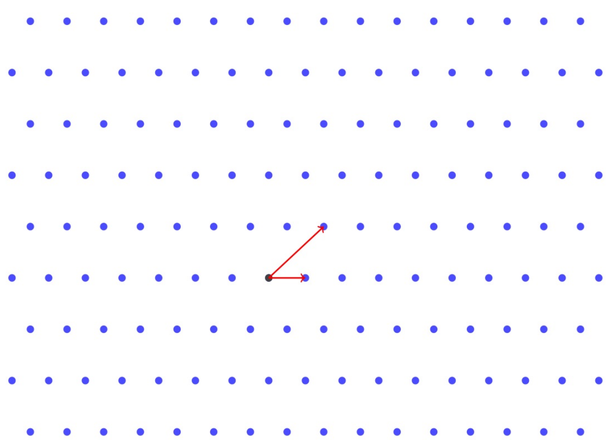
Criptografía post-cuántica basada en retículos

Lección 4: Minicurso Mar del Plata, Noviembre 2025

Syllabus

- **Reticulos: generadores y ecuaciones**
- **Problemas SIS y LWE**
- **Esquema de Lyubashevsky**
- **Reject/Accept Von Neumann's sampling**



Índice

4.6 Reticulos

Un reticulo Λ es un subconjunto discreto de $\mathbb{R}^n$ cerrado por combinaciones lineales con coeficientes enteros. Es decir, si $\mathrm{v}_1, \cdots, \mathrm{v}_k$ son vectores del reticulo y $c_1, \cdots, c_k \in \mathbb{Z}$ entonces el vector

$$c_1 \cdot \mathrm{v}_1 + \cdots + c_k \mathrm{v}_k$$

pertenece al reticulo Λ. Esto en particular implica que el vector cero 0 de $\mathbb{R}^n$ pertenece al reticulo.

NOTE 4.6.0.1

Observar la analogia con los subespacios de un espacio vectorial $\mathbb{V}$ sobre un cuerpo $\mathbb{K}$: los subspacios se definen como subconjutos cerrados respecto las combinaciones lineales usando coeficientes el cuerpo $\mathbb{K}$.

Ser discreto, un concepto topologico, significa que cerca del vector cero 0 no hay ningun otro vector del reticulo.

Ejercicio 4.6.0.2

Dado un numero $\alpha \in \mathbb{R}$ el conjunto de multiplos enteros de α es un reticulo de $\mathbb{R}$.

En cambio el conjunto de numeros racionales $\mathbb{Q}$ no es un reticulo de $\mathbb{R}$. Por que?

4.6.1 Generadores y/o ecuaciones

Tradicionalmente hay dos modos de definir un particular reticulo Λ:

- usando generadores, explicito o por enumeracion,

- usando ecuaciones, modo implicito o por condicion.

Usando **generadores** se define Λ diciendo que Λ es el conjunto de combinaciones lineales enteras de los generadores $\mathrm{b}_1, \cdots, \mathrm{b}_n$. Notar que esto es analogo con el modo de definir un subespacio dando los generadores.

Usando **ecuaciones** se define Λ diciendo que Λ consiste de los vectores v, con coordenadas numeros enteros, que satisfacen las ecuaciones del sistema:

$$\mathbf{A} \cdot \mathrm{v} = 0 \pmod{N}$$

Si el reticulo Λ se define con generadores $\mathrm{b}_1, \cdots, \mathrm{b}_n$ y con la matriz $\mathbf{A}$ entonces necesariamente

$$\mathbf{A} \cdot \mathrm{b}_j = 0 \pmod N$$

para $j = 1, \cdots, n$.

NOTE 4.6.1.1

Si los vectores $\mathrm{b}_1, \cdots, \mathrm{b}_n$ son las columnas de la matriz $\mathbf{B}$ entonces las ecuaciones anteriores expresan la ecuacion:

$$\mathbf{A} \cdot \mathbf{B} = 0 \pmod N$$

Observar tambien que un reticulo Λ admite infinitos modos de ser definido por matrices $\mathbf{A}$ y $\mathbf{B}$.

