El AES (Advanced Encryption Standard)

Félix Delgado - Ana Núñez

Universidad de Valladolid

Curso 2020 - 2021

El AES (Advanced Encryption Standard)

En 1997 el NIST (National Institute for Standars and Technology) convocó públicamente un concurso para la adopción de un nuevo estándar de cifrado en bloque simétrico que sustituyese al DES.

El ganador del mismo fue el sistema llamado SHAPE RIJNDAEL, desarrollado por V. Rijmen y J. Daemen (Univ. de Lovaina). Describiremos brevemente como funciona este nuevo estándar.

El AES (Advanced Encryption Standard)

AES es un sistema de cifrado por bloques, diseñado para manejar longitudes de clave y de bloque variables, ambas comprendidas entre los 128 y los 256 bits.

Realiza varias de sus operaciones internas a nivel de byte (8 bits), interpretando éstos como elementos del cuerpo finito

$$\mathbb{F}_{256} = \mathbb{F}_{2^8} = \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1)\ .$$

El resto de operaciones se efectúan en términos de registros de 32 bits (4 bytes). Sin embargo, en algunos casos, una secuencia de 32 bits se toma como un polinomio de grado inferior a 4, cuyos coeficientes son a su vez polinomios en $\mathbb{F}_{256}$.

Bloques y claves

- Cifra bloques de longitudes 128, 192 o 256 bits. (El estandar final fija bloques de 128 bits).
- Claves de las mismas longitudes: 128, 192 o 256 bits.
- Los tamaños de ambas se fijan independientemente.
- Maneja toda la información en bytes. Luego, cifra mensajes de longitudes 16, 24 o 32 bytes con claves de longitud 16, 24 o 32 bytes.
- Mensajes y claves se manejan en forma de matrices con 4 filas.

Por tanto:

- Los mensajes son matrices de $4 \times Nb$ bytes, siendo $Nb = 4, 6, 8$.
- La clave es una matriz de $4 \times Nk$, siendo $Nk = 4, 6, 8$.

Bloques y claves

Ejemplo: El caso $Nb = 6$, corresponde a un mensaje de longitud 192 bits, es decir, 24 bytes:

$$M = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{pmatrix} \iff a_{00}, a_{10}, \ldots, a_{35} \in \mathbb{F}_2^{192} (= \mathbb{F}_{256}^{24})$$

El caso $Nk = 4$, corresponde a una clave de longitud 128 bits (16 bytes):

$$K = \begin{pmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{pmatrix} \iff k_{00}, k_{10}, \ldots, k_{33} \in \mathbb{F}_2^{128} (= \mathbb{F}_{256}^{16})$$

Algoritmo de cifrado

El cifrado consiste esencialmente en un número variable de vueltas r ($r = 10, 12, 14$) de un algoritmo básico que utiliza como entradas (para la vuelta i- ésima) una clave de vuelta K_i del mismo tamaño que el mensaje y la matriz saliente de la vuelta anterior.

Algoritmo: **Entrada:** El mensaje M y la clave K.

1. Se calculan las claves de vuelta: $K_0, K_1, ..., K_r$ (del mismo tamaño que M).
2. $S := M \bigoplus K_0$.
3. Para $i = 1, ..., r-1$ hacer $S := R[i](S, K_i)$.
4. Se hace una operación más en S, semejante a las anteriores pero un poco más sencilla: $S := R[r](S, K_r)$.
5. Devolver S.

Número de vueltas

El número de vueltas r depende de las longitudes de la clave y el mensaje y se puede ver en la siguiente tabla, donde N_b representa el número de bytes del mensaje y N_k el de la clave.

	$Nb = 4\,(128\ bits)$	$Nb = 6\,(192\ bits)$	$Nb = 8\,(256\ bits)$
$Nk = 4\,(128\ bits)$	10	12	14
$Nk = 6\,(192\ bits)$	12	12	14
$Nk = 8\,(256\ bits)$	14	14	14

(En nuestro ejemplo sería $r = 12$).

Las claves de vuelta

En cada vuelta se aplica el algoritmo básico, $R[i]$, que utiliza como entradas, en la vuelta i-ésima), una *clave de vuelta* K_i y la matriz saliente de la vuelta anterior.

Las sucesivas claves de vuelta se construyen mediante un algoritmo recursivo, relativamente complejo y que incluye en particular un proceso de expansión hasta el tamaño de clave requerido (para adaptarlo al tamaño del mensaje).

Detallaremos el algoritmo de generación de dichas claves más adelante. Por tanto suponemos que, a partir de K y del tamaño del mensaje M, se han formado claves $K_0, \ldots, K_r \ldots$, siendo K_i (la clave de la vuelta i-ésima) una matriz del mismo tamaño que el mensaje original.

El algoritmo de cifrado (cifrado Rijndael)

Entrada: El mensaje M y las claves $K_0, \ldots, K_r$.

- Iniciamos con $S := M + K_0$
- Para $i = 1, \ldots, r-1$ hacemos:
  1. $S := \text{ByteSub}(S)$.
  2. $S := \text{ShiftRow}(S)$.
  3. $S := \text{MixColumn}(S)$.
  4. $S := \text{AddRoundKey}(S, K_i)$.
- $S := \text{ByteSub}(S)$
- $S := \text{ShiftRow}(S)$
- $S := \text{AddRoundKey}(S, K_r)$

El algoritmo de cifrado (cifrado Rijndael)

Obsérvese que hay una operación previa, después $r-1$ vueltas (rondas) idénticas (con claves de vuelta diferentes) y, finalmente, una vuelta más pero sin una de las etapas.

