Revista Cubana de Ciencias Informáticas
Vol. 15, No. 1, Enero-Marzo, 2021
ISSN: 2227-1899 | RNPS: 2301
Pág. 114-128
http://rcci.uci.cu

Tipo de artículo: Artículo original
Temática: Matemática Computacional
Recibido: 23/12/2020 | Aceptado: 30/01/2021

# Algoritmo para la generación de polinomios primitivos sobre extensiones de campos finitos de característica dos

## An algorithm for the generation of primitive polynomials over finite field's extensions of characteristic two

Evaristo José Madarro Capó [1] http://orcid.org/0000-0002-5226-5946

Oristela Cuellar Justiz [2*] http://orcid.org/0000-0002-6685-8013

[1] Universidad de La Habana. Instituto de Criptografía. Dirección postal. ejmadarro@nauta.cu

[2] Universidad de Ciencias Informáticas. Dirección postal. oristelacj@uci.cu

*Autor para la correspondencia. (oristelacj@uci.cu)

**RESUMEN**

En este trabajo se presenta un algoritmo para la generación de polinomios primitivos sobre extensiones de campos finitos de característica dos. Para la construcción del algoritmo se utilizan algunos conceptos de la teoría de números y los campos finitos como los cosetos q-ciclotómicos y los elementos primitivos. Además, se efectúa un análisis de la complejidad computacional del algoritmo expuesto. Finalmente, se realiza una comparación entre el algoritmo desarrollado y varios algoritmos reportados en la literatura, para la construcción determinística de polinomios primitivos.

**Palabras clave:** polinomios primitivos; extensiones de campos finitos.

Revista Cubana de Ciencias Informáticas
Vol. 15, No. 1, Enero-Marzo, 2021
ISSN: 2227-1899 | RNPS: 2301
Pág. 114-128
http://rcci.uci.cu

**ABSTRACT**

In this paper an algorithm for primitive polynomial generation over finite field extension of characteristic 2 is presented. For the construction of the algorithm some concepts of number theory and finite fields such as q-cyclotomic cosets and primitive elements are used. A computational complexity analysis over the exposed algorithm it's carried out. Finally, a comparison between the developed algorithm and several algorithm reported in the literature, for primitive polynomials generation, is performed.

**Keywords:** primitive polynomial; finite fields extension.

# Introducción

La teoría de los campos finitos (Lidl, Niederreiter, 1994), (Mullen, Panario, 2013) es una rama del álgebra moderna que desempeña un papel importante en diversas aplicaciones dentro de las áreas de ingeniería eléctrica (Golomb, Gong, 2005), ciencias de la computación (Menezes, Van Oorschot, Vanstone, 1997), matemáticas (Mullen, Panario, 2013), entre otras.

En las últimas décadas los campos finitos han sido utilizados como una herramienta poderosa en la construcción de sistemas computacionales con aplicación específica en diversas áreas tales como la criptografía (Menezes, Van Oorschot, Vanstone, 1997) y la codificación algebraica (MacWilliams, Sloane, Sloane, 1983) . Entre las concepciones más importantes sobre campos finitos se encuentra la teoría de los polinomios sobre los campos finitos, la cual resulta fundamental para la investigación de las estructuras algebraicas de los campos finitos y sus aplicaciones. La utilización de una clase especial de polinomios, los llamados polinomios primitivos, es de gran utilidad en el diseño de algunos componentes de algoritmos criptográficos (Daemen, Rijmen, 2002) debido a sus características matemáticas (Junod, Vaudenay, 2004).

En los sistemas de cifrado en flujo estos se utilizan como polinomio característico en los registros de desplazamiento con retroalimentación lineal (Golomb, 1982) para la generación de sucesiones recurrentes lineales de período máximo. El valor del período de estas sucesiones está determinado por las propiedades de su polinomio mínimo. Para que la sucesión recurrente lineal tenga período máximo, es necesario y suficiente que su polinomio mínimo sea un polinomio primitivo.

Por otro lado, en los sistemas de cifrado en bloques, las matrices MDS *(Maximum Distance Separable*) (Junod, Vaudenay, 2004) son uno de los componentes más utilizados en los últimos diseños de estándares de cifrado (Daemen, Rijmen, 2002) debido a que garantizan altos valores de difusión. Los polinomios primitivos sobre extensiones de campos finitos son utilizados en la construcción de este tipo de matrices (Freyre, Díaz, Diaz, Perez, 2014).

