
**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA

MAESTRÍA EN INGENIERÍA ELECTRÓNICA, OPCIÓN INSTRUMENTACIÓN ELECTRÓNICA

MATERIA DE:

PROCESAMIENTO DE SEÑALES

---

# ” PROYECTO FINAL ”

---

ALUMNO:

LUIS CARLOS LUJANO HERNÁNDEZ*

PROFESOR:

DR. JESÚS MANUEL MUÑOZ PACHECO

PUEBLA,PUE.,DICIEMBRE 2020

# Introducción

El proyecto final consiste en realizar una encriptación de un texto. El poema de Nezahualcóyotl llamado *Amo el canto del cenzontle*, se tomará como uno de los ejemplos en la realización del proyecto final y obtener como resultado final un mensaje cifrado.

*Amo el canto del cenzontle,*

*pájaro de cuatrocientas voces.*

*Amo el color del jade*

*y el enervante perfume de las flores,*

*pero más amo a mi hermano: el hombre.*

Poema de Nezahualcóyotl (1402-1472).

El siguiente ejemplo que se realizará en éste proyecto final será un texto tomado del libro *Las aventuras de Sherlock Holmes* por Sr. Arthur Conan Doyle.

*" Mi mente se subleva ante el estancamiento. Proporcióneme usted problemas, proporcióneme trabajo, déme los más abstrusos criptogramas o los más intrincados análisis, y entonces me encontraré en mi ambiente. Podré prescindir de estimulantes artificiales. "*

El proceso de cifrado consta de dos bloques importantes, la permutación y la sustitución.

## Permutación

La permutación consiste en intercambiar filas y columnas de la matriz de datos. Las filas y las columnas que se intercambian se indican aleatoriamente con un generador de números aleatorios. El número aleatorio se le llama llave y es un valor que se encuentra entre 0 y 1, para indicar el número de fila y columna es necesario dos llaves. Las llaves se mapean y se redondean en un rango y para este proyecto final será de 1 a 16, esto se debe a que se construye una matriz de datos de tamaño 16x16.

La función de mapeo que se utilizará en este proyecto es la siguiente:

$$y = x * (b - a) + a \tag{1}$$

Donde $y$ indica el número de fila o columna, $x$ el número aleatorio de entre 0 y 1, $a$ el límite inferior que en este caso será $a = 1$ y $b$ el límite superior siendo $b = 16$.

## Sustitución

La sustitución se realiza a partir de una matriz llamada S-box y se denota como $S$. El valor de entrada se representa en un número binario de 8 bits, los primeros 2 bits y los últimos 2 bits indicarán el número de fila y los 4 bits centrales indican el número de columna. Es decir, tenemos la siguiente expresión:

$$S[0] : (1,0,0,1,1,0,1,0) \tag{2}$$

El número binario se divide para obtener el número de fila:

$$1,0,1,0 \rightarrow 10 \tag{3}$$

Se realizará el proyecto en MATLAB y por lo tanto se le suma un 1 al final de la conversión binario a decimal. Obtenemos la Fila número 11.

Los 4 bits centrales indican el número de columna:

$$0,1,1,0 \rightarrow 6 \tag{4}$$

El número de columna seria la 7.

