Ciencias Matemóticas, Vol. 33, No. 1, Pag. 8-12, 2019

Generación de matrices circulantes invertibles y su aplicación al criptosistema McEliece
Generating invertible circulant matrices and their application to the McEliece cryptosystem

Ernesto Dominguez Fiallo[1]*, Frank E. Acosta Fonseca[1], Luis R. Piñeiro Díaz[1]

Resumen En este artículo se propone un algoritmo para generar matrices circulantes invertibles sobre $\mathbb{F}_2$ de orden primo. La generación de tales matrices se realiza actualmente de forma aleatoria. Se muestra como emplear el algoritmo propuesto para generar llaves en el criptosistema McEliece basado en códigos QC-LDPC (el cual es una de las variantes más importantes propuestas para estándar asimétrico post cuántico), proponiendo otros dos algoritmos para construir las matrices S y Q que componen la llave privada y con las cuales se genera la llave pública. Estos algoritmos reducen considerablemente el proceso más costoso del esquema: la generación de llaves.

Abstract In this paper an algorithm is proposed to generate invertible circulant matrices over $\mathbb{F}_2$ of prime order. The generation of such matrices is currently carried out randomly. It is shows how to use the proposed algorithm to generate keys in the McEliece cryptosystem based on QC-LDP codes (which is one of the most important variants proposed for post quantum asymmetric standard), proposing two other algorithms to build the matrices S and Q which make up the private key and with which the public key is generated.

Palabras Clave
matrices circulantes invertibles, criptosistema McEliece, criptografía post cuántica

Keywords
invertible circulant matrices, McEliece cryptosystem, post-quantum cryptography

[1] *Instituto de Criptografía, Universidad de La Habanaó, La Habana, Cuba, edominguezfiallo@nauta.cu, frankorazonero@nauta.cu, lrp@matcom.uh.cu*
***Autor para Correspondencia, Corresponding Author**

Introducción

El criptosistema McEliece es una de las alternativas más importantes a estandar post cuántico [1]. Una de sus variantes más prominentes es la basada en códigos Cuasi-Cíclicos con Baja Densidad en la Matriz de Control (QC-LDPC *siglas en inglés*)[7]. Para generar las llaves se emplean matrices circulantes (ver definición 7) invertibles. En la actualidad no existen algoritmos eficientes para generar tales matrices y se construyen de forma aleatoria chequeando su invertibilidad.

En este artículo se propone un algoritmo para generar matrices circulantes invertibles sobre $\mathbb{F}_2$ de orden primo. A partir de dicho algoritmo, se proponen otros dos algoritmos: uno probabilístico (con probabilidad cercana a 1) para generar matrices circulantes invertibles densas (ver definición 10) y uno determinístico para generar matrices circulantes invertibles sparse (ver definición 9). Para cada algoritmo propuesto se realiza un análisis de su complejidad algorítmica demostrando la eficiencia computacional de los mismos. Estos algoritmos mejoran el proceso de generación de llaves de la variante del criptosistema McEliece considerado.

1. Preliminares

Sean $\mathbb{F}_q$ el campo finito de q elementos y $\mathbb{F}_q^n$ el $\mathbb{F}_q$ espacio vectorial cuyos elementos son vectores de n componentes en $\mathbb{F}_q$.

Definición 1 *Sean $k,n \in \mathbb{N}$ tales que $1 \leq k < n$. Un código lineal $\mathscr{C}$ es subespacio vectorial de $\mathbb{F}_q^n$ de dimensión k. A $\mathscr{C}$ se le llama código lineal sobre $\mathbb{F}_q$ con longitud n y dimensión k y se denota por $[n,k]_q$. A los elementos de $\mathscr{C}$ se les denomina palabras de código.*

Definición 2 *Sea $\mathscr{C}$ un $[n,k]_q$ código lineal. Una matriz G de tamaño $k \times n$ cuyos vectores filas formen una base de $\mathscr{C}$ como espacio vectorial de dimensión k es llamada matriz generadora de $\mathscr{C}$. Si G tiene la forma $G = (I_k|A)$, donde I_k es la submatriz identidad de orden k y A es una submatriz de tamaño $k \times (n-k)$, entonces se dice que G está dada en forma estandar o en forma sistemática.*

La codificación con códigos lineales es muy simple debido a su descripción algebraica. Dado un mensaje $u \in \mathbb{F}_q^k$, se

codifica en la palabra de código c mediante la multiplicación $c = uG$.

