


Escuela Técnica Superior de
Ingeniería Informática

TRABAJO FIN DE GRADO

# Sobre el nuevo estándar de encapsulado de clave (KEM) postcuántico

Realizado por

**Gabriel Vacaro Goytia**
**Ignacio Warleta Murcia**

**Para la obtención del título de**
Grado en Ingeniería Informática - Ingeniería del Software

**Dirigido por**
José Andrés Armario Sampalo

**En el departamento de**
Matemáticas Aplicadas I

**Convocatoria de junio, curso 2024/25**

*Este trabajo está dirigido a los estudiantes de cuarto curso de ingeniería informática interesados en la criptografía. Queremos que sirva de conector entre los conocimientos adquiridos durante el grado y la actualidad de la criptografía postcuántica*

# Agradecimientos

Queremos agradecer a nuestro tutor, don Andrés Armario Sampalo, por introducirnos al mundo de la criptografía de una forma apasionada y contagiarnos su entusiasmo por la misma.

También queremos agradecer a don Antonio Muñoz Matute, nuestro profesor de Física en Bachillerato del Colegio Claret Sevilla, por habernos inculcado su ética de trabajo y por haber sembrado semillas cuyos frutos recogemos ahora.

# Resumen

Este trabajo de investigación tiene por objetivo ilustrar y dar a conocer de manera didáctica el paradigma actual relacionado con la criptografía postcuántica (PQC), más en concreto, trataremos un tema que sin lugar a duda está ganando relevancia debido a su actualidad e innegable importancia: el nuevo estándar de encapsulado de clave (KEM) postcuántico. Para ello se emplean cuadernos interactivos en Python que permiten experimentar con conceptos teóricos y visualizar de forma práctica el funcionamiento de algoritmos. En otras palabras, hablaremos sobre las defensas criptográficas que se alzan contra los ordenadores cuánticos, los problemas intratables que las subyacen y que podemos esperar en estos años venideros.

**Palabras clave:** Criptografía Postcuántica (PQC), Cuántica, Learning With Errors (LWE), Mecanismo de encapsulado de claves (KEM) , Esquema de Cifrado de clave pública (PKE), Retículo, Anillo, CRYSTALS-Kyber, Kyber-KEM.

# Abstract

This research paper aims to illustrate and present in a didactic manner the current paradigm related to Post-Quantum Cryptography (PQC). More specifically, it addresses a topic that is undoubtedly gaining relevance due to its timeliness and undeniable importance: the new post-quantum Key Encapsulation Mechanism (KEM) standard. To this end, interactive Python notebooks are used to experiment with theoretical concepts and practically visualize the functioning of algorithms. In other words, we will discuss the cryptographic defenses rising against quantum computers, the hard problems they are based on, and what we can expect in the years to come.

**Keywords:** Post-Quantum Cryptography (PQC), Quantum, Learning With Errors (LWE), Key Encapsulation Mechanism (KEM), Public Key Encryption Scheme (PKE), Lattice, Ring, CRYSTALS-Kyber, Kyber-KEM.

# Índice general

# Índice de figuras

# Índice de tablas

# Lista de algoritmos

# 1. Introducción

## 1.1. El origen del engaño

Desde el inicio de los tiempos, el ser humano ha tenido la necesidad de esconder cierta información para que otros individuos no sean capaces de acceder a ella, empezando por el homo neanderthalensis, eslabón que se cree que ya era capaz de razonar (pensamiento simbólico). Estos individuos mostraban comportamientos tales como ocultar su comida, sus utensilios de combate o su refugio personal de otros posibles competidores/depredadores, dejando así constancia de que la necesidad de engañar a otros individuos proviene de cientos de miles de años en el pasado.

Alrededor del siglo V a.C, la invención de la estenografía [1], que podemos definir como la ocultación de la existencia de un mensaje secreto, véase por ejemplo el uso de tinta que solo se hace visible al aplicar calor, tatuajes en cabezas rapadas las cuales se dejan crecer el pelo antes de viajar, y al llegar al destino son rapadas de nuevo para leer su contenido, mensajes dentro de mensajes, mensajes escondidos en el medio de emisión, y un largo etcétera de estratagemas cuyo objetivo es el engaño.

La lista que aunaría a todos los intentos de la humanidad por esconder información sin duda sería muy extensa, podemos recopilar algunos de los más importantes [2] como la estenografía, el cifrado César, el cifrado de sustitución monoalfabética, el cifrado Vigenère o el cifrado por transposición, pero si tienen todos algo en común, es que dejaron de usarse progresivamente a medida que se descubría el secreto del método de cifrado o un criptoanálisis eficiente.

Cuando esto sucede, la búsqueda de un nuevo sistema se convierte en norma. Obsérvese el paralelismo con el surgimiento de la computación cuántica: actualmente, nuestra seguridad depende de problemas que consideramos intratables, como el problema del logaritmo discreto (DLP) o la factorización de números enteros grandes (RSA). Aunque ya se han propuesto algoritmos cuánticos, como el de Shor, que permitirían resolver estos problemas en tiempos polinómicos, su aplicación práctica requiere ordenadores cuánticos de gran escala que todavía no existen. No obstante, el simple hecho de que estos métodos teóricos sean conocidos ha puesto en marcha la búsqueda de nuevos sistemas criptográficos resistentes a la computación cuántica, del mismo modo que nuestros antepasados desarrollaban nuevas técnicas al descubrirse debilidades en las anteriores.

## 1.2. Contexto: La amenaza cuántica

Es un hecho que la cuántica no es un descubrimiento rompedor en 2025, ya que, a principios del siglo XX, ya se estaba comenzando a estudiar los principios que rigen la física cuántica. En nuestro caso, es de especial interés la década de 1980, donde C. H. Bennett y G. Brassard crearon el primer protocolo de distribución de claves cuántico (QKD), conocido como BB84. En el año 2000, Peter W. Shor (del cual hablaremos más adelante) y John Preskill, lanzaron un artículo llamado Simple Proof of Security of the BB84 Quantum Key Distribution Protocol [3], donde se pretende demostrar la seguridad del protocolo. Esto es de especial interés, pues se nos presenta como una alternativa segura de distribución de claves; pero si este es el caso, ¿por qué no se está poniendo en uso?, la respuesta es sencilla: no tenemos la infraestructura necesaria, y para cuando la tengamos, ¿qué llegará primero, los ciberataques o la implementación del protocolo?, ¿acaso estaremos dispuestos a ser vulnerables contra los atacantes cuánticos hasta que cada usuario pueda tener un dispositivo cuántico capaz de implementar el protocolo?

Todas estas preguntas tienen una respuesta sencilla: no debemos esperar a que la tecnología cuántica se establezca por completo, sino adelantarnos e implementar un criptosistema que sea resistente tanto a los ataques clásicos como a los cuánticos. Aquí es donde entra la criptografía postcuántica, que pretende aunar una serie de estándares con el propósito de blindarnos contra estas amenazas.

Antes de comentar por qué la computación cuántica supone una amenaza, será necesario comprender algunas de las propiedades elementales de un computador cuántico. En primer lugar, a diferencia de nuestros ordenadores actuales que usan bits (1 ó 0), estos usan cubits, los cuales existen en un continuo estado de superposición [4], lo que quiere decir que representan los valores 1 y 0 al mismo tiempo hasta el momento de la medición, que tomará el valor 1 con un porcentaje p y 0 con un porcentaje 1-p.

Por otra parte, estos cubits pueden estar correlacionadas entre sí por pares mediante el fenómeno denominado como entrelazamiento, en el cual el estado de un cúbit está ligado al de otro, aun cuando la distancia que los separa es grande, y siguiendo el principio de interferencia, las probabilidades de los resultados de los cubits pueden interferir entre sí. Por último, un cúbit no puede ser clonado de forma exacta.

El cómo se aprovechan estas propiedades con objetivo de generar métodos de comunicación seguros, no es menester de este documento, mas sí veremos los principales algoritmos que, sin duda, haciendo uso de estas propiedades, ponen en jaque todo el paradigma criptográfico clásico.

Aproximadamente 20 años más tarde, nacen el algoritmo de Shor y de Grover, siendo este primero el pilar fundamental y la razón por la que instituciones internacionales tales como el NIST [5] (EE. UU), ETSI [6] (UE) o el CCN [7] (ESP) están realizando grandes esfuerzos en generar nuevos estándares criptográficos basados en problemas no afectados por este algoritmo.

La cuestión en este asunto radica en que este algoritmo afecta directamente a las bases de los estándares actuales de intercambio de clave asimétrico, estos son: D H, ElGamal, ECC y RSA. De los cuales, los tres primeros basan su seguridad en el DLP, mientras que RSA se basa en factorización de enteros en primos grandes respectivamente, esto nos deja sin armas para combatir la amenaza cuántica; sin embargo, en previsión de esta situación, el NIST ha realizado un proceso de selección de nuevas propuestas de protocolos o criptosistemas que basen su seguridad en problemas para los que no se conozcan (y se piense que no existen) algoritmos que lo resuelvan en tiempo polinomial tanto para el paradigma de computación cuántica como clásica [8].

Por otra parte, es un hecho que existe información que, si bien es segura actualmente, será comprometida debido al efecto Harvest now, Decrypt later [9] (Recolecta ahora, desencripta luego), pues cualquier información cifrada con criptosistemas vulnerables a los ataques cuánticos generan resultados igual de vulnerables a estos, por lo que en el caso de guardar información encriptada con sistemas basados en problemas tratables por algoritmos cuánticos, un atacante poseedor de dicha tecnología cuántica sería capaz de realizar un criptoanálisis y hallar el mensaje en claro. Y aunque esto es cierto, la información pierde valor con el tiempo, véase el ejemplo de conocer un número ganador de lotería antes y después de que den el premio; o de ser capaz de robar los documentos actuales más secretos del gobierno de los EE. UU ahora y dentro de mil años (donde probablemente sea todo ya contenido de los libros de historia). Este fenómeno no es algo nuevo, ya que lleva sucediendo desde tiempos inmemoriales, pues actualmente somos capaces de descifrar casi todos los sistemas que se usaban hace cientos de años, y es de esperar, que, en cientos de años en el futuro, resolver los problemas que actualmente nos parecen tan intratables se conviertan en materia didáctica.

Por este motivo, lo que a algunos expertos en seguridad informática y criptografía llaman ≪Y2Q≫ (momento en el que los ordenadores cuánticos sean capaces de penetrar en los criptosistemas actuales) [10], puede materializarse en cualquier momento, pues los avances en este ámbito son en el mejor de los casos erráticos e impredecibles. Un informe sobre computación cuántica publicado en 2019 por las Academias Nacionales de Ciencias, Ingeniería y Medicina de EE. UU. predijo que un potente ordenador cuántico que ejecutara el algoritmo de Shor sería capaz de descifrar una implementación de RSA de 1024 bits en menos de un día. Trabajos más recientes sugieren que un ordenador con 20 millones de cubits podría hacer ese trabajo en tan solo ocho horas [11]. Actualmente, esta cifra de cubits queda fuera del alcance de la máquina cuántica más poderosa: el IBM Osprey, el cual cuenta con 433 cubits. No obstante, los avances en computación cuántica son impredecibles, de hecho, la misma IBM junto a el AIST japonés, ya trabajan en una máquina que alcance los 100.000 cubtis para 2029 [12] [13]. Sin defensas criptográficas seguras contra ataques cuánticos, cualquier infraestructura, desde los vehículos autónomos hasta el hardware militar, por no mencionar los servicios como las transacciones y comunicaciones financieras online, podrían ser vulneradas por hackers con acceso a ordenadores cuánticos. Es por ello por lo que la comunidad criptográfica ha aceptado que la computación cuántica supone una

seria amenaza para la criptografía actual y, desde 2017, el Instituto Nacional de Estándares y Tecnología (NIST) de EE. UU. ha liderado una competencia para estandarizar algoritmos postcuánticos, acelerando la investigación y desarrollo en este campo. En 2022, se seleccionaron algoritmos prometedores para convertirse en estándares futuros de la seguridad postcuántica, un ejemplo de estos fue CRYSTALS-Kyber [14]. Además, en 2024 se publicaron las especificaciones FIPS preliminares de estos nuevos estándares, marcando un paso clave hacia su adopción oficial [15]:

- FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard, basado en el algoritmo CRYSTALS-Kyber.
- FIPS 204: Module-Lattice-Based Digital Signature Standard, basado en CRYSTALS-Dilithium.
- FIPS 205: Stateless Hash-Based Digital Signature Standard, basado en SPHINCS+.

## 1.3. Objetivos, alcance y metodología

### Objetivo

El objetivo principal de este trabajo es explorar, desde un punto de vista didáctico y orientado a ingenieros informáticos, el estándar Kyber–KEM, basado en el esquema de cifrado de clave pública CRYSTALS–Kyber, y estudiar la primitiva matemática en la que se basa, conocida como Learning With Errors (LWE). Se pretende analizar su viabilidad y efectividad para proteger los datos frente a las amenazas que plantea la computación cuántica, adoptando una perspectiva didáctica y divulgativa.

Para ello, en determinados apartados, se ha priorizado la claridad expositiva sobre el rigor formal, con el objetivo de facilitar la comprensión a los estudiantes de informática que son el público objetivo de este trabajo, asegurando que, aunque se simplifiquen algunos detalles, los conceptos clave y las bases matemáticas sean comprendidos de manera adecuada.

### Objetivos específicos

- Estudiar las bases teóricas del problema LWE y su aplicación en la seguridad de Kyber.
- Analizar el funcionamiento de Kyber–KEM y los algoritmos implicados en sus fases PKE y KEM.
- Desarrollar implementaciones con un carácter didáctico, facilitando así la comprensión de los conceptos relativos al estándar Kyber.

- Revisar el proceso de estandarización llevado a cabo por el NIST y la evolución de los estándares FIPS.
- Evaluar los desafíos técnicos y prácticos de su adopción como alternativa resistente a ataques cuánticos.

## Alcance

Este trabajo se centra exclusivamente en la familia CRYSTALS, y dentro de ella, en el algoritmo Kyber–KEM. Se excluyen del análisis otros candidatos postcuánticos como NTRU o Classic McEliece. Se han implementado algoritmos con fines didácticos, como el Shortest Vector Problem, Closest Vector Problem, Learning With Errors, Ring–LWE y Module–LWE, así como aquellos relacionados con el PKE y el KEM, además de simulaciones de diversos ataques a este criptosistema. Asimismo, se analizarán únicamente los aspectos criptográficos y matemáticos, dejando en una escueta mención las cuestiones relacionadas con hardware cuántico o sistemas operativos. Todas las implementaciones se encuentran recogidas en: https://github.com/PQC-standards.

## Metodología

La metodología que se ha seguido durante la realización de este trabajo comenzó en octubre de 2024 con la investigación bibliográfica y la adquisición de conceptos generales. Se han utilizado fuentes académicas y oficiales, como las del NIST, para guiar la investigación, así como herramientas de simulación disponibles en repositorios públicos, tales como PQClean y LattPy, las cuales facilitaron la implementación y validación de los algoritmos y conceptos estudiados.

Se adoptó un enfoque gradual, comenzando con los fundamentos matemáticos y avanzando a los problemas CVP, SVP y LWE, seguido de los algoritmos Kyber-KEM y, finalmente, los apartados sobre seguridad y el futuro de la criptografía postcuántica.

A lo largo del proceso, se realizaron implementaciones prácticas antes de pasar a la siguiente sección, lo que permitió consolidar la teoría y la práctica. Para mantener una comunicación fluida, se celebraron reuniones mensuales a través de Microsoft Teams, y se mantuvo una tabla de historial de versiones para seguir el progreso y planificar las tareas. Cada apartado fue revisado antes de continuar con el siguiente, asegurando un desarrollo continuo y de calidad.

A continuación, se presenta el cronograma estimado del proyecto. Es importante mencionar que, en algunos momentos, se retrocedió en el proceso para realizar correcciones y ajustes, lo cual hace que las fechas presentadas no sean estrictamente lineales.

| Fase | Fechas |
|---|---|
| Búsqueda bibliográfica inicial | Octubre 2024 |
| Estudio de fundamentos matemáticos | Noviembre 2024 - Enero 2025 |
| Implementaciones de Retículos y Anillos | Diciembre 2024 - Enero 2025 |
| Problemas CVP, SVP y LWE | Noviembre 2024 - Enero 2025 |
| Implementaciones de los problemas CVP, SVP y LWE | Diciembre 2024 - Febrero 2025 |
| Estudio de Complejidad | Diciembre 2024 - Enero 2025 |
| Análisis del estándar Kyber–KEM | Enero - Febrero 2025 |
| Implementación de algoritmos Kyber-KEM | Febrero - Marzo 2025 |
| Apartados de Seguridad, Futuro de la Criptografía y Conclusiones | Marzo - Abril 2025 |
| Maquetar el documento y anexos | Marzo 2025 |
| Revisión, correcciones y entrega final | Abril 2025 |

**Tabla 1.1:** Cronograma estimado del proyecto

## 1.4. Estructura

Este trabajo se encuentra estructurado en distintos capítulos que permiten comprender de manera progresiva el contexto, los fundamentos teóricos, los aspectos prácticos y el futuro de Kyber–KEM dentro del ámbito de la criptografía postcuántica.

Una vez conocido el contexto en el que nos encontramos, analizaremos mejor la amenaza a la que se enfrenta la criptografía postcuántica, explorando el impacto de los ordenadores cuánticos en la criptografía clásica y los algoritmos de Shor y Grover. De este modo, introduciremos definitivamente la criptografía postcuántica, examinando la importancia de los retículos en este campo y los esfuerzos del NIST en la estandarización de nuevos algoritmos resistentes a ataques cuánticos.

A continuación, en el apartado tres, se presentan todas las definiciones de términos clave, acrónimos y la notación matemática utilizada a lo largo del documento. Esta información será necesaria para el siguiente apartado, donde profundizaremos en la teoría de retículos y anillos algebraicos con un enfoque didáctico, proporcionando las bases matemáticas necesarias para comprender los problemas en los que se sustenta Kyber–KEM, los cuales veremos en el siguiente apartado.

Una vez comprendido los fundamentos matemáticos y los problemas en los que se basa este estándar postcuántico, estudiaremos la estructura del esquema de intercambio de claves, abordando los algoritmos involucrados en la generación de claves (PKE), cifrado, descifrado, generación de claves (KEM), encapsulado y desencapsulado.

Seguidamente, analizaremos los ataques clásicos y cuánticos a los sistemas criptográficos basados en retículos, así como ataques de canal lateral y ataques de texto en claro elegido. En el siguiente apartado explicaremos el futuro de la PQC, donde abordaremos los retos y oportunidades de la transición a esta nueva criptografía y analizaremos la nueva convocatoria del NIST. Y, en el siguiente apartado, presentaremos las principales conclusiones del trabajo y plantearemos posibles líneas futuras de investigación.

Finalmente se incluyen una serie de anexos con implementaciones en Python y Java relacionados con los temas tratados a lo largo del documento. Estos anexos son referenciados en los apartados pertinentes para que la implementación complemente lo que se explica en el documento y resuelte más fácil de comprender. Para poder trabajar con estas implementaciones, es útil consultar el manual de instalación del Anexo 1 o, para ejecutarlos sin necesidad de ninguna instalación, en el repositorio que se desee utilizar, presionar el botón «Binder», el cual abrirá una página donde se podrá usar el cuaderno sin ningún tipo de instalación. Cabe mencionar que, al ser Binder una herramienta ajena que depende del tráfico en línea, es posible que la ejecución sea más lenta en algunos momentos.

## 1.5. Horas invertidas

A continuación se muestran las gráficas de las horas invertidas de cada uno de los miembros registradas con la aplicación Clokify. Cabe mencionar que las horas dedicadas a la presentación son estimadas, ya que la memoria debía entregarse el día 19. Se calcula que se invertirán aproximadamente 20 horas en la preparación y el estudio de la presentación.

En las imágenes 1.1 y 1.2 se muestran las horas invertidas por Gabriel e Ignacio respectivamente y en la imagen 1.3 se muestran las horas totales.

## Gabriel

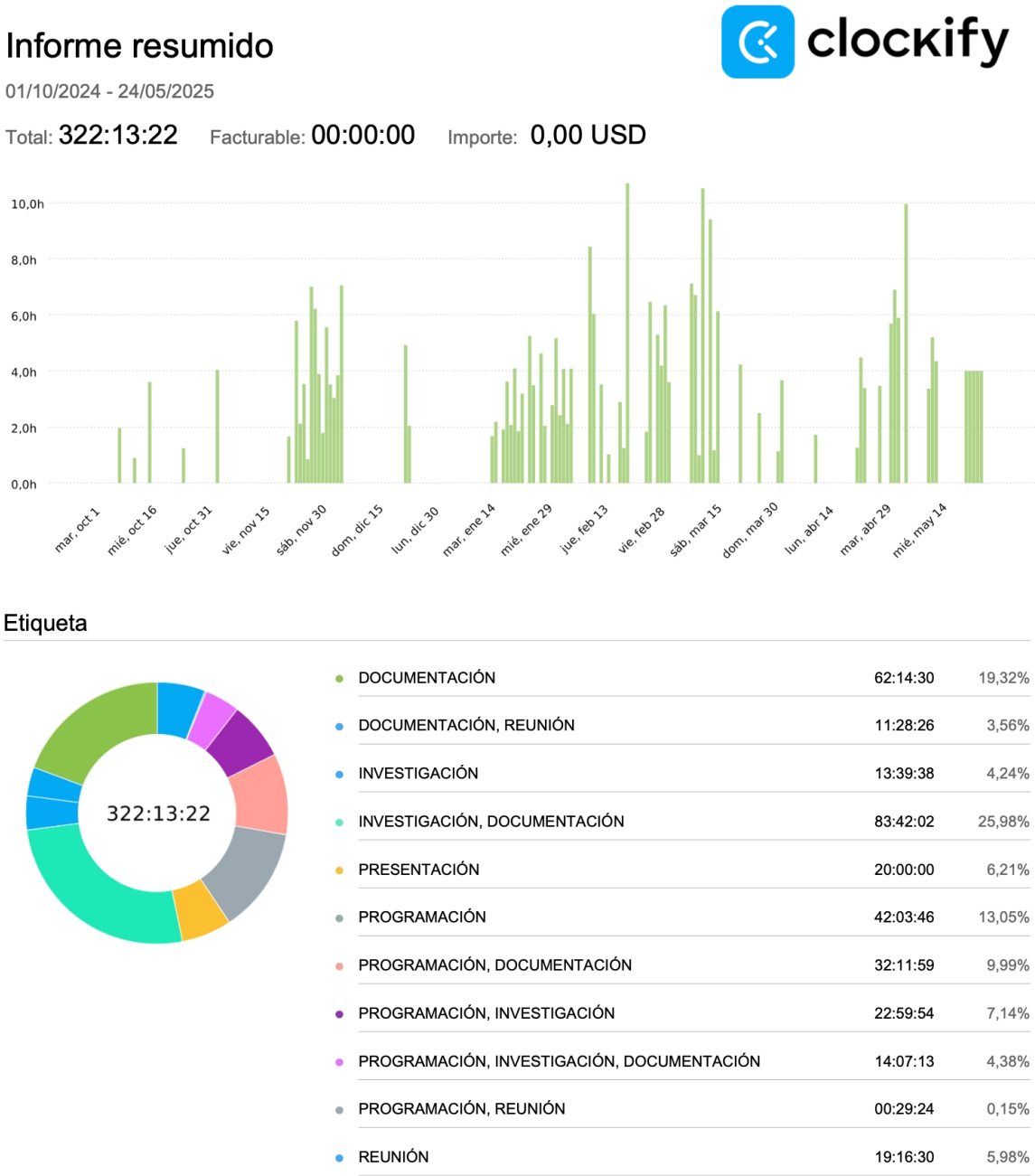


**Figura 1.1:** Informe Gabriel

## Ignacio

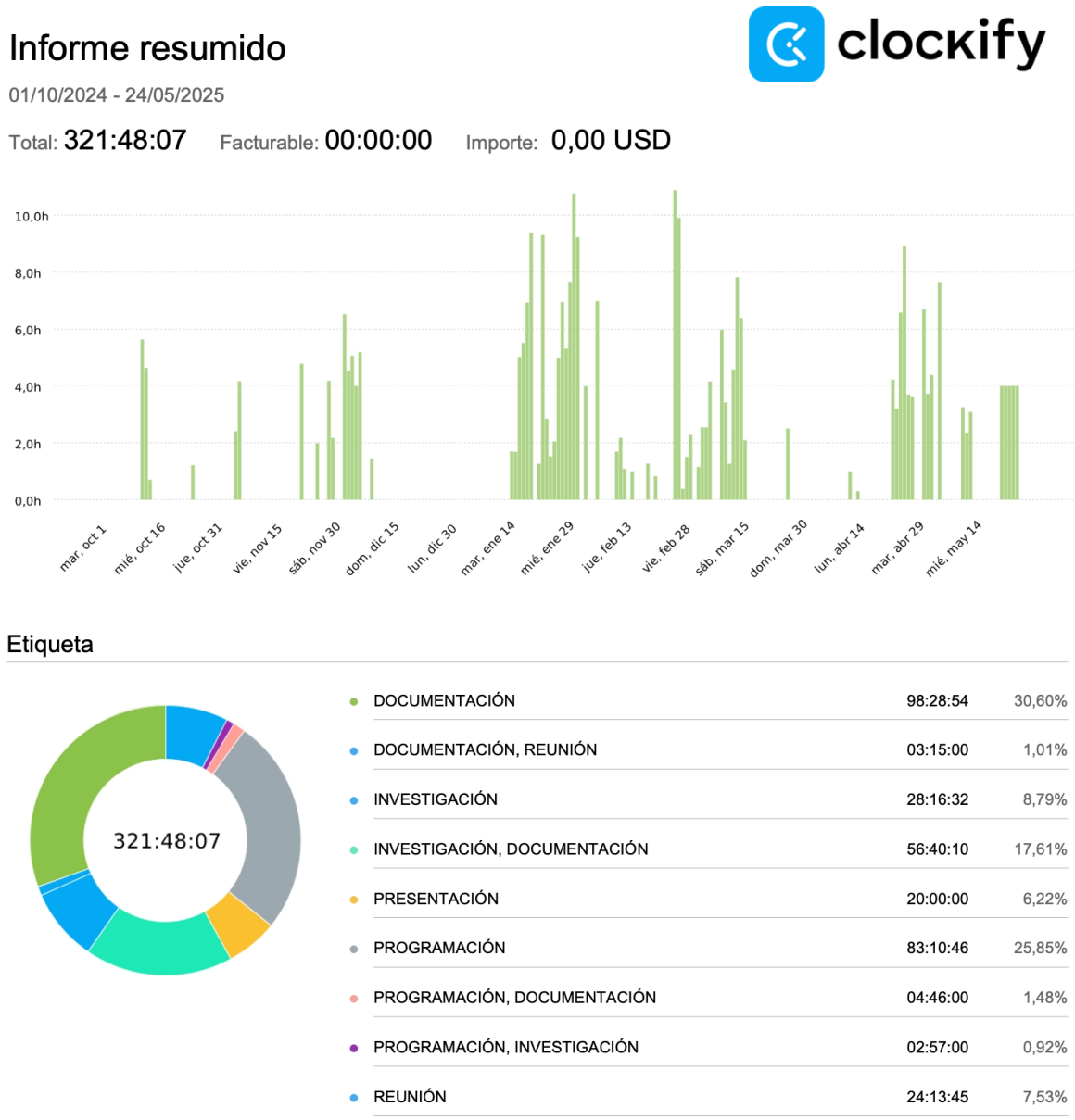


**Figura 1.2:** Informe Warleta

## Grupal

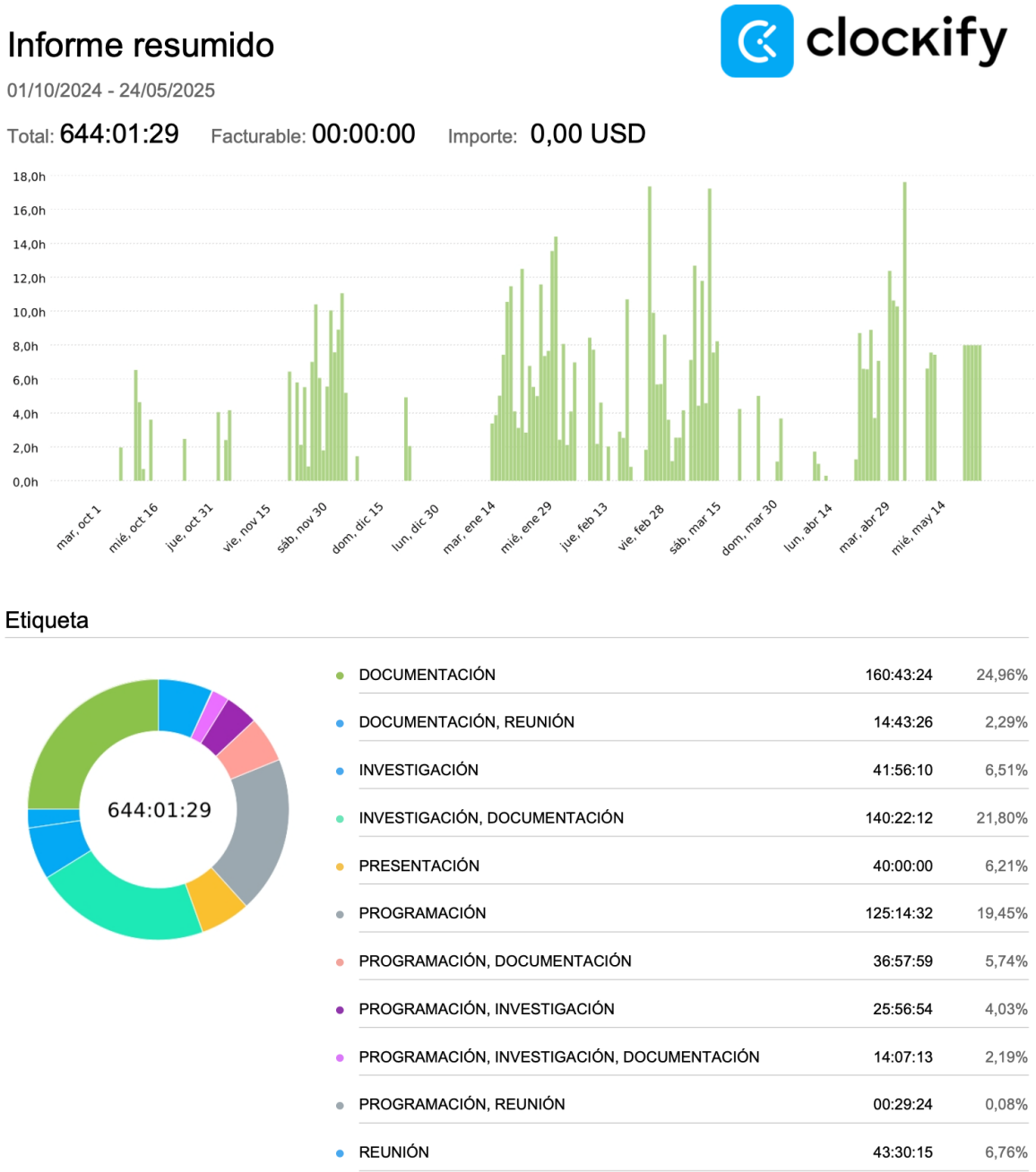


**Figura 1.3:** Informe Grupal

# 2. Criptografía postcuántica: Un nuevo Paradigma

Como ya hemos comentado en el apartado anterior, la llegada de la computación cuántica sumada a ciertos algoritmos ha despertado a los gigantes tecnológicos y puesto en marcha los mecanismos de defensa a nivel mundial, este apartado tiene como objetivo dar a conocer por qué estos algoritmos son tan peligrosos y cómo es que los ordenadores convencionales no han sido capaces de beneficiarse de ellos, mientras que los cuánticos tienen la capacidad de vulnerar nuestros sistemas más seguros.