Supongamos que tenemos la siguiente matriz como S-box:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 249 | 214 | 235 | 206 | 145 | 251 | 241 | 51 | 193 | 48 | 166 | 204 | 180 | 208 | 222 | 246 |
| 2 | 219 | 215 | 6 | 1 | 15 | 13 | 234 | 69 | 209 | 107 | 28 | 213 | 44 | 190 | 54 | 155 |
| 3 | 122 | 173 | 230 | 186 | 238 | 94 | 123 | 169 | 158 | 250 | 53 | 233 | 89 | 211 | 244 | 189 |
| 4 | 200 | 228 | 177 | 198 | 130 | 132 | 29 | 221 | 237 | 144 | 220 | 77 | 172 | 82 | 57 | 45 |
| 5 | 128 | 71 | 67 | 218 | 87 | 64 | 21 | 242 | 126 | 127 | 36 | 42 | 243 | 133 | 247 | 43 |
| 6 | 248 | 5 | 99 | 20 | 33 | 79 | 154 | 38 | 223 | 124 | 149 | 74 | 225 | 3 | 121 | 197 |
| 7 | 83 | 73 | 152 | 11 | 65 | 115 | 88 | 85 | 18 | 188 | 176 | 205 | 199 | 35 | 120 | 101 |
| 8 | 26 | 103 | 118 | 0 | 245 | 170 | 75 | 116 | 192 | 210 | 106 | 84 | 55 | 143 | 229 | 168 |
| 9 | 9 | 150 | 185 | 231 | 146 | 119 | 17 | 138 | 24 | 37 | 156 | 93 | 179 | 10 | 81 | 178 |
| 10 | 125 | 30 | 148 | 151 | 80 | 90 | 207 | 183 | 255 | 196 | 16 | 164 | 7 | 142 | 31 | 201 |
| 11 | 25 | 135 | 12 | 97 | 253 | 167 | 165 | 171 | 68 | 61 | 232 | 111 | 162 | 41 | 160 | 184 |
| 12 | 129 | 137 | 27 | 236 | 226 | 46 | 203 | 19 | 117 | 131 | 58 | 66 | 217 | 195 | 102 | 60 |
| 13 | 224 | 23 | 47 | 212 | 174 | 34 | 78 | 239 | 4 | 113 | 175 | 194 | 70 | 182 | 187 | 32 |
| 14 | 134 | 92 | 104 | 8 | 59 | 140 | 14 | 49 | 254 | 95 | 227 | 52 | 181 | 216 | 159 | 86 |
| 15 | 147 | 76 | 39 | 112 | 98 | 163 | 141 | 2 | 202 | 252 | 100 | 110 | 56 | 157 | 153 | 62 |
| 16 | 22 | 191 | 63 | 105 | 91 | 72 | 139 | 108 | 114 | 240 | 136 | 109 | 161 | 40 | 50 | 96 |

Figura 1: S-box

Por lo tanto, el valor que se sustituye en la posición 0 de la matriz de datos será el número 165.

# Desarrollo

El desarrollo de este proyecto se basó en el siguiente algoritmo.

```
INICIO
Texto de Entrada
Convertir de Caracteres a ASCII
Generar la Matriz de Datos
Generar Llave
Permutar 1
Generar Llave
Permutar 2
Construir S-box
Sustitución
Generar Vector de Datos Final
Convertir de ASCII a Caracteres
FIN
```

1. Inicia el código.
2. Ingresamos el TEXTO en una variable.
3. Convertir de caracteres a valores ASCII de 0 a 255.

4. Se construye la matriz de datos de 16x16.
5. Generar llaves para indicar la fila y columna.
6. Se permutan fila y columna.
7. Generar otras dos llaves.
8. Se permutan fila y columna.
9. Se construye la S-box.
10. Se sustituyen valores de la matriz de datos con la S-box.
11. Retomar el mismo número de caracteres ingresados en el TEXTO de la matriz final.
12. Convertir de ASCII a caracteres.
13. Fin del código.

Comenzamos por definir los tres principales bloques del código, Generador de números aleatorios, Permutar y S-box. La primera función será la construcción del generador de números aleatorios por cáos. El código de la función en MATLAB es el siguiente:

```
function [ NumRand ] = CaosNum()
%Parametros
d = 100;
r = 4;
xi = rand; %semilla
%Funcion
A = d*xi*exp(((r+9)*(1-xi))-((r+5)*(xi)*(1-xi)));
NumRand = mod(A,1);
end
```

1. Se declara una función llamada *CaosNum* y toma como variable de salida *NumRand*.
2. Comentario para indicar los parámetros.
3. Se declara la variable *d*.
4. Se declara la variable *r*.
5. Se declara la semilla como un número aleatorio de MATLAB.
6. Comentario para indicar la función del generador de números aleatorios por caós.
7. Se calcula la variable *A*.
8. Obtenemos el número aleatorio en la variable *NumRand*-
9. Fin de la función.