Definición 3 *A una matriz H de tamaño $(n-k)\times n$ que satisface $Hc^T = 0\ \forall c \in \mathscr{C}$ se le denomina matriz de control del $[n,k]_q$ código lineal $\mathscr{C}$.*

Definición 4 *La distancia de Hamming entre dos vectores $u, v \in \mathbb{F}_q^n$ es el número de coordenadas en las cuales u y v son diferentes. El peso de Hamming w de un vector $u \in \mathbb{F}_q^n$ es el número de coordenadas distintas de cero, o sea, $w(u) = d(u,0)$.*

Definición 5 *La distancia mínima $d_{\text{mín}}$ de un código lineal $\mathscr{C}$ es la distancia de Hamming más pequeña entre dos palabras de código cualesquiera y diferentes $d_{\text{mín}}(\mathscr{C}) = \text{mín}_{c_i \neq c_j}\{d(c_i,c_j) : c_i, c_j \in \mathscr{C}\}$.*

Cuando se conoce la distancia mínima del código $\mathscr{C}$, entonces se dice que $\mathscr{C}$ es un $[n,k,d_{\text{mín}}]_q$ código lineal. La distancia mínima permite determinar el número de errores que puede corregir el código.

Teorema 6 *[6] Si $\mathscr{C}$ es un $[n,k,d_{\text{mín}}]_q$ código lineal, entonces $\mathscr{C}$ puede corregir hasta $\lfloor \frac{d_{\text{mín}}-1}{2} \rfloor$ errores.*

Definición 7 *Una matriz circulante de orden p es una matriz de la forma* $A = \begin{bmatrix} a_0 & a_1 & \dots & a_{p-1} \\ a_{p-1} & a_0 & \dots & a_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \dots & a_0 \end{bmatrix}$ *Si los elementos de dicha matriz pertenecen al campo $\mathbb{F}_2$, entonces se denomina matriz circulante binaria.*

Si se considera el anillo de polinomios $\mathbb{F}_2[x]/(x^p+1)$, entonces la correspondencia

$$A \to a(x) = \sum_{i=0}^{p-1} a_i \cdot x^i$$

es un isomorfismo de anillos y permite asociar cada matriz circulante con un polinomio que tiene como coeficientes a los elementos de la primera fila de la matriz.

En lo adelante, solo se considerarán códigos lineales binarios, o sea, el campo finito $\mathbb{F}_q$ será el campo binario $\mathbb{F}_2$.

1.1 Códigos QC-LDPC

Los códigos QC-LDPC son la unión de dos familias de códigos lineales: los códigos QC [8] y los códigos LDPC [3].

Definición 8 *Un código QC es un código lineal de longitud $n = p\cdot n_0$ y dimensión $k = p\cdot k_0$ en el que cada desplazamiento cíclico de una palabra de código en n_0 posiciones es una palabra de código. Su matriz de control es de la forma* $H = \begin{bmatrix} H_{00} & H_{01} & \dots & H_{0(n_0-1)} \\ H_{10} & H_{11} & \dots & H_{1(n_0-1)} \\ \vdots & \vdots & \ddots & \vdots \\ H_{(r_0-1)0} & H_{(r_0-1)1} & \dots & H_{(r_0-1)(n_0-1)} \end{bmatrix}$ *donde cada submatriz $H_{ij},\ 0 \le i \le r_0-1,\ 0 \le j \le n_0-1$ es una matriz circulante de orden p.*

En particular, es de interés el caso en que $r_0 = 1$. Se cumple entonces que $k_0 = n_0 - 1$, la matriz de control del código tiene la forma $H = \begin{bmatrix} H_0 & H_1 & \dots & H_{n_0-1} \end{bmatrix}$ y H es de rango completo (caso de interés) si al menos, uno de los bloques $H_i,\ i = 0,\dots,n_0-1$, es no singular.