Además, los polinomios primitivos son útiles en la teoría de la codificación algebraica (MacWilliams, Sloane, 1983) debido que a partir de estos se construyen códigos correctores de errores como los códigos Goppa (Valentijn, 2015), utilizados en la construcción de algoritmos asimétricos resistentes a la computación post cuántica.

En la literatura existen varios algoritmos para la generación de polinomios primitivos sobre campos finitos (Madarro, 2017). Algunos de estos algoritmos poseen una complejidad de orden cúbico (Shoup, 2009) o superior (Saxena, McCluskey, 2004). Mientras el propuesto en (Rifà, Borrell,1995) basa su diseño a la utilización de hardware específico y el presentado en (da Silva Filho, Lima Filho, 1993) hace depender su complejidad en la selección de algoritmos para la factorización de polinomios.

Otras interesantes propuestas son los algoritmos propuestos en (Shoup, 1999) y en (Di Porto, Guida, Montolivo, 1992), ambos están basados en la teoría de los registros de desplazamiento y las sucesiones recurrentes lineales. No obstante, existe otra vía teórica relacionada con la generación de polinomios primitivos basada en la obtención de los cosetos $q$-ciclotómicos y la construcción de polinomios mínimos (Gordon, 1976) (Lidl, Niederreiter, 1994) (Shoup, 2009) (Shoup, 1999) que permite un sencillo acople con cualquier esquema de software y un costo computacional razonable. En (Rifà, Borrell,1995), cuya propuesta posee la mejor complejidad, se utiliza una idea similar, pero desde un punto de vista de hardware.

En este trabajo se presenta, utilizando conceptos y propiedades del álgebra en campos finitos y la teoría de números, un algoritmo para la generación de polinomios primitivos sobre campos finitos $\mathbb{F}_{2^n}$.

Revista Cubana de Ciencias Informáticas
Vol. 15, No. 1, Enero-Marzo, 2021
ISSN: 2227-1899 | RNPS: 2301
Pág. 114-128
http://rcci.uci.cu

## Preliminares

Un polinomio $f \in \mathbb{F}_q[x]$ se dice que es irreducible sobre $\mathbb{F}_q$ si $f$ tiene grado positivo y $f = bc$ con $b, c \in \mathbb{F}_q[x]$ .implica que $b$ o $c$ es un polinomio constante.(Lidl, Niederreiter, 1994) (Mullen, Panario, 2013).

Si $f$ es un polinomio irreducible en $\mathbb{F}_q[x]$ de grado $m$, entonces $f$ tiene una raíz $\alpha$ en $\mathbb{F}_{q^m}$. Además, todas las raíces de $f$ son simples y están dadas por los $m$ distintos elementos $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{m-1}}$ de $\mathbb{F}_{q^m}$. (Lidl, Niederreiter, 1994)

Sea $\beta$ un elemento de $\mathbb{F}_{q^m}$, el polinomio mínimo de $\beta$ sobre $\mathbb{F}_q$ es el polinomio mónico $\mathbb{F}_q[x]$ de menor grado que tiene a $\beta$ como raíz. Sea $g_\beta(x)$ el polinomio mínimo de $\beta \in \mathbb{F}_{q^m}$ sobre $\mathbb{F}_q$, entonces:

1. El polinomio $g_\beta(x)$ es irreducible sobre $\mathbb{F}_q$.
2. El grado de $g_\beta(x) = d$ con $d$ $divisor$ $de$ $m$ y sus raíces son $\beta$ y sus conjugados $\beta^q, \beta^{q^2}, \ldots, \beta^{q^{d-1}}$ respecto a $\mathbb{F}_{q^s}, con\ s = \frac{m}{d}$

De esta manera, se puede escribir $g_\beta(x)$ como

$$g_\beta(x) = (X - \beta)(X - \beta^q)(X - \beta^{q^2}) \ldots (X - \beta^{q^{d-1}})$$

Para $\alpha \in \mathbb{F}_{q^m}$, el menor entero e tal que $\alpha^e = 1$ se denomina orden de $\alpha$ y se denota por $ord(\alpha)$. Si $ord(\alpha) = q^m - 1$, entonces $\alpha$ es llamado elemento primitivo $\mathbb{F}_{q^m}$ y todos los elementos distintos de cero de $\mathbb{F}_{q^m}$ pueden ser expresados como potencias de $\alpha$. Los elementos primitivos de $\mathbb{F}_{q^m}$ son los $\alpha^k$ tal que $gcd(k, q^m - 1) = 1$.

