ANEXO B. AES

El AES se compone de rondas formadas por tres capas diseñadas para no permitir el criptoanálisis lineal y diferencial. Cada capa está constituida por 4 funciones invertibles diferentes: *DesplazarFila*, *MezclarColumnas*, *ByteSub* y *capa de adición de clave*, las cuales se explicarán mas adelante. El número de rondas depende del tamaño de bloque y el tamaño de clave, estos valores pueden variar entre 128, 192 y 256 bits. La tabla B.1 muestra el número de rondas dependiendo de los anteriores valores [3].

Tabla 2.8 Número de rondas para AES en función de los tamaños de clave y bloque

	N_b = 4 (128 bits)	N_b = 6 (192 bits)	N_b = 8 (256 bits)
N_k = 4 (128 bits)	10	12	14
N_k = 6 (192 bits)	12	12	14
N_k = 8 (256 bits)	14	14	14

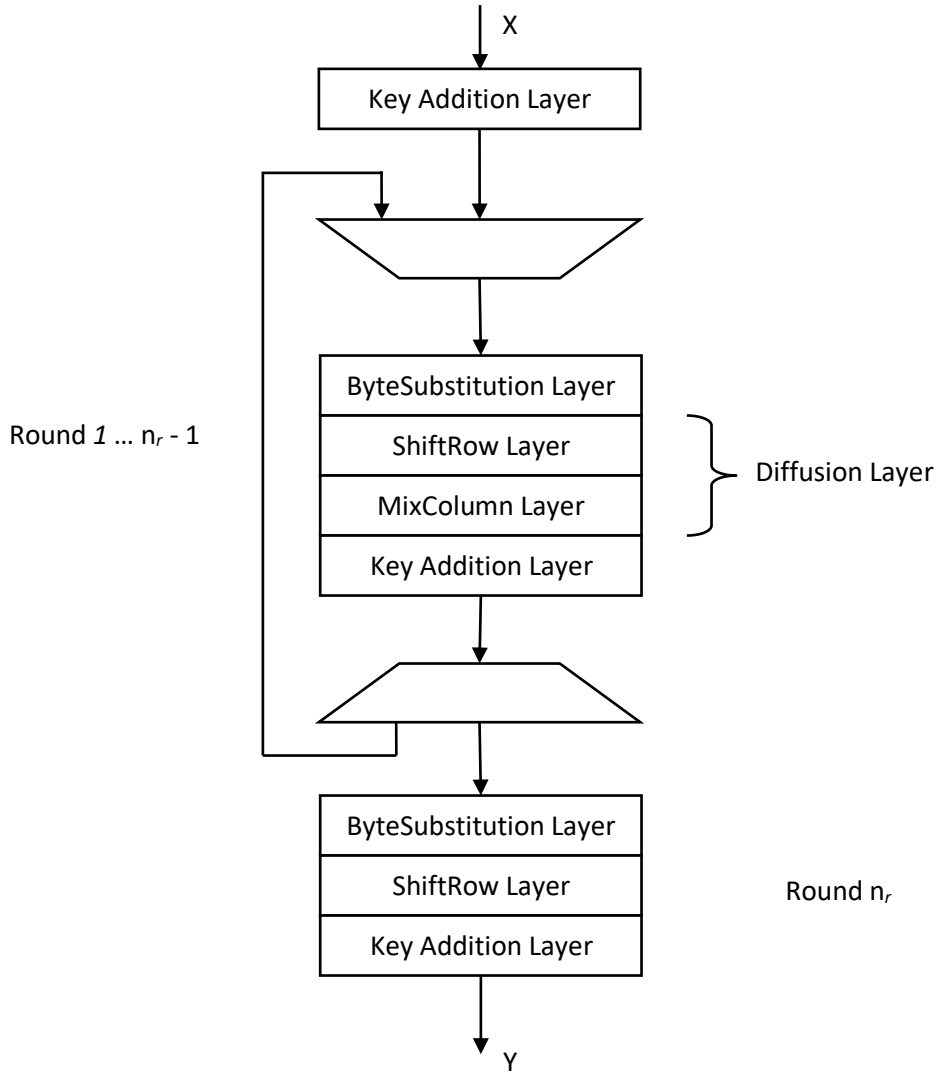
Tomado de [13]

La estructura general del AES es mostrada en la figura B.1.

