Universidad de Oviedo

Facultad de Ciencias

Trabajo Fin de Grado en Matemáticas

"BIKE, Classic McEliece y HQC: esquemas de códigos correctores en la estandarización post-cuántica del NIST"

Realizado por:

David Ambrosi Jalón

Tutelado por:

Dr. Ignacio Fernández Rúa

Oviedo, Junio, 2024

Agradecimientos

A mi tutor Iñaki, por enseñarme la interesante, al menos para mí, área de la criptografía, con usos prácticos, indispensables y constantes de conocimientos comúnmente más abstractos y teóricos de las matemáticas. Y, sobre todo, por haber logrado sacar tiempo de su muy ajetreada agenda para poder resolver de forma clara mis dudas y ayudarme en este Trabajo de Fin de Grado, con el que cierro mi ciclo como estudiante de Matemáticas con un buen sabor de boca.

A mi compañero de piso, por la buena convivencia en esta última etapa como estudiante y, junto a mis amistades de siempre, por las risas y conversaciones cuando más necesitaba un respiro.

Y finalmente a mi familia: a mi padre, a mi hermano y, en especial, a mi madre, por celebrar mis logros en las buenas, calmarme en las malas y apoyarme en todo momento. Espero que estéis tan orgullosos de mí como yo de vosotros.

Índice

1. Introducción

Desde la antigüedad, los seres humanos hemos buscado formas de transmitir información confidencial entre nosotros sin que caiga en manos ajenas, empleando para ello la criptografía. Ésta ha evolucionado desde pequeños e ingeniosos trucos con las letras de un mensaje, como la escítala de los espartanos o el cifrado César de los romanos, hasta sistemas complejos basados en el álgebra y la teoría de números en la actualidad.

La forma más fácil de establecer una comunicación segura es fijando una única clave secreta entre las dos partes en persona, para después poder transmitir información cifrada desde la distancia, siendo ésta la base de la criptografía de clave simétrica o privada. Sin embargo, tras la aparición de Internet, buscamos constantemente acceder y enviar información a grandes distancias. Para ello, surge la criptografía de clave asimétrica o pública (PK), con una clave pública para encriptar mensajes y otra secreta o privada para desencriptarlos, otorgando seguridad sin necesidad de un contacto previo cara a cara para fijar una única clave simétrica de forma segura. En este último caso, un atacante también puede buscar suplantar nuestra identidad, surgiendo las firmas digitales para evitarlo.

En la actual era de la información, su protección mediante la criptografía de clave pública se ha vuelto un pilar fundamental en nuestras vidas, siendo una componente indispensable en la comunicación digital. Estamos empleándola constantemente cada vez que accedemos a Internet y, en última instancia, sustenta nuestra economía e incluso seguridad.

Continuamente se desarrollan nuevos ataques que buscan romper los criptosistemas actuales para poder acceder a información restringida, basándose la criptografía en un incesante tira y afloja entre nuevas mejoras y debilidades. Estos cambios pueden deberse a descubrimientos de nuevos ataques con el hardware actual o al desarrollo de nuevos ordenadores capaces de resolver problemas antes inabordables. En esta última línea hay una mejora constante del poder de computación de los ordenadores actuales siguiendo la famosa ley de Moore; sin embargo, recientemente se está produciendo un gran cambio de paradigma, debido a la posible amenaza del desarrollo de ordenadores cuánticos, los cuales podrían dejar obsoleta la criptografía pública actual.

Otro problema es la posibilidad de almacenar mensajes cifrados ahora y poder descifrarlos cuando estos ordenadores cuánticos estén disponibles. Para evitarlo surge la idea de la "criptografía postcuántica" o PQC (*Post-Quantum Cryptography*), pretendiendo

desarrollar criptosistemas implementables con ordenadores clásicos actuales y que sean seguros frente a ordenadores tanto clásicos como cuánticos.

En este trabajo trataremos sobre los tres finalistas de la cuarta ronda del NIST para la estandarización de PQC (ver *Sección 1.2* más abajo): Classic McEliece, BIKE y HQC, siendo todos ellos criptosistemas de clave pública para encriptar e intercambiar claves. Se introducirán las bases teóricas necesarias y se realizará un enfoque autocontenido, buscando que cualquier lector con una cierta base matemática sea capaz de vislumbrar el funcionamiento y la implementación de cada uno de estos tres criptosistemas.

1.1. Amenaza de los ordenadores cuánticos

Una parte fundamental de los protocolos de comunicación de clave pública actuales se sustentan sobre tres procesos criptográficos: encriptación, firmas digitales e intercambio de claves. Sus implementaciones suelen basarse en el intercambio de claves de Diffie-Hellman [1], el criptosistema RSA (Rivest-Shamir-Adleman) [2] y criptosistemas de curvas elípticas [3, 4], cuya seguridad se basa en la dificultad de una serie de problemas teóricos, como el de la factorización en números primos y el del logaritmo discreto [5].

Sin embargo, Peter Shor en 1994 [6] demostró que dichos problemas, difíciles para ordenadores clásicos, podían ser resueltos en tiempo polinomial con ordenadores cuánticos suficientemente potentes [7]. Ésto supondría una gran amenaza para toda la infraestructura de comunicación digital, ya que destruye los cimientos de la criptografía de clave pública actual sobre la que se sustenta.

Debido a este descubrimiento sobre las posibles mejoras de computación con ordenadores cuánticos para determinados problemas, se impulsó enormemente su desarrollo, tanto en la fabricación de dichos dispositivos como en la teoría de algoritmos cuánticos como el de Shor [5].

Así, en 1996 se desarrolló el algoritmo de búsqueda de Grover [8, 9], el cual otorga una aceleración cuadrática en problemas de búsqueda no estructurados [5]. Esta mejora, aunque no deje obsoletas ciertas tecnologías criptográficas, a diferencia del algoritmo de Shor, sí exige el uso de claves más largas, incluso dentro de la criptografía de clave privada o simétrica, como en AES (*Advanced Encryption Standard*) [10].

Tabla 1: Impacto de ordenadores cuánticos en algoritmos criptográficos comunes [5].

Algoritmo	Tipo	Propósito	Impacto
AES	Privada	Encriptación	Claves más grandes
SHA-2, SHA-3	–	Funciones *hash*	Mayor tamaño de generación
RSA	Pública	Firma e intercambio de claves	No seguro
ECDSA, ECDH (Curvas elípticas)	Pública	Firma e intercambio de claves	No seguro
DSA (Cuerpos finitos)	Pública	Firma e intercambio de claves	No seguro

Sólo estudiaremos criptosistemas PK para encriptar e intercambiar claves, pero se muestran otras implementaciones como funciones resumen (*hash*) y firmas digitales, por dar un mayor contexto.

La *Tabla 1* recoge cuál sería el impacto de ordenadores cuánticos suficientemente potentes sobre algunos algoritmos criptográficos comunes. Se aprecia cómo la criptografía de clave pública actual, a diferencia de la privada, no puede ser reforzada aumentando tamaños de clave, sino que necesita nuevos criptosistemas que resistan los ataques con ordenadores cuánticos, basados en otros problemas difíciles de resolver para éstos.

A día de hoy se está invirtiendo un gran esfuerzo en el desarrollo de ordenadores cuánticos, debido a las posibles grandes mejoras en determinadas tareas respecto a los clásicos. Sin embargo, sigue siendo un gran desafío el aumento de los "qubits físicos" de dichos sistemas, y más todavía el número de éstos que sean funcionales o "qubits lógicos", ya que gran parte de ellos necesitan ser empleados íntegramente en un proceso de corrección de errores cuántica [11].

Aunque teóricamente esta tarea parece factible, es extremadamente complicado desde un punto de vista práctico, debido a la gran sensibilidad e inestabilidad de los qubits físicos, aumentando el ruido producido entre éstos con su número.

Desconocemos cuándo llegarán los temibles ordenadores cuánticos capaces de romper toda la criptografía de clave pública (PK) actual, pero no parece un futuro demasiado lejano y es conveniente buscar soluciones lo antes posible. Algunos motivos son:

1. El desarrollo de nuevos criptosistemas para PQC puede ser arduo y extenso.

2. Una vez desarrollados, puede suponer un gran esfuerzo la migración segura a éstos desde los criptosistemas ampliamente usados actualmente.

3. Cuanto antes se produzca dicha migración, menor cantidad de información encriptada podrá ser recogida y guardada para ser desencriptada con ordenadores cuánticos a gran escala más adelante.

En definitiva, necesitamos realizar la migración a un nuevo ecosistema seguro frente a ordenadores cuánticos a gran escala antes de que éstos se desarrollen [12]. Además, como es costumbre para establecer una seguridad en la comunicación a nivel global, no basta con desarrollar criptosistemas de PQC, sino que éstos deben ser estandarizados.

1.2. Estandarización de criptografía postcuántica por el NIST

Con el objetivo de establecer las bases de una nueva criptografía de clave pública, a finales de 2016 el NIST (*National Institute of Standards and Technology*) comenzó un proceso de solicitud, evaluación y estandarización de algoritmos de criptografía pública resistentes frente a ordenadores tanto clásicos como cuánticos, denominado "*Post-Quantum Cryptography Standarization*" [13].

En la primera ronda en 2017 se aceptaron un total de 69 candidatos, conteniendo tanto algoritmos de firma digital como de establecimiento de claves. Algunas de las familias desarrolladas para poseer seguridad cuántica fueron [5]:

- Criptografía basada en retículos: Se basan en el problema SVP (*Shortest Vector Problem*), que consiste en encontrar el menor vector no nulo dentro de un retículo, siendo un problema NP-difícil para el cual no se conoce ningún algoritmo cuántico [14]. Son muy rápidos, cuentan con claves de pequeño tamaño y presentan una seguridad homogénea (de caso medio), a través de una reducción a un caso difícil de SVP.

- Criptografía basada en códigos correctores: A lo largo del trabajo comentaremos sus orígenes y fundamentos. Por ahora basta con tener en mente que, aunque son bastante rápidos, la mayoría sufren el hecho de tener unas claves demasiado grandes. Además, sus orígenes son los más antiguos, con un mayor tiempo sin ser vulnerables frente a ataques, atribuyendo una gran confianza en su seguridad.

- Criptografía de polinomios multivariables: Se basan en la dificultad de resolver sistemas de dichos polinomios sobre cuerpos finitos.

- Firmas basadas en funciones *hash*: Presentan una seguridad muy consolidada, pero tienen como pegas la necesidad de un seguimiento del número de firmas que se va generando y un límite de firmas que pueden generar, aunque este último problema puede solucionarse aumentando al tamaño de las mismas.

- Otras: Evaluación de isogenias en curvas elípticas supersingulares, problemas en conjuntos de trenzas no abelianos, ...

Tras un arduo proceso de mejora, escrutinio y filtrado, en 2022 se anunciaron los ganadores de la tercera ronda del NIST sobre estandarización PQC. En el caso que nos incumbe, de algoritmos para el intercambio de claves, CRYTALS-Kyber [15] se alzó como el único vencedor, basándose en problemas difíciles sobre retículos modulares.

La seguridad de la criptografía de clave pública, tanto la clásica más asentada como la nueva postcuántica en desarrollo, se basa sobre diversos problemas que se asumen difíciles, pero que pueden dejar de serlo ante la aparición de un nuevo algoritmo o esquema de ataque, tanto clásico como cuántico. De este modo, un criptosistema se considera seguro hasta que se demuestre lo contrario.

Debido a ésto, a pesar de que CRYTALS-Kyber ya ha sido aceptado para su estandarización, es preferible tener algún as bajo la manga en vez de apostarlo todo a una única carta. Por ello, a día de hoy se está llevando a cabo una cuarta ronda para establecer alternativas, previniendo el hipotético caso en el que aparezca un nuevo ataque que rompa el criptosistema seleccionado.

Los candidatos propuestos con este fin fueron BIKE [16], Classic McEliece [17], HQC [18] y SIKE [19], el cual se basaba en isogenias y dejó de ser seguro frente a un nuevo ataque clásico de recuperación de claves [20]. De esta forma, las últimas tres alternativas en la competición pertenecen a la familia de criptografía basada en códigos correctores, siendo su estudio el objetivo del resto del trabajo.

1.3. Notación

Antes de entrar en materia, vamos a fijar una notación que se mantendrá a lo largo de todo el trabajo, salvo que se indique lo contrario. Las letras en minúscula se reservarán para variables simples o polinomios (*e.g.* $m \in \mathbb{Z}^+, f(x) \in \mathbb{Z}[x]$), añadiendo una flecha encima para indicar vectores o tuplas (e.g. $\vec{x} \in \mathbb{Z}^n$), mientras que las mayúsculas se emplearán para hablar de matrices o conjuntos de elementos (*e.g.* $A \in \mathbb{Z}^{2\times 2}, B = \{a, b, c\}$).

Además, vamos a fijar otra serie de símbolos de interés, que pueden ir apareciendo con cierta frecuencia. Para referirnos al conjunto de matrices cuadradas $n \times n$ inversibles, emplearemos $GL_n(Y)$, donde Y indica el anillo al que pertenecen los elementos que configuran dicha matriz. Del mismo modo, $x \leftarrow Y$ se empleará para indicar que $x \in Y$ ha sido seleccionado de forma uniformemente aleatoria de un conjunto Y. El conjunto de matrices de permutación de tamaño $n \times n$ se denotará como $\mathcal{P}_n$. Por último, $\mathbb{F}_q$ indicará el grupo finito de q elementos, siendo q la potencia de un primo.

2. Criptografía de clave pública basada en códigos correctores

La criptografía de clave pública suele basarse en problemas fáciles de plantear pero difíciles de resolver de forma eficiente. Ésto se articula mediante el uso de funciones de una vía, fáciles de calcular pero difíciles de invertir. Si la inversión se vuelve fácil al conocer una cierta información extra, éstas reciben el nombre de funciones de una vía con trampa, más conocidas en inglés como *trapdoor one-way functions* [21].

Como hemos visto anteriormente, los ordenadores cuánticos, capaces de ejecutar los algoritmos de Shor y Grover, ponen en jaque toda la criptografía de clave pública empleada en la actualidad. Algunas de las soluciones propuestas al NIST son los denominados criptosistemas basados en códigos correctores. Este tipo de esquemas fue propuesto por primera vez por Robert J. McEliece a finales de los años 80 [22].

A continuación se realiza una pequeña introducción a los códigos correctores de errores y se comenta el criptosistema original de McEliece, donde se aprecia fácilmente la idea detrás de todos los criptosistemas basados en códigos correctores.

2.1. Conceptos básicos de códigos correctores de errores [23]

La teoría de códigos correctores de errores o ECC (*Error Correcting Codes*) surgió para proteger la información transmitida frente a alteraciones aleatorias y accidentales producidas por un canal con ruido. Ésto resulta indispensable para comunicaciones a largas distancias, como son las típicas en nuestro día a día a través de Internet e incluso las realizadas con sondas espaciales.

Otro uso extremadamente útil de los ECC es para almacenar datos protegidos frente a perturbaciones en su sistema, presentes en las memorias de los ordenadores y también a simple vista en los códigos de barras o los códigos QR.

Añadiendo cierta información redundante antes de su emisión, los ECC permiten detectar y corregir los errores que aparecen, pudiendo recuperar el mensaje original.

Sean dos números enteros $k, n \in \mathbb{Z}$ tales que $1 \leq k \leq n$. Entonces, diremos que $\mathcal{C}$ es un **$[n, k]$ código lineal** sobre $\mathbb{F}_q$ si es un subespacio vectorial $k-$dimensional de $\mathbb{F}_q^n$. El parámetro n representa la longitud del código $\mathcal{C}$ y los elementos del mismo se denominan "palabras-código", mientras que los de $\mathbb{F}_q^n$ son simplemente "palabras" a secas.

Las tres propuestas de criptosistemas tratadas en este trabajo se basan en la **métrica de Hamming** para medir la distancia entre palabras, la cual se define como sigue:

- El **peso de Hamming** de una palabra $\vec{x} \in \mathbb{F}_q^n$ viene dado por el tamaño de su soporte, es decir, el número de componentes no nulas:

$$wt_H(\vec{x}) = |\{i \in \{1, \ldots, n\} \mid x_i \neq 0\}|$$

- La **distancia de Hamming** entre dos palabras $\vec{x}, \vec{y} \in \mathbb{F}_q^n$ viene dada por el número de componentes no coincidentes:

$$d_H(\vec{x}, \vec{y}) = |\{i \in \{1, \ldots, n\} \mid x_i \neq y_i\}| = wt_H(\vec{x} - \vec{y})$$

Una vez fijada nuestra métrica, podemos determinar cuál es la distancia mínima de un código, definida como la distancia más pequeña entre dos palabras-código suyas distintas. De este modo, la **distancia mínima de Hamming de un código lineal** $\mathcal{C} \subset \mathbb{F}_q^n$ viene dada por:

$$d_H(\mathcal{C}) = min\{d_H(\vec{x}, \vec{y}) \mid \vec{x}, \vec{y} \in \mathcal{C},\ \vec{x} \neq \vec{y}\} = min\{wt_H(\vec{x}) \mid \vec{x} \in \mathcal{C},\ \vec{x} \neq \vec{0}\}$$

Ésta es muy importante en la caracterización de un código $\mathcal{C}$, ya que nos indica la cota superior t de los errores que es capaz de corregir, es decir, el número máximo de errores que pueden ser detectados y corregidos de forma correcta por el receptor del mensaje, siendo $t \leq \lfloor \frac{d_H(\mathcal{C})-1}{2} \rfloor$.

Además, teniendo en cuenta la desigualdad de Singleton [24]: "para un $[n, k]$ código lineal $\mathcal{C}$ sobre $\mathbb{F}_q$ se cumple que $d_H(\mathcal{C}) \leq n - k + 1$", obtenemos que $t \leq \lfloor \frac{n-k}{2} \rfloor$, donde $r := n - k$ es la redundancia añadida. Si un código alcanza la desigualdad de Singleton, se dice que es un código MDS (*Maximum Distance Separable*) [23].