La siguiente función corresponde a la operación de permutar las filas y columnas.

```
function [ MatrizOUT, Fila, Columna ] = Permutar( MatrizIN, num1, num2 )
a = 1;
b = 16;
```

```
%----Mapeo
y1 = (num1)*(b-a) + a;
y2 = (num2)*(b-a) + a;
%----Fila o Columna
Fila = round(y1,0);
Columna = round(y2,0);
%----Fila y Columna de los Datos
Dat_F = MatrizIN(Fila,:);
Dat_C = MatrizIN(:,Columna);
%----Intercambio a Columna y Fila
Per_F = reshape(Dat_C,1,16);
Per_C = reshape(Dat_F,16,1);
%----Sustitucion de Filas y Columnas en los Datos
MatrizIN(Fila,:) = Per_F;
MatrizIN(:,Columna) = Per_C;
MatrizOUT = MatrizIN;
end
```

1. Inicia la función llamada *Permutar* con variables de salida *MatrizOUT, Fila, Columna* y variables de entrada *MatrizIN, num1, num2*.
2. Se declara la variable *a* como límite inferior del mapeo de los números aleatorios.
3. Se declara la variable *b* como límite superior del mapeo.
4. Comentario para indicar el comienzo del mapeo.
5. Se obtiene el primer mapeo indicado en la variable *y1*.
6. Se obtiene el segundo mapeo indicado en la variable *y2*.
7. Comentario para indicar las variables *Fila* y *Columna*.
8. Redondeo de la variable *y1* para Fila.
9. Redondeo de la variable *y2* para Columna.
10. Comentario para indicar la obtención de valores de Fila y Columna.
11. Almacenamos la Fila de la matriz de entrada en la variable *Dat_ F*.
12. Almacenamos la Columna de la matriz de entrada en la variable *Dat_ C*.
13. Comentario para indicar la conversión de Fila y Columna.
14. Convertir Fila en Columna.
15. Convertir Columna en Fila.
16. Comentario para indicar la sustitución.
17. Se sustituyen los valores de la Fila indicada de la matriz de entrada.
18. Se sustituyen los valores de la Columna indicada de la matriz de entrada.
19. Se iguala la matriz de salida con la matriz modificada.
20. Fin de la función.

La tercera función que corresponde a la construcción de la S-box se declara a partir de la Tarea 9. Se construye a partir del algoritmo descrito en la referencia [1]. El código de la función *S_ box _ f* solo tiene una variable de salida *Matriz_ S_ box* y necesita de la función *CaosNum()* para generar los números aleatorios. El código es el siguiente:

```
function [ Matriz_S_box ] = S_box_f(  )
%INICIO
%Paso 2
Bits_XOR = zeros(1,16);
LSB = zeros(1,8);
S_box = [];
%Paso 4
l = 1;
while l < 257
    regreso = 0;
    while regreso == 0
        %Paso 5
        %------------- X1 y X2
        x1 = CaosNum();
        x2 = CaosNum();
        %Paso 6
        %-----------------Binario a 16bits
        x1b = round(x1*65535,0);
        x2b = round(x2*65535,0);
        Binx1 = de2bi(x1b,16);
        Binx2 = de2bi(x2b,16);
        %Paso 7
        %-----------------XOR
        for i=1:16
            Bits_XOR(1,i) = xor(Binx1(1,i),Binx2(1,i));
        end
        %Paso 8
        for i=1:8
            LSB(1,i) = Bits_XOR(1,i);
        end
        %Paso 9
        Dec = bi2de(LSB);
        %Paso 10-15
        k = find(S_box == Dec);
        [~,c] = size(k);
        if c >= 1
            regreso = 0;
        else
            S_box(1,l) = Dec;
            regreso = 1;
            l = l + 1;
        end
    end
    %Paso 16
end
%Paso 17
Matriz_S_box = reshape(S_box,[16,16]);
%--------- FIN -----------
end
```

Siguiendo el algoritmo y utilizando la funciones anteriores construimos la encriptación del texto. Se muestra el código completo y se explican las líneas de código más relevantes. El código es el siguiente:

```
clear
clc
Datos = zeros(1,256);
prompt = 'Texto: ';
x = input(prompt,'s');
%Convierte de Caracteres a Unicode
ValUnicode = double(x);
%-------Matriz de Datos
[~,c] = size(ValUnicode);
for i=1:c
    Datos(1,i) =  ValUnicode(1,i);
end
Datos = (reshape(Datos,16,16)).';
%-------Permutar
%----Key 1
Key1_1 = CaosNum();
Key1_2 = CaosNum();
%----Permutar 1
[Perm1, F1, C1] = Permutar(Datos,Key1_1,Key1_2);
%----Key 2
Key2_1 = CaosNum();
Key2_2 = CaosNum();
%----Permutar 2
[Perm2, F2, C2] = Permutar(Perm1,Key2_1,Key2_2);
%----S-box
Sbox = S_box_f();
%-------Sustitucion Sus
Sus = zeros(16,16);
for s1 = 1:16
    for s2 = 1:16
        Dat = Perm2(s1,s2);
        Bin = dec2bin(Dat,8);
        Fil_Bin = [Bin(1,1) Bin(1,2) Bin(1,7) Bin(1,8)];
        Col_Bin = [Bin(1,3) Bin(1,4) Bin(1,5) Bin(1,6)];
        Fil = bin2dec(Fil_Bin);
        Col = bin2dec(Col_Bin);
        Sus(s1,s2) = Sbox(Fil+1,Col+1);
    end
end
%%-------Dato Final
Vector = zeros(1,c);
resto = rem(c,16);
rep = (c - resto)/16;
for i = 1:rep
    for l = 1:16
        Vector(1,l+(16*(i-1))) = Sus(i,l);
    end
end
[~,d] = size(Vector);
for i = 1:resto
    Vector(1,(c-resto+i)) = Sus(rep+1,i);
end
%----Mensaje Final
cifrado = char(Vector);
display(cifrado)
```