## 2.1. Impacto de ordenadores cuánticos en criptografía clásica: Algoritmos de Shor y Grover

### 2.1.1. Algoritmo de Shor y sus implicaciones

Peter Shor, profesor de matemáticas del MIT e investigador famoso por su contribución al ámbito de la computación cuántica, fue uno de los principales responsables de que, en la actualidad, exista una innegable necesidad de cambiar los problemas que subyacen bajo la criptografía moderna. No obstante, no siempre fue tan claro que la computación cuántica pudiera si quiera competir contra la clásica.

En los años 80, P.A. Benioff [16] demostró que la mecánica cuántica podía superar la computación clásica mediante un fenómeno conocido como la evolución unitaria reversible (en el cual no ahondaremos). Sus estudios indicaban que valiéndose de los principios cuánticos, la computación clásica podría verse igualada e incluso superada por su contraparte cuántica. No sería hasta 1992 cuando Deutsch y Jozsa publicaron investigaciones clave que demostraron que las computadoras cuánticas podían resolver ciertos problemas exponencialmente más rápido que las clásicas, marcando un hito en el desarrollo de la computación cuántica.

En este punto es donde entra el algoritmo de Shor, que, desarrollado en 1994 por Peter Shor, tiene como objetivo descomponer números enteros en sus factores primos en un tiempo polinómico [17]. Esto significa que, mientras que factorizar un número grande, como un encriptado con RSA de 1024 bits podría tomar años con un ordenador convencional, el algoritmo de Shor haciendo uso de un ordenador cuántico con la suficiente potencia, podría tardar segundos.

Este avance es especialmente problemático para los sistemas de cifrado basados en la dificultad de la factorización de números grandes, como el cifrado de

clave pública RSA (usado en protocolos como TLS 1.2/SSL, https, ssh, etc.), y los basados en el cálculo del logaritmo discreto, como lo es la ECC (usado en protocolos como TLS 1.3, Bitcoin, Apple Pay, WhatsApp, etc.). En estos sistemas, la seguridad radica en la dificultad computacional de resolver estos problemas en un tiempo razonable (tiempo polinómico). Sin embargo, un ordenador cuántico capaz de ejecutar el algoritmo de Shor a gran escala podría resolver estos problemas en tiempos significativamente cortos, rompiendo así la seguridad de RSA y ECC. De hecho, si un ordenador actual necesita $O(2^{\sqrt[3]{\log n}})$ operaciones bit para romper un esquema de cifrado basado en la factorización de números grandes (como RSA), un ordenador cuántico, usando el algoritmo de Shor, reduciría ese número de operaciones bit a $O((\log n)^3)$ con un almacenamiento de memoria de $O(\log n)$ bits [18] (Sección B.1).

El algoritmo se basa en dos procesos principales: primero, la exponenciación modular, un procedimiento clásico que se utiliza para preparar el estado inicial del registro cuántico. Aunque existen versiones reversibles de esta operación implementadas en circuitos cuánticos, la lógica subyacente sigue siendo clásica; como se comenta en la sección 3: *Reversible Logic and Modular Exponentiation* de [17].

Una vez preparado el registro cuántico, se realiza la Transformada Cuántica de Fourier (QFT), que constituye la parte cuántica utilizada en el algoritmo de Shor: donde, se describe una técnica para construir una transformación unitaria (transformada discreta de Fourier) en tiempo polinómico en computadoras cuánticas, utilizando una matriz indexada por estados en representaciones binarias de enteros (sección 4. QFT de [17]). De forma simplista, es una herramienta matemática que transforma el estado cuántico de modo que el registro cuántico contenga información sobre el período de la función periódica $f(x) = a^r \mod N$ (donde $N$ es el número a factorizar y $r$ el exponente), lo cual es clave para la posterior factorización.

El hecho de que $f(x)$ sea una función periódica es muy importante, pues esto significa que, a partir de un valor de $x$, los valores de $f(x)$ se repiten. El número de elementos contenidos en el conjunto de estos valores es el periodo $r$, es decir, $f(x+r) = f(x) \mod N;\ \forall x \in \mathbb{Z}$.

Conociendo el valor $r$, podemos calcular factores no triviales de $N$, pues al ser $a^r \mod N = 1$, esto implica que $a^r \equiv 1 \mod N$, y como $r$ es el periodo de la función, entonces podemos deducir que $a^r - 1$ es un múltiplo de $N$. Esto proporciona una pista crucial para obtener los factores de $N$, pues si calculamos el MCD$(a^r - 1, N)$, tendremos dos opciones: que hayamos encontrado un factor no trivial (un divisor de $N$) o, por otra parte, en el caso de que el resultado sea 1 o $N$, repetiremos el proceso iterativamente (encontrando otro $r$) hasta finalizar la factorización.

A pesar de su gran potencial, el algoritmo de Shor enfrenta desafíos significativos. Uno de los principales es el requisito tecnológico: necesita una cantidad considerable de cúbits cuánticos estables y de alta calidad, los cuales aún no están disponibles en la actualidad. Además, la eficiencia práctica del algoritmo se ha visto limitada, ya que, aunque es teóricamente poderoso, su implementación real ha sido restringida a números relativamente pequeños. Hasta ahora, el mayor

número que se ha logrado factorizar utilizando el algoritmo de Shor es 21, lo que resalta las limitaciones actuales de la infraestructura cuántica. Esto se debe, en gran medida, a las limitaciones asociadas con la corrección de errores y la escasez de cúbits en los sistemas cuánticos actuales [19].

### 2.1.2. Algoritmo de Grover y sus implicaciones

Lov Grover fue un informático estadounidense de origen indio que, en 1996, publicó un artículo sobre la optimización de problemas de búsqueda en bases de datos no estructuradas. El producto resultante de este estudio fue el algoritmo de búsqueda cuántica, más conocido como el algoritmo de Grover. Para entender el propósito de este algoritmo, imagínese una base de datos (DB) no estructurada de $N$ elementos. Es intuitivo que, para una computadora convencional, este problema tendría una complejidad de consulta de $O(N)$ llamadas (con $n$ llamadas tendremos la certeza de acertar), la cual, asumiendo que la DB se genera de forma aleatoria, nos deja con un promedio de $N/2$ llamadas por consulta.

Por otra parte, utilizando el algoritmo de Grover, mediante conceptos como el oráculo y la amplificación de amplitud, los cuales pueden estudiar con más detalle en la sección 2 de [20], conseguimos una drástica reducción del tiempo necesario para realizar una búsqueda exhaustiva (ataque de fuerza bruta) a la raíz cuadrada del tiempo que tomaría en un sistema clásico. Su algoritmo cuántico permite localizar un elemento objetivo con una eficiencia superior a los métodos clásicos conocidos, reduciendo el número de pasos requeridos de $O(N)$ a $O(\sqrt{N})$ [20] [18] [21].

Esto implica que, si un ordenador cuántico lo suficientemente avanzado estuviera disponible, el tamaño de la clave en un sistema de cifrado simétrico debería duplicarse para mantener el mismo nivel de seguridad. Dicho esto, es importante aclarar que este algoritmo puede ser utilizado para potenciar otros algoritmos, proporcionándoles una aceleración cuadrática, lo cual, si bien reduce considerablemente el tiempo de ejecución, todo parece indicar que este no represente un riesgo significativo para los cifrados actuales más exigentes, en particular los simétricos, como AES con claves de 256 bits en adelante.

Esto se debe a que, si bien la reducción es cuadrática, los espacios de búsqueda en la criptografía actual son exponenciales; pongamos el ejemplo de un criptoanálisis de una clave de 128 bits mediante fuerza bruta. Las combinaciones requeridas por un ataque clásico serían de $2^{128}$, mientras que, usando el algoritmo de búsqueda cuántica, se reducirían a $2^{64}$. Contando con un ordenador de última generación que realizara $10^9$ operaciones por segundo, y simplificando el hecho de probar una clave como realizar una sola operación, tardaríamos aproximadamente 585 años en recorrer el espacio de claves, lo cual es un tiempo peligrosamente corto para los estándares criptográficos. Pero ¿qué pasaría si la supercomputadora *El Capitán*, ubicada en Estados Unidos y propiedad del Lawrence Livermore National Laboratory (LLNL), capaz de realizar $2,79 \times 10^{18}$ operaciones/segundo (del orden de los quintillones), decidiera probar suerte? El resultado sería que, en el peor caso, tardaría 6.6 segundos. Por otra parte, en el caso de utilizar claves de 256 bits

(reducido a 128 por Grover) y 512 bits (reducido a 256), el tiempo que necesitaría sería de 3.863 y 1,31 millones de años respectivamente, lo cual resulta inviable.

Para implementar este algoritmo de búsqueda, Grover definió el problema de búsqueda cuántica mediante la función $f\colon \{0,1\}^n \to \{0,1\}$ (veáse la ecuación 2.1), donde $n$ es el tamaño de los bits del espacio de búsqueda:

$$f(x) = \begin{cases} 1, & \text{si } x = x_0 \\ 0, & \text{si } x \neq x_0 \end{cases} \tag{2.1}$$

donde $x_0$ es el valor específico que estamos buscando. El objetivo es entonces identificar este valor $x_0$, que es el único elemento en el dominio de $f$ que hace que la función devuelva 1. Este enfoque permite que el algoritmo de Grover concentre la probabilidad en el estado correspondiente a $x_0$, facilitando su identificación mediante la amplificación de amplitud cuántica [22] [23] [24].

En otras palabras, Grover observó que, en los sistemas cuánticos, mediante la interferencia cuántica, las amplitudes de probabilidad se redistribuyen, lo que conlleva a un aumento de la probabilidad de encontrar un estado particular. Para entender este concepto, haremos uso de una analogía: supongamos que frente a usted hay un estanque completamente tranquilo, y decide lanzar varias piedras en diferentes posiciones. Las ondas generadas por estas piedras se cruzarán, y dependiendo de las propiedades de cada una, pueden darse varios sucesos: algunas ondas aumentarán la oscilación de otras, por lo tanto, la amplitud aumentará (interferencia constructiva), mientras que otras colisiones resultarán en una disminución de la amplitud (interferencia destructiva). La idea es que Grover se dio cuenta de que si modificamos estratégicamente las amplitudes de probabilidad, era capaz de amplificar las probabilidades de encontrar el estado deseado y, por lo tanto, reducir las de encontrar los no deseados.

Por otra parte, las amplitudes de probabilidad tienden a concentrarse en estados en particular, especialmente en aquellos que son soluciones óptimas de ciertos problemas, acumulando así una mayor amplitud de probabilidad en el estado buscado. Este algoritmo también se apoyó en la ecuación de Schrödinger para describir la evolución de estos estados cuánticos, lo que permitió predecir y explicar el comportamiento de su algoritmo en términos de mecánica cuántica [25].

Aunque el algoritmo de Grover no es tan devastador como el de Shor para la criptografía asimétrica, también tiene un impacto significativo en la criptografía clásica, particularmente en los algoritmos de cifrado simétrico como AES y Triple DES. Debido a su capacidad de reducir el tiempo de búsqueda de forma cuadrática, este algoritmo resalta la necesidad de adoptar cifrados de clave más largos en un mundo donde la computación cuántica puede llegar a ser ampliamente accesible.

## 2.2. La criptografía postcuántica como respuesta: NIST y estandarización de algoritmos postcuánticos

Ante la amenaza de la computación cuántica, la PQC se plantea como un paradigma fundamental para la protección de la información frente a los futuros ordenadores cuánticos. Para dar respuesta a esta necesidad, el Instituto Nacional de Estándares y Tecnología (NIST) lanzó en 2017 una convocatoria internacional para la identificación y estandarización de nuevos algoritmos criptográficos sólidos frente tanto a los futuros ataques cuánticos como convencionales. Esta iniciativa busca desarrollar y establecer estándares de seguridad que puedan salvaguardar los datos en un entorno donde la tecnología cuántica tenga capacidades avanzadas.

Es importante recordar que los algoritmos de la PQC no se basan en principios de la mecánica cuántica (pues deben ser implementados en los computadores convencionales), sino en problemas matemáticos que hasta ahora han demostrado ser resistentes a los ataques cuánticos y convencionales. La seguridad de los algoritmos considerados en la convocatoria del NIST se fundamenta en determinados problemas matemáticos basados en cinco primitivas:

1. Códigos correctores de errores, que dan lugar a la criptografía basada en códigos (Code–based cryptography)
2. Funciones resumen o funciones hash, que proporcionan la criptografía basada en resúmenes (Hash-based cryptography)
3. Polinomios multivariantes cuadráticos, que dan lugar a la llamada criptografía multivariante cuadrática (Multivariate Quadratic Cryptography, MQC)
4. Retículos, dando pie a la criptografía basada en retículos (Lattice–based cryptography)
5. Isogenias definidas sobre curvas elípticas, que proporcionan la criptografía basada en isogenias (Isogeny–based cryptography)

En este documento, nos centraremos en la criptografía basada en retículos, la cual supone casi el 50 % de los algoritmos candidatos [18].

## 2.3. Retículos y su importancia en criptografía

La criptografía basada en retículos ha emergido como una de las alternativas más prometedoras frente a la computación cuántica, posicionándose como un pilar fundamental para la criptografía postcuántica. A diferencia de los esquemas criptográficos tradicionales, como RSA y ECC (criptografía de curvas elípticas), que dependen de problemas matemáticos difíciles de resolver con los ordenadores clásicos, estos métodos se vuelven vulnerables frente al poder de las computadoras cuánticas. En cambio, los problemas en los que se basa la criptografía de retículos, como el Problema del Vector Más Corto (SVP), el Problema del Vector Más Cercano (CVP) y el Aprendizaje con Errores (LWE), son considerados resistentes tanto a los

ataques cuánticos como a los ataques clásicos, especialmente en dimensiones elevadas, lo que les da una ventaja significativa en un futuro cuántico. Esto se debe a que no existen algoritmos que aprovechen los principios cuánticos para vulnerar los problemas relacionados con retículos en tiempo polinómico (así como el algoritmo de Shor vulnera la factorización de enteros y el DLP).

A lo largo de los años, la criptografía basada en retículos ha demostrado una capacidad notable para ofrecer soluciones que no solo son resistentes a las amenazas derivadas de la computación cuántica, sino que también permiten desarrollar esquemas criptográficos avanzados y versátiles. Entre ellos destaca la encriptación totalmente homomórfica (FHE, por sus siglas en inglés), que permite realizar operaciones sobre datos cifrados sin necesidad de descifrarlos. Esta propiedad resulta fundamental en áreas como la computación en la nube, donde la privacidad de los datos debe ser preservada mientras se procesan. Esta propiedad también está presente en algunos criptosistemas clásicos, como ElGamal o RSA, aunque de manera parcial, ya que únicamente permiten realizar operaciones homomórficas de multiplicación. En contraste, en los esquemas basados en retículos, podemos realizar tanto sumas como multiplicaciones sobre datos cifrados.

Además, los retículos no solo ofrecen resistencia frente a la computación cuántica, sino que también presentan una gran flexibilidad. Permiten la construcción de sistemas criptográficos como firmas digitales, intercambio de claves y esquemas de autenticación, todos con niveles de seguridad y eficiencia que los hacen ideales para ser implementados en un mundo con tecnologías emergentes y capacidades de cálculo cada vez mayores.

La seguridad de la criptografía basada en retículos no depende de una única suposición matemática, sino que está construida sobre una familia de problemas matemáticos de complejidad intrínseca. Esto hace que la resistencia a ataques no se base en la factorización de grandes números o la solución de logaritmos discretos (como en los métodos tradicionales), sino en la dificultad de resolver problemas geométricos complejos en espacios de alta dimensión, algo que sigue siendo un desafío incluso para los ordenadores cuánticos. Este enfoque de seguridad «resiliente» convierte a la criptografía basada en retículos en una de las principales candidatas para el futuro de la seguridad informática [26] [27] [28] [29].

# 3. Terminología, acrónimos y notación

## 3.1. Términos y definiciones

### Esquemas criptográficos y mecanismos generales

**Mecanismo de encapsulado de claves (KEM):** Conjunto de tres algoritmos criptográficos (KeyGen, Encaps y Decaps) que pueden ser utilizados por dos partes para establecer una clave secreta compartida a través de un canal público.

**Esquema de Cifrado de clave pública (PKE):** Un conjunto de tres algoritmos criptográficos (KeyGen, Encrypt y Decrypt) que pueden ser utilizados por dos partes para enviar datos secretos a través de un canal público. También conocido como un esquema de cifrado asimétrico.

### Algoritmos y procesos

**KeyGen:** Algoritmo de generación de claves.

**Encapsulado (Encaps):** Proceso de aplicar el algoritmo de encapsulación de un KEM. Este algoritmo toma como entrada una clave pública, usa aleatoriedad interna, y produce como salida una clave secreta compartida junto con un texto cifrado asociado.

**Desencapsulado (Decaps):** El proceso de aplicar el algoritmo de desencapsulación de un KEM. Este algoritmo acepta como entrada un texto cifrado de KEM y la clave de desencapsulación, y produce como salida una clave secreta compartida.

**(KEM) cifrado:** Una cadena de bits que se produce mediante la encapsulación y se utiliza como entrada para la desencapsulación.

## Claves y tipos de claves

**Clave:** Una cadena de bits que se utiliza en conjunto con un algoritmo criptográfico, como las claves de encapsulación y desencapsulación (de un KEM), la clave secreta compartida (producida por un KEM) y las claves de cifrado y descifrado (de un PKE).

**Clave de encriptado:** Una clave criptográfica que se utiliza con un PKE para cifrar textos en claro en textos cifrados. La clave de cifrado puede hacerse pública.

**Clave de descifrado:** Una clave criptográfica que se utiliza con un PKE para descifrar textos cifrados en textos en claro. La clave de descifrado debe mantenerse privada y debe ser destruida cuando ya no sea necesaria.

**Clave de encapsulado:** Una clave criptográfica producida por un KEM durante la generación de claves y utilizada durante el proceso de encapsulación. La clave de encapsulación puede hacerse pública.

**Clave de desencapsulado:** Una clave criptográfica producida por un KEM durante la generación de claves y utilizada durante el proceso de decapsulación. La clave de decapsulación debe mantenerse privada y debe ser destruida cuando ya no sea necesaria.

## Propiedades de seguridad

**IND–CPA:** Es una propiedad de seguridad en criptografía que garantiza que un esquema de cifrado es seguro contra ataques donde un adversario puede elegir y cifrar múltiples mensajes de su elección.

**IND–CCA2:** Es una propiedad de seguridad en criptografía que garantiza que un esquema de cifrado es seguro incluso si un atacante puede descifrar múltiples textos cifrados elegidos adaptativamente.

| | |
|---|---|
| **Fujisaki-Okamoto Transform:** | Es una técnica criptográfica diseñada para convertir un esquema de cifrado semánticamente seguro bajo ataques pasivos (IND–CPA) en un esquema seguro contra ataques de texto cifrado elegido adaptativamente (IND–CCA2). |

## Modelos y herramientas criptográficas

| | |
|---|---|
| **Quantum Random Oracle Model:** | Este modelo permite analizar la seguridad de los algoritmos frente a adversarios que disponen de computadoras cuánticas. |
| **SHA3–256:** | Función hash criptográfica de la familia SHA–3 que genera un hash de 256 bits. |
| **SHA3–512:** | Función hash criptográfica de la familia SHA–3 que genera un hash de 512 bits. |
| **Key Derivation Function:** | Función criptográfica que toma como entrada una clave o valor secreto inicial y genera una o varias claves derivadas, adecuadas para su uso en otros algoritmos criptográficos. |
| **KDF (SHA3–256):** | Función derivadora de claves (Key Derivation Function) basada en SHA–256. |

## Conceptos matemáticos

| | |
|---|---|
| **Norma de un vector:** | Una medida que cuantifica la longitud o magnitud de un vector en el espacio. En el caso de la norma euclidiana, se calcula como la raíz cuadrada de la suma de los cuadrados de sus componentes. En este documento consideraremos la distancia euclídea. |
| **Argumento de un número Complejo:** | Es el ángulo que forma el vector correspondiente a ese número con el eje real en el plano complejo. Su notación es arg. |

## 3.2. Acrónimos

| | |
|---|---|
| **AES** | Advanced Encryption Standard |
| **CCA** | Chosen Ciphertext Attack |
| **D∼H** | Diffie–Hellman Key Exchange (Intercambio de claves Diffie–Hellman) |
| **DLP** | Discrete Logarithm Problem (Problema del logaritmo discreto) |
| **ECC** | Eliptic Curve Cryptography (Criptografía de curvas elípticas) |
| **FIPS** | Federal Information Processing Standard |
| **FOT** | Fujisaki-Okamoto Transform |
| **IND–CCA2** | Indistinguishability under Adaptive Chosen Ciphertext Attack |
| **IND–CPA** | Indistinguishability under Chosen Plaintext Attack |
| **IoT** | Internet of Things (Internet de las cosas) |
| **KDF** | Key Derivation Function |
| **KEM** | Key Encapsulation Mechanism (Mecanismo de Encapsulación de Claves) |
| **LBC** | Lattice–Based Cryptography |
| **LWE** | Learning With Errors |
| **ML–KEM** | Module Lattice Key Encapsulation Mechanism |
| **MLWE** | Module Learning With Errors |
| **MQC** | Multivariate Quadratic Cryptography |

| | |
|---|---|
| **NIST** | National Institute of Standards and Technology |
| **NTT** | Number-Theoretic Transform |
| **PKE** | Public Key Encryption (Cifrado de clave pública) |
| **PQC** | Post–Quantum Cryptography |
| **PRF** | Pseudorandom Function |
| **QFT** | Quantum Fourier Transform (Transformada cuántica de Fourier) |
| **QKD** | Quantum Key Distribution (Distribución cuántica de claves) |
| **QROM** | Quantum Random Oracle Model |
| **RBG** | Random Bit Generator |
| **RSA** | Rivest-Shamir-Adleman Algorithm (Algoritmo de Rivest-Shamir-Adleman) |
| **SHA** | Secure Hash Algorithm |
| **SHAKE** | Secure Hash Algorithm KECCAK |

## 3.3. Notación matemática

| | |
|---|---|
| $N$ | Constante equivalente al entero 256. |
| $p$ | Constante equivalente al primo entero $3329 = 2^8 \cdot 13 + 1$. |
| $\Theta$ | Se utiliza para describir el comportamiento asintótico de funciones. |

| | |
|---|---|
| $\Theta(f(n))$ | Describe el crecimiento exacto de una función $g(n)$ en términos de $f(n)$, de manera que $g(n)$ crece a la misma velocidad que $f(n)$ cuando $n$ es suficientemente grande. |
| $L(B)$ | Retículo $L$ generado por la base $B$. |
| $\|\|a\|\|$ | Norma de un vector $a$. En este documento se usa la norma euclídea. |
| $R_q$ | Anillo $R$ módulo $q$. |
| $R_q^k$ | Conjunto de vectores de dimensión $k$ con coeficientes en el anillo $R_q$. |
| $\mathbb{Z}$ | El conjunto de los números enteros. |
| $\mathbb{Z}_m$ | El anillo de enteros módulo $m$ (es decir, el conjunto $\{0, 1, \ldots, m-1\}$ con suma y multiplicación módulo $m$). |
| $\mathbb{Z}_m^n$ | Conjunto de vectores de dimensión $n$ cuyos elementos están en $\mathbb{Z}_m$. |
| $(R, +)$ | El grupo abeliano formado por el anillo $R$ bajo la operación de suma. |
| $R[x]$ | Anillo de polinomios con coeficientes en $R$. |
| $\otimes_R$ | Producto tensorial sobre el anillo $R$, utilizado para extender módulos o espacios vectoriales sobre $R$. |
| $M$ | Módulo $M$ sobre un anillo $R$. |
| $N$ | Submódulo $N$ de un módulo $M$. |

# 4. Fundamentos matemáticos

Para entender la criptografía basada en retículos, sus algoritmos y los problemas en los que se basan es primordial conocer la primitiva matemática en la que se sustenta. En los siguientes apartados estudiaremos los retículos y los retículos sobre anillos con el objetivo de poder entender posteriormente los problemas en los que se apoyan.

Para facilitar la familiarización de estos fundamentos matemáticos se ha realizado una implementación en Python que se puede encontrar en los Anexos 2 y 3 dentro del repositorio «Fundamentos Teóricos». Para trabajar con estas implementaciones se ha utilizado la librería para Python, «LattPy», para su instalación, vaya al Anexo 1 o, alternativamente, para su despliegue, presione en el botón «Binder» en el repositorio o este enlace https://mybinder.org/v2/gh/PQC-standards/Fundamentos-Teoricos/main.

Para la elaboración de este apartado se han utilizado, además del documento CCN-STIC 221 [18] el documento Lattice-based Cryptography de Daniele Micciancio y Oded Regev [30], así como diversas páginas web de divulgación matemática que, aunque no siempre resultan rigurosas, son útiles para comprender estos fundamentos matemáticos sin requerir conocimientos avanzados.

## 4.1. Retículos

Un retículo (*lattice* o celosía), $\mathcal{L}$, es una estructura algebraica compuesta por un conjunto de puntos en un espacio, donde cada punto puede ser descrito como una combinación lineal de vectores base con coeficientes enteros. Estos puntos están organizados de manera regular y se extienden infinitamente en todas las direcciones, formando una red de puntos que se repite periódicamente.

Es decir, dada una base de vectores $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$, un retículo $\mathcal{L}$ es un subespacio vectorial formado por las combinaciones lineales enteras de los elementos de la base $\mathbf{b}_i$ [31]. Se muestra en la ecuación 4.1

$$\mathcal{L} = \left\{ \sum_{i=1}^{m} a_i \cdot \mathbf{b}_i \ : \ a_i \in \mathbb{Z} \right\} = \{ B \cdot \mathbf{a} \ : \ \mathbf{a} \in \mathbb{Z}^m \} \tag{4.1}$$

El conjunto $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$ se denomina *base del retículo*. El retículo generado por la base $B$ se denota por $\mathcal{L}(B)$ [18].

Por ejemplo, supongamos como vectores base en $\mathbb{R}^2$ aquellos que forman una base ortonormal estándar:

$$\mathbf{b}_1 = (1, 0), \quad \mathbf{b}_2 = (0, 1)$$

El retículo $\mathcal{L}(B)$ generado es el conjunto de todas las combinaciones lineales enteras de estos vectores:

$$\mathcal{L} = \{a_1 \cdot \mathbf{b}_1 + a_2 \cdot \mathbf{b}_2 \mid a_1, a_2 \in \mathbb{Z}\}$$

Representándolo quedaría:

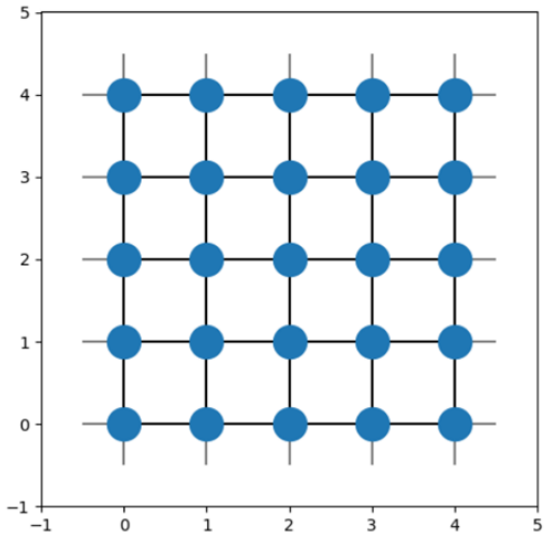


**Figura 4.1:** Representación en 2D de retículos

Si utilizamos los vectores ortonormales estándar de $\mathbb{R}^3$:

$$\mathbf{b}_1 = (1,0,0), \quad \mathbf{b}_2 = (0,1,0), \quad \mathbf{b}_3 = (0,0,1)$$

Obtendríamos la representación gráfica:

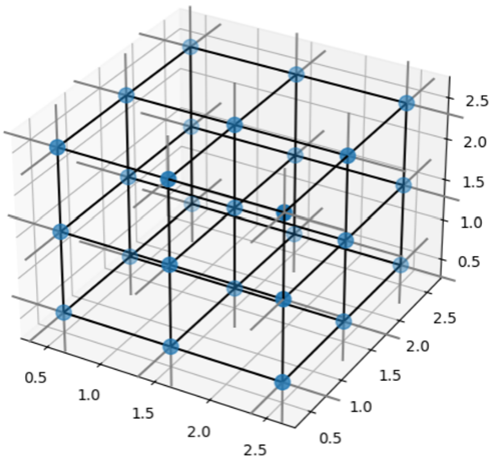


**Figura 4.2:** Representación en 3D de retículos

En el Anexo 2 se presentan las funciones utilizadas para generar las imágenes 4.1 y 4.2, así como otras herramientas que refuerzan los conceptos expuestos, mediante ejemplos prácticos que ilustran cómo visualizar y manipular los parámetros de los retículos en diversas dimensiones.

