.

[5] O. Goldreich, D. Micciancio, S. Safra, and J.-P. Seifert, "Approximating shortest lattice vectors is not harder than approximating closest lattice vectors," *Information Processing Letters*, vol. 71, no. 2, pp. 55-61, 1999.

[6] Wikipedia contributors, "Lattice problem", Wikipedia, The Free Encyclopedia. [En línia]. Disponible: `https://en.wikipedia.org/wiki/Lattice_problem`. [Data d'accés: 12-abril-2024].

[7] A. J. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515-534, 1982.

[8] M. Ajtai, R. Kumar, and D. Sivakumar, "A sieve algorithm for the shortest lattice vector problem," *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pp. 601-610, 2001.

[9] D. Micciancio and P. Voulgaris, "Faster exponential time algorithms for the shortest vector problem," *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA '10), pp. 1468-1480, 2010.

[10] R. Kannan, "Improved algorithms for integer programming and related lattice problems," *Proceedings of the 15th ACM Symposium on Theory of Computing* (STOC), pp. 193-206, 1983.

[11] L. Babai, "On Lovász lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1-13, 1986.

[12] S. Galbraith, *Mathematics of Public Key Cryptography*. Cambridge: Cambridge University Press, 2012.

[13] V. Lyubashevsky and D. Micciancio, "On bounded distance decoding, unique shortest vectors, and the minimum distance problem,"S. Halevi, Ed., *Advances in Cryptology - CRYPTO 2009*, CRYPTO 2009, Lecture Notes in Computer Science, vol. 5677, Berlin, Heidelberg: Springer, 2009. https://doi.org/10.1007/978-3-642-03356-8_34.

[14] M. R. Albrecht, R. Fitzpatrick, and F. Göpfert, "On the efficacy of solving LWE by reduction to unique-SVP,ïn H.-S. Lee and D.-G. Han, Eds., *ICISC 2013*, vol. 8565, pp. 293-310, 2014. `https://doi.org/10.1007/978-3-319-12160-4_18`

[15] Y. Wang, Y. Aono, and T. Takagi, cAn experimental study of Kannan's embedding technique for the search LWE problem," *ICICS 2017*, LNCS 10631, pp. 541-553, 2018.

[16] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, p. 34, 2009. Preliminary version in STOC'05.

[17] D. Micciancio and O. Regev, D. Micciancio and O. Regev, "Lattice-based cryptography,"Post-Quantum Cryptography, D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds. Heidelberg: Springer, pp. 147-191, 2009. https://doi.org/10.1007/978-3-540-88702-7_5

[18] O. Regev, "The Learning with Errors Problem," *Journal of the ACM*, vol. 56, no. 6, p. 34, 2009.

[19] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pp. 97-106, 2011.

[20] C. Vincent, *Math 259 Class Notes*, 2019.