Definición 9 *Una matriz de tamaño $m\times n$ se denomina sparse si $\frac{\#elementos\ \neq 0}{m\cdot n} \approx \frac{\log(m\cdot n)}{m\cdot n}$.*

Definición 10 *Una matriz de tamaño $m\times n$ se denomina densa si $\frac{\#elementos\ \neq 0}{m\cdot n} \approx \frac{3}{4}$.*

Definición 11 *Un código es LDPC si su matriz de control es sparse.*

Definición 12 *Un código QC-LDPC es una clase particular de un código QC caracterizada por matrices de control que son sparse.*

1.2 Criptosistema McEliece basado en códigos QC-LDPC

Generación de llaves: Dado un código LDPC capaz de corregir t errores, el mismo se representa por su matriz de control H en la cual se tiene que $n = n_0\cdot p,\ k = k_0\cdot p$ y se toma $k_0 = n_0 - 1$. Lo anterior significa que $n-k = p$ y que la matriz H es de la forma $H = \begin{bmatrix} H_0 & H_1 & \dots & H_{n_0-1} \end{bmatrix}$ donde cada submatriz $H_i,\ i \in [0, n_0-1]$ es una matriz circulante de orden p con peso d_v en cada fila y columna. Como H es de rango completo, se puede asumir sin pérdida de generalidad que H_{n_0-1} es invertible.

A partir de la matriz H, se obtiene la matriz generadora G que tendrá la forma $G = \begin{bmatrix} \mathrm{I}_{k\times k} | \mathrm{A}_{k\times(n-k)} \end{bmatrix}$ donde $\mathrm{I}_{k\times k}$ es la matriz identidad de orden k y $A_{k\times(n-k)} = \begin{pmatrix} \left(H_{n_0-1}^{-1}H_0\right)^T \\ \left(H_{n_0-1}^{-1}H_1\right)^T \\ \vdots \\ \left(H_{n_0-1}^{-1}H_{n_0-2}\right)^T \end{pmatrix}$.

Se seleccionan dos matrices S y Q ambas invertibles con estructura QC siendo S una matriz densa de orden k y Q una matriz sparse de orden n con peso $m > 1$ en cada fila. La matriz QC generadora del código público G' se obtiene a partir de G del siguiente modo: $G' = S^{-1}GQ^{-1}$.

El uso de la matriz Q influye directamente en la capacidad de corregir errores del código. Si el código secreto puede corregir hasta t errores, para que la decodificación sea correcta, el número máximo de errores intencionales t' que se pueden introducir en el esquema durante el cifrado es $t' \le t/m$.

La llave pública es el par (G', t') y su tamaño es $(n_0-1)\cdot p$ bits, mientras que la llave secreta son las matrices (H, S, Q).
Cifrado: Sea $u \in \mathbb{F}_2^k$ el mensaje a transmitir. Se cifra del siguiente modo: $y = uG' + e$, donde e es el vector error introducido seleccionado aleatoriamente tal que $w(e) = t' \le t/m$.
Descifrado: Dado el vector recibido y, se calcula $y\cdot Q = x = u\cdot S^{-1}\cdot G + e\cdot Q$. Se aplica el algoritmo eficiente de decodificación $\mathscr{D}$ y se obtiene $\mathscr{D}(x) = u\cdot S^{-1}$. Luego se multiplica a

la derecha por la matriz S y se recupera el mensaje original $\mathscr{D}(x)\cdot S = u$.

Detalles sobre la seguridad del esquema y selección de parámetros se dan en [2]. El punto clave aquí es que para garantizar seguridad, p debe ser impar, por lo que en particular puede ser primo.

2. Algoritmo para generar matrices circulantes invertibles de orden primo

2.1 Resultados preliminares

Proposición 13 *[4] Una matriz circulante binaria de orden n es invertible si y solo si su correspondiente polinomio en el anillo $\mathbb{F}_2[x]/(x^n+1)$ es primo relativo a x^n+1.*

De la proposición anterior se deduce que para investigar la invertibilidad de las matrices circulantes, es muy útil conocer la factorización del polinomio x^n+1 en $\mathbb{F}_2[x]$, lo cual conduce a la definición general de polinomio ciclotómico.

Definición 14 *Sea k impar y ζ una k-ésima raíz primitiva de la unidad sobre $\mathbb{F}_2$. Entonces el polinomio $Q_k(x) = \prod_{s:\gcd(s,k)=1,\ s\le k} (x-\zeta^s)$ es llamado k-ésimo polinomio ciclotómico sobre $\mathbb{F}_2$.*

En lo adelante $o_k(2)$ denota el orden de 2 en el grupo $\mathbb{Z}_k^*$ y $\phi(d)$ denota la función de Euler.

