ALGORITMO DE CIFRADO SIMÉTRICO AES.
ACELERACIÓN DE TIEMPO DE CÓMPUTO SOBRE ARQUITECTURAS MULTICORE.

Trabajo Final de Arquitecrtura del Computador
Carrera de Ingenieria de Sistemas

Autores: Luis Carlos Bolaños Villalobos / Esteban Costo.

Profesor: Andrés López

Curso: Arquitectura del Computador

Universidad Católica de Costa Rica

Agosto de 2018

Contenido

Objetivo 1

Introducción 3

Capítulo 1

Criptografía 5

1.1 Definición 5

1.2 Sistemas de cifrado simétricos 5

1.3 Sistemas de cifrado asimétricos 6

Capítulo 2

Algoritmo AES 9

2.1 Historia del algoritmo AES 9

2.2 Mención sobre los fundamentos matemáticos 9

2.3 Bloques AES 10

2.4 Claves 10

2.5 Cifrado AES: Rondas y operaciones 14

2.6 Descifrado AES 17

Capítulo 3

Arquitecturas multicore y herramientas paralelas 19

3.1 Evolución hacia las arquitecturas multicore 19

3.2 Herramientas paralelas 22

Capítulo 4

Algoritmo AES Implementaciones 25

4.1 Implementación y software utilizado 25

4.2 Implementación secuencial 26

4.3 Implementaciones paralelas 26

Capítulo 5

Análisis de rendimiento 29

5.1 Hardware utilizado 29

5.2 Tiempos de ejecución 30

5.3 Aceleración (SpeedUp) 31

Capítulo 6

Conclusiones y trabajo a futuro 33

6.1 Conclusiones..... 33

6.2 Trabajo a futuro..... 34

Bibliografía básica 37

Objetivo

El objetivo de este trabajo es mostrar la aceleración en el tiempo de cómputo del algoritmo criptográfico Advanced Encryption Standard (AES) con clave de tamaño 128bits, que se obtiene al aprovechar el paralelismo que proveen las arquitecturas multicores actuales utilizando herramientas de programación paralela.

AES es uno de los algoritmos de criptografía más usados en la actualidad, con el crecimiento de las redes y la información que se maneja hoy en día puede ser necesario cifrar un volumen muy grande de información para lo que se requiere mayor velocidad en los procesadores, pero esto actualmente no es posible debido a que los procesadores han llegado al límite de velocidad por problemas térmicos y de consumo, por esta razón se está incrementando la cantidad de procesadores en los equipos.

Como aporte de la concreción de este trabajo se pretende presentar un análisis de rendimiento que muestre cómo a pesar de las limitaciones de velocidad de los procesadores, es posible, mediante herramientas de programación paralela, aprovechar las arquitecturas multicore para acelerar el cómputo del algoritmo AES y así reducir el tiempo de cifrar información ya sea para almacenarla o enviarla por la red.

Introducción

Desde mucho antes que existieran las computadoras los seres humanos han tenido la necesidad de intercambiar mensajes de forma segura, de manera que solo puedan ser leídos por las personas a quienes van dirigidos. A partir de esta necesidad es que nace la criptografía, esta provee técnicas de codificación y decodificación de información para un intercambio seguro. Desde la aparición de las computadoras y más aun con el crecimiento de las redes, la necesidad de intercambiar información de manera segura fue mayor y es donde aparecen implementaciones de distintos sistemas de cifrado.

Existen dos sistemas de cifrado que involucran distintos tipos de algoritmos, los sistemas simétricos y los sistemas asimétricos, ambos con sus ventajas y sus desventajas.

Dentro de los sistemas simétricos se encuentra el algoritmo Advanced Encryption Standard (AES), que es uno de los algoritmos más utilizados en la actualidad, considerado por el gobierno de los Estados Unidos como un algoritmo seguro para protección nacional de información y del cual, aun no se conocen ataques eficientes que puedan vulnerarlo.

Hoy en día la el volumen de información que se maneja en ocasiones es muy grande y en algunos casos es necesario almacenar esta información o transmitirla en forma segura, AES es un algoritmo simple y rápido, pero aun así, el tiempo de computo de cifrar un gran volumen de información puede ser importante, es necesario en estos casos contar con procesadores más veloces.