En el caso de las ecuaciones escrivo $\Lambda := \ker(\mathbf{A})$ y en el caso de generadores escribo $\Lambda := \mathrm{im}(\mathbf{B})$

Ejercicio 4.6.1.2

Sea Λ el reticulo definido por la ecuacion

$$3x + 5y = 0 \pmod 7 \ .$$

Encontrar una matriz $\mathbf{B}$ tal que $\Lambda = \mathrm{im}(\mathbf{B})$.

Ejercicio 4.6.1.3

Dado un reticulo $\Lambda := \ker(\mathbf{A}) = \mathrm{im}(\mathbf{B})$ verificar que existen infinitas matrices $\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}$ tales que:

$$\Lambda := \ker(\widetilde{\mathbf{A}}) = \mathrm{im}(\widetilde{\mathbf{B}})$$

Ademas del algebra hay tambien una explicacion geometrica de este hecho.

4.6.2 Problemas SIS y LWE : Ajtai one-way function

Los dos problemas computacionalmente dificiles mas utilizados en la criptografia basada en reticulos son :

- Short Integer Solution (SIS)
- Learning With Errors (LWE)

El problema SIS es el de encontrar un vector short $\mathbf{s}$ de un reticulo Λ. Normalmente, en la literatura, se encuentra utilizando la matriz $\mathbf{A}$ que define el reticulo y se enuncia como: Encontrar $\mathbf{s}$ tal que

$$\mathbf{A} \cdot \mathbf{s} = 0 \pmod N$$

La version no homogenea de SIS es, dado $\mathbf{t}$ encontrar una vector short $\mathbf{s}$ tal que:

$$\mathbf{A} \cdot \mathbf{s} = \mathbf{t} \pmod N \tag{1}$$

Esta version no homogenea fue propuesta por el matematico Ajtai como función one-way: pensando a la matriz $\mathbf{A}$ como una funcion cuyo dominio $\mathcal{S}$ son los vectores short la ecuación anterior expresa el problema de encontrar una pre-imagen del vector $\mathbf{t}$.

4.6.3 Claves publicas y privadas

En la criptografia basada en reticulos la **clave publica** suelen ser la matriz $\mathbf{A}$ y el vector $\mathbf{t}$ en la ecuación fundamental (1). El modulo N suele ser parte del dominio de parametros del esquema.

La **clave secreta** es o el vector short $\mathbf{s}$ o una base de vectores short en forma de matriz $\mathbf{B}$ que permite encontrar vectores short soluciones de la ecuación fundamental (1).

4.7 Firmas digitales segun V. Lyubashevsky

4.7.1 Protocolo Sigma de Identificacion

Tanto el Probador como el Verificador ($\mathcal{V}$) conocen $\mathbf{A}, \mathbf{t}$ y el modulo N.

$\mathcal{P}$ quiere demostrar a $\mathcal{V}$ que conoce un short vector $\mathbf{s}$, que resuelve la ecuación fundamental (1), sin revelarlo.

4.7.1.1 Protocolo Sigma de Lyubashevsky

$$
\begin{array}{rcl}
\mathcal{P}robador\,(\mathbf{A}, \mathbf{s}, \mathbf{t} = \mathbf{A} \cdot \mathbf{s}) & & \mathcal{V}erificador\,(\mathbf{A}, \mathbf{t}) \\
\mathrm{y} \xleftarrow{\$} \mathrm{ShortMasks} & & \\
\mathbf{I} \leftarrow \mathbf{A} \cdot \mathrm{y} & & \\
 & \xrightarrow{\quad\quad\quad \mathbf{I} \quad\quad\quad} & \\
 & & \mathrm{c} \xleftarrow{\$} \mathrm{ShortCoeff} \\
 & \xleftarrow{\quad\quad\quad \mathrm{c} \quad\quad\quad} & \\
\mathrm{z} \leftarrow \mathrm{y} + \mathrm{c} \cdot \mathbf{s} & & \\
\textbf{Reject/Accept}\,\mathrm{z}: & & \\
1)\ \mathrm{z\ es\ short} & & \\
2)\ \mathrm{z\ v.a.\ indep.\ di}\ \mathrm{c} \cdot \mathbf{s} & & \\
 & \xrightarrow{\quad\quad\quad \mathrm{z} \quad\quad\quad} & \\
 & & \mathrm{identifica\ si}: \\
 & & \quad \mathbf{A} \cdot \mathrm{z} = \mathbf{I} + \mathrm{c} \cdot \mathbf{t} \\
 & & \quad \mathrm{z\ es\ short}
\end{array}
$$