Teorema 15 *[5] Para n impar se tiene:*

1. $x^n+1 = \prod_{k|n} Q_k(x)$ *en* $\mathbb{F}_2[x]$.
2. *los coeficientes de $Q_k(x)$ están en $\mathbb{F}_2$.*

Teorema 16 *[5] Los polinomios $Q_k(x)$ se factorizan en $\frac{\phi(k)}{o_k(2)}$ polinomios mónicos distintos e irreducibles de grado $o_k(2)$ en $\mathbb{F}_2[x]$.*

Sea $f \in \mathbb{F}_2[x]$ y $\psi(f)$ el número de polinomios de grado menor que son primos relativos a f en $\mathbb{F}_2[x]$. En [4] se demuestra que si $\gcd(f,g)=1$ entonces $\psi(fg)=\psi(f)\psi(g)$. El número de matrices circulantes binarias e invertibles de orden n es, según la proposición 13, $\psi(x^n+1)$ y utilizando los teoremas anteriores se tiene la siguiente proposición.

Proposición 17 *[4] Sea $n = 2^\alpha m$, donde m es impar y α es un entero positivo. Entonces se tiene que $\psi(x^n+1) = 2^n \prod_{k|m} \left(1-2^{-o_k(2)}\right)^{\phi(k)/o_k(2)}$.*

A continuación se dan dos proposiciones que serán muy útiles para el análisis de las matrices circulantes binarias que son invertibles.

Proposición 18 *Sea $f \in \mathbb{F}_2[x]/(x^n+1)$. Entonces f tiene peso par si y solo si es múltiplo de $x+1$.*

Demostración. Un polinomio f tiene peso par si y solo si $f(1)=0$ pero $f(1)=0$ si y solo si f es múltiplo de $x+1$. ■

Proposición 19 *Toda matriz circulante de orden n sobre $\mathbb{F}_2$ con un número par de unos en una fila es singular.*

Demostración. Si una matriz circulante tiene un número par de unos en una fila, entonces su polinomio correspondiente tiene peso par. Por la proposición 18, dicho polinomio es múltiplo de $x+1$. Aplicando la proposición 13, se tiene el resultado enunciado. ■

2.2 Algoritmo

Sea n un número primo tal que $o_n(2) = n-1$. Los resultados anteriores implican que el polinomio x^n+1 tiene solo dos factores irreducibles: Q_1, Q_n. Por la proposición 19, todos los polinomios de peso impar excepto Q_n corresponden a matrices circulantes invertibles. De esta forma se tiene un algoritmo para generar matrices circulantes invertibles binarias de orden primo con peso d_v en cada fila y columna del conjunto de $\binom{n}{d_v}$ matrices de este tipo.

Algoritmo 1: Generación de matrices circulantes invertibles binarias de orden n primo con peso d_v en cada fila y columna

Data: n primo tal que $o_n(2) = n-1$, d_v impar y $d_v < n$
Result: un polinomio f correspondiente a una matriz circulante invertible binaria de orden n con d_v unos en cada fila y columna

1 Seleccionar $t_0, \ldots, t_{d_v-1}$ del conjunto $\{0,\ldots,n-1\}$ de forma independiente e igualmente distribuidos sin reemplazo.
2 Devolver el polinomio $f(x) = \sum_{i=0}^{d_v-1} x^{t_i}$

2.3 Aplicación al McEliece basado en códigos QC-LDPC

Actualmente no existe un procedimiento para generar las matrices S y Q en el criptosistema McEliece basado en códigos QC-LDPC, por lo que dicho proceso se hace de forma aleatoria lo cual hace más costoso la generación de llaves. Basados en el algoritmo 1, se proponen dos algoritmos: uno probabilístico para generar la matriz S y otro determinístico para generar la matriz Q.

Se propone usar el algoritmo 2 para construir S durante el proceso de generación de llaves del esquema.