La línea 1 a la 5 corresponde a la declaración al ingreso del TEXTO en la variable *x*. La línea 7 convierte los caracteres a valores ASCII extendido que va de decimales de 0 hasta 255. En las líneas 9 a 13 se sustituyen los ceros almacenados en la variable Datos por los valores que corresponden

al TEXTO y finalmente se construye una matriz de 16x16. Las líneas 16, 17, 21 y 22 se declaran las primeras 4 llaves para el proceso permutación. En la línea 19 y 24 se llama la función *Permutar()*. En la línea 26 construimos la S-box. En la línea 28 a 39 se realiza el proceso de sustitución con la S-box. En la línea 41 a 52 obtenemos un vector con el mismo tamaño que el TEXTO, es decir, si el texto es 'HOLA' el vector final tomará los primeros 4 valores de la matriz *Sus*. Las líneas 54 y 55 convierten los valores ASCII extendido en caracteres y los muestra en la variable *cifrado*.

# Resultados

Se presenta como resultado el mensaje cifrado de un poema y de un texto tomado de un libro.

## Poema.

Generamos la S-box como:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 191 | 6 | 232 | 114 | 129 | 78 | 214 | 160 | 24 | 118 | 203 | 189 | 220 | 27 |
| 2 | 81 | 225 | 80 | 123 | 127 | 110 | 156 | 135 | 46 | 40 | 102 | 50 | 31 | 92 | 62 | 33 |
| 3 | 30 | 242 | 199 | 121 | 13 | 181 | 124 | 164 | 98 | 153 | 63 | 148 | 71 | 122 | 97 | 210 |
| 4 | 60 | 221 | 21 | 196 | 244 | 150 | 247 | 103 | 32 | 231 | 14 | 84 | 183 | 167 | 70 | 113 |
| 5 | 56 | 229 | 130 | 217 | 174 | 208 | 248 | 48 | 162 | 126 | 7 | 134 | 139 | 18 | 109 | 137 |
| 6 | 238 | 75 | 55 | 235 | 254 | 90 | 193 | 190 | 115 | 236 | 88 | 136 | 207 | 49 | 233 | 53 |
| 7 | 197 | 209 | 57 | 213 | 73 | 245 | 177 | 44 | 112 | 182 | 171 | 132 | 28 | 151 | 133 | 116 |
| 8 | 255 | 180 | 206 | 17 | 34 | 240 | 155 | 47 | 91 | 253 | 43 | 237 | 158 | 141 | 149 | 142 |
| 9 | 154 | 111 | 230 | 29 | 165 | 202 | 67 | 219 | 95 | 94 | 125 | 194 | 15 | 61 | 143 | 42 |
| 10 | 2 | 41 | 249 | 93 | 234 | 101 | 228 | 211 | 54 | 36 | 224 | 185 | 72 | 66 | 38 | 106 |
| 11 | 100 | 82 | 37 | 179 | 138 | 131 | 51 | 65 | 16 | 212 | 204 | 128 | 79 | 186 | 35 | 172 |
| 12 | 68 | 120 | 11 | 69 | 218 | 19 | 77 | 169 | 59 | 76 | 117 | 187 | 87 | 52 | 184 | 192 |
| 13 | 201 | 10 | 239 | 161 | 227 | 205 | 241 | 198 | 96 | 216 | 3 | 188 | 74 | 22 | 152 | 9 |
| 14 | 246 | 176 | 39 | 140 | 252 | 25 | 146 | 108 | 147 | 163 | 222 | 243 | 23 | 170 | 4 | 144 |
| 15 | 175 | 89 | 5 | 145 | 12 | 20 | 119 | 200 | 83 | 26 | 157 | 45 | 168 | 58 | 99 | 250 |
| 16 | 166 | 159 | 107 | 85 | 215 | 223 | 251 | 226 | 86 | 178 | 195 | 173 | 105 | 8 | 64 | 104 |