NOTE 4.7.1.2

La eleccion de las distribuciones de probabilidades de $\mathbf{A}$, y, c, $\mathbf{s}$ es crucial para la seguridad y la eficiencia (uso realistico) del protocolo. Por ejemplo, cuantas iteraciones son necesarias antes de aceptar z ? si son demasiadas el resultado seria un tiempo de espera alto para completar la identificacion.

Es en esta eleccion de las distribuciones donde se deben combinar adecuadamente los problemas SIS y LWE mencionados anteriormente. Por ejemplo, que la clave publica $(\mathbf{A}, \mathbf{t})$ sea indistinguible de un par completamente aleatorio $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{t}})$ es equivalente a LWE (by Regev's).

Los parametros, las desigualdades, las normas, que controlan los conjuntos $\mathrm{ShortMask}, \mathrm{ShortCoeff}$ tienen tambien un fuerte impacto en el balance eficiencia/seguridad i.e. cuanto grandes, en bytes, van a ser las claves publicas y privadas.

4.7.2 Firma digital : Fiat-Shamir

4.7.2.1 KeyGen (1^λ)

$\mathbf{A} \xleftarrow{\$} \text{Reticulos}$ ▷ *(crea el reticulo)*

$\mathsf{sk} \leftarrow \mathsf{s} \xleftarrow{\$} \text{ShortSecrets}$ ▷ *(crea la clave secreta)*

$\mathbf{t} \leftarrow \mathbf{A} \cdot \mathsf{sk}$

$\mathsf{pk} \leftarrow (\mathbf{A}, \mathbf{t})$ ▷ *crea la clave publica*

return $(\mathsf{sk}, \mathsf{pk})$ ▷ *retorna el par de claves*

4.7.2.2 $\mathsf{Sign}_{\mathsf{sk}}(m)$

$\mathrm{y} \xleftarrow{\$} \text{ShortMasks}$

$\mathbf{I} \leftarrow \mathbf{A} \cdot \mathrm{y}$ ▷ *crea el commitment*

$\mathrm{c} \leftarrow \mathbf{Hash}(\mathbf{I}||\mathsf{pk}||m)$ ▷ *crea el challenge removiendo el Verificador*

$\mathrm{z} \leftarrow \mathrm{y} + \mathrm{c} \cdot \mathsf{sk}$ ▷ *crea una parte del firma*

$$\mathbf{Reject/Accept}\,\mathrm{z} : \begin{array}{l} 1)\ \mathrm{z\ es\,short} \\ 2)\ \mathrm{z\ v.a.\ indep.\ di\ c} \cdot \mathsf{sk} \end{array}$$

$\sigma \leftarrow (\mathbf{I}, \mathrm{z})$ ▷ *firma final*

return σ

4.7.2.3 $\mathsf{Vrfy}_{\mathsf{pk}}(m, \sigma)$

$(\mathbf{I}, \mathrm{z}) \leftarrow \sigma$ ▷ *desempaqueta la firma*

si z no es short **return** False ▷ *primer control de la firma*

$(\mathbf{A}, \mathbf{t}) \leftarrow \mathsf{pk}$ ▷ *desempaqueta la clave publica*

$\mathrm{sI} \leftarrow \mathbf{A} \cdot \mathrm{z} - \mathbf{Hash}(\mathbf{I}||\mathsf{pk}||m) \cdot \mathbf{t}$ ▷ *calculo del supuesto commitment*