Un polinomio $f(x)$ de grado $m$ es llamado primitivo sobre $\mathbb{F}_q$ si es el polinomio mínimo sobre $\mathbb{F}_q$ de un elemento primitivo en $\mathbb{F}_{q^m}$. Sea $f(x)$ un polinomio de grado $m$ sobre $\mathbb{F}_q$, el polinomio reverso $f^*(x)$ de $f(x)$

Revista Cubana de Ciencias Informáticas
Vol. 15, No. 1, Enero-Marzo, 2021
ISSN: 2227-1899 | RNPS: 2301
Pág. 114-128
http://rcci.uci.cu

está definido como $f^*(x) = x^m f(x^{-1})$. Si $f(x)$ es irreducible sobre $\mathbb{F}_q$, $f(x) \neq x$, $f^*(x)$ es irreducible. Si $f(x)$ es primitivo, su reverso también lo es. (Lidl, Niederreiter, 1994) (Mullen, Panario, 2013)

Sea $\alpha$ es un elemento primitivo de $\mathbb{F}_{q^m}$. Sea $g_{\alpha^k}(x)$ el polinomio mínimo del elemento $\alpha^k$. Utilizando los cosetos ciclotómicos modulo $q^m - 1$, denotados como $CC_k = \{k, kq, kq^2, \ldots, kq^{m_k-1}\}$, donde $m_k$ es el menor entero tal que $q^{m_k} \cdot k = k \, mod \, (q^m - 1)$, es posible escribir la siguiente caracterización de polinomios irreducibles y primitivos:

1. Todos los polinomios $g_{\alpha^k}(x)$, tal que $CC_k$ tiene cardinalidad $m$, constituyen los polinomios irreducibles distintos de grado $m$ sobre $\mathbb{F}_q$.
2. Los polinomios irreducibles $g_{\alpha^k}(x)$, tal que $gcd(k, q^m - 1) = 1$, son todos los polinomios primitivos distintos de grado $m$ sobre $\mathbb{F}_q$.

## Generación de polinomios primitivos

La idea del algoritmo (Madarro, 2017) es basar el problema de la generación de polinomios primitivos sobre extensiones de campos finitos en la construcción de los cosetos $q$-ciclotómicos módulo $q^m - 1$.

Algoritmo 1: Generación de polinomios primitivos:

**Entrada**: Polinomio $f$ irreducible sobre $\mathbb{F}_2$ de grado $n$ que define el campo finito de partida $F_q$ y polinomio $g$ irreducible sobre $\mathbb{F}_q$ de grado $m$ que define el campo finito $F_{q^m}$, $\beta$ elemento primitivo de $\mathbb{F}_{q^m}$

**Salida**: Lista $L$ de polinomios primitivos de grado $m$ sobre el campo finito $\mathbb{F}_q$.

1. $x = 0$; $s[m] = \{0\}$; $h = q^m - 1$;
2. *Desde* $k = 1$ hasta $\frac{h+1}{2}$ *hacer*
3. *Si* $mcd(h, k) = 1$ *entonces*
4. *Si* $k \rightarrow$ *coseto líder, entonces*
5. $s_0 = \beta^k$;

Revista Cubana de Ciencias Informáticas
Vol. 15, No. 1, Enero-Marzo, 2021
ISSN: 2227-1899 | RNPS: 2301
Pág. 114-128
http://rcci.uci.cu

6. *Desde* $i = 1$ *hasta* $m - 1$ *hacer*
7. $s_i = {s_{i-1}}^q;$
8. *fin ciclo*
9. $d(x) = \prod_{i=0}^{m-1}(x - s_i);$
10. $L_{x++} = d(x)$
11. $L_{x++} = d^*(x)$
12. *fin condición*
13. *fin condición*
14. *fin del ciclo*
15. *Retornar L*;

Donde $d^*(x)$ es el polinomio mónico reverso del polinomio $d(x)$ y $mcd(\cdot)$ es la función que calcula el máximo común divisor. Se puede observar que el grado $m$ del polinomio de entrada $g$ irreducible sobre $\mathbb{F}_q$ utilizado para definir el campo finito $F_{q^m}$ coincide con el grado de los polinomios primitivos que se desean generar. Por otro lado, si se desea generar una cantidad fija de polinomios primitivos, solo es necesario construir una condición de parada para $x$ cuando su valor llegue a la cantidad deseada.