Teorema 20 *Sea p un número primo tal que $o_p(2) = p-1$. Sea S la matriz construída por el algoritmo 2. Entonces*

$$P(S \text{ sea invertible}) \ge \left(1-\frac{1}{2^{p-1}}\right)^{k_0}$$

Demostración. Sea $\overline{S}$ la matriz generada por los dos primeros pasos del algoritmo 2. Si cada polinomio correspondiente

Algoritmo 2: Construcción de la matriz S

Data: k_0, p
Result: matriz binaria de $k_0 \times k_0$ submatrices circulantes de orden p

1 Generar cada bloque sobre la diagonal principal independiente e igualmente distribuido del conjunto de todas las matrices circulantes binarias de orden p con peso impar;
2 Generar cada bloque fuera de la diagonal principal independiente e igualmente distribuido del conjunto de todas las matrices circulantes binarias de orden p con peso par;
3 Permutar cada fila bloque de la matriz por una permutación seleccionada independiente e igualmente distribuida del espacio de todas las permutaciones de k_0 elementos.

a las submatrices de la diagonal principal es invertible entonces $\overline{S}$ es invertible. La probabilidad de que cada polinomio de las submatrices de la diagonal principal sea invertible en $\mathbb{F}_2[x]/(x^p+1)$ es $\frac{\psi(x^p+1)}{2^p}$. Trabajando se tiene que $\frac{\psi(x^p+1)}{2^p} = 1 - \frac{1}{2^{p-1}}$. Como se tienen k_0 de estos polinomios elegidos al azar y de forma independiente y como no necesariamente para que $\overline{S}$ sea invertible los polinomios correspondientes a las submatrices de la diagonal principal deben ser invertibles, se tiene el resultado deseado. ■

Es de notar que aunque el algoritmo 2 es probabilístico, la probabilidad de éxito es muy próxima a 1 para todos los valores reales de los parámetros [2].

La matriz Q tiene peso m constante en cada fila, tiene que ser invertible y está compuesta de $n_0 \times n_0$ submatrices circulantes de orden p. Luego, m tiene que ser impar y lo escribiremos convenientemente como $m = u(n_0-1)+v$, donde $u \geq 2$ es par y v impar. Bajo estas condiciones se propone el algoritmo 3 para construir Q.

Teorema 21 *Sea p primo tal que $o_p(2) = p-1$. Supongamos que los valores n_0, u y v satisfacen (cierto en la práctica)*

$$n_0! \cdot (max\{u,v\})^{n_0} < p$$

Entonces el algoritmo 3 siempre produce una matriz invertible Q.

Demostración. Aplicando la fórmula de Leibniz[1] para el determinante de Q, el peso del determinante es a lo más $n_0! \cdot (max\{u,v\})^{n_0} < p$. Si la desigualdad del teorema se cumple, entonces el determinante de Q debe ser un polinomio de peso impar distinto de $\sum_{i=0}^{p-1} x^i$. Luego el determinante es invertible y por ende no es un divisor de cero, lo cual garantiza la invertibilidad de la matriz Q ■

Algoritmo 3: Construcción de la matriz Q

Data: $n_0,\ p,\ u,v \in \mathbb{N},\ u \geq 2$ *par*, v *impar*
Result: matriz Q con peso en cada fila y columna $m = u(n_0-1)+v$ y compuesta de $n_0 \times n_0$ bloques circulantes de orden p

1 Para $i = 1$ hasta n_0 hacer:
2 Seleccionar v números diferentes $d_1^i, d_2^i, \ldots, d_v^i$ del conjunto $\{1,2,\ldots,p\}$ de forma independiente e igualmente distribuidos.
3 Crear una matriz circulante D^i de orden p con unos en las posiciones $d_1^i, d_2^i, \ldots, d_v^i$ de la primera fila.
4 Fin
5 Para $i = 1$ hasta $n_0(n_0-1)$ hacer:
6 Seleccionar u números diferentes $b_1^i, b_2^i, \ldots, b_u^i$ del conjunto $\{1,2,\ldots,p\}$ de forma independiente e igualmente distribuidos.
7 Crear una matriz circulante B^i de orden p con unos en las posiciones $b_1^i, b_2^i, \ldots, b_u^i$ de la primera fila.
8 Fin
9 Crear la matriz B compuesta de $n_0 \times n_0$ bloques circulantes de orden p ubicando los bloques D^i en la diagonal principal y los bloques B^i fuera de la diagonal principal.
10 Permutar cada fila bloque de la matriz por una permutación seleccionada independiente e igualmente distribuida del espacio de todas las permutaciones de n_0 elementos.

[1] $det(A) = \sum_{\sigma \in S_n} sgn(\sigma) \prod_{i=1}^{n} a_{\sigma(i),i}$ donde $A = (a_{ij})_{i,j=1,\ldots,n}$ y sgn es la es la función signo de permutaciones en el grupo de permutación S_n que devuelve 1 y -1 para permutaciones pares e impares, respectivamente.