Los procesadores han venido siendo más rápido cada año, pero en la actualidad este crecimiento se ve interrumpido por problemas térmicos y de consumo dentro de los procesadores. Por esta razón es que las arquitecturas actuales tienen más de un procesador de manera de poder aprovechar el paralelismo que proveen, y en la medida que se pueda poder acelerar el tiempo de cómputo de las aplicaciones.

Siempre fue común el uso de arquitecturas como multiprocesadores y clusters dentro del ámbito académico-científico, hoy en día los costos de estas arquitecturas han bajado y es posible encontrarlas fuera de este ámbito; también se le ha dado importancia a las placas para procesamiento grafico (GPU) que han sido usadas con mucho éxito para aplicaciones de propósito general.

Estas arquitecturas junto con el uso de herramientas de programación paralela adecuadas para cada caso, como son OpenMP, MPI y CUDA permiten paralelizar algoritmos de manera de acelerar el tiempo de cómputo.

Como muestra este trabajo, el algoritmo AES puede ser implementado con herramientas paralelas y ejecutado sobre arquitecturas multicores de manera de reducir el costo de cifrado y descifrado de datos ya sea para almacenarlos o para hacer una transferencia importante de datos sensibles sobre una red pública.

Capítulo 1

Criptografía

1.1 Definición

La criptografía (del griego oculta y escribir, literalmente escritura oculta) es el arte o ciencia de cifrar (encriptar) y descifrar (desencriptar) información utilizando técnicas que hagan posible el intercambio de mensajes de manera segura que solo puedan ser leídos por las personas a quienes van dirigidos.

Existen dos tipos básicos de sistemas de cifrado:

- Sistemas de cifrado *simétricos* (o sistemas de clave secreta o de clave privada).
- Sistemas de cifrado *asimétrico* (o sistemas de clave pública).

1.2 Sistemas de cifrado simétricos

Los sistemas simétricos utilizan la misma clave para encriptar y desencriptar. Existen dos modos de operación básicos:

- Cifrado en bloques: La información a cifrar se divide en bloques de longitud fija (8,16, ... bytes) y luego se aplica el algoritmo de cifrado a cada bloque utilizando una clave secreta. Ejemplos: DES, AES.
  Existen distintos modos de operación dependiendo de como se mezcla la clave con la información a cifrar:

  - Modo ECB (Electronic Codebook): El texto se divide en bloques y cada bloque es cifrado en forma independiente utilizando la clave. Tiene la desventaja que puede revelar patrones en los datos.

  - Modo CBC (CBC): El texto se divide en bloques y cada bloque es mezclado con la cifra del bloque previo, luego es cifrado utilizando la clave.

  - Modos CFB (Cipher FeedBack) y OFB (Output FeedBack).

- Cifrado de flujo: Para algunas aplicaciones, como el cifrado de conversaciones telefónicas, el cifrado en bloques es inapropiada porque los datos se producen en tiempo real en pequeños fragmentos. Las muestras de datos pueden ser tan pequeñas como 8 bits o incluso de 1 bit. El algoritmo genera una secuencia pseudoaleatoria (secuencia cifrante o keystream en inglés) de bits que se emplea como clave. El cifrado se realiza combinando la secuencia cifrante con el texto claro. Ejemplo: RC4.

Los sistemas simétricos tienen las siguientes ventajas:

- Gran velocidad de cifrado y descifrado.
- No aumenta el tamaño del mensaje
- Tecnología muy conocida y difundida.

Pero presentan las siguientes desventajas:

- La seguridad depende de un secreto compartido entre el emisor y el receptor.
- La administración de las claves no es "escalable".
- La distribución de claves debe hacerse a través de algún medio seguro.

1.3 Sistemas de cifrado asimétricos

Los sistemas asimétricos utilizan dos claves, una privada y una publica (siendo una la inversa de la otra). Ambas pueden ser usadas para encriptar y desencriptar información. Dichas claves están matemáticamente relacionadas entre sí:

- La clave pública está disponible para todos.
- La clave privada es conocida solo por el individuo.

Existen varios algoritmos muy utilizados por ejemplo Diffie-Hellman , RSA, DSA.