La obtención de los cosetos $q$-ciclotómicos se realiza mediante la obtención del conjunto $E_p$ de los números primos relativos con el orden del grupo multiplicativo $\mathbb{F}_{q^m}^*$ y luego determinar, para cada número $k \in E_p$, el conjunto $\{k, kq, kq^2, \ldots kq^{m-1}\}$ módulo $q^m - 1$. (Cuellar, Morgado, Sosa 2015)

Un problema común es construir cosetos $q$-ciclotómicos repetidos. Para minimizar la ocurrencia de este caso se verifica para cada $k$ si es coseto líder, puesto que el coseto líder es el menor de los elementos del coseto $q$-ciclotómico. Para realizar el paso 4 del algoritmo y determinar si un entero positivo $k$ es un coseto líder se utilizó el siguiente algoritmo.

Algoritmo 2: Verificar coseto líder:

Revista Cubana de Ciencias Informáticas
Vol. 15, No. 1, Enero-Marzo, 2021
ISSN: 2227-1899 | RNPS: 2301
Pág. 114-128
http://rcci.uci.cu

**Entrada**: Entero positivo $k$ a verificar, grado $m$ de la extensión, cardinalidad $q$ del campo finito base $\mathbb{F}_q$.

**Salida**: Verdadero si es líder o Falso en caso contrario.

```
j = k;
Desde i = 1 hasta m − 1 hacer
        j = jq mod (q^m − 1 )
        Si  j < k
                Retornar Falso;
        fin condición
 fin ciclo
Retornar Verdadero;
```

Sea $f \in \mathbb{F}_q[x]$ el polinomio primitivo asociado al coseto $\{k, kq, kq^2, \dots kq^{m-1}\}$. En el coseto $q$-ciclotómico $\{(n-k), (n-k)q, (n-k)q^2, \dots (n-k)q^{m-1}\}$ cada elemento es el inverso multiplicativo de algún elemento del coseto $\{k, kq, kq^2, \dots kq^{m-1}\}$, luego el polinomio asociado a este coseto $q$-ciclotómico es el polinomio reverso de $f$ y por lo tanto también es primitivo.

De esta manera, encontrando cada elemento $k \in E_p$ se estarían determinando dos cosetos $q$-ciclotómicos y a su vez dos polinomios primitivos. Luego, para obtener todos los $k \in E_p$, solo es necesario recorrer la mitad del espacio de búsqueda.

Luego de obtenido cada coseto $q$-ciclotómico, se realiza el cómputo del polinomio primitivo a partir de los elementos de $s = \{\beta^k, \beta^{kq}, \beta^{kq^2}, \dots, \beta^{kq^{m-1}}\}$, cuyos exponentes integran el coseto $q$-ciclotómico $CC_k = \{k, kq, kq^2, \dots kq^{m-1}\}$, con $\beta \in \mathbb{F}_{q^m}$ un elemento primitivo. Esto es debido a que los elementos que pertenecen a $s$ constituyen las raíces de un polinomio primitivo de grado $m$ sobre $\mathbb{F}_q$. Así, cada polinomio primitivo $f_m(x)$ de grado $m$ puede obtenerse mediante la siguiente expresión.

Revista Cubana de Ciencias Informáticas
Vol. 15, No. 1, Enero-Marzo, 2021
ISSN: 2227-1899 | RNPS: 2301
Pág. 114-128
http://rcci.uci.cu

$$f_m(x) = \prod_{i=0}^{m-1} (x - \beta^{kq^i})$$

donde $f_m(x)$ se corresponde con el polinomio mínimo del elemento $\beta$.