En cada vuelta la clave solo interviene en la última operación. Vamos a describir las operaciones en los pasos $1-4$.

Operación ByteSub

• Es una operación sobre cada una de las entradas (bytes) de la matriz S que consiste, para $a = (a_1, \ldots, a_8) \in \mathbb{F}_{256}$ en:

- Sustituir a por su inverso, $a := 1/a$ (si $a \neq 0$).
- Hacer un cifrado afín de Hill, $a \mapsto a \cdot A + b$, siendo A una matriz binaria fija de tamaño 8×8 y b un vector fijo de 8 bits.

El efecto de la operación ByteSub es un cifrado por substitución monoalfabética usando como alfabeto $\mathcal{A} = \mathbb{F}_2^8 = \mathbb{F}_{256}$.

La operación de invertir un elemento no es lineal, por lo que la sustitución en su conjunto no es lineal sobre $\mathbb{F}_2$.

Operación ByteSub

La matriz A y el vector b son los siguientes:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad b = (1, 1, 0, 0, 0, 1, 1, 0)$$

La operación ByteSub se puede ver como una S-caja y se puede implementar como tal o bien mediante la programación de las operaciones (admite simplificaciones importantes).

Operación ShiftRow

• Las filas de la matriz S se permutan por medio de una permutación que depende del número de columnas de S (que es el mismo que el de M).

Denotamos por σ la permutación circular a la izquierda:
$\sigma(a, b, c, d) = (b, c, d, a)$, $\sigma(a, b, c, d, e, f) = (b, c, d, e, f, a)$.

El efecto de la operación sobre la fila i-ésima de S, f_i, es $\sigma^i(f_i)$. Nótese que la primera fila queda inalterada. (Observación: En el caso $Nb = 8$ es ligeramente diferente.)

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \mapsto \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{11} & a_{12} & a_{13} & a_{10} \\ a_{22} & a_{21} & a_{20} & a_{21} \\ a_{33} & a_{30} & a_{31} & a_{32} \end{pmatrix}$$

MixColumn y AddRoundKey

MixColumn

Opera sobre cada columna $c = (c_0, c_1, c_2, c_3)$ de la matriz S.
Escribimos la columna en forma polinómica:
$c(T) = c_0 + c_1 T + c_2 T^2 + c_3 T^3$ y se hace la transformación

$$c(T) := c(T)d(T) \mod (T^4 + 1)$$

siendo $d(T) \in \mathbb{F}_{256}[T]$ un polinomio fijo de grado 3.
Los elementos de $\mathbb{F}_{256}$ se identifican con polinomios en x de grado menor que 8.
O dicho de otra manera, identificamos:

$$\mathbb{F}_{256} \simeq \mathbb{F}_2[x]/(1 + x + x^3 + x^4 + x^8) .$$

De esta forma el polinomio $d(T)$ es:

$$d(T) = x + T + T^2 + (1 + x)T^3 .$$

MixColumn y AddRoundKey

La operación anterior se puede escribir en forma matricial como:

$$(c_0, c_1, c_2, c_3) \mapsto (c_0, c_1, c_2, c_3) \begin{pmatrix} x & 1 & 1 & 1+x \\ 1+x & x & 1 & 1 \\ 1 & 1+x & x & 1 \\ 1 & 1 & 1+x & x \end{pmatrix}$$

$$S \mapsto \begin{pmatrix} x & 1+x & 1 & 1 \\ 1 & x & 1+x & 1 \\ 1 & 1 & x & 1+x \\ 1+x & 1 & 1 & x \end{pmatrix} \cdot S$$

Por lo tanto la transformación es un cifrado de Hill en bloques de tamaño 4 sobre el alfabeto $\mathbb{F}_{256}$.

AddRoundKey: La matriz S se sustituye por la matriz $S + K_i$.

Generación de las claves de vuelta

Partimos de la matriz de clave

$$K = \begin{pmatrix} k_{00} & k_{01} & \dots & k_{0,Nk-1} \\ k_{10} & k_{11} & \dots & k_{1,Nk-1} \\ k_{20} & k_{21} & \dots & k_{2,Nk-1} \\ k_{30} & k_{31} & \dots & k_{3,Nk-1} \end{pmatrix}$$

Se realizan dos operaciones:

1. Una expansión de K hasta completar una matriz W de tamaño $4 \times (r+1)Nb$ añadiendo consecutivamente columnas a la matriz K.
2. Posteriormente se toman como matrices K_i, $i = 0, \dots, r$, las sucesivas $r+1$ cajas de tamañoo $4 \times Nb$ de la matriz $W = (K_0, ..., K_r)$.

Expansión de la clave

Denotamos por $W(i)$, $(0 \leq i < (r+1)Nb)$, las columnas de la matriz W. Recordemos que las Nk primeras coinciden con las de la matriz K. El resto se calculan de acuerdo al siguiente algoritmo:

Algoritmo de Generación de claves

Entrada: Las columnas $W(0), \ldots, W(Nk-1)$.

Para $i = Nk, \ldots, (r+1)Nb$ hacer:

- Si $i \equiv 0 \mod Nk$ entonces:
  1. $t := ByteSub(\sigma(W(i-1))) \bigoplus (x^{i-1}, 0, 0, 0)$
  2. $W(i) := W(i-Nk) \bigoplus t$
- En otro caso $W(i) := W(i-1) \bigoplus W(i-Nk)$.

La operación σ indica una permutación circular a la izquierda.

Se puede encontrar toda la información detallada, en particular el algoritmo para la generación de las distintas claves de vuelta, en la página del NIST

http://csrc.nist.gov/archive/aes/rijndael/wsdindex.html.