### 4.1.1. Mínimo no nulo de un retículo

Dado que un retículo es la versión discreta de un subespacio vectorial, se puede hablar del elemento no nulo más pequeño del retículo en términos de su norma. Este valor se denomina mínimo no nulo (o distancia mínima) del retículo, y se expresa como:

$$\lambda_1(\mathcal{L}) = \min \{ \|\mathbf{x}\| \ : \ \mathbf{x} \in \mathcal{L},\ \mathbf{x} \neq \mathbf{0} \}$$

Rescatando el ejemplo anterior con

$$\mathbf{b}_1 = (1,0,0), \quad \mathbf{b}_2 = (0,1,0), \quad \mathbf{b}_3 = (0,0,1)$$

y considerando la norma euclídea, debemos encontrar el vector no nulo más corto. Para ello, necesitamos el menor valor de $\|\mathbf{x}\|$ cuando $(a_1, a_2, a_3) \neq (0,0,0)$. Esto ocurre cuando uno de los coeficientes es igual a $\pm 1$ y los otros son cero:

$$\begin{aligned}
\text{Si } \mathbf{x} = (1,0,0), &\ \Rightarrow \|\mathbf{x}\| = \sqrt{1^2 + 0^2 + 0^2} = 1 \\
\text{Si } \mathbf{x} = (0,1,0), &\ \Rightarrow \|\mathbf{x}\| = \sqrt{0^2 + 1^2 + 0^2} = 1 \\
\text{Si } \mathbf{x} = (0,0,1), &\ \Rightarrow \|\mathbf{x}\| = \sqrt{0^2 + 0^2 + 1^2} = 1
\end{aligned}$$

Deducimos que el mínimo no nulo de este retículo es:

$$\lambda_1(\mathcal{L}) = 1$$

## 4.2. Retículos basados en anillos

Un retículo sobre un anillo utiliza las propiedades algebraicas de los anillos para representar los puntos del retículo. Mientras que un retículo, como hemos visto en el apartado anterior, está definido en un espacio vectorial, en los retículos de anillos se trabaja dentro de un anillo algebraico, lo que permite una representación más compacta y eficiente. Este enfoque proporciona tanto eficiencia computacional como resistencia frente a ataques.

Para comprender cómo surgen los retículos sobre anillos, primero debemos revisar sus fundamentos matemáticos, como los anillos algebraicos y los anillos cocientes, lo que también nos llevará a aprender sobre el concepto de ideal de un anillo.

A continuación, explicaremos estas nociones matemáticas. Empezaremos conociendo el concepto de anillo algebraico para después poder entender los anillos polinómicos. Posteriormente explicaremos los anillos ideales, necesarios para entender los anillos cocientes, fundamental para entender, en última instancia, los retículos sobre anillos. Finalmente, veremos los módulos sobre anillos, cómo se trabaja con ellos, y su importante relación con los retículos sobre anillos.

Para realizar este apartado se ha usado como referencia el documento Estructuras Algebraicas realizado por el equipo docente de la asignatura Estructuras Algebraicas del grado de Matemáticas de la Universidad de Sevilla [32].

Las implementaciones de estos fundamentos se encuentran alojadas en el Anexo 3 https://github.com/PQC-standards/Fundamentos-Teoricos/blob/main/Anexo3AnillosAlgebraicos.ipynb.

### 4.2.1. Anillos algebraicos

Un anillo, R, es una estructura algebraica que consta de dos operaciones binarias: la suma y el producto, que satisfacen una serie de propiedades fundamentales [33].

#### Propiedades de $(R, +)$: El grupo abeliano

La operación de suma en un anillo satisface las siguientes propiedades al ser $(R, +)$ un grupo abeliano, es decir, un grupo conmutativo respecto a la suma:

- **Operación interna:** Para todo $a, b \in R$, se cumple que $a + b \in R$.
- **Asociativa:** Para todo $a, b, c \in R$, se cumple que $(a + b) + c = a + (b + c)$.
- **Conmutativa:** Para todo $a, b \in R$, se cumple que $a + b = b + a$.
- **Elemento neutro aditivo:** Existe un elemento $0 \in R$ tal que, para todo $a \in R$, se cumple que $a + 0 = a$.
- **Elemento opuesto (inverso aditivo):** Para cada $a \in R$, existe un elemento $-a \in R$ tal que $a + (-a) = 0$.

#### Propiedades del producto en $R$

La operación de producto en un anillo satisface las siguientes propiedades:

- **Cerradura:** Para todo $a, b \in R$, se cumple que $a \cdot b \in R$.
- **Asociativa:** Para todo $a, b, c \in R$, se cumple que $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
- **Distributiva:** Para todo $a, b, c \in R$, se cumplen ambas distributividades:

$$a \cdot (b + c) = a \cdot b + a \cdot c, \quad \text{y} \quad (a + b) \cdot c = a \cdot c + b \cdot c$$

Es importante, además, conocer los términos de anillos conmutativos y unitarios. Un anillo conmutativo es aquel en el que el producto es conmutativo, es decir, $a \cdot b = b \cdot a$ para todo $a, b \in R$. Por otro lado, un anillo unitario es aquel en el que existe un elemento neutro multiplicativo (o elemento unidad) $1 \neq 0 \in R$ [34].

**Ejemplo de anillo:** $(\mathbb{Z}_3, +, \cdot)$

Un ejemplo de anillo podría ser $(\mathbb{Z}_3, +, \cdot)$, basado en los restos de la división entre 3. El conjunto de elementos que lo componen es $\{0, 1, 2\}$, y sus operaciones son la suma y la multiplicación módulo 3.

Es rápidamente deducible que las propiedades anteriormente mencionadas se cumplen. Además, de acuerdo con las definiciones previas, podemos afirmar que se trata de un **anillo conmutativo y unitario**, dado que:

- La operación de multiplicación es conmutativa para todos sus elementos, es decir, $a \cdot b = b \cdot a$ para todo $a, b \in \mathbb{Z}_3$.
- Existe un elemento neutro multiplicativo $1 \in \mathbb{Z}_3$, tal que $a \cdot 1 = a$ para todo $a \in \mathbb{Z}_3$.

### 4.2.2. Anillos polinómicos

Una vez entendido el entramado matemático de un anillo, veremos un caso concreto: los anillos polinómicos. Estos son los anillos que constituyen la base sobre la cual se definen los retículos sobre anillos y los módulos sobre anillos.

Los anillos polinómicos son una estructura matemática formada por polinomios con coeficientes en un anillo dado, combinados bajo las operaciones de suma y multiplicación de polinomios. En esencia, un anillo de polinomios puede considerarse una extensión de conceptos más sencillos, como los números enteros y los polinomios [35].

Matemáticamente, el anillo de polinomios en una variable $x$, denotado por $R[x]$, es el conjunto de expresiones, llamadas polinomios en $x$, de la forma que se muestra en la ecuación 4.2:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \tag{4.2}$$

donde $n \geq 0$ es el grado del polinomio, $a_i \in R$ para $i = 0, 1, \ldots, n$, y $x$ es una variable formal [36].

**Verbigracia:** Consideremos el anillo polinómico $R[x]$, con coeficientes comprendidos en $R$, el conjunto de todos los números reales, y $x$ una variable indeterminada. Un elemento de $R[x]$ podría ser $2x^2 + 3x + 5$, donde los coeficientes son números reales.

Tanto la suma como el producto de polinomios se definen de manera estándar: se suman los coeficientes de cada potencia de $x$ y se multiplican combinando los términos según las reglas de distributividad y conmutatividad del anillo $R$.

**Algunas propiedades importantes:**

- **Conmutatividad:** Si $R$ es un anillo conmutativo, entonces $R[x]$ también lo es.
- **Unidad:** Si $R$ tiene un elemento neutro multiplicativo (elemento unidad), entonces $R[x]$ también lo tiene.

**Anillos de polinomios en varias variables**

Si tomamos $n$ variables $x_1, x_2, \ldots, x_n$, obtenemos $R[x_1, x_2, \ldots, x_n]$, el conjunto de polinomios donde los monomios son productos de potencias $x_1, x_2, \ldots, x_n$ y los coeficientes están en $R$. Por ejemplo, para $R[x, y]$:

$$f(x, y) = a_{2,1}x^2y + a_{1,0}x + a_{0,2}y^2 + a_{0,0}$$

En el apartado 4 del Anexo 3 se presentan implementaciones que permiten generar ejemplos de polinomios pertenecientes a anillos, dada una base, así como realizar operaciones entre ellos y verificar propiedades del anillo, como la conmutatividad y la existencia de un elemento neutro multiplicativo.

### 4.2.3. Ideales en anillos

Un ideal es un conjunto especial dentro de un anillo que permite generalizar la idea de divisibilidad. Tiene dos propiedades clave:

- **Suma cerrada:** Si sumas dos elementos del ideal, el resultado sigue en el ideal.
- **Multiplicación absorbente:** Si multiplicas un elemento del anillo por un elemento del ideal, el resultado también está en el ideal.

El ideal actúa como un «subconjunto absorbente» que ayuda a dividir el anillo en partes y construir estructuras como los *anillos cocientes*, que veremos en el siguiente apartado.

Tomemos como ejemplo el anillo $\mathbb{Z}$, que es el conjunto de los números enteros con las operaciones habituales de suma y multiplicación. Ahora consideremos el conjunto de los múltiplos de un número cualquiera $n$, denotado como $n\mathbb{Z} = \{nk \mid k \in \mathbb{Z}\}$. Este conjunto es un ideal de $\mathbb{Z}$.

Por ejemplo, el conjunto de múltiplos de 3, $3\mathbb{Z} = \{\ldots, -6, -3, 0, 3, 6, \ldots\}$, es un ideal. Lo podemos demostrar apoyándonos en las propiedades de los ideales:

- **Suma cerrada:** Si $a, b \in 3\mathbb{Z}$, entonces $a + b \in 3\mathbb{Z}$.
- **Multiplicación absorbente:** Si $r \in \mathbb{R}$ y $a \in 3\mathbb{Z}$, entonces $r \cdot a \in 3\mathbb{Z}$.

En la implementación alojada en el Anexo 4 encontraremos métodos para construir ideales en anillos y verificar propiedades fundamentales de los ideales, tales como el cierre bajo la adición y la propiedad absorbente respecto a la multiplicación por elementos del anillo.

### 4.2.4. Anillos cocientes

Un anillo cociente se obtiene al dividir un anillo $R$ por un ideal $I$, y se escribe como $R/I$. Los elementos del anillo cociente son grupos de números de $R$ que se consideran iguales si la diferencia entre ellos está en el ideal $I$. Es decir, si tomamos dos números $a$ y $b$ de $R$, se consideran iguales en el anillo cociente si su resta $a - b$ está en el ideal $I$. En este nuevo anillo, las operaciones de suma y multiplicación se realizan entre estos grupos, y el ideal $I$ actúa como el «cero» [37].

Por ejemplo: Tomando el anillo $\mathbb{Z}$ y el ideal $3\mathbb{Z}$, el anillo cociente $\mathbb{Z}/3\mathbb{Z}$ tiene tres elementos: $[0], [1], [2]$, que representan los restos posibles al dividir un número entero entre 3.

**Suma en $\mathbb{Z}/3\mathbb{Z}$:**

$$[1] + [2] = [3] = [0] \quad \text{mód } 3.$$

**Multiplicación en $\mathbb{Z}/3\mathbb{Z}$:**

$$[2] \cdot [2] = [4] = [1] \quad \text{mód } 3.$$

Aquí, $3\mathbb{Z}$ actúa como el cero porque $n \in 3\mathbb{Z}$ equivale a $n \equiv 0 \mod 3$.

En el apartado 6 sobre anillos cocientes del Anexo 4 se presenta un método que permite visualizar todas las operaciones posibles de suma y multiplicación entre elementos de un anillo cociente, dado un módulo $n$. En particular, se continua el ejemplo para el caso del anillo cociente $\mathbb{Z}/3\mathbb{Z}$.

### 4.2.5. Retículos sobre anillos

Tras haber aprendido en el apartado 4.1 lo que son los retículos o lattices y, en los apartados posteriores, los fundamentos y propiedades en los que se sustentan los anillos, exploraremos a continuación los retículos sobre anillos, los cuales usan todo lo aprendido anteriormente.

Esta herramienta matemática combina las propiedades de los retículos tradicionales con la estructura algebraica de los anillos. Esto proporciona herramientas más poderosas y compactas para modelar problemas complejos como el *RLWE* que estudiaremos más adelante.

Los retículos basados en anillos son una variante estructurada en la que los puntos del retículo se definen usando anillos de polinomios. La idea clave es aprovechar la estructura algebraica de un anillo para representar y manipular los vectores de manera más eficiente. Normalmente trabajaremos en un anillo cociente

de polinomios, típicamente $R[x]/(f(x))$, donde $f(x)$ es un polinomio irreducible o ciclotómico y los coeficientes de los polinomios están en $\mathbb{Z}$ o en un anillo finito $\mathbb{Z}_q$.

Un retículo basado en anillos es un subconjunto del anillo $R_q = \mathbb{Z}_q[x]/(f(x))$ con coeficientes en $\mathbb{Z}_q$ y restricciones adicionales que lo hacen actuar como un retículo en el sentido clásico.

Por ejemplo, supongamos el anillo: $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ con $q = 7$, $n = 4$. Un polinomio genérico $p(x) \in R_q$ tiene la forma:

$$p(x) = p_0 + p_1 x + p_2 x^2 + p_3 x^3 \quad \text{con} \quad p_0, p_1, p_2, p_3 \in \mathbb{Z}_7.$$

Construyamos a continuación el retículo basado en este anillo. El retículo se define por un conjunto de polinomios en $R_7$. Supongamos los dos siguientes:

$$g_1(x) = 1 + 2x + 3x^2 + x^3, \quad g_2(x) = 2 + x + 2x^2 + 3x^3.$$

El retículo estará compuesto por todas las combinaciones lineales enteras de estos polinomios, es decir:

$$\mathcal{L} = \{a g_1(x) + b g_2(x) \mid a, b \in \mathbb{Z}_7\}.$$

Este conjunto $\mathcal{L}$ es el retículo generado por $g_1(x)$ y $g_2(x)$ en el anillo ciclotómico $R_7$, donde las combinaciones lineales se toman con coeficientes en $\mathbb{Z}_7$.

En el último apartado del Anexo 4 se implementa un método para construir y visualizar retículos definidos sobre un anillo de polinomios. A partir de un conjunto de generadores, se generan todas las combinaciones lineales posibles dentro del retículo, permitiendo representar su estructura algebraica. Además, se incluye una función que verifica si un polinomio dado pertenece al retículo. La representación gráfica del retículo generado se muestra al final de este apartado, proporcionando una herramienta visual que complementa la comprensión teórica del concepto.

### 4.2.6. Módulo sobre un anillo

Una vez comprendido el concepto de retículo sobre un anillo, debemos abordar una última noción matemática: los módulos sobre anillos. Estos son especialmente importantes porque generalizan los retículos sobre anillos, ofreciendo una estructura más rica. De hecho, un retículo sobre un anillo puede considerarse un módulo sobre un anillo. Este concepto resulta clave para entender cómo se construyen y manipulan los retículos en contextos algebraicos más complejos, como en el caso de los anillos de enteros utilizados en la criptografía basada en el problema RLWE.

A grandes rasgos, un módulo es un conjunto de elementos en el que se pueden hacer sumas y multiplicaciones con elementos de un anillo. Por ejemplo, si

contemplamos un conjunto de números, como los enteros $\mathbb{Z}$, podríamos multiplicarlos por otros números en $\mathbb{Z}$ y sumarlos, y todos los resultados que obtendríamos serían también números enteros. Este conjunto sería un módulo sobre el anillo de los enteros $\mathbb{Z}$.

Matemáticamente, un módulo $M$ sobre un anillo $R$ es una estructura algebraica que generaliza la noción de espacio vectorial, añadiendo la posibilidad de operar con elementos de un anillo. Está dotado de dos operaciones: una adición interna y una acción escalar [38].

La adición interna convierte a $M$ en un grupo abeliano, es decir, un conjunto con una operación de adición que es asociativa, conmutativa y con un elemento neutro. Por otro lado, la acción escalar, definida por los elementos de $R$ sobre los de $M$, cumple propiedades específicas relacionadas con la estructura del anillo.

Estas operaciones cumplen las siguientes propiedades:

**Propiedades de la suma interna en $M$:**

- **Asociativa:** Para todo $m_1, m_2, m_3 \in M$, $(m_1 + m_2) + m_3 = m_1 + (m_2 + m_3)$.
- **Conmutativa:** Para todo $m_1, m_2 \in M$, $m_1 + m_2 = m_2 + m_1$.
- **Existencia de elemento neutro:** Existe un elemento $0 \in M$ tal que, para todo $m \in M$, $m + 0 = m$.

**Propiedades del producto en $R$:**

- **Compatibilidad con la estructura de $R$:** Para todo $a, b \in R$ y $m \in M$, se cumple que $(a \cdot b) \cdot m = a \cdot (b \cdot m)$.
- **Distributiva:** Para $a \in R$ y $m_1, m_2 \in M$, se cumple: $a \cdot (m_1 + m_2) = a \cdot m_1 + a \cdot m_2$.
- **Compatibilidad con la adición en $R$:** Para $a, b \in R$ y $m \in M$, se cumple: $(a + b) \cdot m = a \cdot m + b \cdot m$.
- **Acción del elemento neutro de $R$:** Si $R$ tiene un elemento neutro multiplicativo $1_R$, entonces $1_R \cdot m = m$ para todo $m \in M$.

**Propiedades heredadas y relación con los anillos:**

Los módulos heredan muchas propiedades de los anillos sobre los que están definidos. Por ejemplo:

- Si el anillo $R$ es conmutativo, la multiplicación por escalares en $M$ también será conmutativa.
- La subestructura de un módulo se denomina *submódulos* (denotados como $N$), análogos a los subespacios vectoriales.

- Los homomorfismos de módulos son funciones $f : M \to N$ que preservan la estructura de módulo, es decir, son lineales con respecto a las operaciones de módulo:

$$f(m_1 + m_2) = f(m_1) + f(m_2), \quad f(a \cdot m) = a \cdot f(m),$$

donde $a \in R$ y $m, m_1, m_2 \in M$.

- Un ideal de un anillo $R$ es un subconjunto $r \subseteq R$ que puede considerarse el núcleo de un homomorfismo de anillos.

Veamos un ejemplo considerando el conjunto $\mathbb{Z}_3$ (el conjunto de números enteros módulo 3) y veremos cómo podemos estructurar un módulo sobre el anillo $\mathbb{Z}$ usando $\mathbb{Z}_3$ como conjunto. El conjunto $\mathbb{Z}_3$ está formado por los elementos $\{0, 1, 2\}$, y las operaciones de suma y multiplicación se realizan módulo 3. Definimos el conjunto $M = \mathbb{Z}_3 = \{0, 1, 2\}$, y vamos a ver cómo este conjunto es un módulo sobre el anillo $\mathbb{Z}$ verificando sus propiedades de suma interna y de producto escalar:

**Cerrado bajo la suma (propiedad heredada del anillo $\mathbb{Z}_3$):**

Si tomamos dos elementos de $M = \{0, 1, 2\}$, como $1 + 2$, obtenemos $3 \mod 3 = 0$, y deducimos que el resultado es correcto ya que $0 \in M$.

**Multiplicación por un escalar:**

Si tomamos un escalar $r = 2 \in \mathbb{Z}$ y un elemento $m = 2 \in M$ y multiplicamos: $2 \cdot 2 = 4$, $4 \mod 3 = 1$, siendo $1 \in M$.

Luego, al ver cómo la suma y la multiplicación de escalares funcionan en este conjunto, respetando las propiedades, podemos llegar a la conclusión de que el conjunto $M = \mathbb{Z}_3 = \{0, 1, 2\}$ es un módulo sobre el anillo $\mathbb{Z}$.

**Operaciones y construcciones en módulos**

Disponemos de tres operaciones y construcciones en módulos basados en anillos; no obstante, la última, la factorización, no es esencial para comprenderla en el contexto de la criptografía postcuántica.

**Suma y producto de módulos:** Dados $M_1, M_2 \subseteq M$, su suma $M_1 + M_2$ es el conjunto de elementos que pueden escribirse como $m_1 + m_2$, con $m_1 \in M_1$ y $m_2 \in M_2$. Matemáticamente:

$$M_1 + M_2 = \{m_1 + m_2 \mid m_1 \in M_1, m_2 \in M_2\}.$$

El producto tensorial $M_1 \otimes_R M_2$ permite construir nuevos módulos combinando elementos de $M_1$ y $M_2$ de manera lineal. De la forma:

$$(m_1 + m_1') \otimes m_2 = m_1 \otimes m_2 + m_1' \otimes m_2.$$

**Anulación:** El anulador de un elemento $m \in M$ es el conjunto de escalares $a \in R$ tales que $a \cdot m = 0$. Matemáticamente:

$$\text{Ann}(m) = \{a \in R \mid a \cdot m = 0\}.$$

**Factorización:** Dado un submódulo $N \subseteq M$, el módulo cociente $M/N$ es el conjunto de clases de equivalencia $[m] = \{m + n \mid n \in N\}$, con operaciones inducidas.

Veamos a continuación un ejemplo para cada una de las operaciones empezando por la suma y el producto tensorial:

Supongamos que $M = \mathbb{Z}$, el conjunto de los enteros y que $N_1$ y $N_2$ son dos submódulos de $M$. Definimos los submódulos $N_1 = 2\mathbb{Z} = \{\dots, -2, 0, 2, \dots\}$ y $N_2 = 3\mathbb{Z} = \{\dots, -3, 0, 3, \dots\}$. La suma $N_1 + N_2$ es el conjunto de todos los enteros que se pueden escribir como $n_1 + n_2$, donde $n_1 \in N_1$ y $n_2 \in N_2$, es decir:

$$N_1 + N_2 = \{n_1 + n_2 \mid n_1 \in 2\mathbb{Z}, n_2 \in 3\mathbb{Z}\}.$$

Usemos este mismo ejemplo para calcular un ejemplo de producto tensorial. Si $N_1 = 2\mathbb{Z}$ y $N_2 = 3\mathbb{Z}$, entonces $2 \otimes 3$ representa una base del producto tensorial. Cualquier elemento en $N_1 \otimes_{\mathbb{Z}} N_2$ puede ser una combinación lineal de estos términos. Esencialmente, ambos generan un conjunto mediante combinaciones lineales, pero el producto tensorial lo hace con reglas adicionales para respetar linealidad en cada componente.

Veremos a continuación un ejemplo de anulador. Para ello rescatemos el mismo módulo del ejemplo anterior $M = \mathbb{Z}$ y supongamos un segundo módulo $M_2 = \mathbb{Z}^2$. Tomemos por ejemplo el vector $m = (2, 3)$ contenido en el módulo $M_2$. Nuestro objetivo es encontrar el anulador, es decir, todos los escalares $a \in M$ tales que $a \cdot m = 0$, donde la multiplicación se realiza componente a componente:

$$a \cdot m = a \cdot (2, 3) = (a \cdot 2, a \cdot 3) = (2a, 3a).$$

Debemos conseguir que $a \cdot m = 0$, es decir, $(2a, 3a) = (0, 0)$. Esto significa que ambas componentes deben ser cero:

$$2a = 0; \quad a = 0$$

$$3a = 0; \quad a = 0.$$

Por lo tanto, deducimos que el único valor de $a \in M$ que satisface ambas ecuaciones es $a = 0$ y, en este caso, el anulador de $m = (2, 3)$ es:

$$\text{Ann}((2, 3)) = \{0\}.$$

### 4.2.7. Aplicaciones de los anillos en criptografía

Los retículos sobre anillos han sido una innovación fundamental dentro de la criptografía moderna, especialmente en el contexto de la criptografía postcuántica.

La clave del éxito de esta estructura matemática radica en las propiedades algebraicas de los anillos, que permiten simplificar y optimizar los cálculos necesarios para implementar algoritmos criptográficos de forma más eficiente y segura. Entre las principales ventajas que ofrecen los retículos sobre anillos, se destacan las siguientes:

1. **Eficiencia computacional:** El uso de anillos polinómicos, especialmente aquellos definidos como cocientes de anillos, permite realizar operaciones algebraicas de manera mucho más rápida y eficiente que en los retículos tradicionales. Esto se debe a que las operaciones de suma, multiplicación y reducción en estos anillos se pueden hacer con algoritmos optimizados que manejan las estructuras algebraicas inherentes de forma más compacta.
2. **Seguridad basada en problemas difíciles:** Los problemas fundamentales sobre los que se construye la seguridad de los retículos sobre anillos, como el Ring–Learning With Errors (RLWE), son altamente resistentes a los ataques cuánticos. Estos problemas están basados en la dificultad de encontrar soluciones a sistemas de ecuaciones lineales con errores, un desafío computacionalmente intratable incluso para ordenadores cuánticos. La seguridad en estos sistemas no depende de la factorización de grandes números ni de la resolución de logaritmos discretos, que son vulnerables a los algoritmos cuánticos (como el algoritmo de Shor), sino en la dificultad de resolver problemas geométricos y algebraicos de alta dimensión, que siguen siendo difíciles de resolver incluso con la computación cuántica.
3. **Estructura algebraica robusta:** Al trabajar con anillos, especialmente con anillos de polinomios, los retículos sobre anillos aprovechan una estructura algebraica rica que proporciona una base sólida para una gran variedad de aplicaciones criptográficas. Esto incluye la encriptación homomórfica (tanto parcial como total), que es esencial para realizar cálculos en datos cifrados sin necesidad de descifrarlos previamente, manteniendo la privacidad de la información. Además, los ideales en anillos polinómicos permiten crear sistemas criptográficos más eficientes con mejores propiedades de seguridad, ya que los elementos dentro de los ideales actúan como una base que organiza y restringe las posibles soluciones dentro del anillo, dificultando aún más los ataques.
4. **Escalabilidad y flexibilidad:** Una de las grandes ventajas de los retículos sobre anillos es su escalabilidad. A medida que aumenta la dimensión del retículo y el tamaño del anillo, la dificultad para resolver los problemas en los que se basan estos esquemas criptográficos aumenta exponencialmente, lo que permite ajustar el nivel de seguridad según los requisitos del sistema sin perder eficiencia. Esta flexibilidad también permite la implementación de diversos tipos de criptografía avanzada, como firmas digitales, intercambio de claves y autenticación, adaptándose a las necesidades de seguridad de aplicaciones futuras en un mundo cuántico.

En conclusión, los retículos sobre anillos constituyen un avance clave en la criptografía moderna, al ofrecer eficiencia computacional, seguridad robusta y una estructura algebraica sólida. Proporcionan una base segura y eficiente para

enfrentar los desafíos de la era cuántica, con la flexibilidad necesaria para adaptarse a diversas aplicaciones tecnológicas. Además, la seguridad proporcionada por sus problemas es resistente a los avances de la computación cuántica, ofreciendo una protección robusta frente a ataques que comprometen sistemas tradicionales. A continuación, profundizaremos en los problemas relacionados con estos esquemas criptográficos [39].

# 5. Problemas relacionados

A continuación, se abordarán los problemas relacionados con retículos y anillos en el contexto de la criptografía postcuántica, como el problema del vector más corto (SVP), el problema del vector más cercano (CVP) y el problema de aprendizaje con errores (LWE) y sus variantes en anillo y módulo (RLWE y MLWE), que son clave para el desarrollo de esquemas criptográficos resistentes a ataques cuánticos. En el anexo 3 y 4 del trabajo (los cuales se pueden encontrar en el repositorio «Problemas Relacionados») se ha desarrollado implementaciones para cada uno de estos problemas con sus respectivas explicaciones paso a paso.

## 5.1. Problemas relacionados con retículos

### 5.1.1. Problema del vector más cercano (CVP)

En un retículo $\mathcal{L}$, el problema del vector más cercano [18] (sección B.2. Criptografía Basada en Retículos) [40] consiste en encontrar el vector $v \in \mathcal{L}$ que esté más cerca de un punto dado $t \in \mathbb{R}^n$ en términos de la norma euclidiana. Formalmente, se busca:

$$\text{Encontrar } v \in \mathcal{L} \text{ que minimice } \|v - t\|, \text{ siendo } v = \arg\min_{w \in \mathcal{L}} \|w - t\| \tag{5.1}$$

Entiéndase $w$ como una instancia de cualquier vector del retículo. La importancia del CVP radica en su complejidad computacional, es conocido por ser un problema NP–Hard (al menos tan difícil como cualquier problema de la clase NP) tanto en su forma exacta como aproximada. Resolver el CVP de forma exacta tiene una complejidad exponencial. Los algoritmos de búsqueda por enumeración tienen una complejidad $O(c^n)$, donde $c > 1$. Algoritmos más avanzados, como los basados en reducción BKZ, los cuales veremos más adelante, tienen una complejidad de $O\left(2^{c \cdot n^{1-1/b}}\right)$, con un mejor factor de aproximación dependiendo del tamaño del bloque $b$.