AES funciona a partir de una *matriz de estado S* conformada por los bytes que se van a cifrar/descifrar ordenados de arriba abajo. Análogamente se tiene la matriz de clave en el mismo orden. Siendo B el bloque a cifrar, y S la matriz de estado, el algoritmo AES con n rondas queda como sigue:

1. Calcular K_0, K_1, …, K_n subclaves a partir de la clave K
2. S $\leftarrow$ B $\otimes$ K_0
3. Para *i* = 1 hasta *n* hacer
    a. Aplicar ronda *i*-ésima del algoritmo con la subclave K_i

Figura B.1 Algoritmo AES



Función ByteSub():

La transformación ByteSub es una substitución no lineal aplicada a cada byte de la matriz de estado. Está compuesta de dos transformaciones:

1. Cada byte es considerado como un elemento del GF(2^8) que genera el polinomio irreducible m(x) = $x^8 + x^4 + x^3 + x + 1$, y sustituido por su inversa multiplicativa. El valor cero queda inalterado.

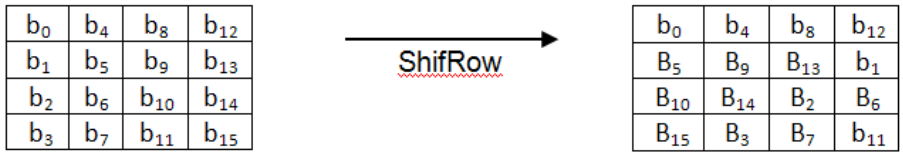
El siguiente paso consiste en aplicar la siguiente transformación afín en GF(2), siendo x_0, x_1, …, x_7 los bits del byte correspondiente, e y_0, y_1, …, y_7 los del resultado:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Función DesplazarColumna (), (ShiftRow):

Consiste en desplazar cíclicamente a la izquierda las filas de la matriz de estado. La primera fila siempre se queda igual y las siguientes se desplazan un valor dependiendo del tamaño de la matriz de estado. La figura B.2 muestra una matriz de estado de 128 bits.

Figura B.2 Transformación ShiftRow para una matriz de 128 bits

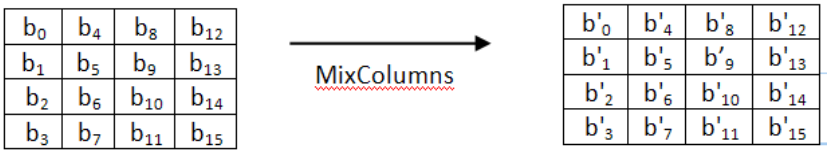


Función MezclarColumna(), (MixColumn):

Esta función mezcla cada columna independientemente haciendo una multiplicación de la columna con el polinomio: $c(x) = 03_x x^3 + 01_x x^2 + 01_x x + 02_x$, el cual pertenece al GF(2^8)^4 módulo el polinomio $x^4 + 1$ y donde los coeficientes del polinomio son valores hexadecimales correspondientes a la concatenación de los valores binarios.

La figura B.3 muestra el resultado luego de aplicar la función.

Figura B.3 Función MixColumn para una matriz de 128 bits



Donde cada columna $(b_i, b_{i+1}, b_{i+2}, b_{i+3}), i \in \{0,4,8,12\}$ es mezclada por:

$$\begin{pmatrix} b'_i \\ b'_{i+1} \\ b'_{i+2} \\ b'_{i+3} \end{pmatrix} = \begin{pmatrix} 02_x & 03_x & 01_x & 01_x \\ 01_x & 02_x & 03_x & 01_x \\ 01_x & 01_x & 02_x & 03_x \\ 03_x & 01_x & 01_x & 02_x \end{pmatrix} \begin{pmatrix} b_i \\ b_{i+1} \\ b_{i+2} \\ b_{i+3} \end{pmatrix}$$

Función AddRounKey():

Esta función simplemente realiza una operación XOR entre la subclave generada a partir de la expansión de la clave principal con los datos.