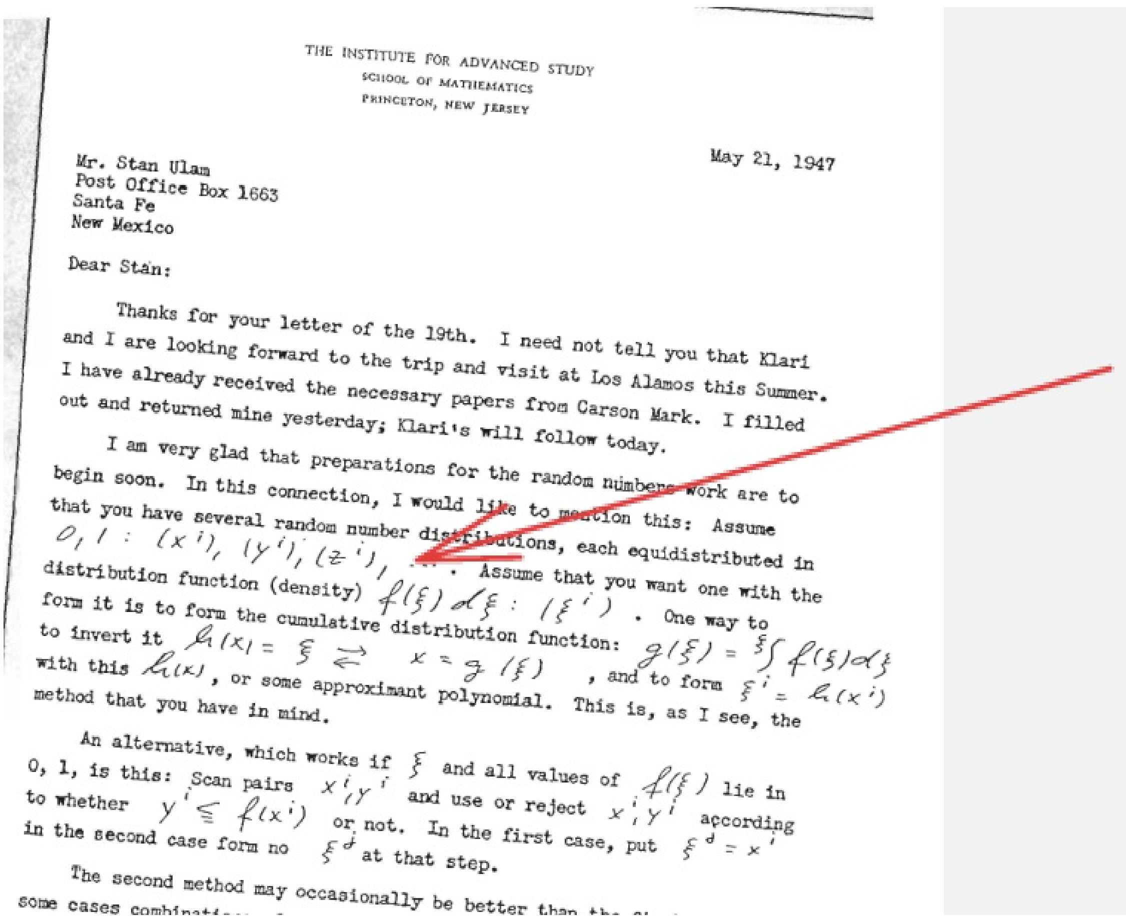
$$\mathbf{return} : \begin{array}{ll} \text{True} & \text{si}\quad \mathrm{sI} = \mathbf{I}\,; \\ \text{False} & \text{en caso contrario} \end{array}$$

4.7.3 Reject-Accept de acuerdo a J. Von Neumann

El segundo problema que aborda Von Neumann en [vN51] es el de crear una muestra $Z_1, Z_2, \cdots$ que siga una distribución dada $\mathcal{D}$ **utilizando** muestras de otras variables aleatorias X, Y con distribuciones conocidas (por ejemplo, la uniforme).

Podría ocurrir que, para generar la muestra Z, en lugar de usar solo dos variables X, Y, se necesiten infinitas (como en el caso del $\mathbb{Z}$-sampler, el último ejemplo mencionado).

Es interesante ver que el propio Von Neumann escribe *several random variables...* cuando le explica a Stan Ulam el método de accept-reject en una carta del 21 de mayo de 1947.