### Ejemplo CVP

Primeramente, debemos definir la base del retículo, por ejemplo: en $\mathbb{R}^3$, usamos la base estándar:

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

conformada por los vectores $(1,0,0)$, $(0,1,0)$ y $(0,0,1)$. Los puntos generados por combinaciones enteras de esta base forman el retículo.

Seguidamente, debemos definir el punto objetivo: Consideremos el punto $t = (6,6,6)$ en el espacio tridimensional.

Por último, intentamos encontrar el vector más cercano al punto $t$ probando de forma sistemática: Supongamos que se elige $v = (5,5,5)$ como la solución. Calculamos la distancia euclidiana:

$$\|v - t\| = \sqrt{(5-6)^2 + (5-6)^2 + (5-6)^2} \approx 1{,}73$$

Aunque no tenemos por qué saberlo, este vector no es la mejor elección, ya que podemos encontrar un vector aún más cercano. No obstante, esta norma será almacenada por el momento como la mínima encontrada hasta que el algoritmo encuentre una mejor.

Dado que la base del retículo es la identidad, el vector correcto es simplemente la componente entera de cada coordenada, ya que este vector pasará justo por el punto $t$: $v = (6,6,6)$.

Calculamos la distancia con respecto al punto $t$:

$$\|v - t\| = \sqrt{(6-6)^2 + (6-6)^2 + (6-6)^2} = 0$$

Y como no puede existir una norma menor, hemos dado con una solución segura.

Véase dos ejemplos gráficos a continuación, tanto en dos como en tres dimensiones, cuyas bases vectoriales son los estándares en $\mathbb{R}^2$ y $\mathbb{R}^3$, donde los puntos elegidos serán $t_{2d} = (6,6)$ y $t_{3d} = (6,6,6)$.

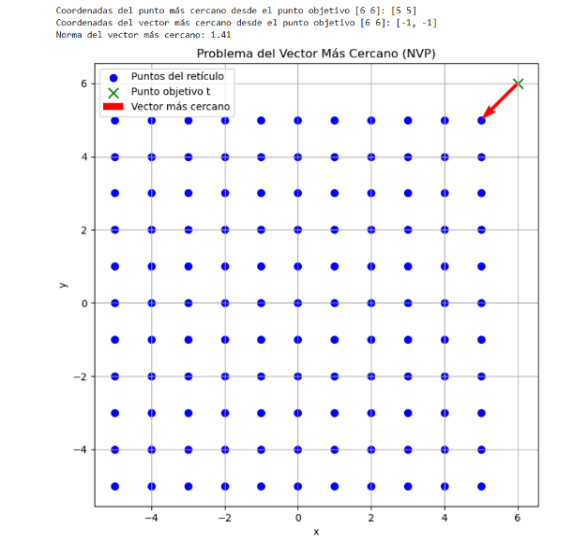


**Figura 5.1:** Representación en 2D de CVP

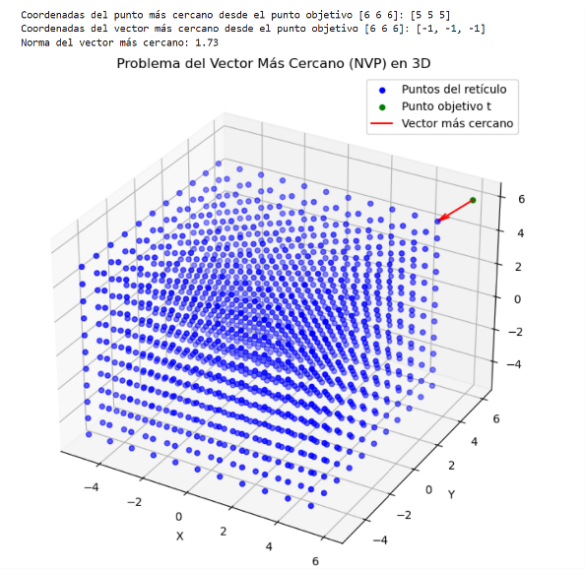


**Figura 5.2:** Representación en 3D de CVP

Como se puede observar en ambas resoluciones gráficas, en el caso de un retículo con base matriz identidad y un punto objetivo $t = (t_1, t_2, \ldots, t_n)$, el vector más cercano a este punto siempre será $v = (t_1, t_2, \ldots, t_n)$.

Es importante tener en cuenta que, aunque en bajas dimensiones y con bases ortonormales, este problema es trivial, la idea es que la elección de los parámetros sea mucho más compleja, y genere un problema no resoluble en tiempo polinomial, como se verá en apartados posteriores. En el anexo 4, donde se han implementado los problemas relacionados con retículos, podemos ver cómo varía la complejidad según los parámetros que se establezcan.

### 5.1.2. Problema del vector más corto (SVP)

En un retículo $\mathcal{L}$, el problema del vector más corto [18] (sección B.2. Criptografía Basada en Retículos) [41] [40] consiste en encontrar un vector no nulo $v \in \mathcal{L}$ que minimice la norma euclidiana $\|v\|$, es decir:

$$v = \arg \min_{w \in \mathcal{L}, w \neq 0} \|w\| \tag{5.2}$$

Este problema pertenece a la clase NP–Hard, ya que, para encontrar el vector más corto, es necesario considerar todos los vectores posibles dentro del retículo. Esto implica una búsqueda exhaustiva en espacios de alta dimensión, lo que conlleva una explosión combinatoria de magnitudes exponenciales. Como resultado, la resolución exacta del problema se vuelve inalcanzable, incluso para computadoras cuánticas. La complejidad teórica es idéntica a la mencionada en el CVP, $O(c^n)$, donde $c > 1$ para una resolución exacta, y $O(2^{c \cdot n^{(1-1/b)}})$ para una resolución aproximada por BKZ.

## Ejemplo SVP

Análogamente al ejemplo anterior, debemos definir la base del retículo: en $\mathbb{R}^3$, usamos la base estándar:

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

conformada por los vectores $(1,0,0)$, $(0,1,0)$ y $(0,0,1)$. Los puntos generados por combinaciones enteras de esta base forman el retículo.

Seguidamente, intentamos encontrar el vector más corto en el retículo probando de forma sistemática: supongamos que se elige $v = (2,1,1)$ como candidato. Calculamos su norma euclidiana:

$$\|v\| = \sqrt{(2)^2 + (1)^2 + (1)^2} \approx 2{,}45$$

Aunque no tenemos por qué saberlo, este vector no es el que tiene la norma más corta, ya que podemos encontrar un vector aún más pequeño. No obstante, esta norma será almacenada por el momento como la mínima encontrada hasta que el algoritmo encuentre una mejor.

Dado que la base del retículo es la identidad, existen tres vectores que ostentan la norma mínima, y estos simplemente serán los que forman la base:

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

Calculamos sus normas:

$$\|v_1\| = \sqrt{(1)^2 + (0)^2 + (0)^2} = 1$$
$$\|v_2\| = \sqrt{(0)^2 + (1)^2 + (0)^2} = 1$$
$$\|v_3\| = \sqrt{(0)^2 + (0)^2 + (1)^2} = 1$$

Cuando la base del retículo es la matriz identidad, el retículo generado es simplemente $\mathbb{Z}^3$ (coordenadas enteras en 3D). En este caso, los vectores más cortos siempre serán los vectores de la base estándar, ya que tienen la menor norma posible distinta de cero.

Si la base fuera diferente, encontrar el vector más corto sería mucho más difícil, ya que implicaría realizar estas operaciones con todos los vectores del retículo, y requeriría algoritmos avanzados como LLL o BKZ.

Véase dos representaciones gráficas a continuación, tanto en dos como en tres dimensiones, cuyas bases vectoriales son los estándares en $\mathbb{R}^2$ y $\mathbb{R}^3$ (como en el ejemplo analítico).

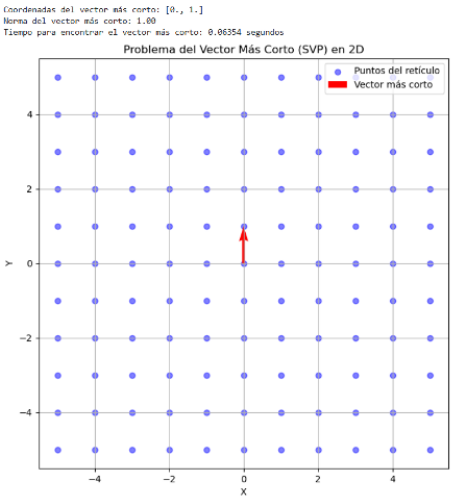


**Figura 5.3:** Representación en 2D de SVP

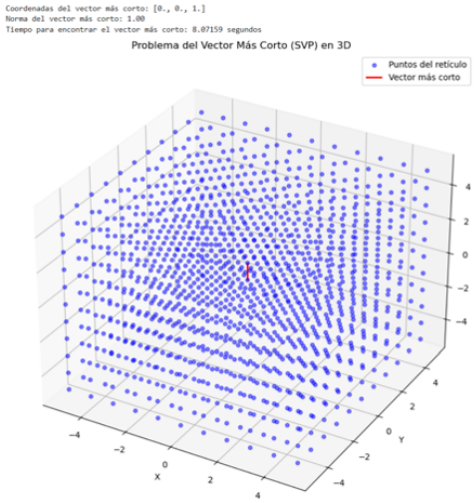


**Figura 5.4:** Representación en 3D de SVP

Aunque a priori, tanto el SVP como el CVP son problemas pertenecientes a la categoría de no resolubles en tiempo polinomial, y ambos tienen una complejidad teórica idéntica, esto no significa que, en la práctica, los tiempos también sean idénticos, como investigaremos más adelante en el apartado del análisis empírico de la complejidad.

### 5.1.3. Learning with errors (LWE)

Para comprender el algoritmo LWE [18] (sección B.2. Criptografía Basada en Retículos) [42], partiremos de un caso base de este llamado LFPWE (Learning From

Parity With Errors). Dado un entero $n \geq 1$ y un número real $\epsilon \geq 0$, el objetivo es encontrar una tupla $s$ perteneciente al conjunto de números $\mathbb{Z}_2^n$ dada una lista de $k$ ecuaciones con errores:

$$\begin{aligned} \langle s, a_1 \rangle &\approx_\epsilon b_1 \pmod 2 \\ \langle s, a_2 \rangle &\approx_\epsilon b_2 \pmod 2 \\ &\vdots \\ \langle s, a_k \rangle &\approx_\epsilon b_k \pmod 2 \end{aligned}$$

Donde:

- $k$ es el número de ecuaciones y $s$ es un vector en $\mathbb{Z}_2^n$ con $n$ componentes. Es común, pero no estrictamente necesario, que $k = n$ en muchos ejemplos de LFPWE, donde hay tantas ecuaciones como incógnitas.
- $\langle s, a_i \rangle = \sum_j s_j (a_i)_j$ es el producto escalar entre $s$ y $a_i$ módulo 2.
- $a_i \in \mathbb{Z}_2^n$ son vectores elegidos de manera uniforme y aleatoria con elementos comprendidos en $\mathbb{Z}_2$.
- $b_i \in \mathbb{Z}_2$ son resultados afectados por ruido con probabilidad $\epsilon$.

El objetivo es recuperar $s \in \mathbb{Z}_2^n$ dado un conjunto de pares $(a_i, b_i)$.

Nótese que el caso $\epsilon = 0$ puede resolverse de manera eficiente mediante, por ejemplo, eliminación gaussiana. Esto requiere $O(n)$ ecuaciones y tiempo polinómico $O(n^3)$. El problema se vuelve significativamente más difícil cuando tomamos cualquier valor positivo $\epsilon > 0$, debido a que las ecuaciones estarán afectadas por errores aleatorios; es decir, el valor observado $b_i$ ya no es exactamente el producto escalar $\langle s, a_i \rangle$, sino una versión ligeramente perturbada de este. En este contexto, métodos clásicos como la eliminación gaussiana dejan de ser eficaces, ya que suponen ecuaciones exactas.

Esto es, en el caso de usar eliminación gaussiana para encontrar un subconjunto $S$ de ecuaciones que permita estimar cada bit de $s$, nos encontraríamos con que, debido al ruido, este enfoque tiene una probabilidad de éxito que decrece exponencialmente con $n$. Por lo tanto, para obtener el primer bit con buena confianza, debemos repetir todo el procedimiento $2^{\Theta(n)}$ veces, lo cual resultaría en un algoritmo que utiliza $2^{O(n)}$ ecuaciones y tardaría $2^{O(n)}$.

El problema LWE extiende LFPWE al considerar cálculos en módulos más grandes y distribuciones de ruido más complejas, convirtiéndose así en un problema de decodificación de códigos lineales aleatorios. Este problema tiene su base en retículos, mientras que sus versiones más eficientes, como Ring–LWE o Module–LWE, están basadas en retículos sobre anillos y módulos sobre anillos respectivamente. Definimos LWE de la siguiente forma:

Sean:

- $p$, un número primo.

- $\chi$, una distribución de probabilidad sobre $\mathbb{Z}_p$, que genera el ruido $e_i$.
- $k$, la cantidad de muestras (ecuaciones) generadas.

Se nos da un conjunto de ecuaciones tal que:

$$\begin{aligned} \langle s, a_1 \rangle &\approx_\chi b_1 \pmod{p} \\ \langle s, a_2 \rangle &\approx_\chi b_2 \pmod{p} \\ &\vdots \\ \langle s, a_k \rangle &\approx_\chi b_k \pmod{p} \end{aligned}$$

O lo que es equivalente:

$$b_i \approx_\chi \langle s, a_i \rangle + e_i \pmod{p}, \quad \forall i \in \mathbb{Z}_k \tag{5.3}$$

Donde:

- $s \in \mathbb{Z}_p^n$ es el vector secreto que recuperar.
- $a_i \in \mathbb{Z}_p^n$ son vectores seleccionados de forma uniforme.
- $e_i \in \mathbb{Z}_p$ es el error añadido generado de acuerdo con $\chi$.
- $b_i \in \mathbb{Z}_p$ es el resultado observado.

Para resolver LWE mediante un ataque de fuerza bruta, necesitaríamos probar todas las posibles soluciones para $s$, y posteriormente verificar cuál de ellas satisface las ecuaciones dadas, teniendo en cuenta los errores asociados. En otras palabras, como $s \in \mathbb{Z}_p^n$, hay $p^n$ posibles valores para $s$. En el caso más simple, si $p = 2$, el número de posibles valores para $s$ es $2^n$, y para cada posible $s$ generado, deberemos calcular si verifica la ecuación:

$$\langle s, a_i \rangle \equiv b_i \pmod{p} \tag{5.4}$$

Con el error $e_i$ tolerado dentro del margen especificado por la distribución de ruido $\chi$, es decir, debido a que las ecuaciones están afectadas por errores, la comparación será tolerante y la solución deberá coincidir con $b_i$ en cada ecuación, considerando el ruido $e_i$.

Es conocido que resolver LWE es al menos tan difícil como resolver los problemas anteriores (CVP, SVP) en el peor caso. Esto refuerza la creencia de que LWE es intrínsecamente resistente incluso frente a algoritmos cuánticos.

## Ejemplo LWE

Para mejor comprensión de este problema, realizaremos un ejemplo sencillo que muestre las bases de este algoritmo.

**Parámetros:**

- $p = 5$ (trabajamos módulo 5).
- $n = 2$ (el vector secreto $s$ tiene dos componentes).
- $k = 3$ (tendremos 3 ecuaciones).
- La distribución de ruido $\chi$ generará errores pequeños entre $-1$ y 1.

**Paso 1:** Generamos el secreto $s$, la dimensión de este vector será la que determine la complejidad del problema:

$$s = (3, 2) \in \mathbb{Z}_5^2$$

**Paso 2:** Generamos 3 vectores aleatorios $a_i$ en $\mathbb{Z}_5^2$, estos vectores compondrán la matriz $A$:

$$a_1 = (1, 4), \quad a_2 = (2, 3), \quad a_3 = (4, 1)$$

**Paso 3:** Calculamos $b_i$ con ruido añadido:

$$b_i = \langle s, a_i \rangle + e_i \pmod 5$$

Donde $e_i$ es el error pequeño generado por la distribución gaussiana $\chi$.

**Cálculo de los productos escalares y suma con error:**

Para $a_1 = (1, 4)$:

$$\langle s, a_1 \rangle = (3 \cdot 1) + (2 \cdot 4) = 3 + 8 = 11$$

Con error $e_1 = -1$:

$$b_1 = (11 - 1) \pmod 5 = 10 \pmod 5 = 0$$

Para $a_2 = (2, 3)$:

$$\langle s, a_2 \rangle = (3 \cdot 2) + (2 \cdot 3) = 6 + 6 = 12$$

Con error $e_2 = 1$:

$$b_2 = (12 + 1) \pmod 5 = 13 \pmod 5 = 3$$

Para $a_3 = (4, 1)$:

$$\langle s, a_3 \rangle = (3 \cdot 4) + (2 \cdot 1) = 12 + 2 = 14$$

Con error $e_3 = 0$:

$$b_3 = (14 + 0) \pmod 5 = 14 \pmod 5 = 4$$

**Paso 4:** El sistema de ecuaciones LWE para un usuario que no conozca la clave secreta $s$ resultaría de la forma:

$$\begin{cases} (1, 4) \cdot (s_1, s_2) + e_1 = 0 \pmod 5 \\ (2, 3) \cdot (s_1, s_2) + e_2 = 3 \pmod 5 \\ (4, 1) \cdot (s_1, s_2) + e_3 = 4 \pmod 5 \end{cases}$$

Donde el objetivo es recuperar el secreto $s = (3, 2)$ sin conocer los errores $e_1, e_2, e_3$.

**Paso 5:** Ataque por fuerza bruta

El atacante probará todas las combinaciones posibles de $s_1, s_2$ en $\mathbb{Z}_5^2$ (es decir, $s_1, s_2$ pueden tomar valores entre 0 y 4) y verificará si las ecuaciones se cumplen aproximadamente, considerando que hay ruido. Generamos todas las combinaciones posibles de $s$, ya que $s_1, s_2 \in \mathbb{Z}_5^2$, hay $5^2 = 25$ combinaciones posibles:

$$(0, 0), (0, 1), (0, 2), \ldots, (4, 4)$$

El atacante probará cada una calculando los valores esperados de $b_i$ sin error y para cada posible $s = (s_1, s_2)$ el atacante calcula:

$$b_i' = \langle s, a_i \rangle \pmod 5$$

Para cada una de las ecuaciones.

Si $b_i'$ está cerca de los valores observados de $b_i$, el atacante considerará esa combinación como posible.

Pongamos el ejemplo, si el atacante prueba con $s = (0, 0)$:

$$\begin{aligned} b_1' &= (1 \cdot 0 + 4 \cdot 0) \pmod 5 = 0 \\ b_2' &= (2 \cdot 0 + 3 \cdot 0) \pmod 5 = 0 \\ b_3' &= (4 \cdot 0 + 1 \cdot 0) \pmod 5 = 0 \end{aligned}$$

Estos valores están fuera del rango de error $\pm 1$ con respecto a $(0, 3, 4)$, por lo que el atacante descarta $(0, 0)$.

Si decide probar con $s = (3, 2)$:

$$\begin{aligned} b_1' &= (1 \cdot 3 + 4 \cdot 2) \pmod 5 = 1 \\ b_2' &= (2 \cdot 3 + 3 \cdot 2) \pmod 5 = 2 \\ b_3' &= (4 \cdot 3 + 1 \cdot 2) \pmod 5 = 4 \end{aligned}$$

Comparando con los valores reales:

$$b = (0, 3, 4)$$

Vemos que hay diferencias, pero son tolerantes al error $\pm 1$ (considerando el ruido). Si el atacante prueba suficientes veces y busca patrones, podrá filtrar candidatos y concluir que $s = (3, 2)$ es una opción altamente probable.

## 5.2. Problemas derivados de LWE

En los siguientes apartados, se explorarán los problemas criptográficos basados en anillos, como el RLWE y el MLWE, es decir, las variantes de LWE en módulo y anillo. Para su completa comprensión, estos problemas han sido implementados en el anexo 5 dentro del repositorio ≪Problemas Relacionados≫. Para desarrollar estos apartados se ha estudiado el documento *Module-LWE versus Ring-LWE, Revisited* de Yang Wang y Mingqiang Wang [43], además de documentos sobre RLWE como *El problema del anillo LWE en la criptografía moderna* [44] y *On ideal lattices and learning with errors over rings* [45].

### 5.2.1. Problema RLWE

El problema de ≪Ring Learning with Errors≫ (RLWE) [18] (sección B.2. Criptografía Basada en Retículos) es una variante del problema LWE que incorpora como estructuras algebraicas adicionales los anillos de polinomios explicados anteriormente.

Algunas similitudes que comparte con el problema LWE son la base teórica basada en la dificultad de resolver problemas en retículos y que ambos cuentan con formulaciones en dos versiones: una de búsqueda, que consiste en encontrar una incógnita dada cierta información, y otra de decisión, cuyo objetivo es determinar si un conjunto de datos sigue una distribución específica o es aleatorio. Sin embargo, existen diferencias clave entre ambos problemas. La más clara es que, mientras que LWE opera en retículos sin incorporar una estructura algebraica adicional, RLWE introduce una estructura de anillo que trabaja con polinomios en un anillo cociente. Esto le permite aprovechar propiedades algebraicas adicionales que no están presentes en LWE.

Otra diferencia importante se encuentra en la eficiencia y el tamaño de las claves. Gracias a su estructura de anillo, RLWE permite representaciones más compactas y operaciones más eficientes en comparación con LWE. Por ejemplo, para un nivel de seguridad de 128 bits, un esquema basado en RLWE utilizaría claves públicas de aproximadamente 7.000 bits, mientras que un esquema basado en LWE requeriría claves de hasta 49 millones de bits para garantizar el mismo nivel de seguridad.

En términos de complejidad, RLWE puede considerarse como un caso particular de LWE, pero con una estructura adicional. Si bien cualquier algoritmo que resuelva LWE puede aplicarse a RLWE, esta variante ofrece ventajas prácticas significativas debido a su estructura algebraica.

En resumen, aunque LWE y RLWE comparten fundamentos teóricos similares, la introducción de la estructura de anillo en RLWE permite lograr mayor eficiencia y reducir considerablemente el tamaño de las claves criptográficas, lo que lo convierte en una opción más práctica en el ámbito de la criptografía postcuántica.

**Versión de búsqueda:** Dado un conjunto de pares de polinomios $(a_i(x), b_i(x))$,

donde $a_i(x)$ es un polinomio conocido, es decir, un polinomio público, y $b_i(x) = a_i(x) \cdot s(x) + e_i(x)$ con $s(x)$ siendo el polinomio secreto que intentamos encontrar y $e_i(x)$ un polinomio de ruido pequeño y aleatorio.

**Versión de decisión:** Dado un conjunto de pares de polinomios $(a_i(x), b_i(x))$, determinar si $b_i(x)$ fue generado como $b_i(x) = a_i(x) \cdot s(x) + e_i(x)$ o si $b_i(x)$ es un polinomio aleatorio.

La dificultad del problema depende de parámetros como el anillo polinómico cociente, el grado $n$ de este anillo, el cuerpo finito $\mathbb{Z}_q$ en el que operan los polinomios y la magnitud del error que se introduce.

A continuación, una definición formal de la versión de búsqueda de RLWE, sean:

- $R = \mathbb{Z}[x]/(f(x))$: un anillo cociente definido por un polinomio irreducible $f(x)$.
- $p$: un número primo que actúa como módulo.
- $\chi$: una distribución de probabilidad sobre $R/pR$ que genera el ruido $e_i(x)$.
- $m$: el número de muestras (ecuaciones) generadas.

Se nos da un conjunto de ecuaciones tal que:

$$\begin{aligned} b_1(x) &\approx_\chi a_1(x) \cdot s(x) + e_1(x) \pmod{p}, \\ b_2(x) &\approx_\chi a_2(x) \cdot s(x) + e_2(x) \pmod{p}, \\ &\vdots \\ b_m(x) &\approx_\chi a_m(x) \cdot s(x) + e_m(x) \pmod{p}, \end{aligned}$$

Donde:

- $a_i(x) \in R/pR$, son polinomios públicos seleccionados uniformemente.
- $b_i(x) \in R/pR$, son los resultados observados.
- $s(x) \in R/pR$, es el polinomio secreto.
- $e_i(x) \in R/pR$, es el polinomio de ruido generado por $\chi$.

El objetivo es recuperar $s(x)$ a partir del conjunto de ecuaciones $(a_i(x), b_i(x))$.

Análogamente al problema en el que se basa, para resolver RLWE mediante un ataque de fuerza bruta, necesitaríamos probar todas las posibles soluciones para $s(x)$ en el espacio $R/qR$, donde $s(x)$ es un polinomio con coeficientes en $\mathbb{Z}_p$ y grado menor que $n$; siendo $n$ el número de dimensiones o tamaño del problema. Esto implica que hay $p^n$ posibles valores para $s(x)$, y por cada posible $s(x)$, deberemos verificar si satisface las ecuaciones anteriormente mencionadas.

### 5.2.2. Problema MLWE

El problema de ≪Module Learning with Errors≫ (MLWE) es una extensión del problema LWE que, en lugar de trabajar con retículos, este se define en módulos sobre anillos enteros, lo que da una estructura matemática más compleja. Una consecuencia directa de esto es que cualquier instancia de LWE puede ser vista como un caso de MLWE, pero no todas las instancias de MLWE son reducibles a LWE.

Algunas similitudes son que comparten la misma base conceptual y que ambos problemas implican descifrar una transformación matemática corrompida por ruido. Sin embargo, existen muchas más diferencias. La primera de ellas, como ya se ha comentado, es la primitiva matemática en la que se basan. Que MLWE se base en módulos aporta una mayor capacidad para modelar sistemas algebraicos más complejos. Esta flexibilidad abre la puerta a aplicaciones criptográficas más sofisticadas, ya que los módulos pueden estar definidos sobre diferentes tipos de anillos (por ejemplo, anillos de enteros, anillos de polinomios, o anillos matriciales), lo que les permite adaptarse a una variedad de contextos y necesidades criptográficas. Por ejemplo, en aplicaciones de criptografía basada en retículos, esta flexibilidad puede ser aprovechada para construir sistemas más resistentes a ciertos ataques, o para diseñar algoritmos criptográficos más eficientes que aprovechen la estructura específica del anillo sobre el cual se definen los módulos. No obstante, esta mayor flexibilidad también implica una mayor complejidad, tanto en la teoría como en la práctica computacional.

En cuanto a sus aplicaciones en criptografía, el problema MLWE debido a su compleja estructura algebraica se emplea en contextos avanzados que requieren soluciones más robustas y eficientes tales como: la criptografía homomórfica, donde se necesita realizar operaciones sobre datos cifrados sin descifrarlos y protocolos de conocimiento cero (ZKP) que permiten verificar la validez de una afirmación sin revelar la información subyacente.

El problema se define de la manera análoga a RLWE, pero en lugar de considerar la estructura de anillo, se considera la de módulo:

Dado un módulo $M$ sobre un anillo $R$, un vector secreto $s \in M$ y un conjunto de vectores $A_1, A_2, \ldots, A_k \in M$, el problema consiste en resolver las ecuaciones de la forma:

$$A_i \cdot s + e_i = b_i \quad \text{para } 1 \leq i \leq k$$

Donde $A_i$ son los vectores públicos, $s$ es el vector secreto que se desea recuperar, $e_i$ es un pequeño error aleatorio en el módulo $M$ y $b_i$ es el valor observado.

El desafío en MLWE radica en que los errores $e_i$ son pequeños y aleatorios, lo que hace difícil recuperar $s$ de manera exacta, incluso cuando se dispone de múltiples ecuaciones.

Como conclusión a los problemas derivados de LWE cabe decir que, aunque el problema RLWE es más eficiente que el LWE gracias a la estructura de anillo

subyacente, esta misma estructura lo hace más susceptible a ciertos ataques. De manera similar, el problema MLWE, al estar definido sobre una estructura de módulo, es más compacto que el LWE, pero menos que el RLWE. Por esta razón, se considera que es menos vulnerable que el RLWE, aunque más propenso a ataques que el LWE [18].

## 5.3. Análisis empírico de la complejidad

A continuación, mediante una implementación en Java de los algoritmos del SVP, CVP y LWE, realizaremos un estudio de la complejidad de los mismos, para observar, mediante una aproximación razonable, cuánto tiempo tardaría un ordenador convencional en realizar un criptoanálisis mediante un ataque a fuerza bruta. Este estudio es posible gracias al repositorio de don Miguel Toro Bonilla, profesor de análisis y estructura de datos y algoritmos, ya que se ha utilizado su repositorio [46] para el análisis de la complejidad automatizado.

Como por el momento no contamos con ordenadores cuánticos, los tiempos que se mostrarán a continuación han sido realizados por un computador trabajando en monohilo, es decir, sin aprovechar la programación en paralelo, la cual puede condicionar la medición del tiempo de los algoritmos de diferentes formas.

Antes de conocer los hiperparámetros utilizados, considérese la siguiente anotación: el parámetro llamado ≪Rango del retículo = [-6,6] ≫ para los algoritmos SVP y CVP implica que las combinaciones lineales que forman el retículo estarán conformadas por los vectores base $B$ y los escalares $a_i$ que varían en el intervalo de $[-6, 6]$. Es decir, los coeficientes de las combinaciones lineales de los vectores base $b_i$ estarán restringidos a valores enteros dentro de este rango. De esta forma, se define un conjunto de posibles combinaciones lineales en el retículo generando un espacio de búsqueda determinado. Por último, la diferencia en el rango utilizado para los algoritmos SVP y CVP con LWE se explicará más adelante.