3. Eficiencia computacional

En esta sección se realiza un análisis de la complejidad computacional de los algoritmos propuestos. El objetivo y principal resultado es demostrar que los tres algoritmos poseen complejidad polinomial y por tanto son computacionalmente eficientes.

Teorema 22 *La complejidad computacional del algoritmo 1 es* $\mathcal{O}(d_v)$.

Demostración. Seleccionar un elemento de un conjunto de n elementos tiene un costo de $\mathcal{O}(1)$ (constante). Como en este caso se seleccionan d_v elementos, el costo del algoritmo 1 es $\mathcal{O}(d_v)$ ■

Teorema 23 *La complejidad computacional del algoritmo 2 es* $\mathcal{O}(k_0^2 p^2)$.

Demostración. El paso que domina el costo en el algoritmo 2 es el paso 3, por lo que estimando el costo de dicho paso se tiene una estimación del costo del algoritmo en general.

Cada fila bloque de la matriz S consta de k_0 matrices de orden p. Una vez seleccionada aleatoriamente una permutación diferente a la identidad del total de $k_0! - 1$ posibles permutaciones, aplicar la permutación sería realizar una copia de cada matriz de orden p en el orden establecido por la permutación en un nueva fila bloque vacía (un nuevo espacio en memoria similar al ocupado).

Como cada matriz tiene orden p, recorrer todos sus elementos para copiar cada uno de ellos requiere la utilización de dos ciclos **for**, uno dentro de otro. Lo anterior conlleva realizar entonces p^2 copias. Como en cada fila bloque se tienen k_0 matrices de orden p, permutar una fila bloque completa necesita $k_0 p^2$ operaciones de copia y como la matriz S tiene k_0 filas bloque, la cantidad de operaciones copias total a realizar es $k_0^2 p^2$ ■

Con un razonamiento muy similar a la demostración del teorema 23, se puede demostrar un resultado análogo para el algoritmo 3.

Teorema 24 *La complejidad computacional del algoritmo 3 es* $\mathcal{O}(n_0^2 p^2)$.

4. Conclusiones

En este artículo se propuso un algoritmo para generar matrices circulantes invertibles sobre $\mathbb{F}_2$ de orden primo y se mostró como aplicarlo a la generación de llaves del criptosistema McEliece basado en códigos QC-LDPC. De dicha aplicación se proposuieron dos algoritmos: uno probabilístico (con probabilidad cercana a 1) para generar matrices circulantes invertibles densas y uno determinístico para generar matrices circulantes invertibles sparse. Se realizó un análisis de la complejidad computacional de cada algoritmo propuesto demostrando así su eficiencia computacional.

Con los resultados obtenidos queda claro la ventaja de los algoritmos propuestos para generar matrices circulantes invertibles y las matrices que componen la llave del criptosistema McEliece basado en códigos QC-LDPC. Ya no es necesario generar de forma aleatoria estas matrices y verificar si cumplen lo requerido, los algoritmos propuestos brindan un procedimiento para construir estas matrices y además los mismos tienen complejidad polinomial; orden cuadrático más concretamente.

Acknowledgments

Los autores agradecen al profesor Raúl Gramatges y al DrC. Camilo Ernesto Nápoles por sus valiosas consideraciones sobre la definición de matriz sparse utilizada.

Referencias

[1] Chen, Lily, Lily Chen, Stephen Jordan, Yi Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner y Daniel Smith-Tone: *Report on post-quantum cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016.

[2] Fiallo, Ernesto D.: *Análisis del criptosistema McEliece basado en códigos QC-LDPC*. Tesis de Maestría, 2019.

[3] Gallager, Robert: *Low-density parity-check codes*. IRE Transactions on information theory, 8(1):21–28, 1962.

[4] Jungnickel, D: *Finite Fields: structure and arithmetics, Mannheim, BI-Wiss*, 1993.

[5] Lidl, Rudolf y Harald Niederreiter: *Introduction to finite fields and their applications*. Cambridge university press, 1994.

[6] MacWilliams, Florence Jessie y Neil James Alexander Sloane: *The theory of error-correcting codes*. Elsevier, 1977.

[7] NIST: *Post-Quantum Cryptography, Round 2 Submissions*. 2019.

[8] Townsend, Richard y E Weldon: *Self-orthogonal quasi-cyclic codes*. IEEE Transactions on Information Theory, 13(2):183–195, 1967.