Los códigos lineales resultan muy prácticos, ya que pueden ser representados fácilmente mediante matrices. Sea $\mathcal{C}$ un $[n, k]$ código lineal sobre $\mathbb{F}_q$:

- $\mathcal{C}$ como imagen de G: Una matriz $G \in \mathbb{F}_q^{k \times n}$ se denomina una **matriz generatriz** de $\mathcal{C}$ si $\mathcal{C} = \{\vec{x}G \mid \vec{x} \in \mathbb{F}_q^k\}$, es decir, si las filas de G forman una base suya.
- $\mathcal{C}$ como núcleo o kernel de H: Una matriz $H \in \mathbb{F}_q^{(n-k) \times n}$ se denomina **matriz de control de paridad** de $\mathcal{C}$ si $\mathcal{C} = \{\vec{y} \in \mathbb{F}_q^n \mid H\vec{y}^\top = \vec{0}\}$. Para cualquier palabra $\vec{y} \in \mathbb{F}_q^n$ se define su síndrome como $\vec{y}H^\top$, de forma que es una palabra-código si y sólo si su síndrome es nulo.

2.2. Criptosistema original de McEliece

Vamos a comenzar explicando el criptosistema propuesto por McEliece en 1978 [22], ya que nos resulta muy conveniente tanto desde un punto de vista histórico como ilustrativo. Es el primer criptosistema de clave pública que se basa en códigos correctores de errores, siendo bastante sencillo y sirviendo como inspiración para esquemas posteriores, como el de Niederreiter [25], sobre el que hablaremos más adelante en la *Sección 3.1.*

Es un criptosistema de clave pública, cuya función de una vía se basa en la existencia de un algoritmo que permite descodificar de forma eficiente un código Goppa cualquiera conocido, mientras que descodificar un código lineal genérico desconocido resulta un problema difícil, incluso empleando hipotéticos ordenadores cuánticos.

Primero describiremos brevemente los códigos Goppa, en los que se basa el criptosistema de McEliece. Después comentaremos su implementación y su situación actual.

2.2.1. Códigos Goppa [23]

Los códigos Goppa fueron introducidos por V. D. Goppa en los años 70 [26], con una estructura algebraica dada por un anillo cociente de polinomios.

Sea $m \in \mathbb{Z}^+$, tomamos $n \leq q^m$ y el cuerpo finito $\mathbb{F}_{q^m}$. Si seleccionamos un polinomio cualquiera $g(x)$ sobre dicho cuerpo, podemos construir el anillo cociente de $\mathbb{F}_{q^m}[x]$ por el ideal generado por $g(x)$ como $Q_m = {}^{\mathbb{F}_{q^m}[x]}/_{< g(x) >}$, pudiendo identificarse como el espacio vectorial de polinomios $p \in \mathbb{F}_{q^m}[x]$ con $deg(p) < deg(g)$ y multiplicación módulo $g(x)$.

Sea $\alpha \in \mathbb{F}_{q^m}$ tal que $g(\alpha) \neq 0$. Entonces el polinomio $(x-\alpha)$ es inversible en el anillo cociente Q_m descrito anteriormente y su inverso viene dado por:

$$(x-\alpha)^{-1} = -\frac{1}{g(\alpha)}\frac{g(x)-g(\alpha)}{x-\alpha}$$

Sea un vector $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_{q^m}^n$ con $\alpha_i \neq \alpha_j \ \forall \ i,j \in \{1,\ldots,n\}$ con $i \neq j$ y tal que $g(\alpha_i) \neq 0 \ \forall \ i \in \{1,\ldots,n\}$. Se define un **código de Goppa $q-$ario** como:

$$GC_q(\vec{\alpha}, g) = \left\{ \vec{c} \in \mathbb{F}_q^n \;\middle|\; \sum_{i=1}^{n} \frac{c_i}{x-\alpha_i} = 0 \text{ en } Q_m \right\}$$

Estos códigos son lineales (Teorema 1 de [27]) y, como vamos a demostrar, presentan una distancia mínima $d_H(GC_q(\vec{\alpha}, g)) \geq deg(g)+1$ y una dimensión $k \geq n - m\ deg(g)$ [23].

Comencemos con su distancia mínima. Según el Teorema 2 de [27], el código de Goppa $q-$ario asociado a un polinomio $g(x)$ con $deg(g) = 2\delta$ es capaz de corregir δ errores, de modo que, si consideramos también la cota de errores t descrita anteriormente en la *Sección 2.1*, tenemos que

$$\frac{deg(g)}{2} = \delta \leq t \leq \lfloor \frac{d_H(GC_q(\vec{\alpha}, g)) - 1}{2} \rfloor \leq \frac{d_H(GC_q(\vec{\alpha}, g)) - 1}{2}$$

de donde se obtiene que $d_H(GC_q(\vec{\alpha}, g)) \geq deg(g) + 1$. $\square$

Para obtener la cota mínima de su dimensión, vamos a seguir un planteamiento similar a [28]. Como hemos visto anteriormente, por construcción, $(x-\alpha_i)$ tiene un inverso en Q_m, el cual podemos escribir como:

$$(x-\alpha_i)^{-1} \equiv g(\alpha_i)^{-1} f_i(x) \equiv \sum_{j=0}^{deg(g)-1} g(\alpha_i)^{-1} f_{i,j} x^j \quad mod\ g(x)$$

Si introducimos esta relación en la condición de las palabras-código $\vec{c} \in \mathbb{F}_q^n$, nos queda que, sobre Q_m:

$$\sum_{i=1}^{n} c_i \left(\sum_{j=0}^{deg(g)-1} g(\alpha_i)^{-1} f_{i,j} x^j \right) = \sum_{j=0}^{deg(g)-1} \left(\sum_{i=1}^{n} g(\alpha_i)^{-1} c_i f_{i,j} \right) x^j = 0$$

de tal forma que, para cada $0 \leq j \leq deg(g) - 1$ se debe cumplir que

$$\sum_{i=1}^{n} g(\alpha_i)^{-1} c_i f_{i,j} = 0$$

teniendo $deg(g)$ condiciones sobre $\mathbb{F}_{q^m}$, ya que $c_i \in \mathbb{F}_q$ pero tanto $g(\alpha_i)^{-1}$ como $f_{i,j}$ pertenecen a $\mathbb{F}_{q^m}$. Podemos escribir los elementos de $\mathbb{F}_{q^m}$ como elementos de $\mathbb{F}_q^m$, fijando previamente una base ordenada $\{b_1, \ldots, b_q\}$ de los elementos de $\mathbb{F}_q$, de modo que cada una de estas condiciones sobre $\mathbb{F}_{q^m}$ supone m condiciones sobre $\mathbb{F}_q$, teniendo un total de $m\ deg(g)$ condiciones sobre $\mathbb{F}_q$.

Dicho número de condiciones sobre las palabras-código, $m\ deg(g)$, y la longitud de las mismas, n, nos indican que el código presenta una matriz de control de paridad $\tilde{H}$ de tamaño $m\ deg(g) \times n$ sobre $\mathbb{F}_q$. Ésto nos dice que $n-k = m\ deg(g)$, donde k es la dimensión del código sobre $\mathbb{F}_q$ (el alfabeto de las palabras-código), de modo que $k = n - m\ deg(g)$. Sin embargo, ésta es una cota inferior, ya que algunas de las filas de $\tilde{H}$ podrían no ser linealmente independientes, teniendo que $k \geq n - m\ deg(g)$. □

Si definimos el vector $\vec{\beta} = \left(g\left(\alpha_1\right)^{-1}, \ldots, g\left(\alpha_n\right)^{-1}\right)$, se tiene que:

$$H = \begin{pmatrix} \beta_1 & \cdots & \beta_n \\ \beta_1\alpha_1 & \cdots & \beta_n\alpha_n \\ \vdots & & \vdots \\ \beta_1\alpha_1^{deg(g)-1} & \cdots & \beta_n\alpha_n^{deg(g)-1} \end{pmatrix}$$

es una matriz $H \in \mathbb{F}_{q^m}^{(n-k')\times n}$ cuyo $\mathbb{F}_{q^m}-$kernel es el código $GC_q\left(\vec{\alpha}, g\right)$ [23]. Ésta se puede considerar como una matriz de control de paridad en sentido no estricto, ya que se define sobre $\mathbb{F}_{q^m}$ en vez de $\mathbb{F}_q$, que es el alfabeto del código.

Para ver ésto, siguiendo el razonamiento de [29], basta con darse cuenta de que $f_i(x)$ puede escribirse de forma explícita. Si definimos $g(x) := g_0 + g_1 x + \cdots + g_{deg(g)} x^{deg(g)}$, obtenemos que:

$$-f_i(x) = \frac{g_1(x-\alpha_i) + g_2(x^2 - \alpha_i^2) + \cdots + g_{deg(g)}(x^{deg(g)} - \alpha^{deg(g)})}{(x-\alpha_i)} =$$
$$= g_1 + g_2(x+\alpha_i) + \cdots + g_{deg(g)}(x^{deg(g)^{-1}} + x^{deg(g)-2}\alpha_i + \cdots + x\alpha^{deg(g)-2} + \alpha^{deg(g)-1})$$

Si introducimos ésto dentro de las $deg(g)$ condiciones sobre $\mathbb{F}_{q^m}$, e interpretamos los factores de las x^j como entradas de vectores columna desde $j = deg(g) - 1$ hasta $j = 0$, construimos una matriz de control de paridad $\hat{H}$ tal que $\vec{c}\hat{H}^\top = 0$ para cualquier palabra-código $\vec{c}$, teniendo la forma:

$$\begin{pmatrix} \beta_1 g_{deg(g)} & \cdots & \beta_n g_{deg(g)} \\ \beta_1(g_{deg(g)-1} + g_{deg(g)}\alpha_1) & \cdots & \beta_n(g_{deg(g)-1} + g_{deg(g)}\alpha_n) \\ \vdots & & \vdots \\ \beta_1(g_1 + g_2\alpha_1 + \cdots + g_{deg(g)}\alpha_1^{deg(g)-1}) & \cdots & \beta_n(g_1 + g_2\alpha_n + \cdots g_{deg(g)}\alpha_n^{deg(g)-1}) \end{pmatrix}$$

Por último, dado que esta matriz se puede escribir como el producto

$$\begin{pmatrix} g_{deg(g)} & 0 & \cdots & 0 \\ g_{deg(g)-1} & g_{deg(g)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_1 & g_2 & \cdots & g_{deg(g)} \end{pmatrix} \begin{pmatrix} \beta_1 & \cdots & \beta_n \\ \beta_1\alpha_1 & \cdots & \beta_n\alpha_n \\ \vdots & \cdots & \vdots \\ \beta_1\alpha_1^{deg(g)-1} & \cdots & \beta_n\alpha_n^{deg(g)-1} \end{pmatrix}$$

y la primera es inversible, nos quedamos con la segunda como la matriz de control de paridad $H \in \mathbb{F}_{q^m}^{(n-k')\times n}$ del $GC_q(\vec{\alpha}, g)$ indicada anteriormente. □

Además, razonando igual que antes, se aprecia cómo el código tiene una dimensión $k' \geq n - deg(g)$ sobre $\mathbb{F}_{q^m}$.

Por último, cabe destacar que cada entrada de H puede escribirse como una columna $m \times 1$ de elementos de $\mathbb{F}_q$, con el paso de $\mathbb{F}_{q^m}$ a $\mathbb{F}_q^m$ comentado anteriormente, para definir una verdadera matriz de control $\tilde{H} \in \mathbb{F}_q^{m(n-k')\times n} = \mathbb{F}_q^{(n-k)\times n}$ sobre el alfabeto $\mathbb{F}_q$.

Como veremos más adelante en la *Sección 3.2*, la construcción de la matriz de control de paridad de un código Goppa binario en Classic McEliece se basa en esta idea.

2.2.2. Implementación

Para cada polinomio irreducible de grado t sobre $\mathbb{F}_{2^m}$ con $m \in \mathbb{Z}^+$ existe un código Goppa binario de longitud $n \leq 2^m$, dimensión $k \geqslant n - tm$ y capaz de corregir cualquier patrón con t errores o menos [22]. Además, con el algoritmo de Patterson [30], éstos pueden descodificarse rápidamente en tiempo $O(nt)$.

A continuación se muestra una aplicación sencilla del criptosistema de McEliece donde Alice envía información confidencial a Bob (ilustrado en la **Figura 1**):

1 Bob genera las claves pública y privada del criptosistema:

1.1) Bob selecciona un valor para n y t, y escoge de forma aleatoria un polinomio irreducible $g(x)$ de grado t sobre $\mathbb{F}_{2^m}$; fijando así un código Goppa binario $\mathcal{C} = GC_2(\vec{\alpha}, g)$ con una cierta $n-$tupla $\vec{\alpha}$.

1.2) Después Bob crea una matriz generatriz G de tamaño $k \times n$ de dicho código $\mathcal{C}$ y la enmascara con una matriz aleatoria $k \times k$ inversible S y otra aleatoria $n \times n$ de permutación P (ésto es, con un uno por fila y columna y el resto ceros), construyendo $G' = SGP$. Ésta será la matriz generatriz pública compartida, con la que se codificarán los mensajes que se quieran enviar a Bob.

1.3) La clave pública será $PK = \{G', t\}$ y la privada $SK = \{g(x), \vec{\alpha}, G, S, P\}$.

2 Alice encripta un mensaje que solo podrá recuperar Bob:

2.1) Alice conoce t y k (de las dimensiones de G') a partir de PK.

2.2) El mensaje se divide en bloques de k bits que serán encriptados.

2.3) Alice genera y envía el cifrado de un bloque $\vec{u}$ como $\vec{x} = \vec{u}G' + \vec{z}$, donde $\vec{z}$ es un vector de longitud n y peso t generado de forma aleatoria para añadir un pequeño error a la palabra-código $\vec{u}G'$.

3 Bob es el único capaz de desencriptar de forma eficiente:

3.1) Calcula $\vec{x}' := \vec{x}P^{-1} = \vec{u}SG + \vec{z}P^{-1}$, con $\vec{u}SG \in \mathcal{C}$ y $wt_H(\vec{z}P^{-1}) = wt_H(\vec{z}) = t$.

3.2) Usando el algoritmo de Patterson para descodificar, se obtiene $\vec{u}' := \vec{u}S$.

3.3) Finalmente se calcula $\vec{u} = \vec{u}'S^{-1}$. Uniendo todos los bloques $\vec{u}$ se recupera el mensaje original.

La seguridad del criptosistema de McEliece, y la de la gran parte de la criptografía basada en códigos correctores, reside en el desafío de descodificar un código lineal aleatorio, el cual es un problema NP-completo [31], y por tanto NP-difícil [23].

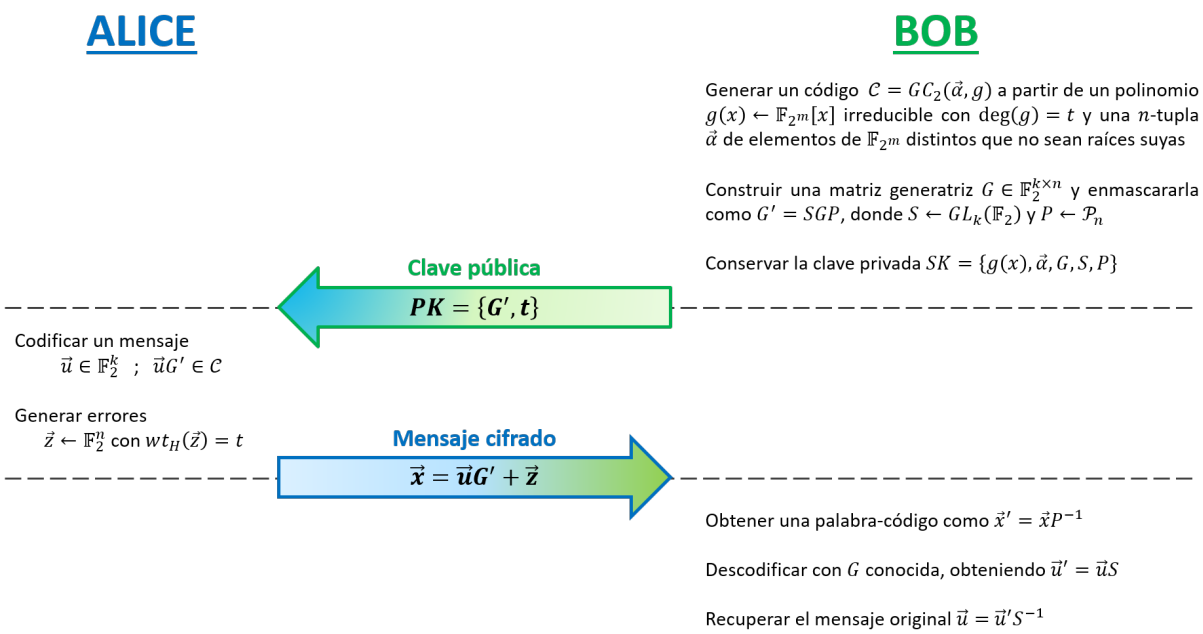

Figura 1: Esquema de aplicación del criptosistema original propuesto por McEliece. Empleado por Alice para transmitir información encriptada a Bob con ayuda de un código Goppa binario $\mathcal{C}$ de longitud $n = 2^m$, dimensión k y capaz de corregir eficientemente hasta t errores. Se aprecia el gran tamaño de PK, debido a G'.

Debido a ésto, aunque el código lineal $\mathcal{C}$ empleado tenga estructura de Goppa, nuestro objetivo será ocultarla de tal forma que cualquier adversario no pueda distinguirlo de uno aleatorio.