Como hiperparámetros, consideraremos los siguientes:

1. Rango del retículo: $\{-6, 6\}$, SVP, CVP, $[0, 13)$, y LWE $\Rightarrow |[-6, 6]| = |[0, 12]|$
2. Número de iteraciones por dimensión = 20
3. Desviación estándar (LWE) = 1
4. Número de Samples (LWE) = 10

El hecho de que ambos rangos no sean iguales en valores (si en cantidad de elementos), se debe a que al tomar módulos en LWE, el espacio de búsqueda del retículo se reduce a la mitad en el caso de tomar números negativos. Véase el ejemplo: si tomamos un rango de $[-6, 6]$ y un punto al azar dentro de este rango, por ejemplo: $(-3, -3, -1, -5)$, este se convertirá en $(3, 3, 5, 1)$, por lo que realmente el espacio de búsqueda está comprendido por el rango $[0, 6]$, reduciéndose de este modo a la mitad.

Se realizarán 20 ejecuciones de cada algoritmo por dimensión y su promedio será el tiempo asignado a la dimensión. La desviación estándar será de 1 unidad, pues al trabajar con módulos tan bajos, un error mayor generaría un sistema incompatible. Por último, generaremos 10 samples para asegurarnos de que el sistema pueda encontrar la correlación entre los parámetros y por tanto una solución.

A continuación, en las tablas 5.1, 5.2 y 5.3, se presentan los tiempos necesarios para llevar a cabo un criptoanálisis exitoso mediante un ataque de fuerza bruta en cada uno de los tres algoritmos.

Cabe destacar que no se ha podido realizar una estimación en computadores cuánticos, dado que para este tipo de problemas, no es aplicable el algoritmo de shor o el de grover (por eso precisamente se han elegido). Además, actualmente no se dispone de ordenadores cuánticos capaces de ejecutar algoritmos de resolución relevantes para este contexto, y la implementación detallada de una simulación cuántica excede los objetivos planteados en este trabajo.

Estos tiempos representan estimaciones basadas en la mejor comprensión actual de la complejidad teórica y la velocidad de procesamiento de los computadores actuales. Sin embargo, los valores pueden variar dependiendo de avances en hardware y optimización de algoritmos. Nota: (C) significa calculado, y (E) significa estimado.

| **Dimensión** | **Tiempo promedio de resolución exitosa** | **Complejidad** |
|---|---|---|
| 1 (C) | $5{,}18 \times 10^{-6}$ s | $T = O(c^n), c > 1$ |
| 2 (C) | $3{,}17 \times 10^{-4}$ s | $P = 5,88 \times 232,32^n$ |
| 3 (C) | 0,073 s | |
| 4 (C) | 17,12 s | |
| 5 (E) | 1,10 h | |
| 32 (E) | $9{,}67 \times 10^{57}$ siglos | |
| 128 (E) | $1{,}35 \times 10^{278}$ eones | |
| 256 (E) | Infinito | |

**Tabla 5.1:** Tiempos de resolución del algoritmo SVP

| Dimensión | Tiempo promedio de resolución exitosa | Complejidad |
|---|---|---|
| 1 (C) | $1{,}08 \times 10^{-6}$ s | $T = O(c^n), c > 1$ |
| 2 (C) | $4{,}16 \times 10^{-6}$ s | $P = 643,67 \times 8,87^n$ |
| 3 (C) | $9{,}40 \times 10^{-5}$ s | |
| 4 (C) | $1{,}05 \times 10^{-3}$ s | |
| 5 (C) | 0,036 s | |
| 6 (C) | 0,31 s | |
| 32 (E) | $4{,}4 \times 10^{16}$ años | |
| 128 (E) | $4{,}41 \times 10^{106}$ siglos | |
| 256 (E) | $9{,}52 \times 10^{219}$ eones | |

**Tabla 5.2:** Tiempos de resolución del algoritmo CVP

| Dimensión | Tiempo promedio de resolución exitosa | Complejidad |
|---|---|---|
| 1 (C) | $4{,}90 \times 10^{-7}$ s | $T = O(q^n), q > 1$ |
| 2 (C) | $2{,}2 \times 10^{-6}$ s | $P = 372,1 \times 6,95^n$ |
| 3 (C) | $2{,}94 \times 10^{-5}$ s | |
| 4 (C) | $3{,}12 \times 10^{-4}$ s | |
| 5 (C) | $6{,}21 \times 10^{-3}$ s | |
| 6 (C) | 0,042 s | |
| 32 (E) | $1{,}03 \times 10^{13}$ años | |
| 128 (E) | $7{,}00 \times 10^{91}$ siglos | |
| 256 (E) | $4{,}17 \times 10^{192}$ eones | |

**Tabla 5.3:** Tiempos de resolución del algoritmo LWE

A continuación, una comparativa gráfica de los tres algoritmos en las dimensiones calculadas de la tabla anterior. Cabe destacar que, en el caso del SVP, al tener una combinatoria tan explosiva, no se ha podido calcular en dimensiones mayores a tres.

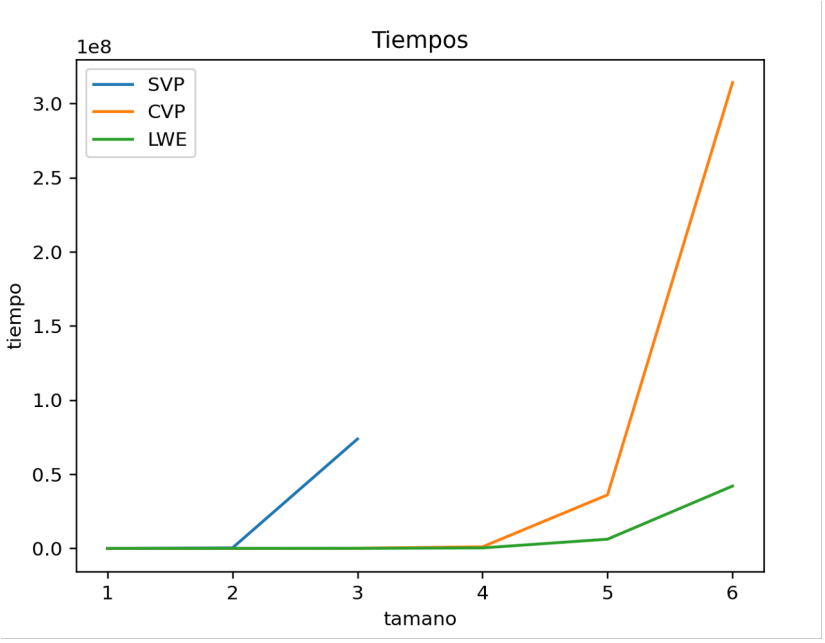


**Figura 5.5:** Gráfica de tiempos de los problemas CVP, SVP y LWE.

Como conclusión de este estudio podemos dilucidar que, en un principio, el SVP cuenta con una mayor explosividad combinatoria, pues debe realizar un mayor número de operaciones en el retículo, pero, aunque esto sea cierto, en un mayor número de dimensiones, el resultado podría invertirse, pues en el CVP, al poder elegir un punto fuera del retículo, estaríamos añadiendo una dimensión adicional, lo cual en altas dimensiones le conferiría un tamaño superior al SVP.

Por otra parte, LWE tiene la ventaja de contar con una distribución de errores, que en altas dimensiones y con módulos más altos, sin lugar a duda hace de este problema una barrera inexpugnable. Finalmente, hay que comentar que, dejando a un lado la rivalidad entre estos algoritmos, tienen una complejidad similar de orden exponencial y, por tanto, resultan intratables a criptoanálisis mediante fuerza bruta.

En el caso de querer abordar este problema mediante otros enfoques distintos a fuerza bruta, hay que destacar que, actualmente no existe ningún algoritmo cuántico que sea capaz de resolverlo en tiempo polinomial de forma exacta, en otras palabras, al no depender su seguridad de factorización de enteros en primos, o resolver el problema del logaritmo discreto, el algoritmo de Shor no es capaz de reducir su complejidad, y por tanto permanece seguro. Y así será hasta que nazca un nuevo algoritmo cuántico capaz de resolver el problema de decodificación de números aleatorios.

# 6. Kyber–KEM

## 6.1. La importancia de CRYSTALS–Kyber

Para que un criptosistema funcione de forma segura, tenemos que asegurarnos de que el problema que subyace en el algoritmo sea intratable para la potencia computacional existente, debido al nuevo paradigma cuántico ya mencionado, es necesario un nuevo estándar que sea capaz de plantar cara a esta amenaza con un nuevo problema intratable tanto para ordenadores cuánticos como convencionales.

Este protocolo KEM basado en retículos surge como respuesta por parte de la iniciativa CRYSTALS (Cryptographic Suite for Alegbraic Lattices) y ha ganado notoriedad, pues de los 81 candidatos iniciales, ha sido el elegido para ser estandarizado por el NIST, siendo el 13 de agosto de 2024 la publicación del estándar FIPS 203: ≪Module–Lattice–Based Key–Encapsulation Mechanism Standard≫. CRYSTALS–Kyber, propuso una modificación con relación a otras propuestas basadas en el problema RLWE, en concreto, el uso del problema MLWE. En resumen, se trabaja en $R_q^k$ en lugar de en $R_q$, siendo $k = 2, 3, 4$ y la dimensión $n$ de $R_q$ es fija e igual a 256. En la versión de la Ronda 1 de Kyber, el módulo era $q = 7681$, pero en la versión de la Ronda 2, los autores lograron reducir el módulo a 3329, consiguiendo una aceleración de la transformada teórica de números (NTT).

El uso del problema LWE en Kyber–KEM se manifiesta de manera práctica en tres pasos fundamentales dentro del proceso criptográfico:

- **Generación de claves:** Kyber crea una clave pública y una clave privada empleando distribuciones de ruido gaussianas, características del problema LWE, sobre un anillo polinomial. Esto asegura que las claves sean extremadamente difíciles de descifrar, pero simples de generar y manejar.
- **Encapsulación de claves:** Cuando un usuario necesita enviar una clave cifrada de manera segura, Kyber usa LWE para añadir un nivel controlado de ruido, protegiendo la clave compartida contra posibles ataques, incluso en un entorno con computadoras cuánticas.
- **Desencapsulación de claves:** El receptor, con su clave privada, elimina el ruido añadido y recupera la clave original, aprovechando las propiedades matemáticas de LWE en retículos para garantizar la precisión y la seguridad.

Gracias a estas características, CRYSTALS–KEM garantiza seguridad criptográfica sólida en el proceso de encapsulación y desencapsulado de claves, aprovechando la robustez matemática del problema MLWE. Además, mediante la transformación Fujisaki–Okamoto (FO), cuenta con una seguridad IND–CCA2, que está soportada por una prueba de seguridad en el modelo del oráculo aleatorio cuántico (QROM).

El esquema ML–KEM (Module–Lattice–KEM), conocido formalmente como

Kyber–KEM, surge como un renombramiento de CRYSTALS–Kyber, cuyo nombre hace referencia a la famosa saga de ficción *La guerra de las galaxias*, en concreto a los cristales de luz que portan los sables láser (cristales kyber), lo que no es apropiado para un estándar gubernamental, por lo que le fue asignada una denominación más técnica y descriptiva.

En el siguiente subapartado, presentaremos todos los mecanismos de Kyber–KEM, en primer lugar, los algoritmos relativos al PKE, y posteriormente los propios del KEM. Además, estos han sido implementados en los anexos 6 y 7, dentro del repositorio «Algoritmos Kyber–KEM», para que se pueda ver paso a paso su funcionamiento. Pero antes de detallar los mecanismos de Kyber–KEM, es útil revisar las notaciones y parámetros empleados en los esquemas. En la tabla 6.1, se describen los principales elementos matemáticos utilizados:

| **Elemento** | **Descripción** |
|---|---|
| $k$ | Número de dimensiones del problema (dimensión del secreto). |
| $d$ | Número de muestras (número de ecuaciones del sistema). |
| $seed\_A$ | Semilla aleatoria utilizada para generar la matriz pública A de manera determinista. |
| $A$ | Matriz pública de tamaño $d \cdot k$ con elementos en $R_q$, generada mediante el proceso de muestreo PSR. |
| $s$ | Vector secreto de longitud $k$, generado muestreando cada elemento de $R_q$ según la distribución $\beta(\mu_1)$, que introduce aleatoriedad y ruido controlado. |
| $e$ | Vector de error de longitud $d$, generado mediante la distribución $\beta(\mu_1)$, esencial para la seguridad del esquema LWE. |
| $b$ | Clave pública, vector de longitud $d$ calculado como $b = As + e$, dificultando la recuperación de $s$ sin conocer $e$ de forma exacta. |
| $pk$ | Clave pública, formada por la concatenación de $b$ y $seed\_A$. |
| $sk$ | Clave secreta, compuesta únicamente por el vector secreto $s$. |
| $A^T$ | Matriz transpuesta de $A$, utilizada en el cifrado. |
| $r$ | Vector aleatorio de longitud $k$, generado con la distribución $\beta(\mu_1)$ para introducir ruido en el cifrado. |
| $e_1$ | Vector de error de longitud $k$, generado mediante la distribución $\beta(\mu_2)$. |
| $e_2$ | Escalar de error, generado con la distribución $\beta(\mu_2)$, usado en el cifrado. |
| $u$ | Primer componente del mensaje cifrado, calculado como $u = A^T r + e_1$. |
| $v$ | Segundo componente del mensaje cifrado, obtenido como $v = b^T r + e_2 + m$. |
| $c$ | Mensaje cifrado, formado por la concatenación de $u$ y $v$. |

| Elemento | Descripción |
|---|---|
| $m$ | Mensaje original, recuperado en el descifrado mediante $m = v - s^T u$. |
| $z$ | Vector aleatorio de 256 bits, usado en la generación de claves KEM para mejorar la seguridad. |
| $H$ | Función hash utilizada en la generación de claves y encapsulado (SHA3–256). |
| $G$ | Función hash utilizada en el encapsulado (SHA3–512). |
| $KDF$ | Función de derivación de claves (SHAKE–256), usada para obtener la clave final compartida. |
| $\bar{K}$ | Clave intermedia generada en el encapsulado, y luego procesada con el KDF para derivar la clave compartida final. |
| $h$ | Valor derivado de la clave secreta $sk$, usado en el desencapsulado para verificar la autenticidad del cifrado. |

**Tabla 6.1:** Elementos presentes en Kyber-KEM

Cabe mencionar, con relación al PKE, que los parámetros $e_1$ y $e$ definen los errores y ambos tienen el mismo valor para los conjuntos de parámetros de Kyber–768 y Kyber–1024, pero diferentes para Kyber–512 (variantes del esquema Kyber en función del nivel de seguridad que ofrecen según el tamaño de las claves en bits).

La representación esquematizada de los algoritmos se ha obtenido del documento del CCN, CCN–STIC 221 [18]. Aunque también se ha usado como referencia el documento FIPS 203 del NIST [47].

## 6.2. Algoritmos relativos al PKE

### 6.2.1. Generación de claves $\mathcal{G}'$

Primeramente, se genera una semilla aleatoria que se utiliza para inicializar y generar la matriz pública $A$ de manera determinista. La aleatoriedad de la semilla garantiza que cada ejecución produzca claves diferentes, pero una misma semilla produciría siempre la misma matriz.

En el segundo paso se define $A$ como una matriz pública de tamaño $k \cdot k$ cuyos elementos son tomados de un conjunto de enteros módulo $q$ (del conjunto $R_q$). PSR es un proceso de muestreo que genera valores aleatorios para $A$.

En el tercer y cuarto pasos se generan los vectores $s$ y $e$ (en LWE, este muestreo de ruido es esencial para que el sistema sea intratable). El vector secreto $s$ es un vector aleatorio de $k$ elementos, cuyos valores provienen de una distribución $\beta_{\mu_1}$. Esta distribución controla la cantidad de ruido o error en el sistema. De manera

**Algorithm 1** Generación de Claves $G'$

1: **Genera** $seed\_A$

2: $A \leftarrow \text{PSR}(R_q^{k \times k})$

3: $s \leftarrow \beta_{\mu_1}(R_q^k)$

4: $e \leftarrow \beta_{\mu_1}(R_q^k)$

5: $b \leftarrow As + e$

6: $pk \leftarrow (b \parallel seed_A)$

7: $sk \leftarrow s$

8: **Devuelve** $(pk, sk)$

similar, el vector $e$ también se genera a partir de la misma distribución. El vector $e$ representa el error que se agrega a la multiplicación de $A$ y $s$ para asegurar la seguridad del sistema.

Seguidamente, se calcula la clave pública $b$ como el resultado de multiplicar la matriz $A$ por el vector secreto $s$ y luego agregar el vector de error $e$ (nótese que $b$ es un ejemplo de instancia/muestra LWE). Esto crea un error en el producto para hacer que sea difícil de resolver para un atacante que conoce la matriz $A$, el vector $b$ y que $e$ se rige por una distribución gaussiana. Con todo esto, $s$ sería computacionalmente difícil de recuperar debido a la aleatoriedad que produce el error $e$ en el sistema de ecuaciones.

En los pasos sexto y séptimo se construyen las claves pública y privada. En primer lugar, la clave pública, $pk$, se forma concatenando el vector $b$ con la semilla $seed\_A$. Esto asegura que cualquier persona que conozca $pk$ pueda verificar la autenticidad de $b$ y generar su propia clave compartida, pero no pueda conseguir fácilmente $s$. Por otro lado, la clave privada, $sk$, es simplemente el vector secreto $s$.

Por último, el algoritmo devuelve la clave pública $pk$ y la clave privada $sk$.

En suma, el núcleo de este esquema está en la ecuación $b = As + e$. Su dureza, es decir, la imposibilidad de resolverla conociendo solo $(A, b)$, es la que fundamenta la seguridad.

### 6.2.2. Cifrado $\mathcal{E}(pk, m, r)$

**Algorithm 2** Cifrado $\mathcal{E}(pk, m, r)$

1: **Genera** $seed\_A$

2: $A^T \leftarrow \text{PSR}(R_q^{k \times k})$

3: $r \leftarrow \beta_{\mu_1}(R_q^k)$

4: $e_1 \leftarrow \beta_{\mu_2}(R_q^k)$

5: $e_2 \leftarrow \beta_{\mu_2}(R_q)$

6: $u \leftarrow A^T r + e_1$

7: $v \leftarrow b^T r + e_2 + m$

8: $c \leftarrow (u \parallel v)$

9: **Devuelve** $c$

El primer paso consiste en generar la semilla $seed\_A$ utilizada previamente en el algoritmo de generación de claves (la obtenemos gracias a $pk$), para, a continuación, generar la matriz $A^T$, siendo esto la matriz transpuesta de la matriz pública $A$, generada también en el proceso anterior.

Seguidamente generamos un vector $r$ aleatorio de longitud $k$ cuyos valores se extraen de una distribución $\beta_{\mu_1}$ para generar ruido en el cifrado.

En los pasos cuarto y quinto generamos los vectores de error $e_1$ y $e_2$. El vector $e_1$ es de longitud $k$ y se genera de acuerdo con la distribución $\beta_{\mu_2}$, que controla el error o ruido en la multiplicación de matrices durante el proceso de cifrado. Por otro lado, el vector $e_2$ es un vector de longitud 1, generado también a partir de la misma distribución. Este se utilizará en el cálculo del componente de la clave en el mensaje cifrado.

En el siguiente paso calculamos $u$ y $v$ (obsérvese su semejanza con la construcción LWE). El primer componente del mensaje cifrado, $u$, se calcula multiplicando la matriz $A^T$ por el vector $r$ y luego sumando el vector de error $e_1$. Esto asegura que $u$ dependa de la clave pública y del ruido.

En cambio, $v$, el segundo componente del mensaje cifrado, se calcula tomando el producto $b$ (la clave pública) con el vector $r$, sumando el error $e_2$ y finalmente añadiendo el mensaje $m$. Esto garantiza que $v$ contenga la información del mensaje cifrado de manera que solo la persona con la clave privada pueda descifrarlo. Cabe destacar que esta expresión es, en esencia, otra instancia LWE, pues incorporamos el mensaje $m$ a la parte de ruido.

Por último, concatenamos $u$ y $v$ como el mensaje cifrado final $c$, el cual devuelve el algoritmo. Debemos fijarnos en que, si alguien no conoce $s$, no podrá aislar el mensaje $m$ a partir de $u$, $v$ sin resolver un problema MLWE.

### 6.2.3. Descifrado $\mathcal{D}(sk, c)$

**Algorithm 3** Descifrado $\mathcal{D}(sk, c)$

1: $m \leftarrow v - s^T u$

2: **Devuelve** $m$

El propósito del algoritmo de descifrado es recuperar el mensaje original a partir del mensaje cifrado $c$ y la clave secreta $sk$. La clave secreta $sk$ es el vector secreto $s$ que el receptor mantiene de forma privada. Y el mensaje cifrado $c$, como hemos visto en el esquema anterior, es el resultado del algoritmo de cifrado y consiste en la concatenación de los componentes $u$ y $v$.

En primer lugar, debemos descomponer el mensaje cifrado $c$ obteniendo $u$ y $v$, necesarios para obtener $m$ aplicando la ecuación indicada en el paso primero y usando la clave secreta, obtener $m$. A continuación, podemos verificar que esto es cierto sustituyendo $v$ y $u$ en la ecuación (6.1):

$$m = v - s^T u \tag{6.1}$$

Sustituyendo los valores de $v$ y $u$ obtenemos:

$$m = (b^T r + e_2 + m) - s^T (A^T r + e_1)$$

Expandiendo los términos:

$$m = b^T r + e_2 + m - s^T A^T r - s^T e_1$$

Recordemos que $b$ está relacionado con $A$ y $s$ de manera que $b = As + e$, por tanto, $b^T r$ se puede escribir como:

$$b^T r = (As + e)^T r = s^T A^T + e^T r$$

Entonces:

$$m = (s^T A^T + e^T r) + e_2 + m - s^T A^T r - s^T e_1$$

Observamos que $s^T A^T r$ aparece con signo positivo y negativo, por lo que se cancelan entre sí. Además, el término $e^T r$ es un error, pero no afecta al mensaje $m$ porque está relacionado con el error que introducimos durante el cifrado. Por lo que nos queda:

$$m = m + (e_2 - s^T e_1)$$

Dado que $e_2$ y $s^T e_1$ son errores que fueron introducidos en el cifrado, el valor final de $m$ será el mensaje original más un pequeño error relacionado con los vectores de error $e_1$ y $e_2$. Sin embargo, $m$ se puede recuperar casi con exactitud, ya que los errores permanecen acotados de modo que, al decodificar (por ejemplo, redondeando o modulando), se recupera el mensaje original.

## 6.3. Algoritmos relativos al KEM

A continuación, se muestran los algoritmos relacionados con el proceso de encapsulado KEM: Generación de claves, encapsulado y desencapsulado. Referenciados como algoritmos 4, 5 y 6 respectivamente.

Las funciones hash $G$, $H$ y la función de clave $KDF$ que se utilizan a continuación, son, en general, las siguientes: $H$ es SHA3–256, $G$ es SHA3–512 y $KDF$ es SHAKE–256.

### 6.3.1. Generación de claves $\mathcal{G}'$

**Algorithm 4** Generación de Claves G

1: $z \leftarrow \{0,1\}^{256}$

2: $(pk, sk') \leftarrow G'$

3: $sk \leftarrow (sk' \parallel pk \parallel H(pk) \parallel z)$

4: **Devuelve** $(pk, sk)$

Primeramente, generamos un vector $z$ de 256 bits aleatorios. Este valor se usa como «entropía» o «salto aleatorio», y es esencial para mantener la seguridad del sistema, ya que se utiliza en la clave secreta final $sk$. Se opta por usar 256 bits para que proporcione suficiente aleatoriedad para asegurar que la clave secreta sea impredecible.

En el siguiente paso llamamos al algoritmo de generación de claves $G'$ descrito anteriormente. Recibimos de este la clave pública $pk$ y la clave secreta intermedia generada, $sk'$, que solo se utiliza para obtener una nueva clave secreta final $sk$.

En el tercer paso, construimos la clave secreta final $sk$ concatenando varios componentes: La clave secreta intermedia $sk$; la clave pública, $pk$ y el resultado de aplicar una función hash, $H$, sobre la clave pública, $H(pk)$. El propósito de este último paso es incluir en la clave secreta un resumen de la clave pública, lo que mejora la seguridad, ya que, de esta manera se consigue que la clave secreta dependa de la pública, evitando ciertos tipos de ataques. Por último, concatenamos también el vector $z$ y se devuelven las claves $pk$ y $sk$.

### 6.3.2. Encapsulado $\mathcal{E}c(pk)$

**Algorithm 5** Encapsulado $\mathcal{E}c(pk)$

1: $m' \leftarrow \{0,1\}^{256}$

2: $m \leftarrow H(m')$

3: $(\bar{K}, r) \leftarrow (m \parallel H(pk))$

4: $c \leftarrow E(pk, m, r)$

5: $K \leftarrow \text{KDF}(\bar{K} \parallel H(c))$

6: **Devuelve** $(c, K)$

El primer paso del algoritmo de encapsulado consiste en generar un valor $m'$ de 256 bits aleatorios, para, en el siguiente paso, producir un valor $m$ al aplicarle al valor aleatorio $m'$ una función hash $H$. El propósito de este paso es generar una ≪clave derivada≫ a partir de los 256 bits, pero que tenga una forma que sea adecuada para el cifrado (es decir, una longitud estándar) la cual conseguimos gracias a la función hash $H$.

En el tercer paso construimos una clave $\bar{K}$ y un valor $r$ concatenando $m$ y el hash de la clave pública, $H(pk)$. El valor $\bar{K}$ es una clave compartida que se utiliza para la comunicación posterior. La concatenación de $m$ y $H(pk)$ se hace para garantizar que $\bar{K}$ dependa tanto del valor aleatorio inicial como de la clave pública del destinatario. Esto hace que el proceso de cifrado sea seguro y dependiente de la clave pública.

En el siguiente paso, realizamos el cifrado (Algoritmo 2) utilizando la clave pública y los valores $m$ y $r$ como parámetros de entrada. El resultado de este paso es el mensaje cifrado $c$.

Seguidamente, realizamos una función de derivación de claves (KDF, Key Derivation Function) para generar la clave compartida final $K$. Primero, se concatena la clave $\bar{K}$ generada en el paso 3 y el hash de $c$. La función de derivación de claves KDF toma esta concatenación y produce una clave derivada final $K$. La razón para usar KDF es asegurarse de que la clave derivada final $K$ tenga una longitud adecuada y sea segura para ser utilizada como una clave secreta compartida entre el emisor y el receptor. La clave $K$ será utilizada para la comunicación segura.

Finalmente, el algoritmo devuelve el mensaje cifrado $c$ y la clave compartida $K$.

### 6.3.3. Desencapsulado $\mathcal{D}c(sk, c)$

**Algorithm 6** Desencapsulado $D_c(sk, c)$

1: $h \leftarrow sk + 24 \cdot k \cdot n/8 + 32$

2: $z \leftarrow sk + 24 \cdot k \cdot n/8 + 64$

3: $\tilde{m} \leftarrow D(sk, c)$

4: $(\bar{K}', r') \leftarrow G(\tilde{m} \parallel h)$

5: $c' \leftarrow E(pk, \tilde{m}, r')$

6: **if** $c = c'$ **then**

7: **Devuelve** $K = \text{KDF}(\bar{K}' \parallel H(c))$

8: **else**

9: **Devuelve** $K = \text{KDF}(z \parallel H(c))$

10: **end if**

El objetivo del desencapsulado es que el receptor recupere la clave compartida $K$ a partir de un mensaje cifrado $c$. Para ello usa la clave secreta $sk$ y realiza los siguientes pasos de verificación para asegurarse de que no ha habido manipulaciones en el mensaje cifrado.

El receptor comienza calculando un valor $h$ a partir de su clave secreta $sk$. Este valor $h$ se deriva utilizando parámetros específicos como $k$ y $n$ (tamaño de la matriz y número de dimensiones), además de una constante $+32$. Este valor lo usaremos en el proceso de verificación del mensaje cifrado.

Luego, se calcula el segundo valor, $z$, que es también una combinación de la clave secreta $sk$ y los parámetros $k$ y $n$, pero con la constante $+64$. Este valor, $z$, es un valor intermedio que se utilizará en caso de que el mensaje cifrado haya sido alterado.

A continuación, el receptor utiliza su clave secreta $sk$ para descifrar el mensaje $c$ utilizando el algoritmo de descifrado $D(sk, c)$ (Algoritmo 3). Obtenemos el mensaje $\tilde{m}$.

Con el mensaje recuperado $\tilde{m}$ y el valor $h$ calculado en el primer paso, el receptor genera una clave compartida $\bar{K}$ y un valor $r'$ mediante la función hash $G(\tilde{m} \parallel h)$. El valor generado $\bar{K}$ es una clave compartida generada a partir de $\tilde{m}$ y $h$, mientras que $r'$ es un valor adicional usado en el siguiente paso para garantizar la integridad. Además, la función hash $G$ se asegura de que el proceso sea seguro y único (dada su

naturaleza de función resumen), $\bar{K}$ y $r'$ serán variables deterministas pero difíciles de predecir sin conocer $\tilde{m}$ y $h$.

Después de obtener $\bar{K}$ y $r'$, el receptor genera un nuevo mensaje cifrado $c'$ utilizando la función de cifrado $E(pk, \tilde{m}, r')$.

El receptor compara el mensaje cifrado $c'$ con el mensaje cifrado $c$ que recibió. Si $c' = c$, significa que el proceso fue correcto y que $\bar{K}$ es la clave compartida correcta. A continuación, genera la clave final, $K$, mediante la función de derivación de claves KDF concatenando la clave derivada $\bar{K}$ con el valor hash $H(c)$, que ayuda a asegurar la integridad y autenticidad del mensaje cifrado.

Si la verificación falla, el receptor utiliza el valor alternativo $z$ para generar la clave $K$ de forma diferente, siendo esta totalmente inútil para un atacante.