THE INSTITUTE FOR ADVANCED STUDY
SCHOOL OF MATHEMATICS
PRINCETON, NEW JERSEY

May 21, 1947

Mr. Stan Ulam
Post Office Box 1663
Santa Fe
New Mexico

Dear Stan:

Thanks for your letter of the 19th. I need not tell you that Klari and I are looking forward to the trip and visit at Los Alamos this Summer. I have already received the necessary papers from Carson Mark. I filled out and returned mine yesterday; Klari's will follow today.

I am very glad that preparations for the random numbers work are to begin soon. In this connection, I would like to mention this: Assume that you have several random number distributions, each equidistributed in $0, 1$: $(x^i), (y^i), (z^i), \ldots$. Assume that you want one with the distribution function (density) $f(\xi)\,d\xi$: (ξ^i). One way to form it is to form the cumulative distribution function: $g(\xi) = \int^{\xi} f(\xi)\,d\xi$ to invert it $h(x) = \xi \rightleftarrows x = g(\xi)$, and to form $\xi^i = h(x^i)$ with this $h(x)$, or some approximant polynomial. This is, as I see, the method that you have in mind.

An alternative, which works if ξ and all values of $f(\xi)$ lie in $0, 1$, is this: Scan pairs x^i, y^i and use or reject x^i, y^i according to whether $y^i \leq f(x^i)$ or not. In the first case, put $\xi^j = x^i$; in the second case form no ξ^j at that step.

The second method may occasionally be better than the
some cases

El formalismo del *reject/accept* es consecuencia directa de la probabilidad condicional y de su cálculo.

Quizás didácticamente, para entender, conviene comenzar con la **siguiente pregunta**:

Sean X, Y dos variables independientes y Z otra variable tal que:

$$\text{Prob}(Z \in A) = \text{Prob}(X \in A \,|\, (X, Y) \in S) \tag{2}$$

¿ Cuál es la distribución $\mathcal{D}$ de Z?

Esta pregunta debería conducir al *reject/accept* para muestrear la distribución $\mathcal{D}$ utilizando las muestras de X, Y, es decir, a partir de la muestra

$$(X_1, Y_1), (X_2, Y_2), \cdots$$

se producirá la muestra

$$Z_1, Z_2, \cdots$$

aceptando $X = Z$ cuando $(X_i, Y_i) \in S$, es decir, la distribución de Z sería la de X condicionada a la aceptación...

Luego, en un segundo momento, siempre para entender, uno se pregunta cómo determinar S para obtener una distribución $\mathcal{D}$ dada.

Ejemplo 1: X, Y uniformes e independientes en $[0,1]$ y $S \subset [0,1] \times [0,1]$ definido por

$$S := \{(x, y) : y \leq F(x)\}$$

¿ Cuál es entonces la distribución $\mathcal{D}$ de Z en la ecuación (2)?
He aquí el cálculo:

$$\begin{aligned} \text{Prob}(Z \in A) = \text{Prob}(X \in A \,|\, (X,Y) \in S) = \\ = \frac{\text{Prob}(X \in A \text{ and } (X,Y) \in S)}{\text{Prob}((X,Y) \in S)} \end{aligned}$$

Ahora notamos que

$$\begin{aligned} \text{Prob}((X,Y) \in S \text{ and } X \in A) = \int_{\substack{Y \leq F(X) \\ X \in A}} dx\, dy = \\ = \int_A F(x)\, dx \end{aligned}$$

Por lo tanto,

$$\text{Prob}(Z \in A) = \frac{\int_A F(x)\, dx}{c}$$

donde $c = \int_{[0,1]} F(x)\, dx$. Así, $\mathcal{D}$ tiene densidad $\frac{F(x)}{c}$.

Entonces, si quiero muestrear de una distribución $\delta(x)$, hago

$$F(x) := \delta(x)$$

y el *reject-accept* produce la muestra deseada Z con densidad δ.

Ejemplo 2: X, Y independientes en $[0,1]$, Y uniforme y X con densidad δ_X, y S definido por

$$S := \{(x,y) : y \leq F(x)\}$$

Procediendo como antes, se obtiene que la densidad de Z es

$$\frac{F(x)\delta_X(x)}{c}$$

donde $c = \int_{[0,1]} F(x)\delta_X(x)\,dx$.