Figura 2: S-box generada en MATLAB para el primer ejemplo

Obtuvimos como resultado el siguiente mensaje:

```
Texto: Amo el canto del cenzontle, pájaro de cuatrocientas voces. Amo el color del jade y el enervante perfume de las flores, pero más amo a mi hermano: el hombre

cifrado =

    'îˆiÕì†Ö[s„ í‡aì†Ö[ì„…í„ ‡ìvÖÖÕ«s íÕ~ìÖ[1s  í[ìì„ s Ö–í[ì "Õîí†ÕìÖ‹[ˆ«„sˆˆ   †Ö  Õéëì†Õì„ì –sí ìÖ‹ì ¶1ˆìÕ~ìÕ† íÕ¶‡í ì vÖ‹ì íÕÕˆ Õsˆìësõˆ XÖ ì ~p„í~“Xí   “s   '
```

Figura 3: Evidencia del primer mensaje cifrado.

Podemos observar que algunos símbolos se repiten en el mensaje cifrado, esto se debe a que hay letras que se repiten dentro del mensaje y por consiguiente son reemplazadas por el mismo valor de la S-box.

## Texto.

La S-box que se generó dentro del código es el siguiente:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 144 | 224 | 93 | 161 | 175 | 36 | 178 | 62 | 254 | 8 | 11 | 160 | 249 | 226 | 98 | 205 |
| 2 | 185 | 128 | 80 | 149 | 207 | 169 | 171 | 166 | 120 | 121 | 162 | 232 | 7 | 126 | 154 | 72 |
| 3 | 129 | 247 | 76 | 115 | 164 | 83 | 127 | 248 | 18 | 138 | 179 | 33 | 14 | 240 | 54 | 244 |
| 4 | 0 | 145 | 251 | 51 | 118 | 53 | 48 | 197 | 19 | 222 | 193 | 176 | 135 | 119 | 141 | 90 |
| 5 | 181 | 77 | 40 | 12 | 191 | 113 | 227 | 17 | 151 | 225 | 26 | 186 | 131 | 56 | 199 | 139 |
| 6 | 177 | 28 | 44 | 114 | 208 | 58 | 97 | 27 | 95 | 67 | 117 | 187 | 174 | 140 | 107 | 22 |
| 7 | 69 | 245 | 153 | 66 | 37 | 246 | 63 | 142 | 189 | 217 | 167 | 78 | 157 | 233 | 255 | 88 |
| 8 | 219 | 204 | 173 | 55 | 81 | 75 | 43 | 106 | 150 | 38 | 212 | 202 | 236 | 210 | 168 | 182 |
| 9 | 137 | 218 | 57 | 216 | 65 | 132 | 60 | 42 | 180 | 183 | 237 | 147 | 228 | 2 | 86 | 6 |
| 10 | 165 | 47 | 220 | 238 | 148 | 96 | 9 | 211 | 112 | 89 | 68 | 94 | 134 | 195 | 64 | 31 |
| 11 | 215 | 45 | 34 | 104 | 192 | 87 | 73 | 184 | 84 | 250 | 85 | 243 | 92 | 213 | 82 | 203 |
| 12 | 122 | 206 | 10 | 152 | 52 | 105 | 74 | 136 | 230 | 3 | 4 | 110 | 109 | 198 | 214 | 24 |
| 13 | 235 | 103 | 158 | 39 | 79 | 143 | 41 | 111 | 13 | 50 | 29 | 70 | 61 | 188 | 123 | 20 |
| 14 | 1 | 25 | 163 | 209 | 108 | 146 | 71 | 231 | 156 | 229 | 170 | 159 | 196 | 30 | 32 | 116 |
| 15 | 172 | 133 | 194 | 49 | 241 | 155 | 21 | 201 | 23 | 130 | 252 | 5 | 91 | 253 | 46 | 15 |
| 16 | 102 | 16 | 223 | 125 | 59 | 124 | 101 | 239 | 221 | 242 | 190 | 35 | 200 | 99 | 100 | 234 |