Para el cómputo del polinomio mínimo $f_m(x)$ (paso 9 del Algoritmo 1) normalmente se utilizan $O(m^2)$ operaciones sobre $\mathbb{F}_{q^m}$. No obstante, en (Gordon, 1976) se proporciona una forma de obtener el polinomio mínimo de una forma más eficiente en cuanto a la cantidad de operaciones, mediante la expresión

$$f_m(\beta) = \prod_{i=0}^{m-1} (\beta - \beta^{kq^i})$$

Utilizando el Algoritmo 3 basado en la teoría descrita en (Gordon, 1976) se realizan solo $O(m)$ operaciones sobre $\mathbb{F}_{q^m}$.

Algoritmo 3: Cómputo del polinomio mínimo:

**Entrada**: Polinomio $f$ irreducible sobre $\mathbb{F}_2$ de grado $n$ que define el campo finito de partida $\mathbb{F}_q$ y polinomio $g$ irreducible sobre $\mathbb{F}_q$ de grado $m$ que define el campo finito $\mathbb{F}_{q^m}$, conjunto $s$ y elemento primitivo $\beta$ de $\mathbb{F}_{q^m}$.
**Salida**: Polinomio primitivo de grado $m$ sobre el campo finito $\mathbb{F}_q$.

1. $\tau = 1;\ P[m];$
2. *Desde* $i = 0$ *hasta* $m - 1$ *hacer*
3. $\tau = \tau \cdot (\beta - s_i);$
4. *fin ciclo*
5. *Desde* $i = 0$ *hasta* $d - 1$ *hacer*
6. $P_i = (g_i - \tau_i);$
7. *fin ciclo*
8. *Desde* $i = d$ *hasta* $m - 1$ *hacer*
9. $P_i = g_i;$
10. *fin ciclo*

Revista Cubana de Ciencias Informáticas
Vol. 15, No. 1, Enero-Marzo, 2021
ISSN: 2227-1899 | RNPS: 2301
Pág. 114-128
http://rcci.uci.cu

11. *Retornar* $P$;

Donde $d$ es el grado del elemento $\tau$, en base polinomial, obtenido en los pasos 2-4.

## Complejidad computacional

Los pasos 6 y 9 son los de mayor costo computacional en el Algoritmo 1, correspondiente al cálculo del coseto $q$-ciclotómico y el cómputo del polinomio mínimo, respectivamente. Es conocido que los elementos del conjunto $s$ se pueden obtener en $O(m \log q)$ operaciones sobre $F_{q^m}$ (Von Zur Gathen, Panario, 2001) (Shoup, 2009), mediante sucesivas potencias a la $q$ y un costo mínimo de almacenamiento, mientras el polinomio mínimo en $O(m)$ operaciones sobre $F_{q^m}$ utilizando el método propuesto en (Gordon, 1976).

Para el cómputo de polinomios primitivos se utilizó la representación en base polinomial de $F_{q^m}$ sobre $F_q$, donde cada elemento de $F_{q^m}$ es representado como un polinomio en $F_q[x]$ de grado menor que $m$. En tal representación el costo de ejecución del Algoritmo 1 está dado por $O(m \log q\ M)$ operaciones sobre $F_q$, donde $M$ es el costo en operaciones de la multiplicación. Utilizando el mejor método de multiplicación (Schönhage, Strassen, 1971), con $O(m \log m \log\log m)$ operaciones sobre $F_q$, se realizan $O(log\ q\ \ m^2 \log m \log\log m)$ operaciones sobre $F_q$. En este trabajo $q = 2^n$, por lo que el costo de ejecución se puede expresar como $O(n\ m^2 \log m \log\log m)$ operaciones sobre $F_q$.

# Resultados y discusión

En la Tabla 1 se presenta un resumen de la complejidad computacional, en base a la cantidad de operaciones sobre $F_q$, de los algoritmos referenciados incluyendo el algoritmo propuesto.