Este criptosistema de clave pública que acabamos de ver se basa en una función de una vía con trampa consistente en la facilidad de codificar un mensaje $\vec{u}$ con G' y añadirle una pequeña perturbación $\vec{z}$, y la dificultad de descodificar $\vec{u}G' + \vec{z}$, salvo que se conozcan G, S y P de la clave privada SK, ya que G' parece generar un código lineal aleatorio. En este caso la trampa consiste en conocer la forma de la matriz generatriz, que permite aplicar el algoritmo eficiente de descodificación.

El objetivo de la clave pública de Bob es transmitir una matriz G', en apariencia aleatoria, pero que permita cifrar mensajes dentro de un determinado código lineal que sólo él conoce. Para ello, escoge una matriz generatriz cualquiera G de dicho código y la enmascara, siguiendo el esquema anterior, como G'. De este modo, cualquiera puede encriptar mensajes con G' pero sólo Bob puede descifrarlos, siendo el único que sabe la estructura del código empleado al conocer G.

2.2.3. Situación actual

Desde sus orígenes en 1978, todavía sigue sin encontrarse un ataque algebraico que rompa el criptosistema propuesto por McEliece, ya que no se ha encontrado ninguna propiedad algebraica de los códigos Goppa binarios empleados que permita diferenciarlos de códigos completamente aleatorios [23].

Sin embargo, los códigos Goppa binarios sólo pueden corregir un pequeño número de errores t, necesitando unos parámetros n y k enormes para lograr ciertos niveles de seguridad en el criptosistema [23]. Ésto implica una claves públicas de tamaño muy elevado, ya que contienen matrices generatrices de tamaño $k \times n$, siendo éste el mayor obstáculo para su implementación.

3. Classic McEliece

Una vez introducido el criptosistema original de McEliece, vamos a pasar a estudiar las características de uno de sus descendientes en la actualidad, el criptosistema Classic McEliece o CM.

Para intentar solucionar el principal problema de su predecesor, el elevado tamaño de las claves públicas que genera, se han intentado realizar diversas modificaciones en el criptosistema original de McEliece, como proponer el uso de otros tipos de códigos correctores de errores, pero la mayoría no consiguen esconder la estructura del código que se emplea como clave privada [23]. Algunos de ellos son:

- Basados en códigos Goppa no binarios [32], distinguibles de códigos aleatorios, con un ataque en tiempo polinomial que revela su estructura algebraica [33].
- El criptosistema de Sidelnikov [34], basado en códigos Reed-Muller (*Sección 5.1*) binarios. Inseguro, con un ataque que crea la clave privada con la pública [35].
- Varios intentos con códigos derivados de los Reed-Solomon (*Sección 5.1*) generalizados o GRS [36, 37, 38, 39], de gran interés al presentar la mayor capacidad de corrección de errores posible [23], pero poco fructíferos. Todos son vulnerables frente a ataques que explotan los códigos formados por productos por entradas de dos palabras-código suyas para recuperar su estructura secreta [40, 41].
- Basados en códigos cuasicíclicos o QC (*Sección 4.1*) [42], LDPC [43, 44] y MDPC (*Sección 4.2*) [45]. Permiten claves mucho más compactas, pero a cambio son demasiado estructurados, surgiendo ataques con los que recuperarlas [46, 47].

Otra forma de emplear códigos correctores dentro de criptosistemas de clave pública fue propuesta por Niderreiter en 1986 [25], en el cual los códigos se representan mediante matrices de control de paridad. La idea de esta aproximación es obtener tamaños de textos cifrados más pequeños sin comprometer la seguridad del esquema.

Aunque existe un ataque (de Sidelnikov-Shestakov [48]) frente a este criptosistema con los códigos GRS propuestos originalmente por Niederreiter [25], se puede emplear el mismo esquema con códigos de Goppa binarios, siendo equivalente al de McEliece [49].

Si se hace uso de la seguridad prolongada en el tiempo del criptosistema de McEliece original y el menor tamaño de textos cifrados que permite la estructura de Niederreiter, surge la propuesta del criptosistema Classic McEliece, al que nos referiremos como "CM". A continuación explicaremos en qué consiste un esquema de Niederreiter genérico y después entraremos en detalle sobre distintos aspectos de la implementación de CM.

3.1. Esquema de Niederreiter [23]

La estructura es muy similar a la de McEliece (*Sección 2.2.2*), sólo que ahora los códigos correctores se representan mediante matrices de control de paridad y los mensajes enviados son síndromes en lugar de palabras-código. De nuevo, la función de una vía consiste en que es fácil descodificar el código únicamente si se conoce su estructura, mientras que es difícil si luce completamente aleatorio.

Vamos a comentar el modelo de aplicación de un criptosistema con estructura de Niederreiter, sin especificar el tipo de código corrector, en el que Bob establece una clave pública con la que Alice procede a mandarle un mensaje de forma confidencial, quedando esquematizado en la **Figura 2**.

1 Bob genera las claves pública y privada del criptosistema:

1.1) Bob genera un $[n, k]$ código lineal $\mathcal{C}$ sobre un alfabeto $\mathbb{F}_q$ con capacidad de corregir eficientemente t errores.

1.2) Construye una matriz de control de paridad $H \in \mathbb{F}_q^{(n-k)\times n}$ de dicho código $\mathcal{C}$ y la camufla multiplicándola por una matriz de paridad $P \in \mathcal{P}_n$ por la derecha y una matriz inversible $S \in GL_{n-k}(\mathbb{F}_q)$ por la izquierda, ambas escogidas de forma aleatoria. Así, obtiene una matriz de control de paridad $H' = SHP \in \mathbb{F}_q^{(n-k)\times n}$ que representa un código lineal equivalente a $\mathcal{C}$ que parece aleatorio, otorgando seguridad.

1.3) La clave pública compartida será $PK = \{H', t\}$ y la privada $SK = \{H, S, P\}$.

2 Alice encripta un mensaje que solo podrá recuperar Bob:

2.1) Selecciona un mensaje $\vec{m} \in \mathbb{F}_q^n$ con un peso de Hamming menor o igual que t.

2.2) Encripta dicho mensaje como su síndrome con la matriz de control de paridad H' como $\vec{c} := \vec{m}H'^{\top}$ y envía su traspuesto.

3 Bob es el único capaz de desencriptar de forma eficiente:

3.1) Calcula $S^{-1}\vec{c}^{\top} := HP\vec{m}^{\top} = \left((\vec{m}P^{\top})H^{\top}\right)^{\top}$, con $wt_H(\vec{m}P^{\top}) = wt_H(\vec{m}) \leq t$.

3.2) Aplica un algoritmo de descodificación de síndromes y obtiene $\vec{m}P^{\top}$.

3.3) Recupera el mensaje multiplicando por P por la derecha, ya que $P^{\top} = P^{-1}$.

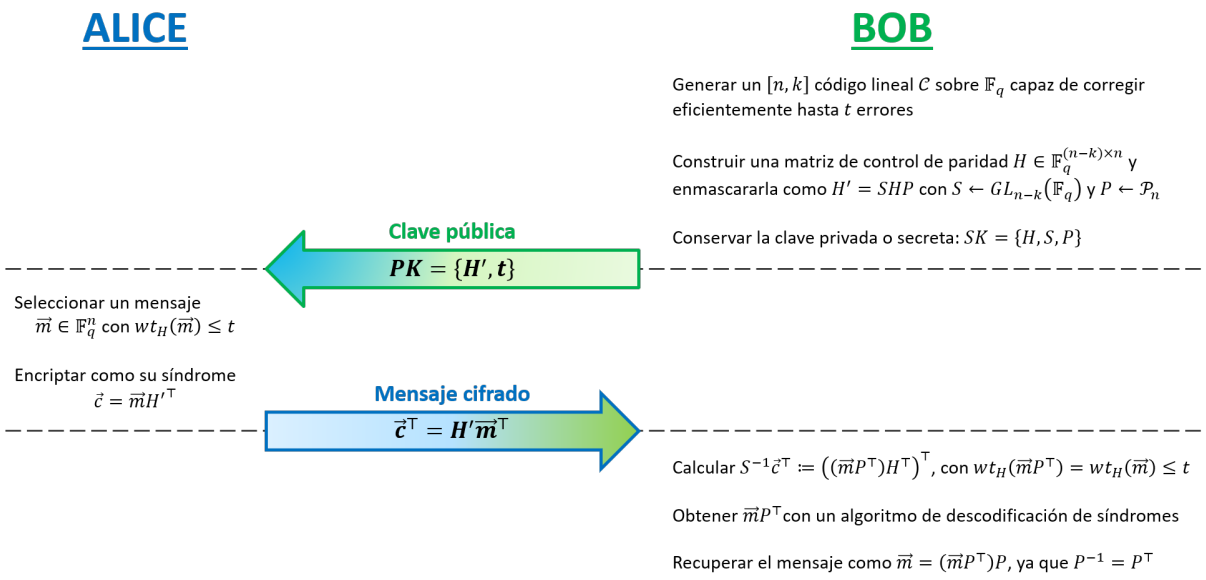

Figura 2: Esquema de un criptosistema con estructura de Niederreiter. Empleado por Alice para transmitir información encriptada a Bob mediante un código secreto $\mathcal{C}$ de longitud n, dimensión k y capaz de corregir eficientemente hasta t errores.

Si comparamos esta estructura (**Figura 2**) con la del criptosistema propuesto originalmente por McEliece (**Figura 1**), se aprecian las siguientes mejoras:

- Bob genera claves de menor tamaño si $k > n/2$, ya que las dimensiones de las matrices de control de paridad H y su enmascarada H' son $(n-k) \times n$, mientras que las matrices generatrices G y G' empleadas en McEliece eran $k \times n$. Este caso es habitual para no tener mucha redundancia.

- Alice realiza una acción completamente determinística, mientras que con McEliece generaba aleatoriamente vectores con pocos errores. Además, ahora encripta mensajes de longitud n en síndromes de longitud $n-k$, en vez de enviar mensajes de

longitud k dentro de palabras-código de longitud n; permitiendo transmitir mensajes mayores con un menor ancho de banda de comunicación, siendo ésta la principal ventaja del esquema de Niederreiter sobre el de McEliece.

Cabe destacar que los mensajes de Alice están limitados a un peso menor o igual que t, pudiendo enviar un número inferior a q^n, que es el número de elementos de $\mathbb{F}_q^n$.

Por último, vamos a ver un pequeño ejemplo sencillo para acabar de ilustrar este esquema criptográfico de Niederreiter, el cual va a ser la base de los criptosistemas Classic McEliece (*Sección 3.2*) y BIKE (*Sección 4.3*).

Usaremos uno de los primeros y más sencillos códigos correctores, un $[7, 4]$ código lineal de Hamming [50]. Nos basta con saber que es un tipo de código binario capaz de detectar y corregir un error y que su matriz de control de paridad H viene dada por:

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{F}_2^{(n-k)\times n} = \mathbb{F}_2^{3\times 7}$$

Para completar lo que será nuestra clave privada, seleccionamos una matriz inversible $S \in GL_3(\mathbb{F}_2)$ y una matriz de permutación $P \in \mathcal{P}_7$:

$$S = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad ; \quad P = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad ; \quad SK = \{HSP\}$$

La clave pública PK consistirá en la capacidad correctora de nuestro código, $t = 1$, y una matriz de control de paridad H' del mismo código pero que esconda su estructura:

$$H' = SHP = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \quad ; \quad PK = \{H', t\}$$

Ahora que hemos compartido nuestra clave pública, supongamos que un amigo, que también conoce el criptoesquema de Niederreiter, escoge como mensaje un vector $\vec{m} \in \mathbb{F}_2^7$ con un peso $wt_H(\vec{m}) \leq t = 1$, por ejemplo $\vec{m} = (0,0,1,0,0,0,0)$, y lo cifra como su síndrome a través de H':

$$\vec{c} = \vec{m}\,(H')^\top = (0,0,1,0,0,0,0) \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = (0,1,0)$$

Tras recibir nuestro ansiado mensaje cifrado como $\vec{c}^\top$, y conociendo SK, computamos

$$S^{-1}\vec{c}^\top = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

Sabemos que dicho vector es igual a $H(P\vec{m}^\top) = \left((\vec{m}P^\top)H^\top\right)^\top$, pudiendo aplicar un método de descodificación de síndromes para nuestro código de Hamming. El síndrome obtenido coincide con la primera columna de H, de modo que $\vec{m}P^\top = (1,0,0,0,0,0,0)$.

Por último, ya que $P^\top = P^{-1}$ recuperamos el mensaje original como:

$$\vec{m} = \left(\vec{m}P^\top\right) P = (1,0,0,0,0,0,0) \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} = (0,0,1,0,0,0,0)$$

En este ejemplo sólo se pueden enviar 7 mensajes con peso $t = 1$ utilizando criptogramas de longitud 3, mientras que su espacio ambiente posee $q^n = 2^7 = 128$ elementos.

3.2. Implementación

A continuación se comenta de forma resumida la estructura del criptosistema de Classic McEliece propuesto en la ronda 3 del NIST, el cual se basa en la estructura de Niederreiter (*Sección 3.1* anterior) con códigos Goppa binarios (*Sección 2.2.1*).

En primer lugar, se escogen $m \in \mathbb{Z}^+$, el cual fija $q = 2^m$, y otros dos enteros positivos $n \leq q$ y $t \geq 2$, de forma que $mt < n$, y se define $k = n - mt$. Además, se toma un polinomio mónico e irreducible $f(x) \in \mathbb{F}_2[x]$ de grado m y se identifica $\mathbb{F}_q$ con $\mathbb{F}_2[x]/_{<f(x)>}$. Así, todo elemento de $\mathbb{F}_q$ puede escribirse como un polinomio $u_0 + u_1 x + \cdots + u_{m-1}x^{m-1}$ mediante un único vector de coeficientes $(u_0, u_1, \ldots, u_{m-1}) \in \mathbb{F}_2^m$ [23].

Una vez descrito nuestro entorno de trabajo, vamos a comentar cómo sería una implementación del criptosistema Classic McEliece con la que Alice pueda compartir un mensaje secreto con Bob de forma segura [51]:

1 **Bob genera las claves pública y privada del criptosistema:**

1.1) Genera de forma aleatoria un polinomio mónico e irreducible $g(x) \in \mathbb{F}_q[x]$ de grado t y n elementos distintos $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ que no sean raíces suyas.

1.2) Computa una matriz de control de paridad $\tilde{H} = \{\tilde{h}_{ij}\}_{ij} \in \mathbb{F}_q^{t \times n}$ para el código Goppa binario asociado a los parámetros $(g, \alpha_1, \ldots, \alpha_n)$, calculando $\tilde{h}_{ij} = \alpha_j^{i-1}/g(\alpha_j)$ para $i = 1, \ldots, t$ y $j = 1, \ldots, n$.

1.3) Se convierte a una matriz $\hat{H} \in \mathbb{F}_2^{mt \times n} = \mathbb{F}_2^{(n-k) \times n}$ sustituyendo cada una de las entradas $u_0 + u_1 x + \cdots + u_{m-1}x^{m-1} \in \mathbb{F}_q = \mathbb{F}_{2^m}$ de H por una columna de m bits $u_0, u_1, \ldots, u_{m-1}$.

1.4) Reduce, si es posible, $\hat{H}$ a forma semisistemática y permuta sus columnas para obtener su única forma sistemática: $H = (Id_{n-k}|T)$, con $T \in \mathbb{F}_2^{(n-k) \times k}$. Además, genera $(\alpha'_1, \ldots, \alpha'_n)$ aplicando las mismas permutaciones a $(\alpha_1, \ldots, \alpha_n)$.

1.5) La clave pública será $PK = \{T, t\}$ y la privada $SK = \{g, \alpha'_1, \ldots, \alpha'_n\}$.

2 **Alice encripta un mensaje que solo podrá recuperar Bob:**

2.1) Toma como mensaje un vector $\vec{e} \in \mathbb{F}_2^n$ de peso t.

2.2) Lo encripta como su síndrome por $H = (Id_{n-k}|T)$, como $\vec{c_0} = H\vec{e}^\top \in \mathbb{F}_2^{n-k}$.

3 Bob es el único capaz de desencriptar de forma eficiente:

3.1) Extiende $\vec{c_0}$ a $\vec{v} = (\vec{c_0}^{\top}, 0, \ldots, 0) \in \mathbb{F}_2^n$ y, como SK nos define un código Goppa conocido, se puede emplear un algoritmo específico para encontrar una palabra-código $\vec{c} \in \mathbb{F}_2^n$ con $d_H(\vec{c}, \vec{v}) \leq t$, en caso de que exista.

3.2) Recupera el mensaje original como $\vec{e} = \vec{v} + \vec{c}$ y comprueba que cumpla que $H\vec{e}^{\top} = \vec{c_0}$ y que $wt_H(\vec{e}) = t$.