Por lo tanto, si quiero muestrear Z con una densidad dada $g(x)$, despejo $F(x)$ de:

$$F(x)\delta_X(x) = g(x)$$

y entonces resulta el cociente $\frac{g(x)}{\delta_X(x)}$ donde en el numerador se observa la distribución *objetivo/deseada* y en el denominador la distribución *fuente/conocida* .

Esto da lugar tambien al modo de hablar *aceptamos* el resultado Z_i con probabilidad $\frac{g(x)}{\delta_X(x)}$ y permite una implementación practica del reject/accept usando una v.a. Bernoulli $\mathcal{B}(p)$ con $p = \frac{g(x)}{\delta_X(x)}$.

Por qué la criptografía basada en retículos es un buen candidato para la PQC ?

1. Reducción del Peor Caso al Caso Promedio (Ajtai, 1996)

- **Teorema (Ajtai):** Resolver problemas del *caso promedio* (e.g., Short Integer Solution, SIS) es tan difícil como resolver problemas del *peor caso* (e.g., aproximar el Shortest Vector Problem, SVP).

$$\text{SIS (caso promedio)} \leq \text{SVP(peor caso)}$$

- **Implicaciones:** Los esquemas criptográficos basados en SIS/LWE heredan la robustez de los problemas de retículos en el peor caso, que se consideran difíciles incluso para computadores cuánticos.

2. Construcciones Eficientes (Babai y otros)

- **Algoritmo de Babai (1986):** Proporciona un método eficiente para resolver el *Closest Vector Problem (CVP)* en retículos "bien estructurados", pero solo para soluciones *aproximadas.*
- **Relevancia criptográfica:** Los esquemas criptográficos usan *retículos aleatorios* donde CVP/SVP siguen siendo difíciles (incluso para algoritmos cuánticos).
- **Eficiencia:** Esquemas como NTRU o FALCON aprovechan retículos estructurados (e.g., retículos construidos usando estructuras matematicas como extensiones de anillos, cocientes de anillos de polinomios, modulos sobre anillos, etc) para tamaños de clave prácticos.

3. Resistencia Cuántica

- **Sin aceleración cuántica conocida:** A diferencia de la factorización/logaritmo discreto (rotos por el algoritmo de Shor), problemas como **LWE** o **SVP** solo admiten mejoras sub-exponenciales cuánticas (e.g. algoritmo de Grover).
- **Mejores ataques conocidos:** Los algoritmos cuánticos reducen la complejidad de 2^n a $2^{n/2}$ (insuficiente para retículos con $n \geq 256$).

4. Otras Ventajas Clave

- **Versatilidad:** Permite cifrado, firmas digitales (e.g., Dilithium), FHE (cifrado homomórfico), Zero Knowledge proofs y más.
- **Seguridad demostrable:** Basada en problemas bien estudiados (SVP, LWE) sin trapdoors conocidas.
- **Flexibilidad:** Escalable a distintos niveles de seguridad (e.g., Niveles 1–5 de NIST) ajustando los parametros (si bien no sea para nada sencillo ajustar los parametros...).

Referencias

[LV12] V. Lyubashevsky, *Lattice signatures without trapdoors*, Advances in cryptology—EUROCRYPT 2012, 738–755, Lecture Notes in Comput. Sci., 7237, Springer, Heidelberg, https://eprint.iacr.org/2011/537.pdf

[Re06] O. Regev, *Lattice-based Cyptography*, https://www.iacr.org/archive/crypto2006/41170129/41170129.pdf

[Pe16] C. Peikert, *A Decade of Lattice Cryptography*, https://eprint.iacr.org/2015/939.pdf

[vN51] John von Neumann. *Various techniques used in connection with random digits*, J. Research Nat. Bur. Stand., Appl. Math. Series, 12:36–38, 1951. https://mcnp.lanl.gov/pdf_files/InBook_Computing_1961_Neumann_JohnVonNeumannCollectedWorks_VariousTechniquesUsedinConnectionwithRandomDigits.pdf.

[QUBIP23] QUBIP: *Transition to Post-Quantum Cryptography*
QUBIP project is co-funded by the European Union under the Horizon Europe framework programme [grant agreement no. 101119746].
https://qubip.eu/
https://github.com/QUBIP
http://www.youtube.com/@qubip_eu