Revista Cubana de Ciencias Informáticas
Vol. 15, No. 1, Enero-Marzo, 2021
ISSN: 2227-1899 | RNPS: 2301
Pág. 114-128
http://rcci.uci.cu

**Tabla 1-** Complejidad de los algoritmos para la generación de polinomios primitivos.

| Algoritmo | Operaciones sobre $F_q$ |
|---|---|
| (Di Porto, Guida, Montolivo, 1992) | $O(km^2)$ |
| (da Silva Filho, Lima Filho, 1993) | Complejidad de la factorización |
| (Rifà, Borrell,1995) | $O(m)$ |
| (Shoup, 1999) | $O\left(M(m)m^{1/2}+m^2\right)$ |
| FactorPower(Saxena, McCluskey, 2004) | $O(tm^4lnm)$ |
| MatrixPower (Saxena, McCluskey, 2004) | $O(m^4lnm)$ |
| (Shoup, 2009) | $O(m^3)$ |
| Algoritmo 1 | $O(n\,m^2\log m\log\log m)$ |

Donde $t$ es el número de factores de $q=2^n$ y $k$ es el exponente de un elemento primitivo del campo finito $F_q$. En el caso del algoritmo (da Silva Filho, Lima Filho, 1993) la complejidad puede sustituirse en dependencia del algoritmo que se utilice para factorizar polinomios sobre campos finitos (Von Zur Gathen, Panario, 2001).

Cuando $n\log m\log\log m\leq k$ la complejidad del Algoritmo 1 es comparable con la del obtenido (Di Porto, Guida, Montolivo, 1992). Mientras que en el caso $n\log m\log\log m\leq m$ el Algoritmo 1 es comparable a la complejidad del algoritmo en (Shoup, 2009) .

A menudo es necesario generar polinomios primitivos de varios grados sobre un mismo campo finito. Este es uno de los escenarios que favorece la utilización del Algoritmo 1 puesto que al fijar el campo finito, el grado $m$ de los polinomios es el valor que determina el costo de generar los polinomios primitivos deseados al poder considerar $n$ constante. Este escenario es una aplicación práctica real debido a que, por ejemplo, los algoritmos criptográficos trabajan sobre un campo finito específico fijo (Daemen, Rijmen, 2002).

En la Figura 1 se puede observar cómo se comporta la complejidad del algoritmo presentado en contraste con la complejidad correspondiente a los algoritmos (Shoup, 2009) y (Di Porto, Guida, Montolivo, 1992) sobre el campo finito fijado $\mathbb{F}_{2^8}$ para varios valores de $m$ y $k$.

Revista Cubana de Ciencias Informáticas
Vol. 15, No. 1, Enero-Marzo, 2021
ISSN: 2227-1899 | RNPS: 2301
Pág. 114-128
http://rcci.uci.cu

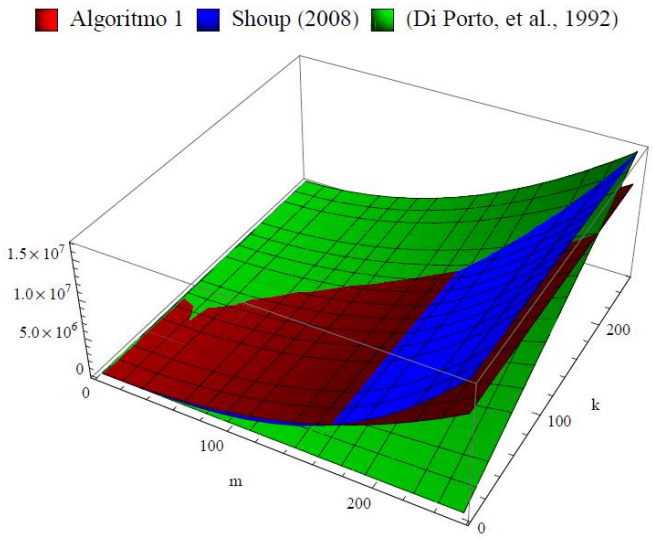


**Fig.1 -** Análisis comparativo de costo computacional para $k = \overline{1,255}$ y $m = \overline{2,256}$ en $\mathbb{F}_{2^8}$ .

Se observa como existen valores de $m$ y $k$ a partir de los cuales el costo computacional del Algoritmo 1 es inferior a los casos (Shoup, 2009) y (Di Porto, Guida, Montolivo, 1992). Estos resultados muestran la importancia de seleccionar adecuadamente el valor de $k$ en (Di Porto, Guida, Montolivo, 1992).

La Tabla 2 muestra el tiempo en segundos de la ejecución del Algoritmo 1, sin la utilización de tablas precomputadas para las operaciones aritméticas sobre campos finitos $\mathbb{F}_{2^n}$, variando la cardinalidad $n$, la cantidad $x$ de polinomios primitivos generados y el grado $m$ de los polinomios primitivos a generar. Lo experimentos se realizaron en una Intel Pentium Dual Core de 1.3 Ghz.