Este sistema tiene además dos modos de cifrado:

- Encripción: el mensaje es encriptado usando la clave pública del receptor, el mensaje encriptado es enviado al destinatario, el mensaje recibido se desencripta usando la clave privada del receptor, garantizando así la confidencialidad del mensaje.

- Autenticación: el mensaje es encriptado usando la clave privada del emisor, el mensaje encriptado se envía a uno o más receptores, el mensaje se desencripta usando la clave pública del emisor. Esto garantiza la autenticidad del emisor y la integridad del mensaje.

Este sistema de cifrado tiene las siguientes ventajas:

- No se intercambian claves.
- Es una tecnología muy difundida.
- Sus modos cubren los requisitos de seguridad de la información.

Pero presentan las siguientes desventajas:

- Requiere potencia de cómputo.
- El tamaño del mensaje cifrado es mayor al del original.

Capítulo 2

Algoritmo AES

2.1 Historia del algoritmo AES

Es un algoritmo de cifrado simétrico desarrollado por los estudiantes Vincent Rijmen y Joan Daemen de la Katholieke Universiteit Leuven en Bélgica, bajo el nombre "Rijndael" fue presentado en 1997 al concurso organizado por el Instituto Nacional de Normas y Tecnologías (NIST) para elegir el mejor algoritmo de cifrado; el algoritmo gano el concurso transformándose en un estándar en el año 2002, con algunos cambios fue posteriormente renombrado AES (Advanced Encryption Standard) y se convirtió en uno de los algoritmos más utilizados en la actualidad.

En 2003, el gobierno de los Estados Unidos anuncio que el algoritmo era lo suficientemente seguro y que podía ser usado para protección nacional de información. Hasta el momento no se conocen ataques eficientes, los únicos conocidos son los denominados ataques de canal auxiliar[1].

2.2 Mención sobre los fundamentos matemáticos

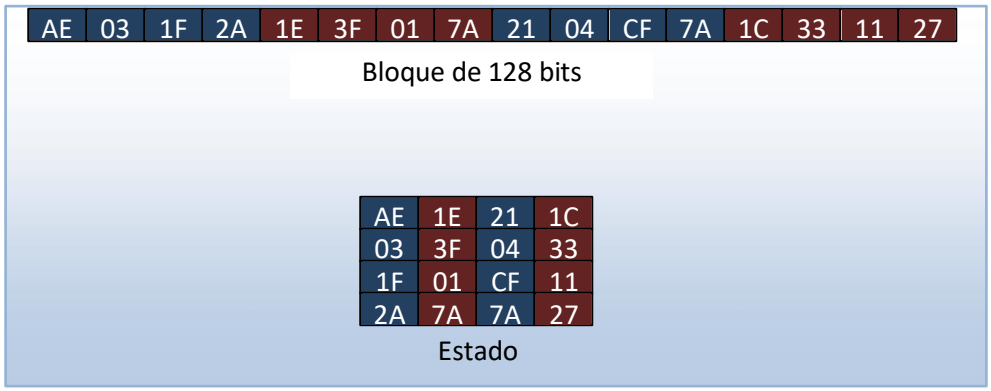
[1] Un ataque de canal auxiliar no ataca al algoritmo de cifrado sino que aprovecha vulnerabilidades de las implementaciones que pueden revelar datos a medida que se realiza el cifrado.[12] [13] [14]

AES toma como elemento básico al byte (8 bits) y ve a los bytes como elementos del campo finito de Galois o GF(2^8), toda operación del algoritmo está basada en operaciones sobre este campo finito, rotaciones de bytes y operaciones de suma módulo 2.

No es objetivo de este trabajo explicar en detalle los extensos fundamentos matemáticos en los que se basa el algoritmo AES.