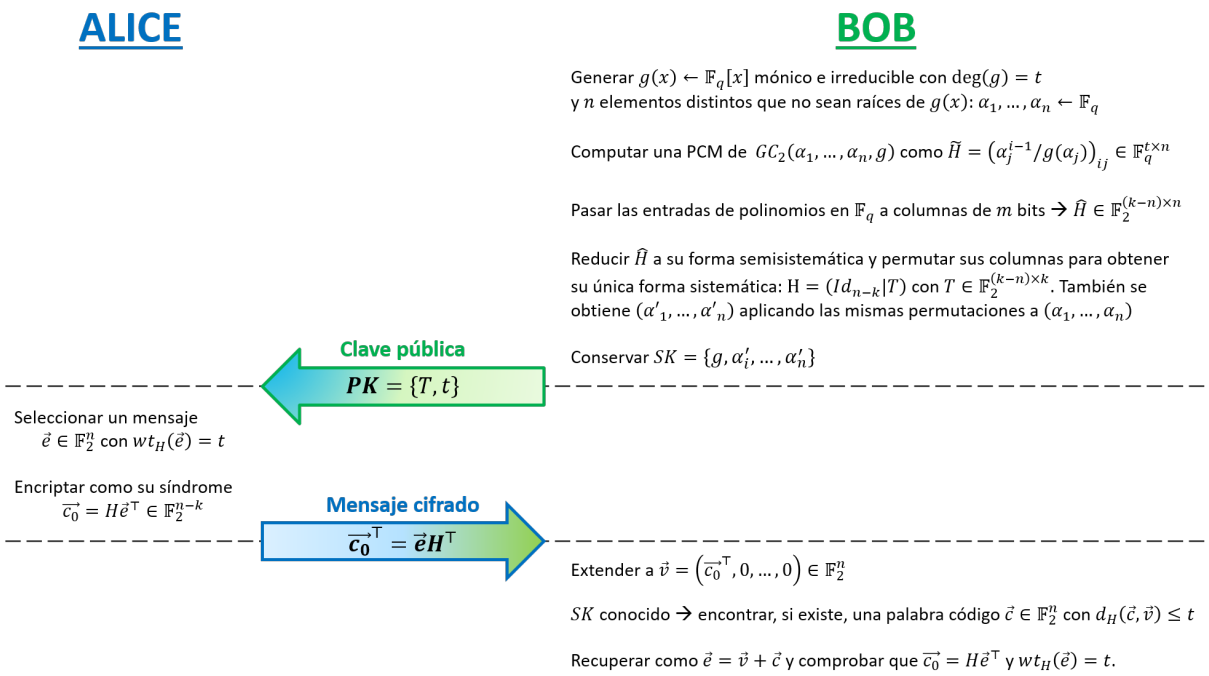

Figura 3: Esquema del criptosistema Classic McEliece. Empleado por Alice para transmitir información encriptada a Bob dentro de un $[n, k]$ código lineal capaz de corregir eficientemente hasta t errores. Se emplea "PCM" para referirnos a una matriz de control de paridad y "$GC_2(\vec{\alpha}, p)$" para un código Goppa binario generado por una $n-$tupla de elementos $\vec{\alpha}$ y un polinomio p; ver *Sección 2.2.1.*

Se basa en el esquema de Niederreiter visto en la *Sección 3.1*, cifrando los mensajes con su síndrome, pero ahora la estructura del código Goppa binario empleado se esconde usando la única forma sistemática de su matriz de control de paridad [52].

La implementación de este criptosistema presenta las siguientes mejoras en el tamaño de las claves pública (PK) y secreta o privada (SK):

- PK: Se envía una matriz $T \in \mathbb{F}_2^{(n-k)\times k}$ de menor tamaño, definiendo completamente una matriz de control de paridad pública H, sin necesidad de mandarla entera.
- SK: El código queda definido por un polinomio y n elementos, en vez de tres matrices.

3.3. Parámetros seleccionados y seguridad

A continuación, en la **Tabla 2**, se recogen los distintos conjuntos de parámetros propuestos, junto al tamaño en bytes de las claves y los textos cifrados y el nivel de seguridad esperado, donde los niveles 1, 3 y 5, se corresponden a 128, 192 y 256 bits respectivamente, es decir, el coste computacional de romper un AES-128, un AES-192 y un AES-256.

Tabla 2: Parámetros propuestos para el criptosistema Classic McEliece [23].

Conjunto	m	n	t	PK	SK	Texto cifrado	Seguridad
mceliece348864	12	3.488	64	261.120	6.492	128	Nivel 1
mceliece460896	13	4.608	96	524.160	13.608	188	Nivel 3
mceliece6688128	13	6.688	128	1.044.992	13.932	240	Nivel 5
mceliece6960119	13	6.960	119	1.047.319	13.948	226	Nivel 5
mceliece8192128	13	8.192	128	1.357.824	14.120	240	Nivel 5

Además, el polinomio de grado m, mónico e irreducible $f(x) \in \mathbb{F}_2[x]$ seleccionado, que define la representación de $\mathbb{F}_q$ como ${}^{\mathbb{F}_2[x]}/_{<f(x)>}$, es $f(x) = x^{12} + x^3 + 1$ para el conjunto "mceliece348864" y $f(x) = x^{13} + x^4 + x^3 + x + 1$ para el resto [51].

4. BIKE

El criptosistema BIKE (*Bit Flipping Key Encapsulation*) emplea matrices de control de paridad circulantes, otorgando tamaños de clave pequeños, y de densidad moderada, permitiendo una descodificación eficiente con un algoritmo BF (*Bit-Flipping*) [23].

A continuación iremos introduciendo los nuevos conceptos en los que se basa BIKE y después comentaremos de forma breve cómo se implementa en la práctica.

4.1. Códigos cuasicíclicos [23]

Sea una tupla o vector $\vec{c} = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_q^n$, se define una **rotación cíclica** suya como $\sigma(\vec{c}) = (c_{n-1}, c_0, \ldots, c_{n-2})$.

Sea $\mathcal{C}$ un $[n, k]$ código lineal sobre $\mathbb{F}_q$, decimos que es un **código cíclico** si $\sigma(\mathcal{C}) = \mathcal{C}$, de forma que la rotación cíclica de cualquier palabra-código es otra palabra-código. Además, dicho código cíclico puede verse como un ideal de ${}^{\mathbb{F}_q[x]}/_{< x^n - 1 >}$, identificando de forma unívoca los coeficientes del único polinomio de grado menor que n que identifica cada clase con las componentes del vector, como se muestra en el siguiente isomorfismo:

$$\varphi : {}^{\mathbb{F}_q[x]}/_{< x^n - 1 >} \to \mathbb{F}_q^n$$
$$c(x) = c_0 + c_1 x \cdots + c_{n-1} x^{n-1} \mapsto (c_0, c_1, \ldots, c_{n-1})$$

Definimos el **polinomio generador** de un código cíclico $\mathcal{C} \subset \mathbb{F}_q^n$ como el único polinomio mónico de menor grado $g(x)$ que genera el ideal correspondiente en ${}^{\mathbb{F}_q[x]}/_{< x^n - 1 >}$. De hecho, si $g(x) = \sum_{i=0}^{r} g_i x^i$ de grado r, se tiene que la dimensión de dicho código $\mathcal{C}$ es $k := n - r$ y una matriz generatriz suya $G \in \mathbb{F}_q^{(n-r)\times n}$ viene dada por:

$$G = \begin{pmatrix} g_0 & \cdots & g_r & & \\ & \ddots & & \ddots & \\ & & g_0 & \cdots & g_r \end{pmatrix}$$

la cual se denomina una **matriz circulante**, siendo cada fila una rotación cíclica de la anterior.

Volviendo a la conexión entre el subespacio vectorial $\mathcal{C} \subset \mathbb{F}_q^n$ y el anillo de polinomios

$\mathbb{F}_q[x]/_{< x^n - 1 >}$, vamos a introducir una nueva operación de multiplicación entre vectores que imite a la de los polinomios.

Sean $\vec{u}, \vec{v} \in \mathbb{F}_q^n$, se define su multiplicación como $\vec{u}\vec{v} := \vec{u}rot(\vec{v})$, donde

$$rot(\vec{v}) = \begin{pmatrix} \vec{v} \\ \sigma(\vec{v}) \\ \vdots \\ \sigma^{n-1}(\vec{v}) \end{pmatrix} \in \mathbb{F}_q^{n \times n}$$

es una matriz circulante denominada **matriz de rotación** de $\vec{v}$.

Así, se devuelve otro vector dentro del espacio vectorial $\mathbb{F}_q^n$, teniendo una nueva operación de multiplicación interna y conmutativa que devuelve el mismo vector que si se multiplicasen los polinomios asociados a los vectores $\vec{u}$ y $\vec{v}$ en $\mathbb{F}_q[x]/_{< x^n - 1 >}$ y se pasase a su vector correspondiente, es decir, que $\vec{u}\vec{v} = \varphi\left(u(x)v(x)\right)$.

Una vez hemos descrito los fundamentos de los códigos cíclicos, podemos definir una variante de los mismos, los códigos cuasicíclicos.

Sean $\vec{c} = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ y $\ell \in \{1, \ldots, n\}$, se define una **rotación ℓ-cíclica** suya como $\sigma_\ell(\vec{c}) = (c_{0+\ell}, c_{1+\ell}, \ldots, c_{n-1+\ell})$ donde las sumas en los subíndices son módulo n.

Por último, sea $\mathcal{C}$ un $[n, k]$ código lineal sobre $\mathbb{F}_q$, diremos que es un **código cuasicíclico (QC)** si existe $\ell \in \{1, \ldots, n\}$ tal que $\sigma_\ell(\mathcal{C}) = \mathcal{C}$. Además, si $n = \ell a$ para algún $a \in \mathbb{N}$, resulta conveniente escribir su matriz generatriz G compuesta por ℓ matrices circulantes $a \times a$.

A modo de ejemplo, consideramos la siguiente matriz generatriz G de un código lineal $\mathcal{C}$, donde cada fila es una rotación $2-$cíclica de la anterior.

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \xrightarrow[\text{Pares e Impares}]{\text{Columnas}} G' = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right)$$

Si agrupamos las columnas en pares e impares mediante permutaciones, surge una matriz G' que define un código $\mathcal{C}'$ equivalente a $\mathcal{C}$, el cual tiene la misma estructura algebraica y conserva su métrica de Hamming, al haber aplicado una isometría suya [23].

4.2. Códigos de densidad moderada [23]

Primero vamos a hablar sobre los **códigos con matriz de paridad de baja densidad o LDPC** (*Low-Density Parity-Check*), introducidos por Gallager en 1962 [53]. Éstos se basan en el empleo de matrices de control de paridad que sean dispersas (más conocidas como *sparse* en inglés), con la mayoría de sus elementos siendo ceros. Esta característica permite mejoras tanto en memoria de almacenamiento como en velocidad de cálculo mediante algoritmos especializados [54].

A partir de ahora, diremos que una matriz tiene un **peso por columna** w, si el peso de Hamming (*Sección 2.1*) de cada una de sus columnas es w, siendo análoga la definición del **peso por fila** de una matriz.

Sean $\lambda, \rho \in \mathbb{N}$. Un $[n, k]$ código lineal $\mathcal{C}$ sobre $\mathbb{F}_2$ se dice que es un **código LDPC $(\boldsymbol{\lambda}, \boldsymbol{\rho})-$regular** si existe una matriz de control de paridad $H \in \mathbb{F}_2^{(n-k)\times n}$ suya con un peso por columna λ y un peso por fila ρ. Generalmente en criptografía solo se indica el peso por fila, siendo un **código $\boldsymbol{\rho}-$LDPC** para el caso anterior.

Los códigos LDPC resultan muy interesantes en criptografía porque no se basan en ninguna estructura algebraica que pueda ser detectada y aprovechada por un atacante. Además, poseen algoritmos de descodificación eficientes. Uno de ellos, llamado *Bit-Flipping* (BF), se remonta al origen de estos códigos [53], y se sigue empleando con diversas variaciones y mejoras. Este algoritmo es iterativo y probabilístico, ya que no siempre consigue descodificar de forma exitosa, dando un "fallo".

Vamos a explicar un algoritmo BF que nos permite recuperar un vector $\vec{e}$ con el que se haya obtenido un síndrome $\vec{s}$ a través de una matriz de control de paridad H, siendo ésta la situación de la descodificación de BIKE, como veremos en la siguiente *Sección 4.3*. Sus pasos son:

① Se inicializa con $\vec{e}' = \vec{0}$ y $\vec{s}' = \vec{s}$

② Se genera una tupla UPC (*Unsatisfied Parity-Checks*) con el número de ecuaciones de control de paridad que no satisface cada bit j de $\vec{e}'$. Éste puede verse como el número de los unos de la columna $j-$ésima de H que coinciden con los de $\vec{s}'^{\top}$.

③ Se invierten los bits de $\vec{e}'$ con un valor de UPC máximo, ya que son los que intervienen en un mayor número de ecuaciones de control de paridad fallidas.

④ Se calcula $\vec{s}' = s - \vec{e}'H^\top$ y se abren dos opciones:

$\boxed{\vec{s}' = \vec{0}:}$ Se devuelve $\vec{e}'$. La idea es que $\vec{s}' = \vec{e}H^\top - \vec{e}'H^\top = (\vec{e} - \vec{e}')H^\top$, anulándose cuando $\vec{e} - \vec{e}'$ sea $\vec{0}$, teniendo una descodificación exitosa con $\vec{e} = \vec{e}'$, u otra palabra-código, con un fallo.

$\boxed{\vec{s}' \neq \vec{0}:}$ Se vuelve al paso ② y se cuenta el número de iteraciones de este bucle, devolviendo un fallo "$\perp$" si se alcanza un máximo fijado previamente.

Si se relaja la condición del peso por fila de los códigos LDPC, surge la familia de los códigos MDPC (*Moderate-Density Parity-Check*). Un $[n, k]$ código lineal $\mathcal{C}$ sobre $\mathbb{F}_2$ se dice que es un **código MDPC** si existe una matriz de control de paridad $H \in \mathbb{F}_2^{(n-k)\times n}$ suya con un peso por fila de $\mathcal{O}(\sqrt{nlog(n)})$ estrictamente; una familia de códigos dependientes de n.

De este modo, la única diferencia entre los códigos LDPC y MDPC es que los últimos tienen unos pesos por fila mayores, lo cual puede suponer un mayor número de iteraciones del algoritmo BF en su descodificación y que sus fallos sean más probables.

Por último, vamos a ilustrar el algoritmo BF previamente explicado con un ejemplo sencillo. Consideremos la siguiente matriz de control de paridad:

$$H = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \in \mathbb{F}_2^{6\times 7} = \mathbb{F}_2^{(n-k)\times n}$$

la cual define un $[7, 1]$ código lineal de repetición binario $\mathcal{C}$ compuesto únicamente por dos palabras-código: $\vec{0} = (0, 0, 0, 0, 0, 0, 0)$ y $\vec{1} = (1, 1, 1, 1, 1, 1, 1)$, pudiendo detectar y corregir hasta $t = 3$ errores por mínima distancia. Basta con darse cuenta de que cada fila, salvo la tercera, impone que las palabras-código tengan parejas de bits iguales. Todas estas condiciones, junto con la de la tercera fila, que enlaza cuatro bits, terminan exigiendo que los $n = 7$ coincidan.

Se aprecia cómo $\mathcal{C}$ es un código LDPC, al ser H una matriz *sparse* con una amplia mayoría de ceros, pero irregular, ya que los pesos por columna y por fila no son constantes.

Supongamos que un amigo nuestro escoge como mensaje el vector $\vec{e} = (0,1,0,0,1,0,0)$, enviándonos su síndrome

$$\vec{s} = \vec{e}H^\top = (0,1,0,0,1,0,0) \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = (1,1,0,1,1,1)$$

Una vez recibido $\vec{s}$, procedemos con el algoritmo BF descrito anteriormente, comenzando por inicializar con $\vec{e}'_{(0)} = (0,0,0,0,0,0,0)$ y $\vec{s}'_{(0)} = \vec{s} = (1,1,0,1,1,1)$. Acto seguido, entramos al bucle de invertir los bits más problemáticos de $\vec{e}'$ y actualizar el síndrome $\vec{s}'$.

Iteración 1

$$H = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \qquad \vec{s}'^{\top}_{(0)} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$UPC_{(0)} = \begin{pmatrix} 1 & 2 & 1 & 1 & 3 & 1 & 1 \end{pmatrix} \quad ; \quad max\left(UPC_{(0)}\right) = 3$$

En este caso, se invierte únicamente el antepenúltimo bit de $\vec{e}'_{(0)} = \vec{0}$ por intervenir en 3 ecuaciones de control de paridad no satisfechas, teniendo que $\vec{e}'_{(1)} = (0,0,0,0,1,0,0)$.

Ahora actualizamos el síndrome como

$$\vec{s}'_{(1)} = \vec{s} - \vec{e}'_{(1)}H^\top = (1,1,0,1,1,1) - (0,0,1,1,1,1) = (1,1,1,0,0,0)$$

y, al ser distinto de $\vec{0}$, seguimos en el bucle.

Iteración 2

$$H = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \qquad \vec{s}'^{\top}_{(1)} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$UPC_{(1)} = \begin{pmatrix} 1 & 3 & 2 & 1 & 1 & 0 & 0 \end{pmatrix} \quad ; \quad max\left(UPC_{(1)}\right) = 3$$

Se invierte el segundo bit de $\vec{e}'_{(1)} = (0,0,0,0,1,0,0)$, obteniendo $\vec{e}'_{(2)} = (0,1,0,0,1,0,0)$.

De nuevo, actualizamos el síndrome como

$$\vec{s}'_{(2)} = \vec{s} - \vec{e}'_{(2)}H^{\top} = (1,1,0,1,1,1) - (1,1,0,1,1,1) = (0,0,0,0,0,0)$$

Salimos del bucle con $\vec{e}'_{(2)} = (0,1,0,0,1,0,0)$, el cual coincide con el mensaje $\vec{e}$ que nos quería hacer llegar nuestro amigo, teniendo un éxito en nuestra descodificación.

4.3. Implementación

En este apartado vamos a explicar de forma esquemática qué pasos se llevan a cabo al implementar BIKE, el cual se basa en el criptoesquema de Niederreiter (*Sección 3.1*) con códigos MDPC (*Sección 4.2*) a partir de matrices circulantes (*Sección 4.1*).

Se comienza seleccionando un número primo r tal que 2 sea raíz primitiva módulo r, *i.e.*, 2 genera el grupo multiplicativo $\mathbb{Z}/r\mathbb{Z}^*$ [23]. Este parámetro r indica el tamaño de bloque, fijando el tamaño del código como $n = 2r$, y la redundancia $n - k$ del código, teniendo dimensión $k = r$. Además, se escoge un peso por fila $w \approx \sqrt{n}$ tal que $w/2$ sea impar y un peso de errores $t \approx \sqrt{n}$.