Figura 4: S-box generada en MATLAB para el segundo ejemplo

El resultado es el siguiente mensaje:

'ruþ»CNþCþìCþì½ºCé_þ_N»Cþ8ºþCì8_N_»uCu8Ê□þ¿□Ê□Ê□□uÈNCþCþ□ì8Cáþ□□Ê½ºC»_ì ì□□Ê□Ê□□
uÈNC»»þ8Ê_½_§Ê þáª»Cþ½Êìu»□ìþ_½ìC!□þ□þ□ìþ8□_u_C_»_ìþÊCºÊ□þ»□ìþuN8□uN□NáÊ□þ_N□ºuìuì
þk8CNuÊN□Cìþ»CþCN□CN8□_□ªþCN88N»_C!ìþ8þÊ½C»uá□ªþ□□þì□ÊNáu□þáCþCì8u¿□º&N8Cìþ_□8
uÙu□Ê_º□ì!'

Figura 5: Evidencia del segundo mensaje cifrado.

El resultado del mensaje cifrado fue copiado a WORD para mostrar la evidencia y por esa razón hay algunos símbolos que no pueden ser registrados dentro del editor de texto.

# Conclusión

Este proceso de cifrado fue basado en el protocolo AES, es un algoritmo simple pero obtenemos como resultado mensajes cifrados, especialmente en los mensajes largo donde es difícil identificar loas letras repetidas del mensaje y relacionarlos con los símbolos generados en el mensaje final.

En general, es un proyecto final completo y donde aplicamos lo visto en clases, desde el generador de números aleatorios caótico, realizamos a partir de un algoritmo una S-box y realizamos un procedimiento más simple pero basado en el protocolo de encriptación AES. Además se revisó el proceso de sustitución y permutación.

El tema de seguridad y cifrado son temas que adquieren una mayor importancia hoy en día, y conocer los protocolos y las pruebas que se usan para éstos métodos nos ayudan para que en el futuro se puedan proponer nuevos métodos de cifrado.

# Referencias

[1] Lai, Qiang and Akgul, Akif and Li, Chunbiao and Xu, Guang-Hui and çavuşoğlu, Ünal. ” A New Chaotic System with Multiple Attractors: Dynamic Analysis, Circuit Realization and S-Box Design ”. *Entropy*, vol. 20, pp. 12, Diciembre 2017. DOI: 10.3390/e20010012

[2] Petavratzis, Eleftherios and Moysis, Lazaros and Volos, Christos and Nistazakis, Hector and Munoz-Pacheco, J.M. and Stouboulos, Ioannis, ” Chaotic Path Planning for Grid Coverage Using a

Modified Logistic-May Map ", *Journal of Automation Mobile Robotics & Intelligent Systems*, Octubre 2020. 10.14313/JAMRIS/2-2020/13
[3] Fei Yu, Lixiang Li, Qiang Tang, Shuo Cai, Yun Song, Quan Xu, .A Survey on True Random Number Generators Based on Chaos", Discrete Dynamics in Nature and Society, vol. 2019, Article ID 2545123, 10 pages, 2019. https://doi.org/10.1155/2019/2545123
[4] Rukhin, A., et al.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800–22. http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf.
[5] Rudolf Ahlswede. " The Mathematical Background of the Advanced Encryption Standard " en *Hiding Data - Selected Topics*, volumen 12. Wolfgang Utschick, Holger Boche, Rudolf Mathar. Ed. Alemania: Springer, 2016, pp. 155-210
[6] Pedro Sanchez Munoz, Nam Tran, Brandon Craig, Behnam Dezfouli, Yuhong Liu. " Analyzing the Resource Utilization of AES Encryption on IoT Devices ". *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* pp. 1-5, 2019.
[7] SYSTEMS SECURITY FOR THE 21st CENTERY. *National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information.* Junio 2003.
[8] Shashi Mehrotra Seth, Rajan Mishra. " Comparative Analysis Of Encryption Algorithms For Data Communication ". Vol. 2, Junio 2011.
[9] Muñoz Pacheco Jesús Manuel. "NIST SP800-22"8 Oct. 2020. Procesamiento Digital de Señales. Benemérita Universidad Autónoma de Puebla. Clase.