## 6.4. Seguridad de Kyber–KEM basada en LWE

La seguridad de Kyber–KEM (y, por extensión, de su variante PKE) radica en la suposición de que no se puede resolver el problema LWE (o sus variantes en anillo y módulo) en un tiempo factible para los parámetros escogidos. Así:

- Si un atacante posee $(A, b)$ con $b = As + e$, no puede deducir $s$ (ni el error $e$) de manera eficiente, pues se encontraría un problema de corrección de errores en retículos, que es extremadamente difícil.
- En la fase de cifrado, los valores $(u, v)$ involucran nuevas instancias de LWE ($u = A^T r + e_1, v = b^T r + e_2 + m$). Sin $s$, el atacante no puede obtener el mensaje $m$.
- La transformación Fujisaki–Okamoto (FO) confiere, sobre esta base, seguridad IND–CCA2 en el modelo del oráculo aleatorio cuántico (QROM), garantizando que incluso ataques de texto en claro elegido, donde el atacante puede solicitar descifrados de textos a su elección, no comprometen la clave. Estos ataques y conceptos se explicarán más detalladamente en el siguiente apartado.

Veamos a continuación por qué la resolución de LWE es intratable:

Para descifrar el mensaje $m$, el atacante debe recuperar $s$. Esto significa resolver el sistema:

$$b = As + e \mod q \tag{6.2}$$

Sin el conocimiento exacto de $e$, el problema se vuelve difícil porque la introducción de ruido al sistema evita que el sistema pueda resolverse directamente con álgebra lineal clásica, y las posibles combinaciones de parámetros abruman cualquier ataque conocido.

Las estrategias de ataque incluyen:

- **Eliminación del Ruido con Algoritmos Clásicos:** Si ignoramos el ruido y tratamos de resolver $b = As$, podríamos usar una descomposición Gaussiana

o eliminación de Gauss; sin embargo, estas no funcionan porque el ruido $e$ lo hace demasiado impreciso.

- **Algoritmos de Reducción de Retículos:** Métodos de reducción de base en retículos, como LLL o BKZ son efectivos para dimensiones pequeñas, pero en Kyber $n = 256$ el ataque requeriría recursos computacionales exponenciales. Estos ataques son actualmente los más fuertes, sin embargo, se estima que resolver Kyber–512 (la versión más débil de Kyber) requiere más operaciones que un ataque de fuerza bruta al AES–128.

En definitiva, cada uno de estos algoritmos PKE (generación de claves, cifrado y descifrado) está construido de forma que la tarea de romper el sistema equivalga a resolver instancias LWE; y es precisamente la dureza de LWE la que protege la confidencialidad de las claves y mensajes en Kyber–KEM.

En el anexo 8 del repositorio Criptoanálisis se han implementado diferentes simulaciones de ataques que podría realizar alguien que pretendiese romper la seguridad de LWE. Otros tipos de ataques a los que pueden verse sometidos estos algoritmos por parte de atacantes se comentan en el siguiente apartado.

# 7. Seguridad

En este apartado se analizarán los ataques clásicos y cuánticos a la seguridad basada en retículos, así como los ataques de canal lateral y los ataques de descifrado de texto en claro conocido.

## 7.1. Ataques clásicos y cuánticos a la seguridad basada en retículos

A continuación, se abordan las principales amenazas a la seguridad basada en retículos, incluyendo técnicas clásicas y cuánticas. Se explorarán métodos como la reducción de retículos mediante los algoritmos LLL y BKZ, los ataques de decodificación de retículos y el impacto de los algoritmos cuánticos en la criptografía basada en estas estructuras.

Para la elaboración de este apartado se han utilizado referencias fundamentales como el artículo original del algoritmo LLL de Lenstra, Lenstra y Lovász [48], el trabajo de Schnorr y Euchner [49] sobre mejoras prácticas, y la revisión de Micciancio y Regev sobre criptografía basada en retículos [50]. Además, se ha consultado el estudio de Gama y Nguyen sobre el algoritmo BKZ [51], fundamental para entender su papel en ataques criptográficos. Para el subapartado de ataques de canal lateral se han visitado los artículos *Countermeasures for Side-Channel Attacks in Cryptographic Systems* [52] y *Quantum Attacks on Lattice-Based Cryptography: The Side-Channel Perspective* [53]. Por último, también se ha usado el documento *Security against Chosen Ciphertext Attacks: Theory and Practice* [54].

### 7.1.1. Reducción de retículos: LLL y BKZ

Existen algoritmos de reducción de bases, como el algoritmo LLL (Lenstra–Lenstra–Lovász), que permiten simplificar la base de un retículo sin alterar su estructura. Este tipo de algoritmo transforma el conjunto de vectores generadores del retículo en una nueva base, en la que los vectores son más pequeños y ortogonales entre sí.

El algoritmo utiliza el proceso de Gram–Schmidt para ortogonalizar los vectores, luego ajusta las longitudes de los vectores para que sean lo más pequeñas posible y elimina redundancias. Además, se basa en condiciones matemáticas para determinar si la base está suficientemente reducida, y en algunos casos intercambia vectores para optimizar la representación del retículo.

La base resultante del algoritmo tiene vectores de longitudes más pequeñas y con ángulos lo suficientemente pequeños entre ellos, lo que facilita la resolución de problemas como el Problema del Vector Más Corto (SVP). Aunque LLL no

encuentra la solución óptima, su complejidad $O(n^3 \log^3 B)$, (siendo $B$ el mayor valor absoluto de las coordenadas de los vectores de entrada) lo hace eficiente en la práctica, especialmente en dimensiones moderadas; por el contrario, los retículos de alta dimensión siguen siendo seguros frente a estas técnicas porque la reducción obtenida no es lo suficientemente fuerte para encontrar los vectores más cortos con la precisión requerida.

Para abordar instancias más difíciles, se emplean algoritmos más potentes como BKZ (Block Korkine-Zolotarev), una extensión de LLL que introduce bloques de mayor tamaño en la reducción de la base. BKZ mejora significativamente la reducción en comparación con LLL, ya que dentro de cada bloque resuelve instancias del problema del vector más corto (SVP), proporcionando vectores aún más pequeños y una base mejor optimizada. Sin embargo, este enfoque implica un costo computacional mucho mayor. La seguridad de muchos esquemas criptográficos basados en retículos se evalúa en función de la resistencia a ataques basados en BKZ, en definitiva, aunque es más efectivo que LLL, sigue siendo computacionalmente costoso en dimensiones suficientemente altas.

### 7.1.2. Ataques de decodificación de retículos

El objetivo principal de estos ataques es resolver problemas fundamentales en retículos, como el Problema del Vector Más Corto (SVP) y el Problema del Vector Más Cercano (CVP). Resolver estos problemas permite a un atacante extraer información sobre claves privadas en esquemas criptográficos basados en retículos. Para llevar a cabo estos ataques, se emplean los algoritmos de reducción de bases vistos previamente, LLL y BKZ.

La reducción de bases es una técnica fundamental para mejorar la eficiencia de los ataques de decodificación, ya que cuanto más corta y ortogonal sea la base del retículo, más fácil será resolver problemas como el SVP y el CVP.

Existen diferentes enfoques para los ataques de decodificación de redes. Uno de ellos es el ataque primal, que intenta resolver una instancia del SVP en la base pública del retículo con el fin de recuperar la clave secreta.

Otro enfoque es el ataque dual, que, en lugar de atacar directamente la base del retículo, busca combinaciones lineales de los vectores del retículo que permitan descartar posibles claves privadas. Además, algunos ataques combinan estrategias primal y dual para mejorar su efectividad.

Estos ataques representan una amenaza para la seguridad de los esquemas criptográficos basados en retículos, especialmente si se encuentran implementaciones con dimensiones insuficientemente altas o con parámetros mal configurados. Sin embargo, en la práctica, los sistemas diseñados con parámetros seguros ofrecen una resistencia significativa frente a estos métodos.

### 7.1.3. Algoritmos cuánticos

Aunque Kyber es resistente a ataques cuánticos como el algoritmo de Shor (que rompe RSA y ECC), aún podría ser vulnerable a mejoras en algoritmos cuánticos diseñados para atacar esquemas basados en retículos. Uno de ellos, el algoritmo de Grover, explicado anteriormente, podría acelerar ciertos ataques de fuerza bruta en esquemas criptográficos, aunque su impacto en Kyber es limitado, ya que los ataques conocidos requieren más que una simple búsqueda de clave.

Otra posible amenaza proviene de las mejoras en la reducción de retículos. Actualmente se han propuesto versiones cuánticas de estos algoritmos, como el BKZ acelerado cuánticamente, que podrían reducir el coste computacional de atacar estos sistemas [55]. Aunque estas mejoras aún no son prácticas, la investigación en esta área continúa, y en el futuro podrían representar una amenaza para la seguridad de Kyber si logran reducir significativamente la complejidad computacional de la reducción de redes.

Además, existen estudios sobre posibles enfoques cuánticos para resolver el problema de Learning With Errors (LWE), en el que se basa Kyber. Un ejemplo es el estudio *Quantum Algorithms for Lattice Problems* [56], que presenta un algoritmo cuántico en tiempo polinómico para resolver LWE, considerando ciertas relaciones entre el ruido y el módulo. Aunque este algoritmo es relevante desde el punto de vista teórico, los parámetros actuales de Kyber, como el tamaño del módulo y la dimensión, siguen siendo seguros frente a este enfoque. Además, en el análisis *Implications of the Proposed Quantum Attack on LWE* [57], Nigel Smart explora las implicaciones de un algoritmo cuántico propuesto para resolver LWE en tiempo polinómico. Aunque promete avances, la viabilidad práctica de este algoritmo con los parámetros de Kyber aún está siendo evaluada. A pesar de que actualmente no existen algoritmos cuánticos que resuelvan LWE en tiempo polinómico, es posible que futuras mejoras en la computación cuántica afecten la seguridad de Kyber si se desarrollan métodos más eficientes para resolver LWE.

Cabe destacar la existencia de herramientas como *LWE-benchmarking* [58], desarrollada por Facebook Research, que permite evaluar empíricamente la resistencia de instancias del problema LWE frente a ataques clásicos. Aunque esta herramienta no contempla ataques cuánticos, proporciona un marco útil para analizar la efectividad de técnicas tradicionales en función de los parámetros del esquema.

Además de probar ser sólido frente a ataques clásicos, tampoco se conocen algoritmos cuánticos que puedan romper Kyber de manera eficiente, y por eso ha sido seleccionado como estándar en la criptografía postcuántica. Sin embargo, la investigación en algoritmos cuánticos sigue avanzando, por lo que es crucial seguir evaluando la seguridad de Kyber frente a posibles mejoras en la computación cuántica y adaptar sus parámetros si fuera necesario. Por este motivo el NIST se esfuerza en reclutar nuevos estándares para no depender solo de un problema intratable, sino un abanico de ellos.

## 7.2. Ataques de canal lateral

Un ataque de canal lateral, a diferencia de los ataques tradicionales, no se enfoca en vulnerabilidades del software o debilidades en los algoritmos, sino en la explotación de las características físicas de un sistema, como el consumo de energía, el tiempo de ejecución de las operaciones o incluso las emisiones electromagnéticas, con el objetivo de obtener información confidencial.

En el contexto de la criptografía postcuántica, estos ataques representan una amenaza significativa, ya que, al ser los algoritmos lo suficientemente seguros como para resistir ataques cuánticos, su implementación física se convierte, inevitablemente, en el principal objetivo para vulnerar el sistema. Por ejemplo, algunas operaciones de reducción de vectores en un retículo pueden presentar tiempos de ejecución o características específicas que revelan el método utilizado, lo que podría comprometer la confidencialidad de las claves privadas. También, en algoritmos como Kyber o Dilithium, las multiplicaciones y reducciones modulares pueden revelar información si no se implementan de forma constante en tiempo o sin optimizar el manejo de caché. Además, se han identificado ataques que aprovechan la propagación de fallos en implementaciones de esquemas basados en el problema de aprendizaje con errores (LWE) [59].

Es por esto por lo que actualmente se implementen una serie de contramedidas especificas tanto en el hardware como en el software. Algunas de estas técnicas son el enmascaramiento, la introducción de ruido y la implementación de diseños resistentes a fallos.

El enmascaramiento es una técnica que consiste en mezclar los datos sensibles con valores aleatorios, para que no sea posible obtener la información al observar las señales físicas del sistema. Por ejemplo, si estamos trabajando con una clave criptográfica, podemos añadirle un valor aleatorio antes de realizar cualquier operación. Esto hace que las señales físicas (como el consumo de energía o el tiempo de procesamiento) no muestren la clave real. Por otro lado, la introducción de ruido implica agregar perturbaciones aleatorias al sistema. Estas alteraciones dificultan que un atacante pueda identificar patrones útiles para obtener información confidencial. Al mezclar las señales relevantes con las aleatorias, el ruido complica los ataques de canal lateral, haciendo más difícil que el atacante pueda distinguir entre lo que es útil y lo que es ruido. Por último, los diseños resistentes a fallos consisten en desarrollar algoritmos y arquitecturas que mantienen su seguridad incluso en presencia de fallos o manipulaciones físicas. Estos diseños están pensados para resistir ataques que intentan inducir errores en el sistema, como variaciones de voltaje o temperatura, que podrían ser explotadas para obtener información confidencial.

En conclusión, la implementación de contramedidas contra ataques de canal lateral es crucial para garantizar la seguridad de la criptografía postcuántica, pues es su único punto débil conocido. Descuidar estas contramedidas podría comprometer la confidencialidad de los datos que se buscan proteger.

## 7.3. Ataques de texto en claro elegido (CCA)

Los ataques de descifrado adaptativo, también conocidos como ataques de texto cifrado elegido (CCA, *(Chosen Ciphertext Attacks)*, son una categoría de ataques criptográficos en los que un atacante tiene la capacidad de elegir textos cifrados y obtener información sobre sus correspondientes textos en claro a través de un oráculo de descifrado, es decir, un sistema que le permite enviar textos cifrados y recibir los mensajes en claro correspondientes. Estos ataques pueden comprometer la seguridad de los esquemas criptográficos si el sistema no está diseñado adecuadamente para resistirlos.

Existen dos variantes principales de los ataques CCA: El CCA1 y el CCA2. El modelo CCA1 *(Non-adaptive Chosen Ciphertext Attack)*, permite al atacante enviar varios textos cifrados al oráculo y obtener sus textos en claro antes de recibir el desafío. Sin embargo, una vez que recibe el texto cifrado objetivo, no puede interactuar con el oráculo para adaptar consultas. Este tipo de ataque es menos potente que el CCA2, pero puede ser útil para analizar debilidades en un sistema criptográfico.

En el modelo CCA2, el atacante tiene acceso al oráculo de descifrado tanto antes como después de recibir el mensaje. Esto le permite modificar estratégicamente textos cifrados y analizar las respuestas del oráculo para extraer información sobre la clase secreta o recuperar el mensaje original. Este es el tipo de ataque más peligroso y el que la mayoría de los esquemas criptográficos modernos buscan prevenir.

Uno de los ataques CCA2 más conocidos es el ataque de Bleichenbacher contra RSA con el oráculo PKCS#1 v1.5. En este ataque, el atacante envía múltiples textos cifrados manipulados y observa si el sistema rechaza o acepta los descifrados. A través de este proceso, puede reconstruir el mensaje original sin necesidad de conocer la clave privada.

El esquema Kyber–KEM incorpora técnicas como el uso de aleatorización y hash de reencapsulación para garantizar la seguridad contra ataques CCA2.

# 8. Futuro de la criptografía postcuántica y nueva convocatoria del NIST

La computación cuántica amenaza la seguridad de los sistemas criptográficos actuales, haciendo necesaria una transición hacia algoritmos postcuánticos. Sin embargo, este cambio implica desafíos, como la actualización de infraestructuras y el mayor coste computacional de los nuevos algoritmos.

A pesar de estas dificultades, la adopción temprana de la criptografía postcuántica permitirá fortalecer la seguridad digital a largo plazo. En este proceso, el NIST juega un papel clave en la estandarización de estos algoritmos, facilitando su integración en los sistemas actuales.

En los siguientes apartados, se analizarán los retos y oportunidades de esta transición, así como la nueva convocatoria del NIST para la selección de algoritmos adicionales. Para redactarlos, se ha investigado el documento *CCN-TEC 009 Recomendaciones para una transición postcuántica segura* [60], además del documento *Migration to Post-Quantum Cryptography* [61] del NIST. Otros documentos que se han usado se indican en cada apartado y para el apartado 8.2.1. sobre los desafíos en enternos IoT (*Internet of Things*) se ha visitado el documento *Challenges and opportunities of post-quantum cryptography for Internet of Things* [62].

## 8.1. Transición a la criptografía postcuántica

La transición a la criptografía postcuántica es un proceso inevitable debido al avance de la computación cuántica. Los sistemas criptográficos tradicionales, como RSA, ECC y DH, dependen de problemas matemáticos que podrían ser resueltos eficientemente por un ordenador cuántico con suficientes cubits. Por ello, como hemos visto a lo largo de este documento, el NIST ha trabajado en la estandarización de algoritmos resistentes a ataques cuánticos. Sin embargo, la transición hacia estos nuevos estándares no es inmediata y requiere una planificación estructurada para garantizar la seguridad y compatibilidad de los sistemas actuales, ya que implica rediseñar y actualizar protocolos y sistemas de seguridad en todo el mundo.

Para facilitar esta migración, Organismos como el Centro Criptológico Nacional (CCN) han emitido recomendaciones como adoptar un enfoque híbrido, donde se combinan algoritmos postcuánticos con esquemas clásicos para evitar interrupciones en la seguridad [60]. Esto es especialmente importante en sectores donde la integridad y confidencialidad de la información deben garantizarse durante largos períodos, como en bancos, defensa y comunicaciones

gubernamentales. La implementación de estos nuevos algoritmos debe realizarse por fases, priorizando los sistemas más críticos y asegurando que las infraestructuras actuales puedan soportar claves y firmas de mayor tamaño sin afectar el rendimiento. Empresas tecnológicas como Google, IBM y Microsoft ya están explorando la integración de estos estándares en sus productos, mientras que organismos internacionales trabajan en la actualización de protocolos como SSH para soportar criptografía postcuántica.

En España, el Ministerio de Defensa, junto con el Ministerio de Ciencia, Innovación y Universidades y el Ministerio para la Transformación Pública, organizó en febrero de 2025 la primera mesa redonda sobre la estrategia de tecnologías cuánticas. En este encuentro se debatió el impacto de estas tecnologías en sectores clave. Con esta iniciativa, España busca afianzar su liderazgo en el ámbito cuántico, un sector que, según el Ministerio de Defensa, «tiene el potencial de transformar industrias y la sociedad en su conjunto, posicionando al país como un referente global en innovación» [63].

## 8.2. Desafíos y oportunidades de la criptografía postcuántica

Este proceso de transición trae consigo diversos desafíos y oportunidades. Uno de los principales retos es la adaptación de hardware y software, ya que muchos dispositivos actuales no están diseñados para manejar las claves más grandes y los tiempos de procesamiento que requieren algunos algoritmos postcuánticos. Además, la seguridad de los nuevos esquemas sigue siendo un área activa de investigación, ya que, aunque han sido seleccionados por el NIST, pueden surgir nuevos ataques que requieran ajustes o incluso su reemplazo en el futuro. Otro aspecto importante es el impacto en la eficiencia, ya que algunas soluciones criptográficas postcuánticas requieren más recursos computacionales, lo que puede ser un obstáculo en dispositivos con limitaciones de memoria o procesamiento.

A pesar de estos desafíos, la criptografía postcuántica también presenta oportunidades significativas. La adopción de estos nuevos estándares no solo permitirá mantener la seguridad frente a amenazas cuánticas, sino que también puede mejorar algunos aspectos de la criptografía actual, como la velocidad de ciertos esquemas de firma digital o el diseño de nuevos protocolos de seguridad más resistentes. Además, la transición hacia estos algoritmos impulsará la investigación en áreas como la criptografía basada en isogenias de curvas elípticas y la criptografía basada en hash, que podrían convertirse en alternativas viables en el futuro.

### 8.2.1. Posibles desafíos en la integración de la criptografía postcuántica en entornos IoT

La integración de algoritmos criptográficos postcuánticos en dispositivos del internet de las cosas (IoT) representa uno de los desafíos más relevantes para la adopción de estas nuevas tecnologías. Los dispositivos IoT suelen estar diseñados bajo fuertes restricciones de recursos, tanto en términos de capacidad computacional como de memoria y consumo energético. Sin embargo, los algoritmos postcuánticos, especialmente aquellos basados en retículos, como ya hemos visto, requieren un manejo de claves y operaciones mayor que los esquemas clásicos, lo que incrementa la demanda de memoria y potencia de cálculo.

Esta limitación resulta aún más problemática en aplicaciones donde es fundamental mantener tiempos de respuesta muy bajos, algo especialmente importante en dispositivos con autonomía limitada o con recursos restringidos. Esto es habitual en sensores industriales, dispositivos médicos implantados y sistemas de control autónomo. Por tanto, el futuro despliegue de criptografía postcuántica en IoT dependerá no solo de la resistencia de los algoritmos frente a ataques cuánticos, sino también de la capacidad de la industria de optimizar implementaciones que equilibren seguridad y eficiencia en dispositivos con recursos limitados.

Por el momento, Telefónica junto a la empresa Halotech, han logrado integrar cifrado postcuántico en dispositivos IoT como cascos y pulseras inteligentes utilizados en entornos industriales críticos, pero queda un largo camino hasta que podamos verlo integrado de manera generalizada en todos los dispositivos IoT con los que interactuamos a diario, cuya seguridad es crucial para garantizar nuestro bienestar y privacidad [64].

## 8.3. Nueva convocatoria del NIST

El National Institute of Standards and Technology (NIST) ha impulsado desde 2016 un proceso de estandarización de criptografía postcuántica para contrarrestar las amenazas de los computadores cuánticos. Como resultado de este proceso, en julio de 2022 se seleccionaron CRYSTALS–Kyber como mecanismo de encapsulación de clave (KEM) y CRYSTALS–Dilithium, FALCON y SPHINCS+ como esquemas de firma digital [65]. Los estándares correspondientes, FIPS 203, 204 y 205, fueron publicados en agosto de 2024, mientras que FALCON sigue en desarrollo [66].

Sin embargo, reconociendo la necesidad de diversificar su portafolio criptográfico, el NIST ha lanzado una nueva convocatoria para completar su conjunto de algoritmos con soluciones adicionales. Esta nueva fase se centra en la evaluación de esquemas de firma digital adicionales, así como algoritmos de cifrado y KEM alternativos a Kyber que puedan ofrecer diferentes ventajas en términos de eficiencia y seguridad.

En septiembre de 2022, el NIST abrió la convocatoria para esquemas de firma

digital adicionales, que cerró el 1 de junio de 2023 con 50 propuestas recibidas, de las cuales 40 fueron aceptadas. Posteriormente, el 24 de octubre de 2024, se anunciaron 14 candidatos en la segunda ronda, permitiendo actualizaciones hasta el 5 de febrero de 2025. Se estima que la evaluación de estos esquemas tomará entre 12 y 18 meses. En esta fase, el NIST busca esquemas que no dependan de redes estructuradas y que presenten firmas más cortas, características clave para aplicaciones como la transparencia de certificados.

Paralelamente, en la cuarta ronda del proceso inicial de estandarización, se evalúan KEM alternativos a Kyber. Entre los candidatos se encuentran BIKE, HQC y Classic McEliece, basados en criptografía de códigos, así como SIKE, basado en curvas isógenas.

En relación a estas nuevas propeustas de KEM, el 11 de marzo de 2025, el NIST anunció la selección de HQC (Hamming Quasi-Cyclic) como un segundo mecanismo de encapsulación de claves, junto a ML-KEM (CRYSTALS-Kyber), fortaleciendo así la diversidad criptográfica del conjunto de estándares. HQC es un esquema basado en códigos cuasi-cíclicos, cuya seguridad se fundamenta en la dificultad del problema de decodificación de síndromes (en el cual no ahondaremos dado que rebasa el alcance del trabajo), y se distingue por no depender de estructuras algebraicas como los retículos. Por otra parte, el NIST ha anunciado que el borrador del estándar para HQC se publicará en 2026, con una versión final prevista para 2027. Mientras tanto, se recomienda a las organizaciones adoptar los estándares FIPS 203, 204 y 205 ya publicados, como parte de una transición proactiva hacia entornos resistentes a la computación cuántica [67].

El futuro de la criptografía postcuántica dependerá en gran medida de cómo evolucione la computación cuántica y de la capacidad de adaptación de las infraestructuras actuales. La estandarización es solo el primer paso; la verdadera prueba vendrá con la implementación a gran escala y la resistencia de estos nuevos algoritmos a ataques futuros. Por ello, la investigación y desarrollo en este campo continuará siendo esencial en los próximos años para garantizar la seguridad digital en la era postcuántica. En este contexto, el NIST había solicitado comentarios para una guía técnica que proporciona recomendaciones sobre la transición a la criptografía postcuántica, el borrador SP 800–227, con fecha límite hasta el 7 de marzo de 2025 [68], reafirmando su compromiso con la evolución de la criptografía postcuántica y su implementación segura en el ecosistema digital.

# 9. Conclusiones

Este trabajo ha ofrecido una visión didáctica y técnica del esquema Kyber-KEM, seleccionado por el NIST como estándar para el intercambio de claves en el contexto de la criptografía postcuántica. A través del análisis realizado, se ha puesto de manifiesto tanto su solidez teórica como su viabilidad práctica, sin perder de vista las limitaciones y desafíos que su adopción plantea en entornos reales.

Un aspecto fundamental que debe destacarse es que Kyber no puede considerarse una solución única ni definitiva. La historia de la criptografía demuestra que la diversidad de algoritmos fortalece la seguridad global. Apostar por una estrategia de diversificación, incorporando alternativas como McEliece o BIKE, contribuirá a una infraestructura más resiliente frente a vulnerabilidades presentes o futuras.

Asimismo, aunque la computación cuántica aún no representa una amenaza inmediata, su desarrollo avanza con rapidez. Adoptar una actitud pasiva supondría un riesgo innecesario, especialmente ante escenarios como el de "Harvest now, decrypt later", donde información cifrada hoy podría ser comprometida en el futuro. Por ello, creemos que la transición hacia esquemas postcuánticos debe iniciarse cuanto antes, comenzando por sistemas críticos y evaluando su integración de forma progresiva y controlada.

También se ha subrayado la importancia de seguir auditando y optimizando estos algoritmos, tanto a nivel teórico como en sus implementaciones prácticas. Su integración en dispositivos con recursos limitados (como los del Internet de las Cosas (IoT)) presenta retos significativos, al igual que su incorporación en protocolos ampliamente utilizados como TLS, SSH o blockchain, que requerirán ajustes técnicos y normativos.

Además de estas conclusiones generales, el trabajo ha permitido identificar problemas abiertos y líneas de investigación futuras que serán clave en la evolución de la criptografía postcuántica y pueden llevar a nuevos trabajos de investigación para licenciados en Ingeniería del Software. Estas áreas deben ser abordadas con prioridad para garantizar una transición sólida y sostenible hacia un entorno digital seguro frente a amenazas cuánticas.

Uno de los desafíos principales será lograr una integración eficiente de los nuevos algoritmos en infraestructuras existentes, sin comprometer el rendimiento ni la interoperabilidad. Este aspecto resulta especialmente relevante en sectores como las comunicaciones seguras, la banca o las redes distribuidas. En este sentido, otro ámbito crítico es el impacto sobre los sistemas financieros y las criptomonedas, donde la seguridad, la eficiencia y la compatibilidad serán aspectos clave que podrían requerir reestructuraciones importantes.

Asimismo, será necesario avanzar en comparativas entre distintos esquemas

postcuánticos, tanto en rendimiento como en robustez, para apoyar procesos de estandarización más precisos y adaptados a diferentes escenarios de uso. Del mismo modo, urge trabajar en la estandarización de esquemas de firma digital que ofrezcan garantías similares a las que Kyber ya proporciona en el ámbito del cifrado de clave pública.

Otra línea prometedora es la exploración de la criptografía basada en pruebas de conocimiento cero (ZKP), una herramienta con gran potencial para complementar o incluso sustituir, en algunos casos, los esquemas postcuánticos actuales. Estas técnicas pueden ofrecer nuevas soluciones orientadas a la privacidad y la verificación segura.

Dado el enfoque pedagógico del trabajo, también se propone como futura línea de acción el desarrollo de materiales educativos y divulgativos sobre criptografía postcuántica, con el objetivo de facilitar su comprensión e impulsar su enseñanza en niveles académicos intermedios y avanzados.

Finalmente, se considera esencial mantener una vigilancia tecnológica constante ante la aparición de nuevos algoritmos cuánticos que puedan poner en peligro incluso los sistemas hoy considerados seguros. Esta monitorización permitirá responder con agilidad ante posibles avances disruptivos en el campo de la computación cuántica.

En conjunto, las conclusiones y líneas propuestas no solo resumen los hallazgos del presente trabajo, sino que reafirman la necesidad de continuar investigando desde un enfoque multidisciplinar, proactivo y adaptativo, con el fin de garantizar la seguridad de la información en una era donde los paradigmas tecnológicos están en plena transformación.