Después, se toma como espacio de trabajo el anillo de polinomios $R := {}^{\mathbb{F}_2[x]}/_{< x^r - 1 >}$, cuya conexión con un código cíclico $\mathcal{C} \subset \mathbb{F}_2^r$ ya ha sido comentada antes en la *Sección 4.1*. Así, cualquier elemento $a \in R$ puede representarse de forma única por un polinomio de grado menor o igual que $r-1$ como $a = \sum_{i=0}^{r-1} a_i x^i$, donde $a_i \in \mathbb{F}_2$ para $\forall\, i \in \{0,1,\ldots,r-1\}$.

Debido a la elección de r, la cual implica que los únicos factores irreducibles de $x^r - 1$ sean $x - 1$ y $x^{r-1} + x^{r-2} + \cdots + 1$, un elemento $a \in R$ es inversible si y solo si $wt(a)$ es impar y distinto de r, donde $wt(a) = wt_H(\vec{a})$ con $\vec{a}$ su representación única como vector de $\mathbb{F}_2^r$, *i.e.*, $\vec{a} = \varphi(a)$; siendo ésta una nueva noción de pesos para elementos de R [23].

Una vez definido nuestro entorno de trabajo, vamos a comentar el mecanismo de cifrado que se lleva a cabo en la práctica con este criptosistema [23]:

1 **Bob genera las claves pública y privada del criptosistema:**

1.1) Selecciona un par de elementos $h_0, h_1 \leftarrow R$ tales que $wt(h_0) = wt(h_1) = w/2$.

1.2) Computa $h = h_1 h_0^{-1} \in R$, donde h_0^{-1} siempre existe por tener h_0 peso $w/2$, que es impar y distinto de r.

1.3) La clave pública será $PK = \{h, t\}$ y la privada $SK = \{h_0, h_1\}$.

2 **Alice encripta un mensaje que solo podrá recuperar Bob:**

2.1) Se codifica como un error descrito por $(e_0, e_1) \in R^2$ con $wt(e_0) + wt(e_0) = t$.

2.2) Lo encripta como $c_0 = e_0 + e_1 h \in R$

3 **Bob es el único capaz de desencriptar de forma eficiente:**

3.1) Computa $s := c_0 h_0 = e_0 h_0 + e_1 h_1$.

3.2) Como h_0 y h_1 tienen densidad moderada, se puede descodificar de forma eficiente mediante un algoritmo *Bit-Flipping* para recuperar (e_0, e_1).

Ésta es una manera de ver el sistema desde el punto de vista del anillo de polinomios. Sin embargo, a la hora de la verdad se trabajan con bits, de modo que vamos a reescribir los pasos desde el punto de vista de vectores y matrices sobre $\mathbb{F}_2 = \{0, 1\}$, valiéndonos de la relación entre R y $\mathbb{F}_2^r$, descrita de forma genérica en la *Sección 4.1*.

Así, se aprecia más fácilmente la base del esquema de Niederreiter (*Sección 3.1*) y la matriz de control de paridad que define nuestro código, siendo un QC-MDPC [55], ya

que, como veremos ahora, es una matriz de densidad moderada (MDPC, *Sección 4.2*) compuesta por dos bloques que son matrices circulantes (QC, *Sección 4.1*).

1 Bob genera claves:

1.1) Selecciona dos elementos $h_0, h_1 \leftarrow R$ tales que $wt(h_0) = wt(h_1) = w/2$, representados unívocamente por $\vec{h}_0 = \varphi(h_0), \vec{h}_1 = \varphi(h_1) \in \mathbb{F}_2^k = \mathbb{F}_2^r \subset \mathbb{F}_2^{2r} = \mathbb{F}_2^n$.

1.2) Computa $h = h_1 h_0^{-1} \in R$ y lo escribe como un vector $\vec{h} = \varphi(h) \in \mathbb{F}_2^r$.

1.3) La clave pública será $PK = \{\vec{h}, t\}$ y la privada $SK = \{\vec{h}_0, \vec{h}_1\}$.

2 Alice encripta un mensaje que solo podrá recuperar Bob:

2.1) Se codifica dentro de un error descrito por $(e_0, e_1) \in R^2$ con $wt(e_0)+wt(e_1) = t$, mediante dos vectores *sparse* de errores: $\vec{e}_0 = \varphi(e_0), \vec{e}_1 = \varphi(e_1) \in \mathbb{F}_2^r$.

2.2) Lo encripta como $\vec{c}_0 = \vec{e_0} + \vec{e_1}\vec{h} := \vec{e}_0 + \vec{e_1} rot(\vec{h}) \in \mathbb{F}_2^r$.

- Cabe destacar que Alice puede seleccionar un mensaje $\vec{e} \in \mathbb{F}_2^{2r}$ de $2r$ bits y mandarlo encriptado en $\vec{c_0} \in \mathbb{F}_2^r$ con la mitad de bits. Ésto resulta muy práctico desde el punto de vista del ancho de banda de comunicación entre los dos usuarios, siendo la razón de que BIKE esté basado en un esquema de Niederreiter y no de McEliece [55].

3 Bob es el único capaz de desencriptar de forma eficiente:

3.1) Computa $\vec{s} = \vec{c}_0\vec{h}_0 := \varphi\left(e_0 h_0 + e_1(h_1 h_0^{-1})h_0\right) := \vec{e}_0 rot(\vec{h}_0) + \vec{e}_1 rot(\vec{h}_1) = \vec{e}H^\top$, donde $\vec{e} = (\vec{e}_0 \mid \vec{e_1}) \in \mathbb{F}_2^{2r}$ con $wt_H(\vec{e}) = t$ y $H \in \mathbb{F}_2^{(n-k)\times n} = \mathbb{F}_2^{r\times 2r}$ es una matriz de control de paridad cuasicíclica (QC), formada por dos matrices circulantes:

$$H = \left(rot\left(\vec{h}_0\right) \,\middle|\, rot\left(\vec{h}_1\right) \right)$$

$$= \left(\begin{array}{ccccc|ccccc} h_{0_0} & h_{0_1} & \cdots & h_{0_{r-2}} & h_{0_{r-1}} & h_{1_0} & h_{1_1} & \cdots & h_{1_{r-2}} & h_{1_{r-1}} \\ h_{0_{r-1}} & h_{0_0} & \cdots & h_{0_{r-3}} & h_{0_2} & h_{1_{r-1}} & h_{1_0} & \cdots & h_{1_{r-3}} & h_{1_2} \\ \vdots & & \ddots & & \vdots & \vdots & & \ddots & & \vdots \\ h_{0_2} & h_{0_3} & \ldots & h_{0_0} & h_{0_1} & h_{1_2} & h_{1_3} & \ldots & h_{1_0} & h_{1_1} \\ h_{0_1} & h_{0_2} & \ldots & h_{0_{r-1}} & h_{0_0} & h_{1_1} & h_{1_2} & \ldots & h_{1_{r-1}} & h_{1_0} \end{array}\right)$$

apreciándose el uso de un código MDPC con un peso por fila de $w/2+w/2 = w$.

3.2) Al tener $\vec{e}$ y H densidad moderada $(t, w \approx \sqrt{n})$, se puede descodificar el síndrome $\vec{s}$ de forma eficiente con un algoritmo BF (*Bit-Flipping*) para recuperar el mensaje $\vec{e}$ con una alta probabilidad de éxito.

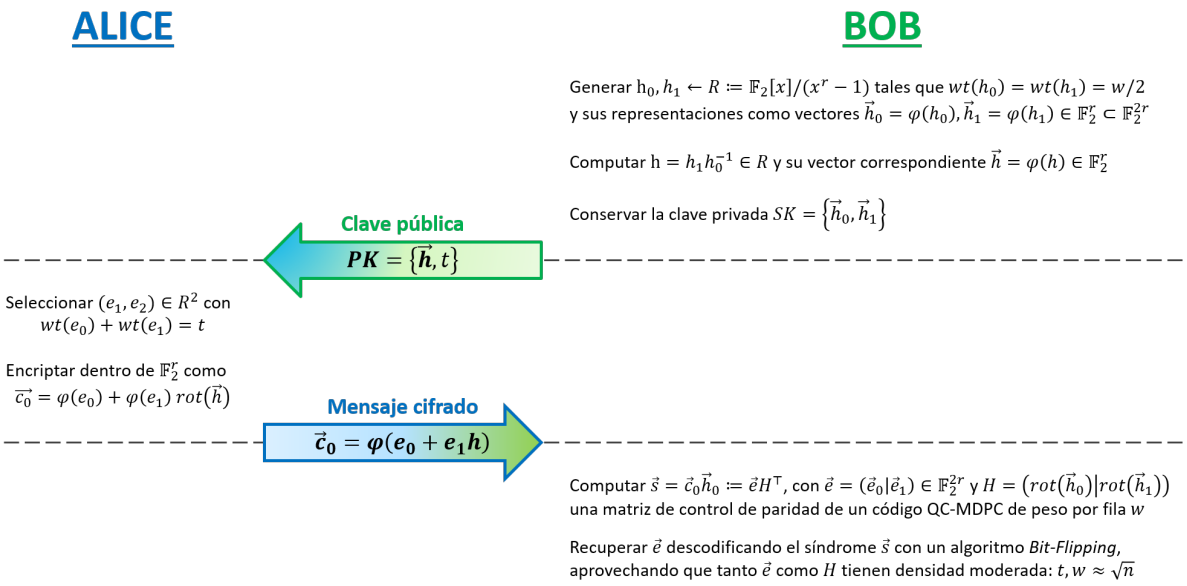

Figura 4: Esquema del criptosistema BIKE. El código empleado, en el intercambio de un mensaje $\vec{e} \in \mathbb{F}_2^{2r}$ entre Alice y Bob, es un QC-MDPC, donde "QC" indica su carácter cuasicíclico (*Sección 4.1*) y "MDPC" que su matriz de control de paridad es de densidad moderada (*Sección 4.2*). Cabe comentar que el algoritmo *Bit-Flipping* no siempre consigue descodificar el síndrome, pudiendo devolver un fallo, pero con muy baja probabilidad.

Si lo comparamos con el criptosistema Classic McEliece o CM antes visto (*Sección 3.2*):

- PK: Ahora, además de la capacidad de corrección de errores t, contiene un vector $\vec{h}$ de r bits de longitud en vez de una matriz $(n-k) \times k$, que sería de r^2 bits con los parámetros empleados en BIKE: $k = r = n/2$. Esta gran mejora en el tamaño de la clave pública es la principal ventaja de un sistema con códigos QC.

- SK: La matriz de control de paridad se define a partir de dos vectores de r bits ($\vec{h_0}$ y $\vec{h}_1$), con un total de $2r$ bits, mientras que en CM se emplean dos vectores de longitud n sobre $\mathbb{F}_{2^m}$ (para el polinomio g y n elementos $\alpha_1, \ldots, \alpha_n$ que no fuesen raíces suyas, definiendo un código Goppa binario), con $2nm$ bits en total. Además, los vectores *sparse* $\vec{h}_0$ y $\vec{h}_1$ pueden guardarse de forma todavía más compacta listando únicamente sus $w/2$ posiciones no nulas.

- Mensaje cifrado: Ambos se basan en un esquema de Niederreiter, que permite encriptar mediante un síndrome de $n-k$ bits un mensaje de n bits, siendo r y $2r$ bits con los parámetros de BIKE; la mitad de lo necesario si se basase en un esquema de McEliece [55].

4.4. Parámetros seleccionados y seguridad

En la *Tabla 3* se muestran los parámetros propuestos para BIKE (tamaño de bloque r, peso por fila w y peso de los errores t) en la ronda 4 del NIST, junto al tamaño en bytes de la clave pública, la clave privada y el texto cifrado, para tres niveles de seguridad. También se recoge su tasa de fallo de descodificación o DFR (*Decoding Failure Ratio*), que indica cómo de probable es que se produzca un fallo en su descodificación.

Tabla 3: Parámetros propuestos para el criptosistema BIKE [23].

Seguridad	r	w	t	PK	SK	Texto cifrado	DFR
Nivel 1	12.323	142	134	1.541	281	1.573	2^{-128}
Nivel 3	24.659	206	199	3.083	419	3.115	2^{-192}
Nivel 5	40.973	274	264	5.122	580	5.154	2^{-256}

Recordemos que los niveles 1, 3 y 5, se corresponden a 128, 192 y 256 bits de seguridad respectivamente. Además, al seleccionar el tamaño del bloque r en BIKE, ya quedan fijados el resto de parámetros del $[n, k]$ código lineal empleado como $n = 2r$ y $k = r$.

5. HQC

El criptosistema HQC (*Hamming Quasi-Cyclic)* se basa en un paradigma cuasicíclico, en lugar del de Niederreiter, y usa una combinación de un código cualquiera, capaz de descodificarse de forma eficiente, con matrices circulantes (*Sección 4.1*). A partir de la ronda 3 del NIST se propone emplear códigos concatenados de Reed-Muller y Reed-Solomon [23].

Una propiedad característica de HQC es que los códigos usados no son secretos, como veremos más adelante en su implementación (*Sección 5.3*).

Como siempre, vamos a comenzar con una base teórica sobre los nuevos elementos en los que se sustenta HQC. Comenzaremos introduciendo los códigos que emplea y después veremos el criptoesquema cuasicíclico en el que se basa.

5.1. Códigos Reed-Solomon y Reed-Muller concatenados

Sean $k \leq n \leq q$ enteros positivos y $\vec{\alpha} \in \mathbb{F}_q^n$ una $n-$tupla de elementos distintos. Un **código Reed-Solomon (RS)** [56] de longitud n y dimensión k se define como [23]:

$$RS_{n,k}(\vec{\alpha}) = \{(f(\alpha_1), \ldots, f(\alpha_n)) \mid f \in \mathbb{F}_q[x],\ deg(f) < k\}$$

siendo un $[n, k]$ código lineal cuya matriz generatriz G viene dada por la matriz de Vandermonde:

$$G = \begin{pmatrix} 1 & \ldots & 1 \\ \alpha_1 & \ldots & \alpha_n \\ \vdots & & \vdots \\ \alpha_1^{k-1} & \ldots & \alpha_n^{k-1} \end{pmatrix}$$

Para codificar un mensaje $\vec{m} = (m_0, m_1, \ldots, m_{k-1}) \in \mathbb{F}_q^k$, éste se interpreta como un polinomio $f(x) = m_0 + m_1 x + \cdots + m_{k-1}x^{k-1} \in \mathbb{F}_q[x]$ y se evalúa en los puntos $\alpha_1, \ldots, \alpha_n$, obteniendo así su palabra-código como $\vec{m}G$ [57].

Además, son códigos MDS, ya que alcanzan la distancia máxima posible para su tamaño y dimensión (*Sección 2.1*). Basta con darse cuenta de que un polinomio $f(x)$ de grado $k-1$ puede tener hasta $k-1$ raíces distintas, de modo que $d_H\left(RS_{n,k}(\vec{\alpha})\right) \geq n-k+1$ y alcanza la desigualdad de Singleton [24], implicando que $d_H\left(RS_{n,k}(\vec{\alpha})\right) = n-k+1$ [57].

Los códigos Red-Muller (RM) [58], se asemejan a los RS que acabamos de describir, construyéndose mediante la evaluación de polinomios, pero siendo ahora multivariables.

Sean m, r enteros positivos, p un primo y $q = p^n$. Denotamos como $\mathbb{F}_q[x_1, \ldots, x_m]_{\leq r}$ el $\mathbb{F}_q-$espacio vectorial de polinomios de m variables y grado total menor o igual que r y fijamos un orden $\{\alpha_1, \alpha_2, \ldots, \alpha_{q^m}\}$ para los elementos de $\mathbb{F}_q^m$. Un **código Red-Muller (RM)** sobre $\mathbb{F}_q$, denotado por $RM_q(m, r)$, se define como la imagen de la aplicación evaluación [23]:

$$\begin{aligned} ev: \ \mathbb{F}_q[x_1, \ldots, x_m]_{\leq r} &\to \mathbb{F}_q^{q^m} \\ f &\mapsto (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_{q^m})) \end{aligned}$$

El grado total de un monomio $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$ es $k_1 + k_2 + \cdots + k_n$, y el grado total de un polinomio es el máximo de los asociados a sus monomios con coeficientes distintos de cero [57].

Un $RM_q(m, r)$ es un código lineal de longitud $n = q^m$ y de dimensión dada por el número de polinomios en $\mathbb{F}_q[x_1, \ldots, x_m]_{\leq r}$, la cual no tiene una expresión sencilla para el caso general [57]. En [29], mediante combinatoria, se obtiene que la dimensión de un $\mathbb{F}_q[x_1, \ldots, x_m]_{\leq r}$ es:

$$k = \sum_{t=0}^{r} \sum_{i=0}^{m} (-1)^i \begin{pmatrix} m \\ i \end{pmatrix} \begin{pmatrix} t - iq + m + 1 \\ t - iq \end{pmatrix}$$

No es para nada trivial, pero se puede demostrar [57, 29] que, para un código $RM_q(m, r)$ con $r = \nu(q-1) + s$ y $0 \leq s < q - 1$, su distancia mínima cumple que $d_H\left(RM_q(m, r)\right) \geq (q - s)q^{m-\nu+1}$.

Por último, vamos a introducir los códigos concatenados [59], cuya idea consiste en emplear un proceso de codificación en dos etapas a través de dos códigos.

Sean $\mathcal{C}_1$ un $[n_1, k_1]$ código lineal sobre $\mathbb{F}_q$ con distancia mínima d_1, al cual llamaremos "código interior", y $\mathcal{C}_2$ un $[n_2, k_2]$ código lineal sobre $\mathbb{F}_{q^{k_1}}$ con distancia mínima d_2, denominado "código exterior". Entonces, el **código concatenado $\mathcal{C} = \mathcal{C}_2 \circ \mathcal{C}_1$** es un $[n_1 n_2, k_1 k_2]$ código lineal sobre $\mathbb{F}_q$ con una distancia mínima mayor o igual que $d_1 d_2$ [23].