**Tabla 2 -** Tiempo promedio de ejecución del algoritmo propuesto en segundos.

| $x$ | 50 | | | 100 | | | 500 | | |
|---|---|---|---|---|---|---|---|---|---|
| $n/m$ | 4 | 8 | 16 | 4 | 8 | 16 | 4 | 8 | 16 |
| $\mathbb{F}_{2^4}$ | 2.8 | 22.4 | 263.7 | 6.2 | 43.9 | 323.4 | 43.5 | 234.9 | 1805.7 |
| $\mathbb{F}_{2^6}$ | 4.4 | 35.5 | 298.6 | 12.1 | 71.8 | 521.3 | 61.8 | 397 | 3,391.5 |
| $\mathbb{F}_{2^8}$ | 7.1 | 51.4 | 341.9 | 17.6 | 108.2 | 623 | 145.6 | 514.3 | 3,517.6 |
| $\mathbb{F}_{2^{16}}$ | 24.7 | 150.8 | 1,341.7 | 54.5 | 503 | 2,004.8 | 287.8 | 1,591 | 10,802 |

Revista Cubana de Ciencias Informáticas
Vol. 15, No. 1, Enero-Marzo, 2021
ISSN: 2227-1899 | RNPS: 2301
Pág. 114-128
http://rcci.uci.cu

La cardinalidad de los campos finitos aumenta exponencialmente en base a la dimensión $n$ del campo finito, de esta manera un aumento en la dimensión $n$ del campo finito influye en el tiempo requerido para generar los polinomios primitivos. No obstante, la mayor influencia, basado en la propia complejidad del Algoritmo 1, es el aumento del grado $m$ de los polinomios primitivos a generar (Ver Tabla 2).

## Conclusiones

En el presente trabajo se propuso un algoritmo para la obtención de polinomios primitivos sobre extensiones de campos finitos de característica dos. Las bases teóricas sobre las cuales se basa el diseño del algoritmo están determinadas por la teoría de los cosetos $q$-ciclotómicos y los elementos primitivos para la construcción de polinomios mínimos. Para la construcción del algoritmo se transformó el problema de la búsqueda de polinomios primitivos sobre extensiones de campos finitos en la obtención de los cosetos $q$-ciclotómicos sobre campos finitos, el cual consiste en la búsqueda de los números primos relativos con el orden del grupo multiplicativo de $F_{q^m}$.

Se determinó la complejidad del algoritmo presentado y se comparó con los algoritmos reportados en la literatura. El algoritmo propuesto tiene una complejidad inferior a varios algoritmos reportados en la literatura y mejora a otros bajo las condiciones descritas. Este algoritmo constituye una herramienta importante en aplicaciones prácticas pues puede ser implementada sin dificultad en software y puede ser utilizada para polinomios primitivos sobre extensiones de campos finitos.

En trabajos futuros se investigará el uso de bases normales y torres de campos finitos sobre el Algoritmo 1 con el objetivo de mejorar su complejidad computacional. Así como, realizar la implementación en paralelo para disminuir su tiempo de ejecución.

# Referencias

Cuellar, Oristela, Morgado, Eberto R. And Sosa, Guillermo, 2015. Relación Entre El Automorfismo De Frobenius Y Los Homomorfismos De Inmersión Del Campo $F_{P^N}$ En El Campo $F_{2^M}$. *Revista Ciencias Matemáticas. Cuba*. 2015. Vol. Vol 29, P. 127–130.

Da Silva Filho, Joel Guilherme And Lima Filho, Dimas De Queiroz, 1993. Construção De Polinômios Primitivos Sobre Corpos Finitos $F_Q$. In: *11no Simposio Brasileño De Telecomunicaiones.* 1993. P. 497–502.

Daemen, Joan And Rijmen, Vincent, 2002. The Design Of Rijndael.Springer.2002.

Di Porto, A., Guida, F. And Montolivo, E., 1992. Fast Algorithm For Finding Primitive Polynomials Over $F_Q$. *Electronics Letters*. 1992. Vol. 28, No. 2, P. 118–120.