# 10. Problemas encontrados

En este apartado, se presentan los principales problemas encontrados durante el desarrollo del proyecto. A lo largo del proceso, nos enfrentamos a diversos desafíos que nos llevaron a evaluar y considerar distintas soluciones. A continuación, detallamos los problemas identificados, el contexto en el que surgieron, las alternativas que planteamos y la opción final elegida, con el objetivo de compartir las decisiones tomadas y las experiencias adquiridas para mejorar futuros desarrollos.

| **Problema n1** | **Autor/es: Ignacio y Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | Facilitar a los usuarios el acceso a la implementación sin necesidad de una instalación local. |
| **Descripción del problema** | La instalación en local no es una alternativa atractiva para el usuario, vemos necesaria otra alternativa más directa para acceder al contenido. |
| **Alternativas consideradas** | 1. Desplegar la aplicación en línea: Utilizar un host como Binder para el acceso en remoto, proporcionaría una alternativa mucho más inmediata e inocua. Pero dependeremos de la disponibilidad del servicio.<br>2. Desarrollar un ejecutable con objetivo de instalar todas las dependencias necesarias de forma intuitiva: Muchos usuarios podrían desconfiar del origen de un ejecutable. Además, la instalación en local mediante línea de comandos es mucho más transparente y sencilla. |
| **Alternativa elegida** | Despliegue en Binder: Tras realizar una breve formación sobre el procedimiento a seguir para desplegar los cuadernos, se lograron exhibir los repositorios. |
| **Conclusión** | Pese a las posibles inconveniencias de la falta de servicio de Binder, priorizamos que sea fácil para el usuario acceder al contenido, ya que consideramos una parte esencial que complementa el TFG de forma práctica y complicar su acceso haría que muchos desistieran en acceder. |

**Tabla 10.1:** Problema 1

| **Problema n2** | **Autor/es: Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | Al intentar implementar el problema LWE en Java. |

| | |
|---|---|
| **Descripción del problema** | Ni los resultados ni la complejidad eran los esperados una vez se implementó el algoritmo, esto se debe a que hubo un error en el planteamiento, pues al tratar en módulo $q$, Java comprende los elementos desde $(-q+1, q-1)$. No obstante, para ser coherentes con la definición del problema se debe trabajar con retículos en el rango $(0, q-1)$. |
| **Alternativas consideradas** | 1. Conformarnos con la implementación del momento, que si bien no era precisa, sí servía para un propósito didáctico.<br>2. Trabajar en encontrar una solución para que el algoritmo sea congruente con la definición proporcionada. |
| **Alternativa elegida** | Trabajar en encontrar una solución: Este error fue solucionado cambiando la forma en que se trataban los módulos, asegurando que los valores estuvieran siempre en el rango $(0, q-1)$. Para ello, se aplicó una corrección al cálculo del módulo en Java, utilizando la expresión: $\text{mod}(x, q) = (x \text{ mód } q + q) \text{ mód } q, \forall x \in \mathbb{Z}$, de modo que los residuos negativos se transformaran en valores dentro del rango adecuado. Con esta modificación, los cálculos se alinearon correctamente con la definición del problema y se obtuvieron los resultados esperados. |
| **Conclusión** | Al ser LWE el pilar fundamental del PKE usado por el estándar, decidimos dedicarle el tiempo que fuese necesario hasta encontrar una implementación que encajara completamente con la definición del problema. |

**Tabla 10.2:** Problema 2

| **Problema n3** | **Autor/es: Ignacio y Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | Durante la implementación de los algoritmos de encapsulado y desencapsulado del estándar Kyber. |
| **Descripción del problema** | Al trabajar en las funciones de encapsulado y desencapsulado en Python, siguiendo la estructura de ambos algoritmos proporcionada en el FIPS-203, nos topamos con que el mensaje no se recuperaba correctamente en algunas ocasiones. |
| **Alternativas consideradas** | 1. Conformarnos con la implementación del momento, que, si bien no era precisa, sí servía para un propósito didáctico.<br>2. Investigar el por qué de este suceso e intentar proponer una alternativa más precisa. |

| | |
|---|---|
| **Alternativa elegida** | Investigar el por qué de este suceso: Tras varias evaluaciones e intentos de optimización de los algoritmos, nos dimos cuenta de que esta imprecisión es intrínseca al estándar, debido a la naturaleza probabilística de Kyber proporcionada por la introducción de errores. Estas imprecisiones no afectan la seguridad ni la funcionalidad del algoritmo, pero sí pueden influir en la exactitud de ciertos cálculos internos. En nuestro caso, nos vimos obligados a reducir los parámetros en gran medida para poder comenzar a vislumbrar una tasa de aciertos elevada. |
| **Conclusión** | A través de nuestra investigación y experimentación, comprendimos que las imprecisiones observadas no eran errores de implementación, sino características inherentes al estándar Kyber debido a su naturaleza probabilística. Si bien logramos mejorar la tasa de aciertos ajustando los parámetros, este ajuste conlleva un compromiso entre precisión, eficiencia y seguridad. Este aprendizaje nos permitió profundizar en el funcionamiento del algoritmo y su comportamiento en distintos escenarios, brindándonos una mejor comprensión de sus limitaciones y aplicaciones prácticas.<br>Al ser en última instancia un trabajo de investigación didáctico, consideramos que, si bien la implementación no es perfecta, sí puede ser utilizada para fines didácticos, lo cual era el objetivo desde el principio. |

**Tabla 10.3:** Problema 3

| **Problema n4** | **Autor/es: Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | Al intentar realizar una estimación del tiempo requerido por un ordenador cuántico para resolver problemas sobre retículos (CVP, SVP, LWE) |
| **Descripción del problema** | Con el objetivo de probar la intratabilidad de los problemas basados en retículos para la infraestructura clásica y cuántica, se midieron los tiempos para diversos tamaños del problema en un computador convencional, mas para la estimación cuántica, no somos capaces de proponer ninguna. |
| **Alternativas consideradas** | 1. Utilizar las estimaciones convencionales y añadirle una reducción equivalente a aplicar previamente el algoritmo de Grover: Descartada por no tratarse de una base de datos no estructurada.<br>2. Utilizar las estimaciones convencionales y añadirle una reducción equivalente a aplicar el algoritmo de Shor: Descartada por no tratarse de un problema de factorización ni de PLD. |

| | |
|---|---|
| | 3. Seguir investigando y manteniéndonos al tanto sobre la aparición de algoritmos cuánticos con el propósito de resolver estos problemas reticulares. |
| **Alternativa elegida** | Seguir investigando y manteniéndonos al tanto sobre la aparición de algoritmos cuánticos: Es la única opción viable y lógica por el momento, consideramos que no es buena idea realizar suposiciones sin pruebas sólidas. |
| **Conclusión** | Teniendo esto en cuenta, la elección de estos problemas por el NIST se hace evidente, pues se debe al hecho de que no existan algoritmos cuánticos capaces de vulnerarlas por el momento. No obstante, los avances en cuántica son impredecibles, y es importante mantenerse al día con los nuevos descubrimientos. |

**Tabla 10.4:** Problema 4

| **Problema n5** | **Autor/es: Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | En los primeros acercamientos a los retículos, al intentar comprender y usar la librería Lattpy. |
| **Descripción del problema** | Para empezar a entender de forma práctica los retículos, se optó por el uso de la librería lattpy, esta decisión tuvo sus pros y contras, uno de estos últimos fue el hecho de tener funcionalidades tangenciales al uso propio de los retículos en el estándar kyber, lo cual se traduce en una inversión de tiempo en algo que no produjo resultados. |
| **Alternativas consideradas** | 1. Haber realizado una implementación de una librería de retículos por nuestra cuenta: sin lugar a duda una opción interesante, mas no está alineado con los propósitos del TFG.<br>2. Utilizar lattpy: de esta forma podemos centrar el trabajo en la explicación didáctica del estándar postcuántico. |
| **Alternativa elegida** | Utilizamos LattPy. Si bien nos tomó algo de tiempo comprender qué funcionalidades necesitábamos y cuáles no, estimamos que la opción 1 nos habría llevado aún más tiempo. |
| **Conclusión** | Creemos que la decisión fue la acertada, pues con una inversión de tiempo razonable, pudimos obtener una base sólida sobre la que trabajar con retículos y desarrollar los cuadernillos complementarios al TFG. |

**Tabla 10.5:** Problema 5

| **Problema n6** | **Autor/es: Ignacio y Gabriel** |
|---|---|

| | |
|---|---|
| **Contexto en el que sucede el problema** | En las primeras tomas de contacto con la PQC. |
| **Descripción del problema** | Como estudiantes de informática, a lo largo de la carrera, si bien hemos recibido nociones de criptografía, los primeros acercamientos a la PQC fueron en cierto modo abrumadores, debido a la ingente cantidad de información nueva que procesar y separar en necesario y no. |
| **Alternativas consideradas** | 1. Pedir una reunión con el tutor don Andrés Armario para que nos asesore sobre el camino a seguir. |
| **Alternativa elegida** | Celebramos una reunión con don Andrés y nos comentó una serie de documentos vitales para entender el contexto de la PQC, así como varias páginas web donde revisar actualidad de este paradigma. |
| **Conclusión** | Las indicaciones del tutor nos encauzaron de forma efectiva y precisa hacia el objetivo del TFG, sin esta guía, el comienzo hubiera sido mucho más tortuoso. |

**Tabla 10.6:** Problema 6

| **Problema n7** | **Autor/es: Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | Al intentar integrar la librería de complejidad realizada por Don Miguel Toro Bonilla. |
| **Descripción del problema** | En pos de realizar un análisis empírico de la complejidad de calidad sobre los algoritmos descritos en el TFG, fue necesario realizar un trabajo de integración entre el repositorio del profesor Miguel Toro y nuestro proyecto en java llamado latticeComplexity. |
| **Alternativas consideradas** | 1. No realizar la integración: en vez de realizar un esfuerzo en conectar ambos proyectos, hacer de forma manual las anotaciones del resultado tamaño/tiempo para luego erigir métricas.<br>2. Realizar el trabajo de integración para poder usar la librería: encontrar la forma de compatibilizar ambos proyectos para automatizar las métricas y hacer experimentos reproducibles. |
| **Alternativa elegida** | Se optó por la segunda opción, ya que es una herramienta muy útil que tuvimos que utilizar para la asignatura de Análisis y Diseño de Datos y Algoritmos, y pensamos que aprovechar los conocimientos adquiridos y reutilizarlos es la mejor opción. |

| | |
|---|---|
| **Conclusión** | Estamos seguros de que la opción electa fue la adecuada, pues se tuvieron que hacer cambios continuos en todos los algoritmos a medida que avanzaba la investigación, y haber hecho manualmente el trabajo de analizar la complejidad en cada release hubiera consumido demasiado tiempo. |

**Tabla 10.7:** Problema 7

| **Problema n8** | **Autor/es: Ignacio y Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | En lo respectivo a la implementación del algoritmo LWE. |
| **Descripción del problema** | Al intentar implementar el algoritmo LWE se encontraron problemas tanto de rendimiento como de eficiencia, había veces en las que se generaba un bucle infinito. |
| **Alternativas consideradas** | 1. Seguir probando el programa hasta encontrar una solución.<br><br>2. Buscar implementaciones parecidas en Github con el objetivo de mejorar y optimizar el algoritmo. |
| **Alternativa elegida** | En este caso se optó por una mezcla de ambas, se realizó un análisis exhaustivo del comportamiento del algoritmo, observando los parámetros cambiar en cada iteración y deduciendo la posible causa del error. Por otra parte, se tomaron ideas de otras implementaciones que sirvieron para mejorar en diversos aspectos el algoritmo. El error radicaba en que existía la posibilidad, que debido al error introducido en las ecuaciones (en el caso de ser excesivo), se conformara un sistema incompatible, del cual ninguna combinación de vector secreto s fuera solución. Lo que hacía que el programa diera vueltas buscando una solución no factible. |
| **Conclusión** | En los problemas que introducen errores, el equilibrio entre los parámetros es vital para la existencia de una solución válida. |

**Tabla 10.8:** Problema 8

| **Problema n9** | **Autor/es: Ignacio** |
|---|---|
| **Contexto en el que sucede el problema** | Durante la fase inicial de la investigación, sin conocimiento previo sobre qué eran los retículos, se dedicó varias horas a estudiar los retículos desde el punto de vista de la teoría del orden. |

| | |
|---|---|
| **Descripción del problema** | El estudio de los retículos desde la teoría del orden resultó ser innecesario para el objetivo del trabajo. Al no conocer la diferencia entre los tipos de retículos, se perdió tiempo y esfuerzo investigando un enfoque incorrecto, ya que lo que se necesitaba eran los retículos algebraicos. |
| **Alternativas consideradas** | 1. Revisión de otros tipos de retículos en teoría del orden.<br><br>2. Continuar la investigación hasta descubrir el tipo de retículo correcto. |
| **Alternativa elegida** | Se detuvo la investigación inicial y, tras una revisión más profunda, se identificaron los retículos algebraicos como los que correspondían al trabajo. |
| **Conclusión** | Este error de enfoque permitió ajustar la dirección de la investigación y mejorar la precisión en la búsqueda de información relevante. |

**Tabla 10.9:** Problema 9

| **Problema n10** | **Autor/es: Ignacio** |
|---|---|
| **Contexto en el que sucede el problema** | Durante el diseño del esquema criptográfico, no se anticipó el comportamiento del ruido bajo ciertos parámetros, lo que resultó en un rendimiento subóptimo cuando el error se propagó a través de los cálculos. |
| **Descripción del problema** | El ruido creció de forma no controlada durante el proceso de descifrado, lo que provocó que los mensajes recuperados contuvieran demasiados errores, haciendo imposible una interpretación correcta del mensaje cifrado. |
| **Alternativas consideradas** | 1. Utilizar técnicas más robustas para la gestión del ruido, como el escalado dinámico.<br>2. Implementar técnicas de reconstrucción de mensajes a partir de errores menores. |
| **Alternativa elegida** | Se decidió aplicar un control más estricto del ruido mediante el ajuste dinámico de los parámetros del algoritmo, mejorando la capacidad de recuperación sin comprometer la eficiencia. |
| **Conclusión** | Controlar el crecimiento del error en cada iteración permitió mejorar la precisión del proceso de descifrado y la estabilidad general del sistema. |

**Tabla 10.10:** Problema 10

| Problema n11 | Autor/es: Ignacio |
|---|---|
| **Contexto en el que sucede el problema** | Durante la implementación de un sistema criptográfico basado en retículos, se observó que las operaciones requerían una cantidad excesiva de recursos computacionales, lo que afectaba la eficiencia y el rendimiento general del sistema. |
| **Descripción del problema** | La eficiencia y el rendimiento del sistema se vieron comprometidos debido a la sobrecarga de operaciones de cálculo intensivo, como la multiplicación de matrices grandes y las operaciones de reducción. Esto causaba que el sistema fuera lento e ineficiente, especialmente cuando se trabajaba con grandes volúmenes de datos. |
| **Alternativas consideradas** | 1. Implementar métodos de optimización como la reducción de la complejidad algorítmica.<br>2. Usar técnicas de paralelización para distribuir la carga de trabajo y acelerar los procesos.<br>3. Reducir la complejidad del sistema mediante la reducción de dimensionalidad de las matrices. |
| **Alternativa elegida** | Se optó por la última alternativa, la más sencilla. Esto simplificó las operaciones sin entrar en técnicas de paralelización u optimización algorítmica avanzada. Esto permitió una mejora en el rendimiento, reduciendo la carga computacional. |
| **Conclusión** | Controlar el crecimiento del error en cada iteración permitió mejorar la precisión del proceso de descifrado y la estabilidad general del sistema. |

**Tabla 10.11:** Problema 11

| Problema n12 | Autor/es: Ignacio |
|---|---|
| **Contexto en el que sucede el problema** | Durante la implementación del esquema de aprendizaje con errores (MLWE) en un sistema de criptografía basado en retículos, surgieron errores en la implementación de las clases dentro del código, lo que afectó la funcionalidad general del sistema. |
| **Descripción del problema** | La implementación incorrecta de las clases en el código provocó que ciertas funciones no se ejecutaran correctamente, lo que causó fallos en el proceso de descifrado. Las clases mal implementadas no gestionaban adecuadamente las estructuras de datos, y en algunos casos, los valores de las matrices no se inicializaban correctamente, afectando el comportamiento general del sistema. |
| **Alternativas consideradas** | 1. Revisión detallada de la implementación de las clases para asegurar que todas las estructuras de datos estén correctamente definidas. |

| | |
|---|---|
| | 2. Reescribir las clases problemáticas para garantizar que las operaciones de MLWE se realicen de forma adecuada y eficiente. |
| **Alternativa elegida** | Se decidió realizar una revisión exhaustiva del código y reestructurar las clases involucradas en MLWE, corrigiendo errores en la inicialización de matrices y mejorando la gestión de las operaciones criptográficas. |
| **Conclusión** | La corrección de los errores en la implementación de las clases permitió que el sistema de MLWE funcionara correctamente, garantizando la correcta ejecución de los procesos de cifrado y descifrado, y mejorando la estabilidad del sistema. |

**Tabla 10.12:** Problema 12

| **Problema n13** | **Autor/es: Ignacio y Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | Durante las primeras fases del trabajo, al intentar comprender las bases matemáticas de los esquemas criptográficos postcuánticos como Kyber, concretamente los conceptos de retículos y anillos. |
| **Descripción del problema** | Nos enfrentamos a una barrera conceptual significativa al intentar estudiar las primitivas matemáticas subyacentes a los esquemas postcuánticos, dado que ni los retículos ni los anillos habían sido tratados en profundidad durante la carrera. Esto derivó en dificultades para entender tanto la formulación teórica como su aplicación práctica en los algoritmos como LWE o MLWE. |
| **Alternativas consideradas** | 1. Intentar avanzar directamente en el estudio del esquema sin una comprensión completa de las bases matemáticas, asumiendo los conceptos como cajas negras.<br>2. Dedicar tiempo adicional a formarnos de manera autodidacta en álgebra abstracta, estructuras algebraicas y teoría de retículos, antes de seguir con el diseño criptográfico. |
| **Alternativa elegida** | Se optó por la segunda alternativa. Aunque más costosa en términos de tiempo, decidimos detener el avance para reforzar los conocimientos teóricos, utilizando libros de texto, artículos académicos y recursos en línea, priorizando la comprensión de retículos euclidianos, módulos y estructuras cíclicas. |
| **Conclusión** | Esta inversión adicional de tiempo fue clave para poder seguir el hilo de los esquemas criptográficos basados en retículos. Nos permitió tener una visión más clara del funcionamiento interno de Kyber y del porqué de su seguridad, resultando en una comprensión más sólida y argumentada para el desarrollo del TFG. |

**Tabla 10.13:** Problema 13

| **Problema n14** | **Autor/es: Ignacio y Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | A lo largo del trabajo, al intentar comprender la rigurosa notación propia de los artículos y papers científicos. |
| **Descripción del problema** | Nos enfrentamos a una barrera intelectual significativa al intentar digerir los conocimientos que nos proporcionan las fuentes de rigor. Con el objetivo de respaldar nuestro trabajo con prestigiosos autores y organizaciones, la terea de comprender los conceptros expuestos y abstraerlos para plasmarlos de una forma didáctica ha sido un gran reto. Por otra parte, la mayoría del contenido existente sobre la PQC esta basado en aplicaciones muy concretas o demostraciones formales de un rigor matemático que se nos escapa por completo a nuestras herramientas matemáticas. |
| **Alternativas consideradas** | 1. Intentar abarcar los temas de una forma inteligente, partiendo de las bases matemáticas más simples y apoyándonos en la documentación inicial que nos proporcionó el tutor [18] [47].<br>2. Comenzar con una búsqueda arbitraria sobre el tema principal del TFG. |
| **Alternativa elegida** | Se optó por la primera alternativa. Los documentos proporcionados por el tutor, en especial [18] nos proporcionó un contexto ideal para entender el propósito de este trabajo. Debido al enfoque general del documento y sus semejanzas en lo que respecta a la criptografía clásica con lo aprendido en la asignatura de criptografía, nos ha servido de trampolín para abarcar temas más complejos. |
| **Conclusión** | Gracias a las indicaciones bibliográficas iniciales del tutor y a sus recomendacioens, fuimos capaces de comenzar la investigación con gran presteza. |

**Tabla 10.14:** Problema 14

| **Problema n15** | **Autor/es: Ignacio y Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | En la etapa final, al recibir un recordatorio del tutor sobre una propuesta que nos realizó para maquetar el TFG en LaTeX. |

| | |
|---|---|
| **Descripción del problema** | Dado que el Trabajo de Fin de Grado fue desarrollado por dos personas, surgió la necesidad de contar con un documento compartido que permitiera realizar aportaciones de forma colaborativa. Este documento fue creciendo progresivamente hasta convertirse en un prototipo del trabajo final. Una vez cumplida esta función, se procedió a su maquetación en LaTeX, a pesar de que ninguno de los autores tenía experiencia previa con esta herramienta. |
| **Alternativas consideradas** | 1. Estructurar el documento de la forma más clara y ordenada posible utilizando Microsoft Word.<br>2. Invertir un breve período en el aprendizaje de LaTeX y seguir la recomendación del tutor, con la confianza de que el resultado final sería considerablemente más profesional que con otras alternativas. |
| **Alternativa elegida** | Se optó por la segunda alternativa, si bien al principio el avance de la transcripción del documento era lento y tedioso, a medida que fuimos ganando soltura, la mejora en calidad de presentación se hizo evidente. |
| **Conclusión** | En definitiva, aunque el uso de LaTeX supuso un desafío inicial debido a nuestra falta de experiencia, la decisión de adoptarlo resultó acertada. No solo permitió una presentación mucho más profesional del trabajo, sino que también nos proporcionó nuevos conocimientos y habilidades útiles para futuros proyectos académicos y profesionales. |

**Tabla 10.15:** Problema 15

| **Problema n16** | **Autor/es: Ignacio y Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | Cada vez que se iniciaba el estudio de una nueva rama de conocimiento de la PQC. |
| **Descripción del problema** | Debido a la gran cantidad de información y líneas de investigación (problemas intratables, algoritmos relativos al PKE y KEM, actualidad, ciberataques, etc.), tendíamos a enfrascarnos en detalles que, si bien resultaban interesantes, nos alejaban del objetivo principal del trabajo. Por ello, fue necesario detenernos en varias ocasiones para reconducir el enfoque y no desviarnos de nuestro propósito didáctico, centrándonos en lo esencial: el estándar FIPS 203 [47], verdadera piedra angular de nuestro proyecto. |

| | |
|---|---|
| **Alternativas consideradas** | 1. Explorar de forma amplia y transversal todos los aspectos relevantes de la criptografía postcuántica, desde fundamentos teóricos hasta casos actuales de ciberseguridad, pasando por distintos esquemas criptográficos y escenarios de ataque.<br>2. Utilizar como base estructural los documentos oficiales del CCN [18] y el estándar FIPS 203 [47], empleándolos como guión para organizar y desarrollar el contenido. Esta opción permitía centrarse en los puntos clave reconocidos institucionalmente como fundamentales, facilitando un enfoque didáctico más claro y alineado con los objetivos del TFG. |
| **Alternativa elegida** | Finalmente, optamos por una combinación de ambas alternativas. Realizamos un estudio riguroso de los fundamentos teóricos, así como algunas implementaciones prácticas propias, para adquirir una comprensión profunda de los conceptos clave. Sin embargo, todo este trabajo se articuló en torno al FIPS 203 [47], que funcionó como hilo conductor y objetivo final. El documento del CCN y el propio estándar nos sirvieron como guía estructural, asegurando que el contenido no se dispersara y que mantuviéramos siempre presente nuestro enfoque didáctico. |
| **Conclusión** | Esta estrategia híbrida nos permitió abordar el trabajo con una base sólida tanto teórica como práctica, sin perder de vista nuestro objetivo principal: explicar y contextualizar el FIPS 203 dentro del marco de la criptografía postcuántica enfocado a estudiantes de cuarto de ingeniería informática. Así, logramos un equilibrio entre profundidad y claridad, garantizando un TFG coherente, útil y alineado con las necesidades actuales del ámbito de la ciberseguridad. |

**Tabla 10.16:** Problema 16

| **Problema n17** | **Autor/es: Ignacio y Gabriel** |
|---|---|
| **Contexto en el que sucede el problema** | Durante el uso de herramientas de inteligencia artificial para apoyar el estudio y desarrollo de contenidos relacionados con Kyber-KEM y la criptografía postcuántica en general. |

| | |
|---|---|
| **Descripción del problema** | La utilización de modelos de IA para obtener explicaciones, ejemplos de código o referencias sobre esquemas criptográficos avanzados como Kyber-KEM resultó ser problemática. En varios casos, la IA generaba respuestas con errores técnicos, explicaciones inconsistentes o incluso fragmentos de código no funcionales. Este fenómeno, conocido como "alucinación de la IA", afectó especialmente las fases en las que se requería precisión matemática o conocimiento profundo de implementaciones reales, haciendo ineficaz su uso para temas complejos sin una supervisión crítica constante. |
| **Alternativas consideradas** | 1. Continuar utilizando la IA como herramienta principal de consulta y generación de contenido, complementándola con una verificación manual exhaustiva de cada resultado obtenido.<br>2. Limitar el uso de la IA únicamente a tareas generales (resúmenes, esquemas, listas de conceptos) y basar el trabajo técnico y las explicaciones detalladas exclusivamente en fuentes primarias, como los documentos del NIST, papers científicos y bibliografía especializada. |
| **Alternativa elegida** | Optamos por la segunda alternativa. La IA se empleó solo como apoyo en tareas de bajo riesgo (estructuración de contenidos, elaboración de resúmenes preliminares), mientras que para los aspectos técnicos y de implementación nos apoyamos estrictamente en documentación oficial y bibliografía reconocida. Esta decisión permitió reducir significativamente el riesgo de introducir errores en el trabajo final. |
| **Conclusión** | Esta estrategia mitigó los problemas asociados a las alucinaciones de la IA, garantizando la fiabilidad del contenido técnico del TFG. Además, nos permitió aprovechar las ventajas de la IA para tareas generales sin comprometer la precisión de los apartados críticos relacionados con Kyber-KEM, el estándar FIPS 203 y la implementación práctica de algoritmos de criptografía postcuántica. El resultado fue un trabajo riguroso, equilibrado y adecuado para estudiantes avanzados en ingeniería informática. |

**Tabla 10.17:** Problema 17

## 10.1. Uso de la IA

En el contexto del proyecto de esta asignatura, la inteligencia artificial (IA) se ha convertido en una herramienta clave para optimizar procesos, mejorar la eficiencia del desarrollo y garantizar la calidad del producto final. A medida que las tecnologías avanzan, la integración de IA en flujos de trabajo no solo permite automatizar tareas repetitivas, sino que también facilita la toma de decisiones, la

depuración de errores y la mejora de la documentación. Sin embargo, es posible que su uso produzca lo que se conoce como «alucinaciones de la IA». Estas consisten en la generación de respuestas que, a pesar de parecer plausibles o técnicamente correctas, son erróneas, inconsistentes o completamente inventadas. Este fenómeno es especialmente problemático en ámbitos que requieren un alto grado de precisión, como es el caso de la criptografía postcuántica.

En el caso de Kyber-KEM y otros esquemas de NIST, detectamos errores recurrentes en las respuestas generadas por IA, tales como:

- Confusión entre conceptos básicos, como diferenciar entre PKE (Public Key Encryption) y KEM (Key Encapsulation Mechanism).
- Generación de ejemplos de código incompletos, sintácticamente incorrectos o directamente inejecutables.
- Referencias a bibliografía inexistente o inventada.
- Suposiciones incorrectas sobre parámetros, como el tamaño de las claves o los niveles de seguridad.

En temas altamente especializados como criptografía, los errores suelen amplificarse, ya que los modelos no han sido entrenados en profundidad sobre literatura técnica específica ni han sido validados para tareas críticas. A pesar del acceso a internet de muchas herramientas de IA esto no garantiza la precisión ni mucho menos la fiabilidad de la información proporcionada. De hecho, la mayoría de los modelos priorizan generar respuestas lingüísticamente coherentes por encima de garantizar su veracidad.

Por esta razón, como se ha comentado en la tabla 10.17, en nuestro trabajo hemos optamos por limitar el uso de IA a funciones auxiliares, evitando apoyarnos en ella para tareas donde un error pudiera comprometer la calidad del proyecto. En su lugar, priorizamos fuentes primarias como el estándar FIPS 203 [47] y documentación oficial del NIST.

En cuanto al código, el uso de herramientas de inteligencia artificial también presenta limitaciones significativas. Aunque la IA puede generar fragmentos de código útiles como punto de partida, a menudo produce implementaciones incompletas, ineficientes o directamente incorrectas. Entre los problemas detectados se incluyen errores sintácticos, uso inapropiado de bibliotecas o confusión entre versiones de algoritmos.