Las palabras-código de $\mathcal{C}$ se generan de la siguiente forma [23]:

① Se codifica cualquier $\vec{u} \in \mathbb{F}_{q^{k_1}}^{k_2}$ con la matriz generatriz $G_2 \in \mathbb{F}_{q^{k_1}}^{k_2 \times n_2}$ del código exterior $\mathcal{C}_2$:

$$\vec{u}G_2 = \left((\vec{u}G_2)_1, \ldots, (\vec{u}G_2)_{n_2} \right) \in \mathcal{C}_2 \subset \mathbb{F}_{q^{k_1}}^{n_2}$$

② Se representa las entradas $(\vec{u}G_2)_i \in \mathbb{F}_{q^{k_1}}$ como vectores $\overrightarrow{(\vec{u}G_2)_i} \in \mathbb{F}_q^{k_1}$, empleando una base ordenada $\{b_1, \ldots, b_q\}$ de los elementos de $\mathbb{F}_q$ previamente fijada.

③ Se codifica cada entrada con la matriz generatriz $G_1 \in \mathbb{F}_q^{k_1 \times n_1}$ del código interior $\mathcal{C}_1$:

$$\left(\overrightarrow{(\vec{u}G_2)_1} G_1, \ldots, \overrightarrow{(\vec{u}G_2)_{n_2}} G_1 \right) \in \mathcal{C} \subset \mathbb{F}_q^{n_1 n_2}$$

donde $\overrightarrow{(\vec{u}G_2)_i} G_1 \in \mathcal{C}_1 \subset \mathbb{F}_q^{n_1}$ $\forall\ i \in \{1, \ldots, n_2\}$.

Ahora queda más clara la notación $\mathcal{C} = \mathcal{C}_2 \circ \mathcal{C}_1$ anterior, ya que las matrices generatrices G_2 y G_1 actúan sobre los vectores a su izquierda. Además, como es lógico, para que una palabra entre a nuestro código, ésta debe ir del "exterior" al "interior".

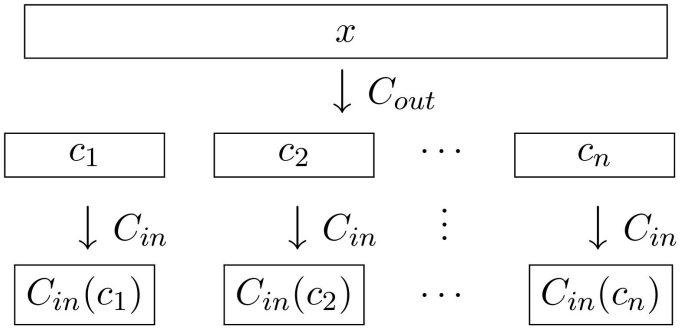

Figura 5: Generación de palabras-código con códigos concatenados. El código exterior $\mathcal{C}_{out}$ convierte el mensaje de entrada x en una palabra-código de n componentes. Después, cada una de éstas se codifica con el código interior $\mathcal{C}_{in}$ en una palabra-código suya. Finalmente, se juntan todas para obtener una palabra-código del código concatenado $\mathcal{C} = \mathcal{C}_{out} \circ \mathcal{C}_{in}$ [57].

5.2. Esquema cuasicíclico [23]

Es un esquema de encriptación con un enfoque probabilístico que emplea un código con descodificación eficiente, el cual no se esconde.

El mensaje se encripta como una palabra-código a la cual se le añade un error demasiado largo como para poder ser descodificado. Después, conociendo la clave privada, se cancelan partes de dicho error, obteniendo con gran probabilidad un vector que puede descodificarse para recuperar el mensaje.

Sea n un entero positivo, q la potencia de un primo y $R = {}^{\mathbb{F}_q[x]}/_{< x^n - 1 >}$, cuyos polinomios pueden identificarse con vectores en $\mathbb{F}_q^n$, como vimos en la *Sección 4.1.* Además, usaremos la multiplicación allí definida como $\vec{u}\vec{v} = \vec{u}rot(\vec{v})$ para elementos de $\mathbb{F}_q^n$.

El esquema cuasicíclico emplea dos tipos de códigos, los cuales se hacen públicos:

① Un $[n, k]$ código lineal $\mathcal{C}$ sobre $\mathbb{F}_q$ capaz de descodificar t errores de forma eficiente, el cual se define por una matriz generatriz $G \in \mathbb{F}_q^{k \times n}$.

② Un código cuasicíclico $[2n, n]$ descrito mediante una matriz de control de paridad

$$H = \left(Id_n \mid rot(\vec{h}) \right) \in \mathbb{F}_q^{n \times 2n}$$

donde $\vec{h}$ se escoge de $\mathbb{F}_q^n$ de forma aleatoria, y que no necesita ser eficientemente descodificable.

Por último, se fijan tres enteros positivos $w, w_e, w_r \approx \sqrt{n}/2$ para controlar errores.

Una vez definidos todos los elementos del esquema, vamos a ver cómo éste es usado para intercambiar un mensaje entre Alice y Bob:

1 Bob genera claves:

1.1) Genera ①, descrito por una G.

1.2) Selecciona dos elementos $y, z \in \mathbb{F}_q^n$ con $wt_H(\vec{y}) = wt_H(\vec{z}) = w$.

1.3) Toma de forma aleatoria un vector $\vec{h} \in \mathbb{F}_q^n$ que define el código ②.

1.4) Computa $\vec{s} = \vec{y} + \vec{z}\vec{h}$, el cual puede verse como un síndrome del código ②:

$$s = (\vec{y}, \vec{z}) \begin{pmatrix} Id_n \\ rot(\vec{h}) \end{pmatrix} = (\vec{y}, \vec{z}) H^\top$$

donde $(\vec{y}, \vec{z})$ juega el papel de un vector de errores.

1.5) La clave pública será $PK = \{G, \vec{h}, \vec{s}\}$ y la privada $SK = \{\vec{y}, \vec{z}\}$.

2 Alice encripta un mensaje que solo podrá recuperar Bob:

2.1) Escoge un mensaje $\vec{m} \in \mathbb{F}_q^k$.

2.2) Toma $\vec{e}, \vec{r}_1, \vec{r}_2 \leftarrow \mathbb{F}_q^n$ con $wt_H(\vec{e}) = w_e$ y $wt_H(\vec{r}_1) = wt_H(\vec{r}_2) = w_r$.

2.3) Computa $\vec{u} = \vec{r}_1 + \vec{r}_2\vec{h}$ como un síndrome de ② y $\vec{v} = \vec{m}G + (\vec{r}_2\vec{s} + \vec{e})$ como una palabra-código de ① a la que se le añade un error, el cual debe ser suficientemente grande para no ser descodificable incluso conociendo ① mediante G; es decir, que $wt_H(\vec{r}_2\vec{s} + \vec{e}) > t$.

2.3) El cifrado enviado viene dado por $\vec{c} = (\vec{u}, \vec{v})$.

3 Bob es el único capaz de desencriptar de forma eficiente:

3.1) Puede emplear un algoritmo de descodificación de ① sobre $\vec{v} - \vec{u}\vec{z}$, ya que:

$$\begin{aligned}\vec{v} - \vec{u}\vec{z} &= \vec{m}G + \vec{r}_2\vec{s} + \vec{e} - (\vec{r}_1 + \vec{r}_2\vec{h})\vec{z} \\ &= \vec{m}G + \vec{r}_2(\vec{y} + \vec{z}\vec{h}) + \vec{e} - \vec{r}_1\vec{z} - \vec{r}_2\vec{h}\vec{z} \\ &= \vec{m}G + (\vec{r}_2\vec{y} - \vec{r}_1\vec{z} + \vec{e})\end{aligned}$$

recuperando $\vec{m}$ de forma exitosa cuando $wt_H\left(\vec{r}_2\vec{y} - \vec{r}_1\vec{z} + \vec{e}\right) \leq t$. Los parámetros se seleccionan de tal forma que ésto ocurra con gran probabilidad, pero el esquema presenta una tasa de fallos o DFR (*Decoding Failure Rate*).

- En la última igualdad se cancelan dos términos del error: $\vec{r}_2\vec{z}\vec{h} - \vec{r}_2\vec{h}\vec{z} = 0$, ya que la operación de multiplicación descrita en $\mathbb{F}_q^n$ es conmutativa (*Secicón 4.1*).
- Por otro lado, la condición de éxito es muy probable debido a que los pesos de los vectores del error $(\vec{r}_2\vec{y} - \vec{r}_1\vec{z} + \vec{e})$ son de orden $\sqrt{n}/2$, mucho menor que su longitud n, esperando pocas coincidencias entre la entradas no nulas del vector y las de la matriz circulante correspondiente en su producto.
- Por último, la seguridad del esquema cuasicíclico recae en que, tanto para determinar $(\vec{r}_1, \vec{r}_2)$ como $(\vec{y}, \vec{z})$, con los cuales recuperar el mensaje $\vec{m}$, un atacante debe resolver un problema de descodificación de síndromes (o SDP) de ②, el cual se considera NP-difícil. De este modo, Bob es el único capaz de aplicar "*3.1)*", ya que sólo él conoce $\vec{z}$.

Cabe destacar que SDP es NP-difícil para un código completamente aleatorio [23]. En este caso, ② presenta una cierta estructura cuasicíclica en su matriz de control de paridad H, de modo que la seguridad se basa en la suposición de que este tipo de códigos son suficientmente aleatorios como para seguir suponiendo un problema NP-difícil.

Finalmente, vamos a ver dos ejemplos sencillos para acabar de ilustrar este esquema cuasicíclico, sobre el que se basa HQC, como veremos en la siguiente *Sección 5.3*. Además, un ejemplo será de éxito y otro de fallo, apreciando así el comportamiento probabilístico de este criptosistema.

Trabajaremos sobre el anillo $R = {}^{\mathbb{F}_2[x]}/_{< x^7+1 >}$ y emplearemos un código binario de repetición de longitud 7, cuya matriz generatriz es

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{1\times 7}$$

De esta forma, tenemos un $[7, 1]$ código lineal

$$\mathcal{C} = \{\vec{x}G \mid \vec{x} \in \mathbb{F}_2^1\} = \{(0,0,0,0,0,0,0), (1,1,1,1,1,1,1)\} := \{\vec{0}, \vec{1}\}$$

capaz de corregir hasta 3 errores, devolviendo las palabras-código $\vec{0}$ o $\vec{1}$ cuando la mayoría de bits de una palabra sean ceros o unos respectivamente.

Así, ya tenemos fijados los parámetros $n = 7$, $k = 1$ y $t = 3$ del código $\mathcal{C}$, y seleccionamos el resto, asociados a los pesos de distintos elementos, como $w = w_e = w_r = 1$.

◦ Primer ejemplo:

Vamos a considerar el polinomio $h(x) = 1 + x + x^2 \in R$, con su representación vectorial $\vec{h} = (1,1,1,0,0,0,0) \in \mathbb{F}_2^7$.

Acto seguido, seleccionamos dos elementos de R con peso $w = 1$, por ejemplo $y(X) = 1$ y $z(x) = x^3$, representados como $\vec{y} = (1,0,0,0,0,0,0)$ y $\vec{z} = (0,0,0,1,0,0,0)$, los cuales componen nuestra clave privada $SK = \{\vec{y}, \vec{z}\}$.

Por último, computamos $\vec{s} = \vec{y} + \vec{z}\vec{h} := \vec{y} + \vec{z}rot(\vec{h})$ como

$$\vec{s} = (1,0,0,0,0,0,0) + (0,0,0,1,0,0,0) \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = (1,0,0,1,1,1,0)$$

y publicamos la clave pública $PK = \{G, \vec{h}, \vec{s}\}$.

Ahora que hemos compartido nuestra clave pública, supongamos que un amigo, que también conoce el criptoesquema cuasicíclico, escoge para compartir con nosotros un mensaje $\vec{m} \in \mathbb{F}_2^1$, por ejemplo $\vec{m} = (1)$.

Además, genera aleatoriamente $e(x) = x \in R$, representado por $\vec{e} = (0,1,0,0,0,0,0)$ con peso $wt_H(\vec{e}) = w_e = 1$, y $r_1(x) = r_2(x) = x^2 \in R$, representados por $\vec{r}_1 = \vec{r}_2 = (0,0,1,0,0,0,0)$ con pesos $wt_H(\vec{r}_1) = wt_H(\vec{r}_2) = w_r = 1$.

Luego, computa $\vec{u} = \vec{r}_1 + \vec{r}_2\vec{h} := \vec{r}_1 + \vec{r}_2 rot(\vec{h})$ y $\vec{v} = \vec{m}G + (\vec{r}_2\vec{s} + \vec{e})$, que puede verse como una palabra-código $\vec{c}_* = \vec{m}G$ más un error $\vec{e}_* = \vec{r}_2\vec{s} + \vec{e} := \vec{r}_2 rot\,(\vec{s}) + \vec{e}$, teniendo:

$$\vec{u} = (0,0,1,0,0,0,0) + (0,0,1,0,0,0,0) \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = (0,0,0,1,1,0,0)$$

$$\vec{c}_* = (1) \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} = (1,1,1,1,1,1,1)$$

$$\vec{e}_* = (0,0,1,0,0,0,0)\begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} + (0,1,0,0,0,0,0) = (1,1,1,0,0,1,1)$$

$$\vec{v} = \vec{c}_* + \vec{e}_* = (1,1,1,1,1,1,1) + (1,1,1,0,0,1,1) = (0,0,0,1,1,0,0)$$

Finalmente, nos envía su mensaje $\vec{m}$, pero escondido dentro de $\vec{c} = (\vec{u}, \vec{v})$.

Cabe destacar que $wt_H(\vec{e}_*) = 5 > 3 = t$, lo cual da seguridad al sistema, ya que un atacante no podrá recuperar $\vec{m}$ aunque intercepte $\vec{v}$ y conozca $G \in PK$. Podrá corregir los errores y obtener una palabra-código, pero incorrecta en este caso, ya que obtendría el vector $\vec{0}$.

A partir de esta información recibida y la previamente conocida, $SK = \{\vec{y}, \vec{z}\}$, calculamos $\vec{v} - \vec{u}\vec{z}$, el cual es igual por construcción a $\vec{m}G + (\vec{r}_2\vec{y} - \vec{r}_1\vec{z} + \vec{e})$, pudiendo aplicar un algoritmo de descodificación para recuperar el mensaje $\vec{m}$ con alta probabilidad, siendo exitoso siempre que $wt_H(\vec{r}_2\vec{y} - \vec{r}_1\vec{z} + \vec{e}) \leq t = 3$.

$$\vec{v} - \vec{u}\vec{z} = (0,0,0,1,1,0,0) - (0,0,0,1,1,0,0)\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} = (1,0,0,1,1,0,1)$$

Ya que $\vec{v} - \vec{u}\vec{z}$ presenta 4 unos y 3 ceros, tenemos que la palabra-código $\vec{m}G$ es $\vec{1}$, lo que nos indica que el mensaje es $\vec{m} = (1)$, recuperándolo exitosamente.

Este acierto era completamente esperado, ya que se tiene que $\vec{r}_2\vec{y} - \vec{r}_1\vec{z} + \vec{e} = (0,1,1,0,0,1,0)$ con peso 3.

◦ Segundo ejemplo:

Vamos a considerar el mismo sistema que el del ejemplo anterior (con un anillo $R = {}^{\mathbb{F}_2[x]}/_{< x^7+1 >}$ y un código binario de repetición de longitud 7), manteniendo las claves pública $PK = \{G, \vec{h}, \vec{s}\}$ y privada $SK = \{\vec{y}, \vec{z}\}$. Los parámetros (n, k, t, w, w_e, w_r) también serán los mismos, salvo w_e, que ahora tomaremos como 2 en vez de 1.

Supondremos que nuestro amigo quiere enviarnos el mismo mensaje $\vec{m} = (1)$, pero ahora genera de forma aleatoria con peso $w_e = 2$ a $\vec{e} = (0, 0, 0, 1, 1, 0, 0)$ y con peso $w_r = 1$ a $\vec{r}_1 = (0, 0, 0, 0, 0, 0, 1)$ y $\vec{r}_2 = (0, 0, 0, 0, 0, 1, 0)$.

De nuevo, computa dos vectores $\vec{u} := \vec{r}_1 + \vec{r}_2 rot(\vec{h})$ y $\vec{v} = \vec{c}_* + \vec{e}_*$, donde $\vec{c}_* := \vec{m}G$ y $\vec{e}_* := \vec{r}_2 rot\,(\vec{s}) + \vec{e}$ que compondrán la información que nos envía mediante $\vec{c} = (\vec{u}, \vec{v})$.

$$\vec{u} = (0,0,0,0,0,0,1) + (0,0,0,0,0,1,0) \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = (1,0,0,0,0,1,0)$$

$$\vec{c}_* = (1) \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} = (1,1,1,1,1,1,1)$$

$$\vec{e}_* = (0,0,0,0,0,1,0) \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} + (0,0,0,1,1,0,0) = (0,1,1,0,1,1,0)$$

$$\vec{v} = \vec{c}_* + \vec{e}_* = (1,1,1,1,1,1,1) + (0,1,1,0,1,1,0) = (1,0,0,1,0,0,1)$$

Volvemos a tener que $wt_H(\vec{e}_*) = 4 > 3 = t$, de modo que la información $\vec{c} = (\vec{u}, \vec{v})$ que nos envía nuestro amigo no permite que posibles adversarios accedan a su mensaje $\vec{m}$.