Freyre, Pablo, Díaz, Nelson, Diaz, Rafael And Perez, Claudia, 2014. Random Generation Of Matrixes Mds. In: *Ctcrypt 2014*. 2014.

Golomb, S. W., 1982. *Shift Register Sequences*. Aegean Park Press.

Golomb, Solomon W. And Gong, Guang, 2005. *Signal Design For Good Correlation: For Wireless Communication, Cryptography, And Radar*. Cambridge University Press. Isbn 0-521-82104-5.

Gordon, John A., 1976. Very Simple Method To Find The Minimum Polynomial Of An Arbitrary Nonzero Element Of A Finite Field. *Electronics Letters*. 1976. Vol. 12, No. 25, P. 663–664.

Junod, Pascal And Vaudenay, Serge, 2004. Perfect Diffusion Primitives For Block Ciphers. In: *International Workshop On Selected Areas In Cryptography*. Springer. 2004. P. 84–99.

Lidl, Rudolf And Niederreiter, Harald, 1994. *Introduction To Finite Fields And Their Applications*. Cambridge University Press.

Macwilliams, Florence J., Sloane, Neil J. And Sloane, Neil J., 1983. The Theory Of Error-Correcting Codes: North Holland Mathematical Library.1983.

Madarro, Evaristo José, 2017. *Generación De Polinomios Primitivos Sobre Extensiones De Campos Finitos* $F_{2^N}$. Universidad Central “Marta Abreu” De Las Villas. Facultad De Matemática.

Menezes, Alfred J., Van Oorschot, Paul C. And Vanstone, Scott A., 1997. Handbook Of Applied Cryptography Crc Press. *Boca Raton*. 1997.

Mullen, Gary L And Panario, Daniel, 2013. *Handbook Of Finite Fields*. Crc Pres.Taylor & Francis Group.

Rifà, Josep And Borrell, Joan, 1995. A Fast Algorithm To Compute Irreducible And Primitive Polynomials In Finite Fields. *Mathematical Systems Theory*. 1995. Vol. 28, No. 1, P. 13–20.

Saxena, Nirmal R. And Mccluskey, Edward J., 2004. Primitive Polynomial Generation Algorithms Implementation And Performance Analysis. *Crc Technical Report 2004*. 2004.

Schönhage, Arnold And Strassen, Volker, 1971. Schnelle Multiplikation Grosser Zahlen. *Computing*. 1971. Vol. 7, No. 3–4, P. 281–292.

Shoup, Victor, 1999. Efficient Computation Of Minimal Polynomials In Algebraic Extensions Of Finite Fields. In: *Proceedings Of The 1999 International Symposium On Symbolic And Algebraic Computation*. 1999. P. 53–58.

Shoup, Victor, 2009. *A Computational Introduction To Number Theory And Algebra*. Cambridge University Press. Isbn 0-521-51644-7.

Valentijn, Ashley, 2015. Goppa Codes And Their Use In The Mceliece Cryptosystems. 2015.

Von Zur Gathen, Joachim And Panario, Daniel, 2001. Factoring Polynomials Over Finite Fields: A Survey. Journal Of Symbolic Computation. 2001. Vol. 31, No. 1, P. 3–17.

## Conflicto de interés

Los autores de este artículo autorizan la distribución y uso de su artículo.

## Contribuciones de los autores

1. Conceptualización: Evaristo José Madarro Capó, Oristela Cuellar Justiz
2. Curación de datos: Evaristo José Madarro Capó
3. Análisis formal: Evaristo José Madarro Capó, Oristela Cuellar Justiz
4. Adquisición de fondos: No procede
5. Investigación: Evaristo José Madarro Capó, Oristela Cuellar Justiz
6. Metodología: Evaristo José Madarro Capó, Oristela Cuellar Justiz
7. Administración del proyecto: Evaristo José Madarro Capó, Oristela Cuellar Justiz

8. Recursos: Evaristo José Madarro Capó, Oristela Cuellar Justiz
9. Software: Evaristo José Madarro Capó
10. Supervisión: Oristela Cuellar Justiz
11. Validación: Evaristo José Madarro Capó
12. Visualización: Evaristo José Madarro Capó
13. Redacción – borrador original: Evaristo José Madarro Capó
14. Redacción – revisión y edición: Oristela Cuellar Justiz