A pesar de ser conscientes de las limitaciones y posibles errores de estas herramientas, decidimos emplear la IA como una herramienta de apoyo puntual en ciertas fases del proyecto. Concretamente, utilizamos *prompts* cuidadosamente formulados que nos permitieran obtener borradores iniciales, ejemplos orientativos o esquemas preliminares, siempre sujetos a posterior validación y corrección manual. En este proceso, empleamos diversas herramientas de inteligencia artificial, como ChatGPT, Claude y Copilot. A continuación, se presenta una selección de los *prompts* utilizados durante el desarrollo del proyecto:

| **Prompt 1** | **Autor: Ignacio** |
|---|---|
| Herramienta Utilizada | Microsoft Copilot |
| Descripción del Uso | Se ha solicitado la búsqueda de bibliografía en la web de artículos relacionados con Kyber-KEM y criptografía basada en retículos. |
| Enlace a la consulta | copilot.microsoft.com/shares |
| Respuesta Obtenida | La IA ha proporcionado después de un segundo prompt tres artículos que pueden proporcionar una base sólida sobre la que empezar a trabajar y a buscar más bibliografía. |
| ¿Se utilizó la respuesta sin cambios? | Sí |
| Modificaciones realizadas | Se realizó un segundo intento puesto que la primera puesta no fue la esperada. |
| Tiempo estimado ahorrado | 1 hora |

**Tabla 10.18:** Prompt 1

| **Prompt 2** | **Autor: Ignacio** |
|---|---|
| Herramienta Utilizada | ChatGPT |
| Descripción del Uso | Se ha solicitado que ordene alfabéticamente la lista de acrónimos. |
| Enlace a la consulta | chatgpt.com/share |
| Respuesta Obtenida | La IA ha proporcionado lista de acrónimos correctamente ordenada; sin embargo, ha añadido en último lugar un acrónimo que no se le había proporcionado originalmente. |
| ¿Se utilizó la respuesta sin cambios? | Sí |
| Modificaciones realizadas | Después de indicarle el fallo de añadir un acrónimo que no se encontraba originalmente en la lista proporcionada en el prompt, lo eliminó. |
| Tiempo estimado ahorrado | 10 minutos |

**Tabla 10.19:** Prompt 2

| **Prompt 3** | **Autor: Ignacio** |
|---|---|
| Herramienta Utilizada | ChatGPT |
| Descripción del Uso | Se ha solicitado una explicación sencilla y didáctica de retículos para asimilar el concepto. |
| Enlace a la consulta | chatgpt.com/share |
| Respuesta Obtenida | La IA ha proporcionado un ejemplo sencillo del concepto de retículo y lo ha relacionado con dos ejemplos de LWE y de Kyber. |
| ¿Se utilizó la respuesta sin cambios? | Sí |
| Modificaciones realizadas | Intencionalmente en blanco. |
| Tiempo estimado ahorrado | 45 minutos. |

**Tabla 10.20:** Prompt 3

| **Prompt 4** | **Autor: Ignacio** |
|---|---|
| Herramienta Utilizada | ChatGPT |
| Descripción del Uso | Se ha solicitado la búsqueda de artículos sobre retículos basados en anillos. |
| Enlace a la consulta | copilot.microsoft.com/shares |
| Respuesta Obtenida | La IA ha proporcionado artículos pero no relacionados con lo que se solicitó. |
| ¿Se utilizó la respuesta sin cambios? | No |
| Modificaciones realizadas | Intencionalmente en blanco. |
| Tiempo estimado ahorrado | Intencionalmente en blanco. |

**Tabla 10.21:** Prompt 4

| **Prompt 5** | **Autor: Gabriel** |
|---|---|
| Herramienta Utilizada | Claude |
| Descripción del Uso | Se ha solicitado una segunda opinión sobre el algoritmo resolutor LWE (LWESolve) implementado en java para posibles mejoras. |
| Enlace a la consulta | claude.ai/share |
| Respuesta Obtenida | La IA proporcionó ciertos comentarios adecuados pero sin tener en cuenta parte del contexto. |
| ¿Se utilizó la respuesta sin cambios? | No, pero fue útil para reforzar el conocimiento sobre el método. |
| Modificaciones realizadas | Intencionalmente en blanco. |
| Tiempo estimado ahorrado | Intencionalmente en blanco. |

**Tabla 10.22:** Prompt 5

| **Prompt 6** | **Autor: Ignacio** |
|---|---|
| Herramienta Utilizada | ChatGPT |
| Descripción del Uso | Solicitud de herramientas y bibliotecas para trabajar con retículos en Python. |
| Enlace a la consulta | chatgpt.com/share |
| Respuesta Obtenida | La IA ha sugerido varias bibliotecas de Python aunque algunas requieren de C++. |
| ¿Se utilizó la respuesta sin cambios? | No |
| Modificaciones realizadas | Intencionalmente en blanco. |
| Tiempo estimado ahorrado | 1 hora |

**Tabla 10.23:** Prompt 6

| **Prompt 7** | **Autor: Gabriel** |
|---|---|
| Herramienta Utilizada | Claude |
| Descripción del Uso | Se ha solicitado la aclaración de un concepto relacionado con el número de ejecuciones necesarias para considerar una estimación empírica de la complejidad válida. |
| Enlace a la consulta | claude.ai/share |
| Respuesta Obtenida | La IA nos proporciono información que confirmaba lo que ya sabíamos. |
| ¿Se utilizó la respuesta sin cambios? | No, pero refuerza la confianza en el trabajo realizado. |
| Modificaciones realizadas | Intencionalmente en blanco. |
| Tiempo estimado ahorrado | Intencionalmente en blanco. |

**Tabla 10.24:** Prompt 7

| **Prompt 8** | **Autor: Gabriel** |
|---|---|
| Herramienta Utilizada | ChatGPT |
| Descripción del Uso | Se ha solicitado una aclaración sobre el algoritmo de Shor, en concreto que parte es la que lo convierte en un algoritmo cuántico. |
| Enlace a la consulta | chatgpt.com/share |
| Respuesta Obtenida | La IA nos proporciono información que confirmaba nuestros conocimientos, y además nos los amplió. |
| ¿Se utilizó la respuesta sin cambios? | No, al ser un tema de tanta importancia, se buscó en artículos científicos para ratificar la información. |
| Modificaciones realizadas | Intencionalmente en blanco. |
| Tiempo estimado ahorrado | Intencionalmente en blanco. |

**Tabla 10.25:** Prompt 8

| **Prompt 9** | **Autor: Gabriel** |
|---|---|
| Herramienta Utilizada | ChatGPT |
| Descripción del Uso | Se ha preguntado por actualidad y como avanzan las rondas del NIST. |
| Enlace a la consulta | chatgpt.com/share |
| Respuesta Obtenida | La IA nos comentó que había un nuevo estándar aceptado desde el 11 de marzo de 2025 basado en la dificultad del problema de decodificación de síndromes en códigos cuasi-cíclicos de paridad de baja densidad (QC-MDPC). |
| ¿Se utilizó la respuesta sin cambios? | No, pero gracias a esta consulta nos enteramos del nuevo estándar y pudimos incluirlo en el TFG a tiempo. |
| Modificaciones realizadas | Intencionalmente en blanco. |
| Tiempo estimado ahorrado | 30 min. |

**Tabla 10.26:** Prompt 9

| **Prompt 10** | **Autor: Gabriel** |
|---|---|
| Herramienta Utilizada | ChatGPT |
| Descripción del Uso | Se ha solicitado una comparativa entre el estandar ML-KEM y el futuro estándar HQC. |
| Enlace a la consulta | chatgpt.com/share |
| Respuesta Obtenida | La IA nos comentó que, si bien ambos son seguros y eficientes, Kyber tiene mayor aceptación y parece ser la opción que se usará ampliamente. |
| ¿Se utilizó la respuesta sin cambios? | No, se decidió no introducir en el trabajo una comparación entre estos estándares. |
| Modificaciones realizadas | Intencionalmente en blanco. |
| Tiempo estimado ahorrado | 20 min. |

**Tabla 10.27:** Prompt 10

# Bibliografía


[1] Santiago Fernández. Criptografía clásica, 2004. URL https://jefepipiolo.wordpress.com/wp-content/uploads/2016/09/9_criptografia_clasica.pdf.

[2] Universidad de Sevilla. Apuntes sobre la criptografía, 2023. Apuntes internos del curso de Criptografía, no disponibles en línea.

[3] Peter W. Shor and John Preskill. Simple proof of security of the bb84 quantum key distribution protocol, 2000. URL https://arxiv.org/pdf/quant-ph/0003004. Accessed: 2025-01-26.

[4] Inna Vasylkovska and Yurii Popov. Essential principles of quantum computing. In *Proceedings of the Scientific and Practical Conference on Technical Sciences*. National Technical University of Ukraine Ïgor Sikorsky Kyiv Polytechnic Institute", 2023. URL https://molodyivchenyi.ua/omp/index.php/conference/catalog/download/42/849/1762-1. Accessed: 2024-10-28.

[5] National Institute of Standards and Technology (NIST). nist.gov, 2025. URL https://www.nist.gov/. Accessed: 2025-04-23.

[6] European Telecommunications Standards Institute (ETSI). etsi.org, 2025. URL https://www.etsi.org/. Accessed: 2025-03-13.

[7] Centro Criptológico Nacional (CCN). ccn-cert.cni.es, 2025. URL https://www.ccn-cert.cni.es/es/. Accessed: 2025-04-21.

[8] National Institute of Standards and Technology (NIST). Nist round 4 submissions, 2024. URL https://csrc.nist.gov/projects/post-quantum-cryptography/round-4-submissions. Accessed: 2025-04-21.

[9] K. Townsend. Solving the quantum decryption 'harvest now, decrypt later' problem, February 2022. URL https://www.securityweek.com/solving-quantum-decryption-harvest-now-decrypt-later-problem/. Accessed: 2024-10.

[10] E. Yndurain. Y2q: el fin de la ciberseguridad actual. *El Mundo*, March 2023. URL https://www.elmundo.es/economia/actualidad-economica/2023/03/25/64146ca0fdddff95788b4597.html.

[11] Dave Taku. Setting the record straight on quantum computing and rsa encryption, October 2024. URL https://www.rsa.com/resources/blog/zero-trust/setting-the-record-straight-on-quantum-computing-and-rsa-encryption/.

[12] R. Holgado. Nuevo ordenador cuántico de ibm con 100.000 qubits. *20 minutos*, January 2024. URL https://www.20minutos.es/tecnologia/actualidad/nuevo-ordenador-cuantico-ibm-100000-qubits-5205508/. Último acceso: 10 2024.

[13] J. C. López. Ibm quiere liderar la computación cuántica a golpe de cúbit: prepara un ordenador de 10.000 cúbits para 2029. *Xataka*, June 2024. URL https://www.xataka.com/ordenadores/ibm-quiere-liderar-computacion-cuantica-a-golpe-cubit-prepara-ordenador-10-000-cu Último acceso: 10 2024.

[14] Radboud University, Ruhr Universität Bochum (RUB), IBM, and CWI. Crystals, cryptographic suite for algebraic lattices, 2024. URL https://pq-crystals.org/kyber/index.shtml.

[15] National Institute of Standards and Technology. nist.gov, 2024. URL https://csrc.nist.gov/news/2024/postquantum-cryptography-fips-approved.

[16] Wikipedia contributors. Paul benioff. https://es.wikipedia.org/wiki/Paul_Benioff, April 2024. Último acceso: 10 marzo 2025.

[17] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999. doi: 10.1137/S0036144598347011. URL https://www.jstor.org/stable/2653075.

[18] Centro Criptológico Nacional (CCN). Guía de seguridad de las tic ccn-stic 221, 2023. URL https://www.ccn-cert.cni.es/es/seguridad-al-dia/novedades-ccn-cert/12396-nueva-guia-ccn-stic-221-sobre-mecanismos-criptograficos-autorizados-por-el-html. Accessed: 2024-10.

[19] M. Tyson. La amenaza cuántica: la computación cuántica y la criptografía. https://www.computerworld.es/article/2120296/la-amenaza-cuantica-la-computacion-cuantica-y-la-criptografia.html, May 2022. Último acceso: diciembre 2024.

[20] María López López. Implementación del algoritmo de grover en los ordenadores cuánticos de ibm. https://rua.ua.es/dspace/handle/10045/107674, June 2020. Trabajo Fin de Grado, Universidad de Alicante, Facultad de Ciencias, Grado en Física. Director: Joaquín Fernández-Rossier. Fecha de defensa: 26 de junio de 2020. Último acceso: 11 marzo 2025.

[21] Utimaco. ¿qué es el algoritmo de grover? https://utimaco.com/es/servicio/base-de-conocimientos/criptografia-postcuantica/que-es-el-algoritmo-de-grover, 2024. Último acceso: octubre 2024.

[22] Microsoft. Conceptos: Algoritmo de grover. https://learn.microsoft.com/es-es/azure/quantum/concepts-grovers, January 2025. Último acceso: octubre 2024.

[23] Microsoft. Tutorial: Búsqueda de grover con qdk. https://learn.microsoft.com/es-es/azure/quantum/tutorial-qdk-grovers-search?tabs=tabid-copilot, January 2025. Último acceso: octubre 2024.

[24] C. C. J. P. Paz. Clase 10: Computación cuántica. http://users.df.uba.ar/paz/pag_comp_cuant/resumenes/clase10.pdf, September 2006. Último acceso: octubre 2024.

[25] Massachusetts Institute of Technology Center for Theoretical Physics. An analog analogue of a digital quantum computation. https://arxiv.org/pdf/quant-ph/9612026, December 1996. Último acceso: marzo 13, 2025.

[26] Nikil Dutt Sandip Ray Francesco Regazzoni Indranil Banerjee Rosario Cammarota Hamid Nejatollahi (University of California), Hamid Nejatollahi. Post-quantum lattice-based cryptography implementations: A survey. *ACM Computing Surveys*, 2024. URL https://www.researchgate.net/publication/330697634_Post-Quantum_Lattice-Based_Cryptography_Implementations_A_Survey. Último acceso: 01-2025.

[27] Robert Relyea. Post-quantum cryptography: Lattice-based cryptography. *Red Hat*, 2023. URL https://www.redhat.com/en/blog/post-quantum-cryptography-lattice-based-cryptography. Último acceso: 01-2025.

[28] Christopher Patton and Peter Schwabe. Prepping for post-quantum: A beginner's guide to lattice cryptography. *Cloudflare Blog*, 2025. URL https://blog.cloudflare.com/lattice-crypto-primer/. Último acceso: marzo de 2025.

[29] Ultimaco. What is lattice-based cryptography? URL https://utimaco.com/service/knowledge-base/post-quantum-cryptography/what-lattice-based-cryptography#:~:text=Lattice%2Dbased%20Cryptography%20explained&text=The%20name%20derives%20from%20the,not%20finite%20in%20any%20way.

[30] Daniele Micciancio and Oded Regev. Lattice-based cryptography. 2008. URL https://cims.nyu.edu/~regev/papers/pqc.pdf. Último acceso: 11-2024.

[31] Lily Hulatt and Gabriel Freitas. Criptografía basada en retículas, 2024. URL https://www.studysmarter.es/resumenes/matematicas/matematicas-discretas/criptografia-basada-en-reticulas/. Accessed: 2024-11-21.

[32] Equipo docente de la asignatura Estructuras Algebraicas de la titulación Grado en Matemáticas. Estructuras algebraicas. *Universidad de Sevilla*, 2024. URL https://asignatura.us.es/estalg/docs/latex.pdf. Último acceso: 12-2024.

[33] EcuRed. Anillo (álgebra). URL https://www.ecured.cu/Anillo_%28%C3%A1lgebra%29. Accessed: 2024-11-01.

[34] Academia Lab. (2025). Teoría del anillo. enciclopedia., . URL https://academia-lab.com/enciclopedia/teoria-del-anillo/. Accessed: 2024-11-01.

[35] Lily Hulatt and Gabriel Freitas. Anillos polinomiales, . URL https://www.studysmarter.es/resumenes/matematicas/matematicas-puras/anillos-polinomiales/. Accessed: 2024-11-21.

[36] Academia Lab. (2025). Anillo polinomial. enciclopedia, . URL https://academia-lab.com/enciclopedia/anillo-polinomial/. Accessed: 2024-11-01.

[37] Fernando Revilla. Anillo cociente. URL https://fernandorevilla.es/2014/04/12/anillo-cociente/. Accessed: 2024-11.

[38] Lily Hulatt and Gabriel Freitas. Teoría de módulos, . URL https://www.studysmarter.es/resumenes/matematicas/matematicas-puras/teoria-de-modulos/. Accessed: 2024-11.

[39] J. F. P. Hartwig and otros. Lattice-based cryptography: An overview. *IACR Cryptology ePrint Archive*, 2015:1046, 2015. URL https://eprint.iacr.org/2015/1046.pdf.

[40] Manuel Betancort Pérez. Estructura de retículos y su aplicación en la criptografía, May 2023. URL https://riull.ull.es/xmlui/bitstream/handle/915/33970/Estructura%20de%20reticulos%20y%20su%20aplicacion%20en%20la%20criptografia.pdf?isAllowed=y&sequence=1. Trabajo de Fin de Grado en Matemáticas. Dirigido por Irene Márquez Corbella (Universidad de La Laguna) y Luis José Santana Sánchez (Universidad de Valladolid).

[41] Daniele Micciancio. The shortest vector problem (svp), 2024. URL https://cseweb.ucsd.edu/~daniele/LatticeLinks/SVP.html. University of California, San Diego.

[42] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):1–40, 2009. doi: 10.1145/1552303.1552304. URL https://arxiv.org/abs/2401.03703.

[43] Yang Wang and Mingqiang Wang. Module-lwe versus ring-lwe, revisited. 2019. URL https://eprint.iacr.org/2019/930.pdf. Último acceso: 12-2024.

[44] Ring lwe cerrando el circulo el problema del anillo lwe en la criptografia moderna. *Faster Capital*, 2025. URL https://fastercapital.com/es/contenido/Ring-LWE--Cerrando-el-circulo--El-problema-del-anillo-LWE-en-la-criptografia-mode.html.

[45] Chris Peikert Vadim Lyubashevsky and Oded Regev. On ideal lattices and learning with errors over rings. 2012. URL https://eprint.iacr.org/2012/230.pdf. Último acceso: 11-2024.

[46] Miguel Toro Bonilla. adda_v5. https://github.com/migueltoro/adda_v5, 2025. Repositorio de código para la asignatura Análisis y Diseño de Datos y Algoritmos (ADDA).

[47] National Institute of Standards and Technology. Fips 203: Module-lattice-based key-encapsulation mechanism standard. Technical report, U.S. Department of Commerce, August 2024. URL https://csrc.nist.gov/pubs/fips/203/final.

[48] Arjen K. Lenstra, Hendrik W. Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[49] Claus-Peter Schnorr and Markus Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 1994.

[50] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-Quantum Cryptography*, pages 147–191. Springer, 2009.

[51] Nicolas Gama and Phong Q. Nguyen. Bkz-reduction algorithm and its application to lattice cryptanalysis. In *Advances in Cryptology–EUROCRYPT 2008*, pages 99–118. Springer, 2008.

[52] Ingrid Verbauwhede and Shih-Hong Saito. Countermeasures for side-channel attacks in cryptographic systems. *IEEE Transactions on Emerging Topics in Computing*, 5(4):553–563, 2017. doi: 10.1109/TETC.2017.2728420.

[53] Jonas Kaiser and Enrique Apon. Quantum attacks on lattice-based cryptography: The side-channel perspective. *Journal of Quantum Cryptography*, 1(2):1–19, 2017. doi: 10.1007/s40940-017-0023-9.

[54] Shanshan Yu, Wei Li, Wei Lou, and Y. Liu. Security against chosen ciphertext attacks: Theory and practice. *Journal of Computer Science and Technology*, 32(5): 847–865, 2017.

[55] Peter Eisert and Lars Wilke. Quantum algorithms for lattice problems. *Journal of Cryptology*, 31(4):703–750, 2018. doi: 10.1007/s00145-018-9293-x.

[56] Yilei Chen. Quantum algorithms for lattice problems. *IACR ePrint Archive*, 2024. URL https://eprint.iacr.org/2024/555.

[57] Nigel Smart. Implications of the proposed quantum attack on lwe, 2024. URL https://nigelsmart.github.io/LWE.html.

[58] Facebook Research. Lwe benchmarking, 2025. URL https://github.com/facebookresearch/LWE-benchmarking. Último acceso: mayo 2025.

[59] A. Karmakar I. Verbauwhede S. Saha y D. Mukhopadhyay S. Kundu, S. Chowdhury. Carry your fault: A fault propagation attack on side-channel protected lwe-based kem. 2024. URL https://eprint.iacr.org/2023/1674. Último acceso: 02 2025.

[60] Centro Criptológico Nacional (CCN). Ccn-tec 009 recomendaciones para una transición postcuántica segura, 2022. URL https://www.ccn.cni.es/eu/docman/documentos-publicos/boletines-pytec/495-ccn-tec-009-recomendaciones-transicion-postcuantica-segura/file. Accessed: 2025-03.

[61] National Institute of Standards and Technology. Migration to post-quantum cryptography, 2022. Disponible en: https://csrc.nist.gov/publications/detail/nistir/8105/final. [Último acceso: 02 2025].

[62] Sushmita Banerjee and Sunil Karforma. Challenges and opportunities of post-quantum cryptography for internet of things: A comprehensive survey. *Computer Networks*, 196:108244, 2021. doi: 10.1016/j.comnet.2021.108244.

[63] InfoDefensa. El ministerio de defensa participa en el desarrollo de la estrategia cuántica española. *InfoDefensa*, 2025. URL https://www.infodefensa.com/texto-diario/mostrar/5190498/ministerio-defensa-participa-desarrollo-estrategia-cuantica-espanola. Último acceso: 02-2025.

[64] Telefónica Tech. Telefónica tech y halotech integran criptografía postcuántica en dispositivos iot industriales, 2024. Disponible en: https://www.telefonicatech.com/noticias/telefonica-halotech-criptografia-postcuantica-iot. [Último acceso: 02 2025].

[65] S. Olivo. Nist descubre cuatro algoritmos criptográficos resistentes a la computación cuántica, jul 2022. Disponible en: https://www.escudodigital.com/ciberseguridad/nist-descubre-cuatro-algoritmos-criptograficos-resistentes-cuantica_52241_102.html. [Último acceso: 02 2025].

[66] D. King. 3 nuevas normas criptográficas postcuánticas del nist, oct 2024. GlobalSign, Disponible en: https://www.globalsign.com/es/blog/3-nuevas-normas-criptograficas-postcuanticas-del-nist. [Último acceso: 02 2025].

[67] Gorjan Alagic, Jessica Alperin-Sheriff, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Dustin Moody Miller, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Status report on the fourth round of the nist post-quantum cryptography standardization process. Technical Report NIST IR 8545, National Institute of Standards and Technology, March 2025. URL https://doi.org/10.6028/NIST.IR.8545.

[68] National Institute of Standards and Technology. Nist, ene 2025. Disponible en: https://csrc.nist.gov/News/2025/draft-sp-800-227-is-available-for-comment. [Último acceso: 02 2025].

[69] D. L. Jones. LattPy. https://lattpy.readthedocs.io/en/latest/index.html, 2022. [Último acceso: 11 2024].

[70] Dylan L. Jones. Lattpy: Simple and efficient python package for modeling bravais lattices. https://github.com/dylanljones/lattpy, 2022. University of Augsburg. Accessed: 2025-05-11.

[71] Giacomo Pope (PhD in Theoretical Physics). kyber-py: Implementación en python puro de ml-kem y crystals-kyber. Repositorio en GitHub, 2025. URL https://github.com/GiacomoPope/kyber-py. Último acceso: mayo de 2025.

[72] G4G4N. post-quantum-crypto-toolkit: Implementación en python de algoritmos criptográficos post-cuánticos basados en retículas. Repositorio en GitHub, 2025. URL https://github.com/G4G4N/post-quantum-cypto-toolkit. Último acceso: mayo de 2025.

# A. Anexo 1: Manual de instalación de LattPy

Para ilustrar el concepto de retículo o Lattice y trabajar con ellos, hemos utilizado la librería LattPy, desarrollada por Dylan Jones [69] [70]. Esta herramienta está diseñada para modelar y analizar estructuras de retículos en Python. Aunque no es una de las librería más conocidas, resulta útil en áreas como la física computacional y el estudio de sistemas complejos, facilitando la representación y manipulación de redes cristalinas y otras estructuras relacionadas. Hemos optado por trabajar con LattPy utilizando la herramienta Jupyter Notebook, ampliamente recomendada durante el grado.

Para instalar Jupyter Notebook debemos, en primer lugar, descargar e instalar Miniconda, una distribución ligera de Python que incluye el administrador de entornos y paquetes Conda. Para ello nos dirigimos al enlace: https://docs.conda.io/projects/conda/en/stable/ y descargamos la versión que mejor se acomode a nuestro sistema operativo. Una vez instalado, iniciaremos el ejecutable AnacondaPrompt o una ventana del terminal y crearemos un entorno específico para trabajar con las librerías deseadas mediante el comando:

```
conda create -n <env-name>
```

A continuación, para cambiar del entorno base que viene por defecto al que acabamos de crear usaremos:

```
conda activate <env-name>
```

Una vez dentro del entorno, descargaremos las dependencias necesarias. En este caso instalaremos las librerías matplotlib, lattpy, numpy y scikit-learn:

```
conda install matplotlib
conda install numpy
pip install lattpy
pip install scikit-learn
```

Teniendo todas las dependencias necesarias preparadas, instalamos jupyter notebook:

```
pip install notebook
```

Y, por último, para ejecutarlo, usaremos el comando:

```
jupyter notebook
```

# B. Anexo 2: Cuaderno sobre teoría de retículos

En este notebook se explican los conceptos básicos relacionados con los retículos, sus propiedades fundamentales, cómo acceder a ellas y la generación de retículos, todo ello acompañado de ejemplos prácticos, tanto analíticos como gráficos. Cabe resaltar que tanto las funciones generadas como el código aparte contienen comentarios explicativos paso a paso, con el objetivo de generar un cuadernillo autocontenido que sea didáctico y fácil de seguir. Este documento se puede encontrar en: https://github.com/PQC-standards/Fundamentos-Teoricos o pinchando aquí.

También se puede desplegar online siguiendo este enlcace: https://mybinder.org/v2/gh/PQC-standards/Fundamentos-Teoricos/main

# C. Anexo 3: Cuaderno sobre teoría de anillos

En este notebook se explican los conceptos básicos relacionados con los anillos algebraicos, sus propiedades fundamentales, los tipos de anillos más importantes para comprender los conceptos relacionados con la criptografía postcuántica y la implementación de estos, todo ello acompañado de ejemplos prácticos que facilitan el aprendizaje. Cabe resaltar que tanto las funciones generadas como el código aparte contienen comentarios explicativos paso a paso, con el objetivo de generar un cuadernillo autocontenido que sea didáctico y fácil de seguir. Este documento se puede encontrar pinchando aquí o en: https://github.com/PQC-standards/Fundamentos-Teoricos

También se puede desplegar online siguiendo este enlcace: https://mybinder.org/v2/gh/PQC-standards/Fundamentos-Teoricos/main

# D. Anexo 4: Cuaderno sobre problemas relacionados

En este notebook se explican los principales problemas que conforman la base de la criptografía reticular, se propone una definición formal, se implementan mediante funciones en Python exhaustivamente comentadas, con el objetivo de que se entienda cada paso que realiza el algoritmo, y se proporcionan ejemplos, ya sean representaciones gráficas en dos y tres dimensiones o soluciones analíticas en distinto número de dimensiones. Cabe destacar que todos los problemas están sujetos a la experimentación del usuario con solo cambiar las variables iniciales. Este documento se puede encontrar pinchando aquí o en: https://github.com/PQC-standards/Problemas-Relacionados

También se puede desplegar online siguiendo este enlcace: https://mybinder.org/v2/gh/PQC-standards/Problemas-Relacionados/main

# E. Anexo 5: Cuaderno sobre problemas derivados de LWE

En este notebook se explican los principales problemas derivados de LWE que conforman la base del estándar Kyber, se propone una definición formal, se implementan mediante funciones en Python exhaustivamente comentadas, con el objetivo de que se entienda cada paso que realiza el algoritmo, y se proporcionan ejemplos, ya sean representaciones gráficas en dos y tres dimensiones o soluciones analíticas en distinto número de dimensiones. Cabe destacar que todos los problemas están sujetos a la experimentación del usuario con solo cambiar las variables iniciales. Este documento se puede encontrar en: https://github.com/PQC-standards/Problemas-Relacionados o pinchando aquí.

También se puede desplegar online siguiendo este enlcace: https://mybinder.org/v2/gh/PQC-standards/Problemas-Relacionados/main

# F. Anexo 6: Cuaderno sobre Algoritmos PKE

En este notebook se explican los tres algoritmos que conforman el PKE (cifrado de clave pública) del estándar Kyber: Generación de claves, cifrado y descifrado. Por cada algoritmo se explicará de forma clara y precisa los pasos a seguir para conseguir que el esquema de cifrado de clave pública sea robusto y resistente a ataques. Además se proporcionará una implementación repleta de comentarios explicativos para poder facilitar su comprensión, y por último se mostrará un ejemplo de uso completo del proceso que seguiría el PKE en una situación habitual. Este documento se puede encontrar pinchando aquí o en: https://github.com/PQC-standards/Algoritmos-Kyber-KEM

También se puede desplegar online siguiendo este enlcace: https://mybinder.org/v2/gh/PQC-standards/Algoritmos-Kyber-KEM/main

# G. Anexo 7: Cuaderno sobre Algoritmos KEM

En este notebook se explican los tres algoritmos que conforman el KEM (mecanismo de intercambio de claves) del estándar Kyber: Generación de claves, encapsulado y desencapsulado. Por cada algoritmo se explicará de forma clara y precisa los pasos a seguir para conseguir que el mecanismo sea robusto y resistente a ataques. Además, se proporcionará una implementación repleta de comentarios explicativos para poder facilitar su comprensión, y por último se mostrará un ejemplo de uso completo del proceso que seguiría el KEM en una situación habitual. Este documento se puede encontrar pinchando aquí o en: https://github.com/PQC-standards/Algoritmos-Kyber-KEM

También se puede desplegar online siguiendo este enlcace: https://mybinder.org/v2/gh/PQC-standards/Algoritmos-Kyber-KEM/main

Como inspiración se han hecho uso de varios repositorios de GitHub, destacando particularmente el trabajo de Giacomo Pope (Doctor en Física Teórica) [71], cuya implementación en Python del esquema Kyber ha servido como referencia técnica para comprender y replicar los pasos del KEM de forma didáctica. Por otra parte, mencionar el repositorio de G4G4N [72], que ofrece una implementación en Python de algoritmos criptográficos postcuánticos basados en retículos, incluyendo Kyber, proporcionando herramientas útiles para la investigación y experimentación con criptografía postcuántica.

# H. Anexo 8: Cuaderno sobre simulación de ataques

Este notebook contiene cuatro implementaciones de ataques con el objetivo de simular los pasos que un atacante seguiría para romper la seguridad de un esquema de cifrado Kyber-KEM, donde por cada ataque se explicará en que consiste y se proporcionará una implementación detallada y comentada, así como la efectividad de cada uno y si se ha logrado o no penetrar la seguridad del estándar. Este documento se puede encontrar en: https://github.com/PQC-standards/Criptoanalisis o pinchando aquí.

También se puede desplegar online siguiendo este enlcace: https://mybinder.org/v2/gh/PQC-standards/Criptoanalisis/HEAD

# I. Anexo 9: Análisis empírico de la complejidad

En este anexo, se proporciona un proyecto java donde se evalúa la complejidad empírica de los problemas fundamentales relacionados con retículos, en concreto: CVP, SVP y LWE. Esta implementación se puede encontrar en: https://github.com/PQC-standards/Complejidad-Problemas-Relacionados o pinchando aquí. Para trabajar con este repositorio siga las instrucciones indicadas en el README.md del repositorio. Cabe resaltar que se ha utilizado el repositorio de Don Miguel Toro Bonilla [46] para la automatización del análisis de la complejidad.