Tras recibirla, computamos $\vec{v} - \vec{u}\vec{z} := \vec{m}G + (\vec{r}_2\vec{y} - \vec{r}_1\vec{z} + \vec{e})$ como:

$$\vec{v} - \vec{u}\vec{z} = (1,0,0,1,0,0,1) - (1,0,0,0,0,1,0)\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} = (1,1,0,0,0,0,1)$$

Al presentar ahora más ceros, obtenemos como palabra-código $\vec{0}$ y, por tanto, que el mensaje era (0), habiéndose producido un fallo del esquema. Observad que se tiene $\vec{r}_2\vec{y} - \vec{r}_1\vec{z} + \vec{e} = (0,0,1,1,1,1,0)$ con peso $4 > 3 = t$, añadiendo demasiados errores a la palabra-código verdadera $\vec{m}G$ como para poder asegurar su buena corrección.

De esta forma queda patente el carácter probabilístico del esquema cuasicíclico y su DFR (*Decoding Failure Rate*), ya que, aunque hayamos variado ligeramente un peso en el segundo ejemplo, los parámetros de ambos casos se acomodan a sus especificaciones, teniendo $w, w_e, w_r \approx \sqrt{n}/2 = \sqrt{7}/2 \sim 1{,}3$. También hay que tener en cuenta que esta condición resulta mucho más significativa a medida que se aumenta el tamaño n y que en la práctica se escogen los parámetros de forma óptima para minimizar la DFR.

5.3. Implementación

Una vez introducido el marco teórico necesario, vamos a comentar la implementación del criptosistema HQC, el cual consiste en una aplicación del esquema cuasicíclico, visto en la *Sección 5.2* anterior, empleando como código con descodificación eficiente a un "RM∘RS", que hace referencia a una concatenación de códigos Reed-Muller y Reed-Solomon (*Sección 5.1*).

Se escoge $n \in \mathbb{Z}^+$ tal que $(x^n - 1)\,/\,(x - 1)$ es irreducible sobre ${}^{\mathbb{F}_2[x]}/_{< x^n - 1 >}$, denominando entonces a n como un "primo primitivo" [60]. Después se selecciona un entero positivo $k < n$ y un $[n, k]$ código lineal $\mathcal{C}$ de tipo RM∘RS y binario, *i.e.*, sobre $\mathbb{F}_2$, capaz de corregir eficientemente t errores.

Empleando un enfoque similar a BIKE (*Sección 4.3*), se trabajará sobre un anillo de polinomios $R := {}^{\mathbb{F}_2[x]}/_{< x^n - 1 >}$, donde cada elemento se identifica como un vector de longitud n sobre $\mathbb{F}_2$, siguiendo la relación descrita en la *Sección 4.1*.

Finalmente se escogen tres pesos para los distintos vectores de error, siendo w, w_e y w_r en el orden de $\sqrt{n}/2$.

Como siempre, una vez definido el entorno de trabajo, vamos a comentar cómo Alice y Bob pueden intercambiarse información empleando este criptosistema, con unos parámetros (n, k, t, w, w_e, w_r) previamente seleccionados.

1 Bob genera claves:

1.1) Escoge una matriz generatriz G de un $[n, k]$ código lineal $\mathcal{C}$ de tipo RM∘RS, capaz de descodificar t errores de forma eficiente.

1.2) Toma de forma aleatoria un elemento $h \in R$, representado por $\vec{h} = \varphi(h) \in \mathbb{F}_2^n$.

1.3) Genera aleatoriamente una pareja de elementos $(y, z) \in R^2$, descritos por dos vectores $\vec{y}, \vec{z} \in \mathbb{F}_2^n$ con $wt_H(\vec{y}) = wt_H(\vec{z}) = w$.

1.4) Computa $\vec{s} = \vec{y} + \vec{z}\vec{h} \in \mathbb{F}_2^n$.

1.5) La clave pública será $PK = \{G, \vec{h}, \vec{s}\}$ y la privada $SK = \{\vec{y}, \vec{z}\}$.

2 Alice encripta un mensaje que solo podrá recuperar Bob:

2.1) Genera aleatoriamente tres elementos $e, r_1, r_2 \in R$, cuyos vectores $\vec{e}, \vec{r}_1, \vec{r}_2 \in \mathbb{F}_2^n$ cumplan que $wt_H(\vec{e}) = w_e$ y $wt_H(\vec{r}_1) = wt_H(\vec{r}_2) = w_r$.

2.2) Encripta un mensaje $\vec{m} \in \mathbb{F}_2^n$ dentro de $\vec{c} = (\vec{u}, \vec{v}) \in \mathbb{F}_2^{2n}$, donde $\vec{u} = \vec{r}_1 + \vec{r}_2\vec{h}$ y $\vec{v} = \vec{m}G + (\vec{r}_2\vec{s} + \vec{e})$.

3 Bob es el único capaz de desencriptar de forma eficiente:

3.1) Aplica un algoritmo de descodificación de $\mathcal{C}$ a $\vec{v} - \vec{u}\vec{z} := \vec{m}G + (\vec{r}_2\vec{y} - \vec{r}_1\vec{z} + \vec{e})$, donde $wt_H\left(\vec{r}_2\vec{y} - \vec{r}_1\vec{z} + \vec{e}\right) \leq t$ con alta probabilidad, recuperando $\vec{m}$ exitosamente cuando ésto ocurra. Recordad que este paso presenta una tasa de fallo de descodificación (DFR), pero muy baja por su diseño.

Tanto su funcionamiento como su seguridad recaen en el esquema cuasicíclico, visto en la *Sección 5.2* anterior, y su implementación se esquematiza en la **Figura 6**.

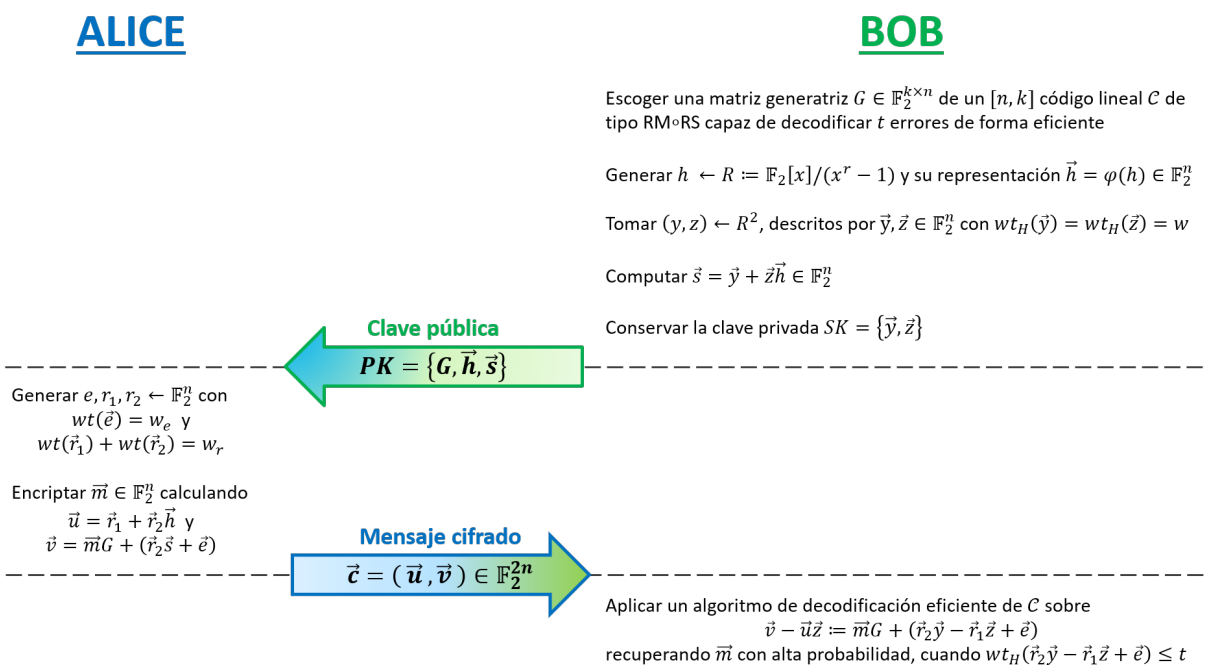

Figura 6: Esquema del criptosistema HQC con parámetros (n, k, t, w, w_e, w_r) previamente fijados. El código empleado, en el intercambio de un mensaje $\vec{m} \in \mathbb{F}_2^n$ entre Alice y Bob, es un RM∘RS, indicando la concatenación de un código Reed-Muller y un código Reed-Solomon (*Sección 5.1*). La descodificación puede ser errónea con una tasa de fallo denominada DFR, cuando $wt_H\,(\vec{r}_2\vec{y} - \vec{r}_1\vec{z} + \vec{e}) > t$.

Por último, vamos a hacer una pequeña comparación entre este criptosistema HQC (**Figura 6**) y su competidor más cercano, BIKE (**Figura 4**):

- PK: En este caso es más grande, siendo una matriz G de tamaño $k \times n$ y dos vectores $\vec{h}$ y $\vec{s}$ de longitud n, mientras que en BIKE solamente se compartía un vector $\vec{h}$ de longitud n y el número de errores t que podía descodificar eficientemente su código.

- SK: Su tamaño es el mismo, pero HQC guarda un par de vectores de error $\vec{y}$ y $\vec{z}$ de longitud n, mientras que BIKE conservaba dos vectores $\vec{h_0}$ y $\vec{h_1}$ de longitud n que ocultaban su código.

- Mensaje cifrado: En BIKE se empleaban $r = n/2$ bits, mientras que con HQC se envían $2n$ bits, requiriendo un mayor ancho de banda en la comunicación.

5.4. Parámetros seleccionados y seguridad

En la *Tabla 4* se describen los parámetros con los que generar los códigos concatenados de Reed-Muller (RM) y Reed-Solomon (RS) empleados en HQC para tres niveles de seguridad 1, 3 y 5 de 128, 192 y 256 bits respectivamente. Estos parámetros son la longitud de los códigos (n_1 para el RS interior, n_2 para el RM exterior y n para el RM∘RS concatenado), su dimensión (k_1, k_2 y k) y su distancia mínima (d_1, d_2 y d). Además, se muestra la máxima capacidad correctora t del código concatenado.

Tabla 4: Parámetros de los códigos concatenados de RM y RS empleados en HQC [60].

Seguridad	RS interior			RM exterior			Concatenado: RM∘RS			
	n_1	k_1	d_1	n_2	k_2	d_2	$n = n_1 n_2$	$k = k_1 k_2$	$d \geq d_1 d_2$	$t \leq \lfloor \frac{d-1}{2} \rfloor$
Nivel 1	46	16	31	384	8	192	17.664	128	5.952	2.975
Nivel 3	56	24	33	640	8	320	35.840	192	10.560	5.279
Nivel 5	90	32	59	640	8	320	57.600	256	15.104	7.551

Los códigos RS interiores tienen como mínima distancia $d_1 = n_1 - k_1 + 1$, siendo MDS, y se encuentran sobre $\mathbb{F}_{2^8}$, mientras que los RM exteriores tienen $d_2 = n_2/2$ y trabajan sobre $\mathbb{F}_2$.

En la *Tabla 5* se recogen los conjuntos de parámetros propuestos para HQC en la ronda 4 del NIST para los niveles de seguridad 1, 3 y 5. Dichos parámetros son la dimensión del espacio ambiente (n, el primer primo primitivo mayor que $n_1 n_2$ para evitar ataques algebraicos [60]) y los pesos de los errores del sistema (w) y del mensaje ($w_e = w_r$). También se incluye el tamaño de las claves y el mensaje cifrado en bytes y una estimación de la cota máxima de DFR (*Decoding Failure Rate*).

Tabla 5: Parámetros propuestos para el criptosistema HQC [23].

Seguridad	n	w	$w_e = w_r$	PK	SK	Texto cifrado	DFR
Nivel 1	17.669	66	75	2.249	40	4.481	2^{-128}
Nivel 3	35.851	100	114	4.522	40	9.026	2^{-192}
Nivel 5	57.637	131	149	7.245	40	14.469	2^{-256}

6. Comparación

En esta sección vamos a hacer una comparativa entre los criptosistemas de encriptación de clave pública de la cuarta ronda de estandarización de PQC del NIST, resumiendo sus distintos elementos y características fundamentales vistas en este trabajo (*Tabla 6*). Después contrastaremos su desempeño y el de Kyber (*Tabla 7*), el único seleccionado para estandarizar en la tercera ronda; buscando en qué contextos pueden ser preferibles los distintos criptosistemas en función de sus ventajas e inconvenientes.

Tabla 6: Comparativa teórica de los candidatos en la ronda 4 de PQC del NIST.

Comparativa PQC - Ronda 4	Classic McEliece	BIKE	HQC
Esquema	Niederreiter	Niederreiter	Cuasicíclico
Código lineal	Goppa binario	QC-MDPC	RM∘RS y QC
PK	T, t	$\vec{h}, t$	$G, \vec{h}, \vec{s}$
SK	$g, \alpha'_1, \ldots, \alpha'_n$	$\vec{h}_0, \vec{h}_1$	$\vec{y}, \vec{z}$
Fundamento	Esconder código	Esconder código	Añadir más errores de los corregibles
Problema difícil	Descodificación de un código lineal aleatorio	Descodificación de un código lineal aleatorio	Descodificación de síndromes
DFR	–	$2^{-bits_{seguridad}}$	$2^{-bits_{seguridad}}$

En primer lugar, tanto Classic McEliece como BIKE se basan en un esquema de Niederreiter, cuya seguridad recae en la dificultad de descodificar un código aleatorio. En ambos casos sus códigos presentan una estructura que permita una descodificación eficiente, pero que debe permanecer oculta frente a adversarios externos:

- **Classic McEliece:** Emplea códigos Goppa binarios, los cuales han sido ampliamente estudiados; una opción conservadora basada en la propuesta original de McEliece, la cual sigue protegida frente a todos los ataques conocidos hasta la fecha [52]. Su estructura se guarda a través de un polinomio (g) y n elementos $(\alpha'_1, \ldots \alpha'_n)$ con los que generar su matriz de control de paridad en forma sistemática $H = (Id \mid T)$. Ésto permite compartirla a través de una matriz T de menor tamaño, junto a su capacidad de corrección de errores t, para poder cifrar mensajes de peso t como sus síndromes, pero sin dar a conocer su estructura algebraica. (*Sección 3.2*)

- **BIKE**: Emplea un $[n, k]$ código lineal cuasicíclico (QC) con una matriz de control de paridad H de densidad moderada (MDPC), formada por dos matrices circulantes generadas por dos vectores $\vec{h}_0$ y $\vec{h}_1$ de pesos moderados ($\approx \sqrt{n}/2$), siendo ésta la estructura a esconder. Se comparte otro vector $\vec{h}$ con el que encriptar mensajes de hasta un peso $t \approx \sqrt{n}$, los cuales pueden transformarse en síndromes de H sólo si se conoce su estructura. Por último su descodificación eficiente se debe a un algoritmo *Bit-Flipping*, aprovechando su densidad moderada. (*Sección 4.3*)

Por otro lado, **HQC** se basa en un esquema cuasicíclio, que emplea dos códigos:

① Un código lineal capaz de descodificar t errores de forma eficiente y descrito por una matriz generatriz $G \in \mathbb{F}_q^{k \times n}$. Para HQC éste será un código RM∘RS, es decir, la concatenación de un Reed-Muller exterior y Reed-Solomon interior.

② Un código cuasicíclico $[2n, n]$ que no necesita ser eficientemente descodificable y se describe mediante una matriz de control de paridad $H = \left(Id_n \mid rot(\vec{h})\right) \in \mathbb{F}_q^{n \times 2n}$, donde $\vec{h}$ se escoge de $\mathbb{F}_q^n$ de forma aleatoria.

Dicho esquema publica la estructura de ambos códigos, mediante G y $\vec{h}$, y un vector $\vec{s} = \vec{y} + \vec{z}rot(\vec{h})$. Los mensajes $\vec{m}$ se cifran a partir de $G, \vec{h}, \vec{s}$ y se les puede aplicar un método de descodificación solamente si se conoce $\vec{z}$, imposible de recuperar si también se guarda $\vec{y}$. La seguridad del sistema se resume en que el texto cifrado consta de un síndrome y una palabra-código con un cierto error añadido, que no pueden ser recuperadas: el síndrome por ser de ②, de origen aleatorio y sin descodificación eficiente, y la palabra-código por ser de ①, pero con un error añadido de peso mayor que t. (*Sección 5.3*)

Por último, cabe destacar que, una vez generadas las claves, la encriptación y desencriptación de **Classic McEliece** es completamente determinística, mientras que tanto en **BIKE** como en **HQC** las encriptaciones se basan en vectores aleatorios, con pesos previamente fijados en sus parámetros, y los métodos de descodificación son probabilísticos, con una cierta tasa de fallo de descodificación o DFR.

Sin embargo, como se aprecia en la *Tabla 6*, ésta es extremadamente baja en la práctica, donde los criptosistemas se diseñan con niveles de seguridad equivalentes a un AES de 128, 192 y 256 bits.

Una vez repasadas las principales diferencias en las bases teóricas de los distintos criptosistemas presentes en la cuarta ronda de PQC del NIST, vamos a analizar su desempeño y compararlo con el del estandarizado CRYTALS-Kyber, para ver bajo qué circunstancias las alternativas propuestas podrían sustituirlo.

Antes de seguir, merece la pena realizar un apunte más práctico de cara a las aplicaciones reales de los criptosistemas de clave pública estudiados. Desde un punto de vista funcional, los criptosistemas de clave privada o simétrica suelen ser más óptimos que los de clave pública o asimétrica para transmitir mensajes grandes [23], y además siguen siendo seguros frente a ataques con ordenadores cuánticos, como comentamos al principio en la *Sección 1.1*. Por ello, resulta más conveniente emplear los criptosistemas de clave asimétrica para establecer una clave simétrica entre dos partes, con la cual poder después intercambiar información de forma más eficiente.

Así, surgen los mecanismos de encapsulación de claves o KEM (*Key-Encapsulation Mechanism*). Éstos consisten en generar y emplear un criptosistema de clave pública para transmitir un mensaje $\vec{m}$, a partir del cual se puede generar una clave simétrica M mediante una función de una vía $\mathcal{K}$ previamente acordada [23]. En este caso, el proceso de encriptar $\vec{m}$ se suele llamar "encapsulación", mientras que el de desencriptar $\vec{m}$ y generar $M = \mathcal{K}(\vec{m})$ se denomina "desencapsulación", siendo ésta la notación de la *Tabla 7*.

Tabla 7: Comparativa del desempeño de los candidatos de la ronda 4 de PQC del NIST y el seleccionado en la ronda 3. (Modificada de [61])

Implementación	PK	SK	Texto cifrado	Generación de claves	Encapsulación	Desencapsulación
mceliece460896f	524.160	13.608	156	117.301	81	264
BIKE	3.082	418	3.144	1.823	223	3.887
HQC	4.522	64	9.042	204	465	755
Kyber-768	1.184	32	1.088	213	249	275

Se muestran los tamaños en bytes de las claves pública PK y privada SK y del texto cifrado, importantes de cara al almacenamiento y el ancho de banda necesario en cada criptosistema, así como los tiempos de ejecución de cada etapa suya como KEM. Se consideran implementaciones con un nivel de seguridad 3.

En primer lugar, como se aprecia en la *Tabla 7*, CRYSTALS-Kyber presenta un conjunto de características óptimo en rasgos generales, como cabría esperar del único vencedor de la ronda 3. No hay demasiadas dudas sobre su mejor desempeño, pero, debido a

la continuidad y entereza prolongada en el tiempo de los criptosistemas de clave pública basados en códigos correctores de errores, como el inamovible criptosistema de McEliece, se le puede considerar más inseguro que los aspirantes Classic McEliece, BIKE y HQC de la ronda 4. De este modo, se plantea la implementación de una de estas alternativas con la intención de ganar seguridad a cambio de algún sacrificio en tamaños y/o velocidad.

- **Classic McEliece (CM)**: Tamaños monstruosos y generación de claves lentas, pero una vez creadas es el más rápido de aplicar (encapsulación y desencapsulación), superando incluso a CRYSTALS-Kyber. Sería la mejor opción en protocolos donde se puedan reusar las claves varias veces, buscando claves estáticas y con mucha seguridad. Además, presenta el texto cifrado más pequeño con gran diferencia, necesitando mucho menos ancho de banda que el resto, por lo que también sería la mejor opción cuando se requieran textos cifrados de poco tamaño o una gran cantidad de los mismos [61].
- **BIKE** y **HQC**: Sus claves son mucho más asequibles por tiempo y espacio que las de CM. Ambas tienen una mejor implementación que CM orientada a KEMs dentro de una PKI (*Public Key Infraestructure*) [61], donde, además de la integridad de la clave simétrica establecida, se busca la autenticación de las partes mediante firmas digitales y certificados de una tercera parte de confianza como una CA (*Certificate Authority*) [62, 63]. En este caso, se buscan claves efímeras para cada sesión, que sean rápidas de generar y distribuir. Este paradigma es ampliamente empleado en Internet, como en todas las URL que comienzan por "*https*".

Dentro de esta línea, HQC parece más prometedor al ser más rápido que BIKE, pero su mucho mayor texto cifrado hace que BIKE sea una mejor opción cuando haya restricciones considerables en el ancho de banda [61]. En cualquier caso, ninguno de los dos puede desbancar a Kyber.

Por último, cabe destacar que **Classic McEliece (CM)** se basa en los clásicos códigos de Goppa binarios, blindados con una robustez ampliamente constatada, mientras que **BIKE** y **HQC** se basan en códigos cuasicíclicos, más recientes y con una mayor probabilidad de que se encuentren ataques capaces de explotar su estructura, por ejemplo a partir de sus DFR, las cuales todavía se siguen estudiando.

A día de hoy, no se puede prever que opciones serán estandarizadas, quedando a la espera de nuevas noticias a lo largo de este año.

7. Conclusiones

En este trabajo se ha introducido la amenaza de la computación cuántica en la ciberseguridad y los protocolos de comunicación empleados a día de hoy, con la aparición de nuevos algoritmos capaces de resolver los problemas difíciles en los que se basa la criptografía de clave pública actual. Se ha constatado la importante labor del NIST en el proceso de estandarización de criptografía postcuántica (PQC) de clave pública, la cual permite desarrollar nuevos sistemas con la tecnología actual que nos protejan de los futuros ordenadores cuánticos a gran escala, comentando la estandarización en 2022 de CRYSTALS-Kyber, basado en retículos, como el único vencedor de encriptación de clave pública en la ronda 3.

Nos hemos centrado en el estudio de las tres alternativas que todavía se proponen en la ronda 4: Classic McEliece, BIKE y HQC, todos ellos pertenecientes a una nueva rama de criptografía basada en códigos correctores de errores; teniendo como proncipal baza frente a CRYSTALS-Kyber una mayor confianza en su seguridad, especialmente relevante ante la gran incertidumbre en el desarrollo de nuevos ataques cuánticos. Esta rama aplica conceptos de álgebra y la teoría de números, como operaciones modulares sobre un anillo de polinomios, para proteger nuestros datos frente a futuros atacantes cuánticos. Las funciones de una vía con trampa, en las que se sustentan estos nuevos criptosistemas de clave pública, giran en torno a problemas difíciles para ordenadores clásicos y cuánticos, como la descodificación de códigos lineales aleatorios y la descodificación de síndromes.

Se ha seguido un enfoque autocontenido y progresivo, desde los conceptos más básicos de códigos correctores de errores hasta los fundamentos e implementaciones de los tres criptosistemas propuestos, pasando por sistemas más sencillos y clásicos, como el de McEliece, e incluso realizando algún pequeño ejemplo práctico. También se han comparado las características tanto teóricas como prácticas de los tres candidatos entre sí, y su desempeño frente al escogido CRYSTALS-Kyber, apreciando que presentan una variedad deseable en sus bases de seguridad, lo cual resultaría útil para futuras implementaciones híbridas, y sus perfiles de desempeño, óptimos en distintos contextos.

Como es habitual en criptografía, se espera la aparición de nuevos ataques tanto clásicos como cuánticos para los nuevos sistemas de PQC y un desarrollo constante de la comunidad, tanto a nivel teórico por matemáticos como práctico por informáticos e ingenieros, que siga mejorando la seguridad y la eficiencia de estos criptosistemas.

Índice de figuras

Índice de tablas

Referencias

[1] W. Diffie y M. Hellman. "New directions in cryptography". En: *IEEE Transactions on Information Theory* **22**, 6 (1976).

[2] R. L. Rivest, A. Shamir y L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems". En: *Communications of the ACM* **21**, 2 (1978).

[3] V. S. Miller. "Use of Elliptic Curves in Cryptography". En: *Advances in Cryptology — CRYPTO '85 Proceedings*. Springer Berlin Heidelberg, 1986. ISBN: 978-3-540-39799-1.

[4] N. Koblitz. "Elliptic curve cryptosystems". En: *Mathematics of Computation* **48**, (1987).

[5] L. Chen *et al. Report on Post-Quantum Cryptography*. Inf. téc. NISTIR 8105. National Institute of Standards and Technology, 2016.

[6] P. W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". En: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994.

[7] P. W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". En: *SIAM Journal on Computing* **26**, 5 (1997).

[8] L. K. Grover. "A fast quantum mechanical algorithm for database search". En: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Association for Computing Machinery, 1996. ISBN: 0897917855.

[9] L. K. Grover. "Quantum Mechanics Helps in Searching for a Needle in a Haystack". En: *Phys. Rev. Lett.* **79**, 325 (1997).

[10] M. Dworkin *et al. Advanced Encryption Standard (AES)*. [online], `https://doi.org/10.6028/NIST.FIPS.197` (Accessed May 24, 2024). 2001.

[11] Wikipedia contributors. *Quantum error correction — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=Quantum_error_correction&oldid=1223384218`. [Online; accessed 24-May-2024].

[12] M. Mosca. "Cybersecurity in an Era with Quantum Computers: Will We Be Ready?" En: *IEEE Security & Privacy* **16**, 5 (2018).

[13] National Institute of Standars and Technology. *Post-Cuantum Cryptography*. https://csrc.nist.gov/projects/post-quantum-cryptography. [Online; accessed 25-May-2024].

[14] M. Campagna *et al. Quantum Safe Cryptography and Security: An Introduction, Benefits, Enablers and Challenges*. Inf. téc. White Paper No. 8. European Telecommunications Standards Institute, 2015.

[15] CRYSTALS Team: R. Avanzi *et al. Cryptographic Suite for Algebraic Lattices*. https://pq-crystals.org/. [Online; accessed 25-May-2024].

[16] BIKE Team: N. Aragon *et al. BIKE - Bit Flipping Key Encapsulation*. https://bikesuite.org/. [Online; accessed 25-May-2024].

[17] Classic McEliece Team: D. J. Bernstein *et al. Classic McEliece*. https://classic.mceliece.org/. [Online; accessed 25-May-2024].

[18] HQC Team: C. A. Melchor *et al. HQC (Hamming Quasi-Cyclic)*. https://pqc-hqc.org/. [Online; accessed 25-May-2024].

[19] SIKE Team: D. Jao *et al. SIKE - Supersingular Isogeny Key Encapsulation*. https://sike.org/. [Online; accessed 25-May-2024].

[20] SIKE Team postscript. *Foreword and postscript: SIKE and SIDH are insecure and should not be used.* https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/sike-team-note-insecure.pdf. In NIST's PQC Round 4 Submissions [Online; accessed 25-May-2024].

[21] Y. Lindell. *Lecture notes in Foundations of Cryptography (89-856)*. 2017.

[22] R. J. McEliece. "A public-key cryptosystem based on algebraic coding theory". En: *DSN PR* **42-44**, 114 (1978).

[23] V. Weger, N. Gassner y J. Rosenthal. *A Survey on Code-Based Cryptography*. 2024. arXiv: 2201.07119 [cs.CR].

[24] R. Singleton. "Maximum distance q-nary codes". En: *IEEE Transactions on Information Theory* **10**, 2 (1964).

[25] H. Niederreiter. "Knapsack-type cryptosystems and algebraic coding theory." En: *Problems of Control and Information Theory* **15**, 159 (1986).

[26] V. D. Goppa. "A new class of linear correcting codes." En: *Problemy Peredachi Informatsii* **6**, 3 (1970).

[27] E. Berlekamp. "Goppa codes". En: *IEEE Transactions on Information Theory* **19**, 5 (1973).

[28] T. Lange. *Code-based cryptography III - Goppa codes: definition and usage.* `https://www.youtube.com/watch?v=qisORKNShvo&t=207s`. 2021.

[29] C. Munuera y J. G. Tena. *Codificacion de la informacion.* (Capítulos 11 y 12). Universidad de Valladolid. Secretariado, 1997.

[30] N. Patterson. "The algebraic decoding of Goppa codes". En: *IEEE Trans. Inf. Theory* **21**, 203 (1975).

[31] E. Berlekamp; R. McEliece y H. van Tilborg. "On the inherent intractability of certain coding problems (Corresp.)" En: *IEEE Transactions on Information Theory* **24**, 3 (1978).

[32] Tanja D. J. Bernstein; T. Lange y C. Peters. "Wild McEliece". En: *Selected Areas in Cryptography.* Springer Berlin Heidelberg, 2011.

[33] A. Couvreur; A. Otmani y J.-P. Tillich. "Polynomial Time Attack on Wild McEliece Over Quadratic Extensions". En: *IEEE Transactions on Information Theory* **63**, 1 (2017).

[34] V. M. Sidelnikov. "A public-key cryptosystem based on binary Reed-Muller codes". En: *Discrete Mathematics and Applications* **4**, 3 (1994).

[35] L. Minder y A. Shokrollahi. "Cryptanalysis of the Sidelnikov Cryptosystem". En: *Advances in Cryptology - EUROCRYPT 2007.* Springer Berlin Heidelberg, 2007.

[36] M. Baldi *et al.* "A variant of the McEliece cryptosystem with increased public key security." En: *WCC 2011- Workshop on coding and cryptography.* 2011.

[37] K. Khathuria; J. Rosenthal y V. Weger. "Weight two masking of the Reed-Solomon structure in conjunction with list decoding". En: *Proceedings of 23rd International Symposium on Mathematical Theory of Networks and Systems.* 2018.

[38] K. Khathuria; J. Rosenthal y V. Weger. "Encryption scheme based on expanded Reed-Solomon codes". En: *Advances in Mathematics of Communications* **15**, 2 (2021).

[39] T. P. Berger *et al.* "Generalized subspace subcodes with application in cryptology." En: *IEEE Transactions on Information Theory* **65**, 8 (2019).

[40] A. Couvreur *et al.* "Distinguisher-based attacks on public-key cryptosystems using Reed–Solomon codes". En: *Designs, Codes and Cryptography* **73**, (2014).

[41] A. Couvreur y M. Lequesne. “On the Security of Subspace Subcodes of Reed–Solomon Codes for Public Key Encryption”. En: *IEEE Transactions on Information Theory* **68**, 1 (2022).

[42] T. P. Berger *et al.* “Reducing Key Length of the McEliece Cryptosystem”. En: *Progress in Cryptology – AFRICACRYPT 2009*. Springer Berlin Heidelberg, 2009.

[43] C. Monico; J. Rosenthal y A. Shokrollahi. “Using low density parity check codes in the McEliece cryptosystem”. En: *2000 IEEE International Symposium on Information Theory (Cat. No.00CH37060)*. 2000.

[44] M. Baldi; M. Bodrato y F. Chiaraluce. “A New Analysis of the McEliece Cryptosystem Based on QC-LDPC Codes”. En: *Security and Cryptography for Networks*. Springer Berlin Heidelberg, 2008.

[45] R. Misoczki *et al.* “MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes”. En: *2013 IEEE International Symposium on Information Theory*. 2013.

[46] V. G. Umaña y G. Leander. “Practical key recovery attacks on two McEliece variants”. En: *Proc. 2nd Int. Conf. on Symbolic Computation and Cryptography*. 2009.

[47] J.-P. Tillich A. Otmani; Tillich y L. Dallot. “Cryptanalysis of Two McEliece Cryptosystems Based on Quasi-Cyclic Codes”. En: *Mathematics in Computer Science* **3**, (2010).

[48] V. M. Sidelnikov y S. O. Shestakov. “On an encoding system constructed on the basis of generalized Reed–Solomon codes.” En: *Diskretnaya Matematika* **4**, 3 (1992).

[49] Y. X. Li, R. H. Deng y X. M. Wang. “Equivalence of McEliece’s and Niederreiter’s Public-Key Cryptosystems.” En: *IEEE Transactions on Information Theory* **40**, 1 (1994).

[50] R. W. Hamming. “Error detecting and error correcting codes”. En: *The Bell System Technical Journal* **29**, 2 (1950).

[51] D. J. Bernstein *et al. ClassicMcEliece: conservative code-based cryptography: cryptosystem specication*. 2022. NIST submission: `Round4` (mceliece-spec-20221023.pdf). URL: `https://classic.mceliece.org/nist.html`.

[52] D. J. Bernstein *et al. ClassicMcEliece: conservative code-based cryptography: design rationale*. 2022. NIST submission: `Round4` (mceliece-rationale-20221023.pdf). URL: `https://classic.mceliece.org/nist.html`.

[53] R. Gallager. "Low-density parity-check codes". En: *IRE Transactions on Information Theory* **8**, 1 (1962).

[54] *Sparse matrix — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Sparse_matrix&oldid=1217667757. [Online; accessed 15-May-2024].

[55] BIKE Website; Specification Document. *BIKE: Bit Flipping Key Encapsulation — Round 4 Submission*. https://bikesuite.org/. NIST PQC Call for Proposals. 2022.

[56] I. S. Reed y G. Solomon. "Polynomial Codes Over Certain Finite Fields". En: *Journal of the Society for Industrial and Applied Mathematics* **8**, 2 (1960).

[57] V. Guruswami y E. Blais. *Introduction to Coding Theory, Notes 6: Reed Solomon, BCH, Reed-Muller, and concatenated codes*. https://www.cs.cmu.edu/~venkatg/teaching/codingtheory/notes/notes6.pdf. Lecture Notes, 2010. [Online; accessed 19-May-2024].

[58] D.E. Muller. "Application of Boolean algebra to switching circuit design and to error detection". En: *Transactions of the I.R.E. Professional Group on Electronic Computers* **EC-3**, 3 (1954).

[59] G. D. Forney. "Concatenated Codes". PhD thesis. Massachusetts Institute of Technology, 1965.

[60] HQC Website; Specification Document. *Hamming Quasi-Cyclic (HQC) — Fourth round version*. https://pqc-hqc.org/documentation.html. NIST PQC Call for Proposals. Updated version 01/10/2024.

[61] A. Robinson; N. Sendrier; C. A. Melchor y E. Persichetti. "BIKE, Classic McEliece, HQC. Comparing and contrasting NIST PQC 4th Round KEMs". En: *5th PQC Standardization Conference*. PANEL: 4th Round - BIKE / HQC / Classic McEliece (https://csrc.nist.gov/Events/2024/fifth-pqc-standardization-conference, Session 3). 2024.

[62] JumpCloud. *What Is Public Key Infrastructure (PKI)?* https://www.youtube.com/watch?v=uVaUgrxjMe0. [Online; accessed 29-May-2024].

[63] D. Kehn and J. Crume. *Tech Talk: What is Public Key Infrastructure (PKI)?* https://www.youtube.com/watch?v=uVaUgrxjMe0. IBM Technology [Online; accessed 29-May-2024].