2.3 Bloques AES

AES es un algoritmo de cifrado por bloques, inicialmente fue diseñado para tener longitud de bloque variable pero el estándar define un tamaño de bloque de 128 bits, por lo tanto los datos a ser encriptados se dividen en segmentos de 16 bytes (128 bits) y cada segmento se lo puede ver como un bloque o matriz de 4x4 bytes al que se lo llama estado, este se organiza de la siguiente forma:

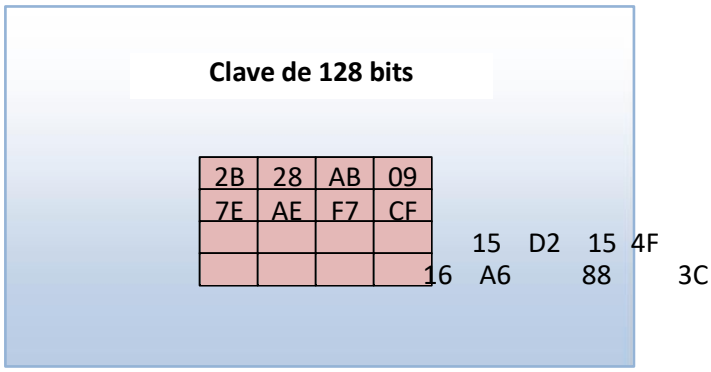


2.4 Claves

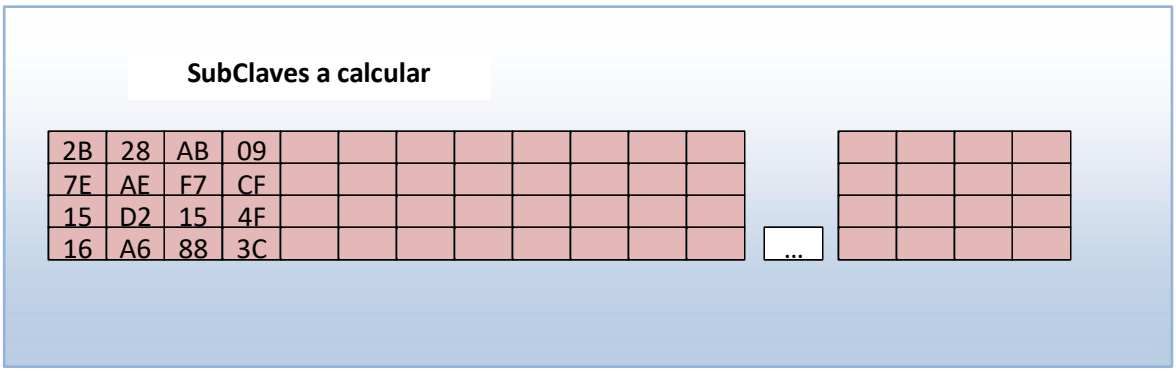
Por ser simétrico, se utiliza la misma clave para encriptar como para desencriptar, la longitud de la clave puede ser de 128, 192 o 256 bits según especifica el estándar, esto permite tres implementaciones conocidas como AES-128, AES-192 y AES-256, el presente trabajo está basado en AES-128.

Partiendo de una clave inicial de 16 bytes (128 bits), que también se la puede ver como un bloque o matriz de 4x4 bytes, se generan 10 claves, estas claves resultantes junto con la *clave inicial* son denominadas *subclaves*.

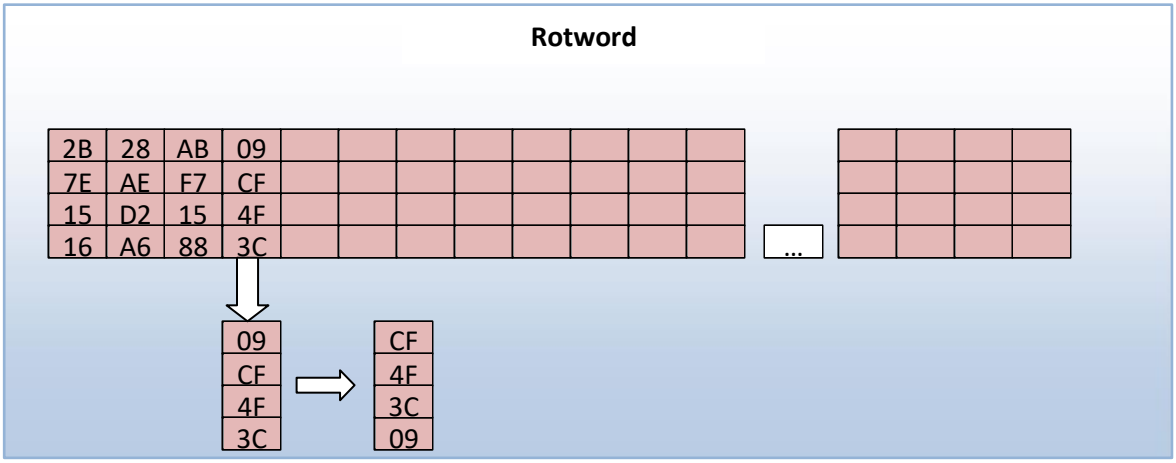
El proceso de generación de subclaves parte de la clave inicial vista como una matriz de 4x4 bytes:



Para mostrar claramente como se calculan las subclaves, el conjunto de subclaves puede verse como una matriz de 4 filas x 44 columnas, o sea una subclave a continuación de otra:

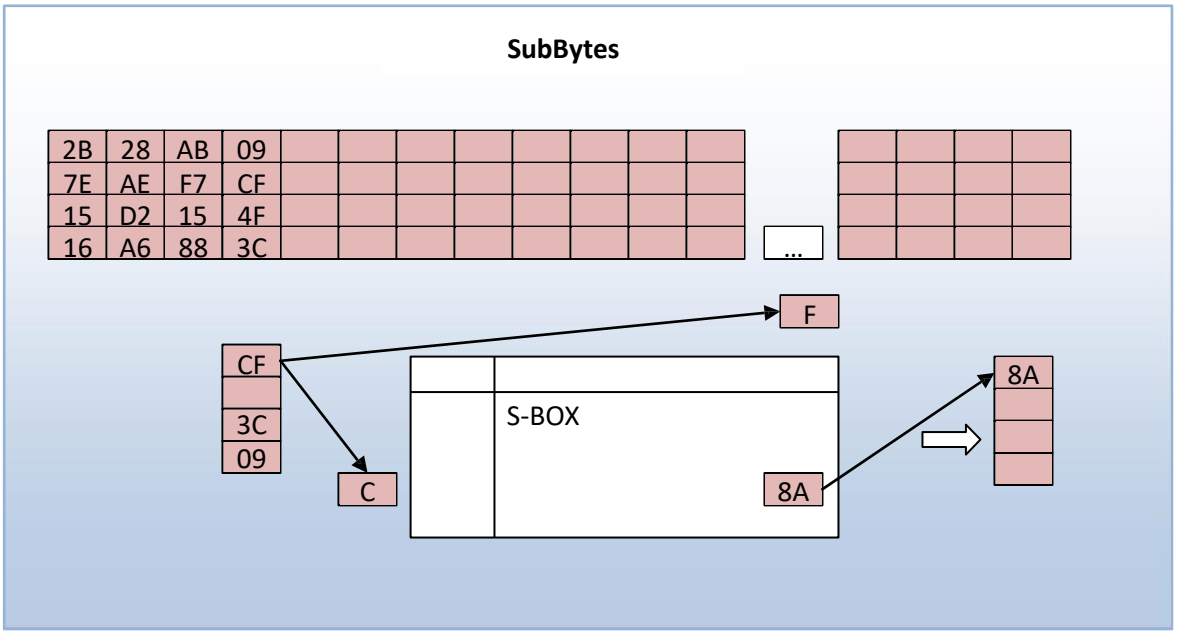


Para calcular la primera columna de la siguiente subclave se toma la última columna de la subclave anterior (en este caso la clave inicial) y se aplica una operación llamada Rotword que consiste en realizar una rotación del primer byte hacia el último lugar en la columna:

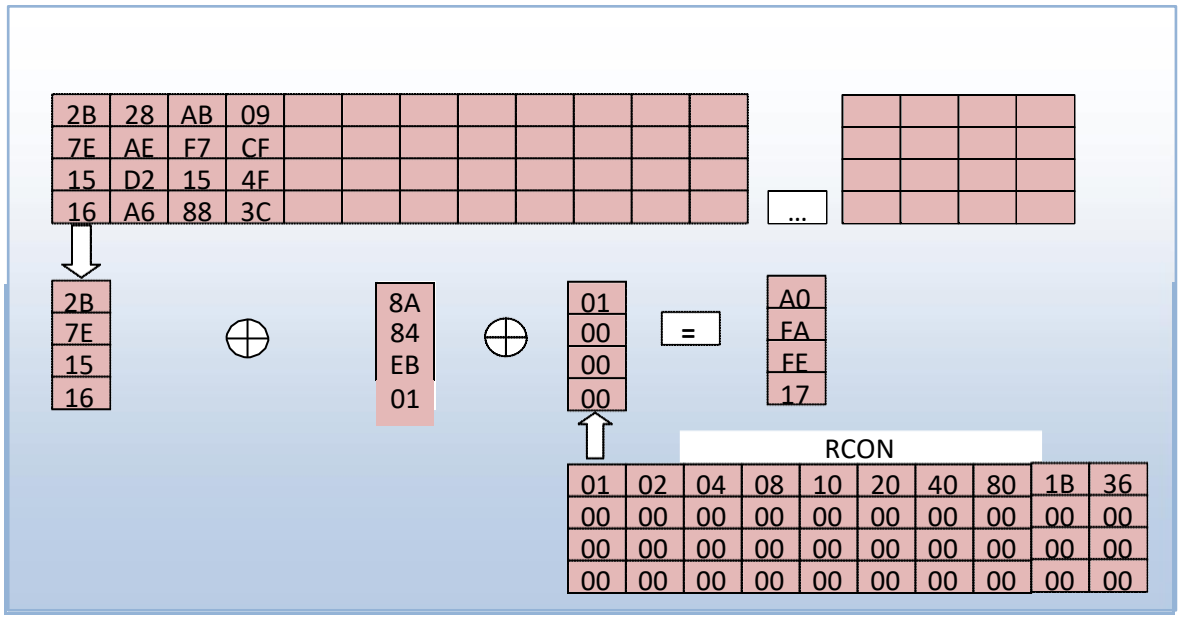


Luego, a la columna resultante, se aplica una operación llamada SubBytes que consiste en

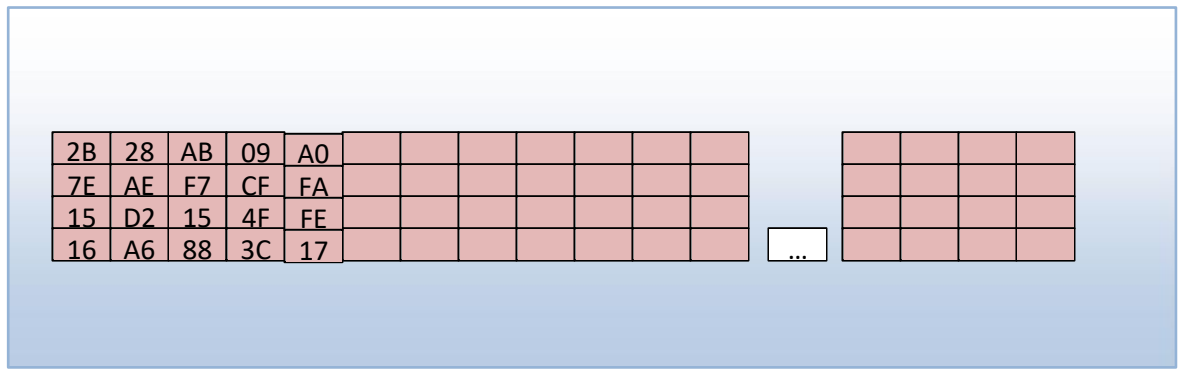
reemplazar cada byte de la columna ya rotada por un byte almacenado en una tabla llamada S-Box, esta tabla contiene pre calculados el resultado de aplicarle a cada byte la inversión en el campo GF y una transformación afín, la dimensión de la tabla es de 16x16 bytes donde los índices tanto de las columnas como de las filas van de 0 a F, para obtener la transformación S-Box de un byte se toman los primeros 4 bits como el índice de la fila de la tabla y los segundos 4 como índice de la columna de la tabla:



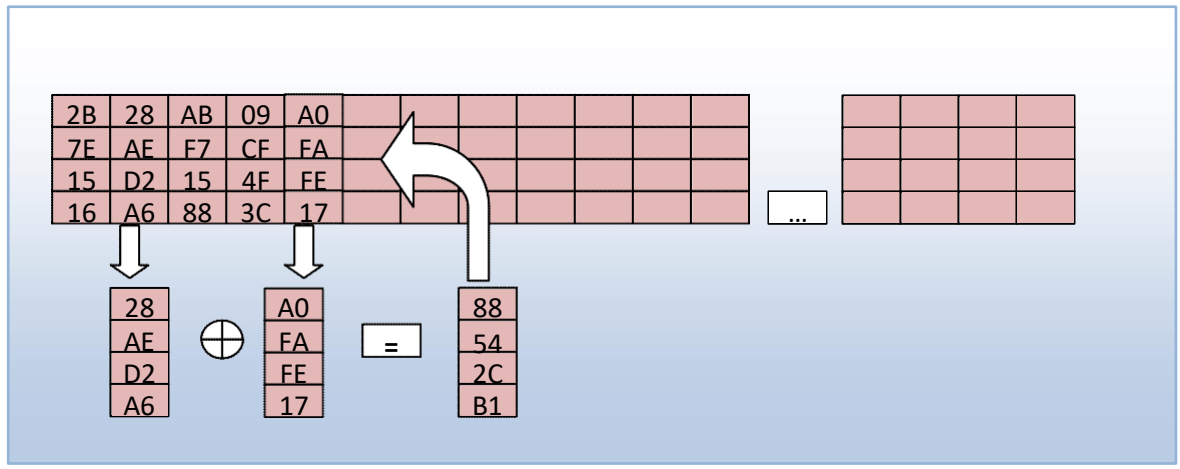
Luego al resultado se le aplica un XOR byte a byte con la columna 4 posiciones atrás (en este caso la primer columna de la clave inicial) y un XOR byte a byte con una columna de una tabla llamada RCON que mantiene en la primer fila constantes 2^i en el campo GF y en las restantes filas 0, por ser la primer subclave la que estamos calculando se toma para el cálculo la primer columna de la tabla RCON, para las siguientes subclaves se toma la próxima columna no utilizada de esta tabla:



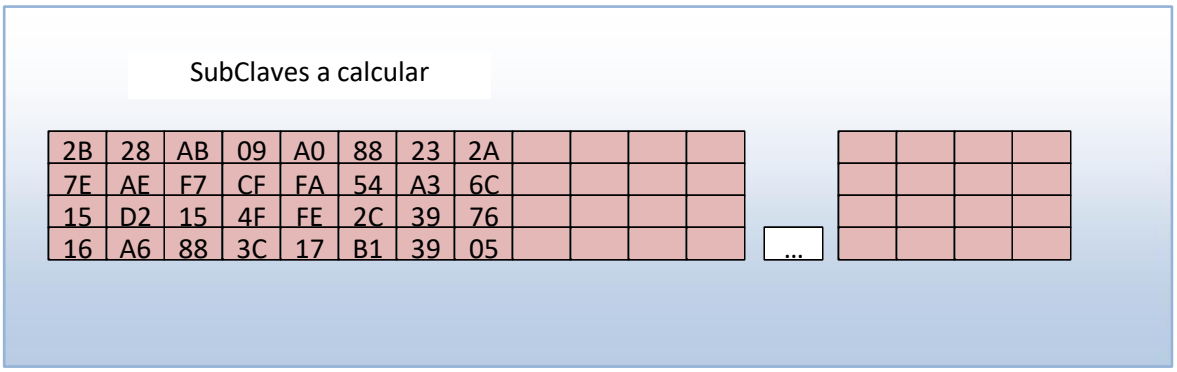
El resultado de esta última operación será la primera columna de la subclave calculada (en este caso la segunda subclave siguiente a la inicial):



Para calcular las tres columnas siguientes se hace un XOR entre la columna anterior y la columna de cuatro posiciones atrás:



Una vez aplicadas estas operaciones se tiene una nueva subclave:



Y se procede de la misma forma para calcular las siguientes subclaves.

Al finalizar se tendrán 11 subclaves, cada una de estas subclaves se aplica en una de las rondas de operaciones que se explican en detalle a continuación.

2.5 Cifrado AES: Rondas y operaciones

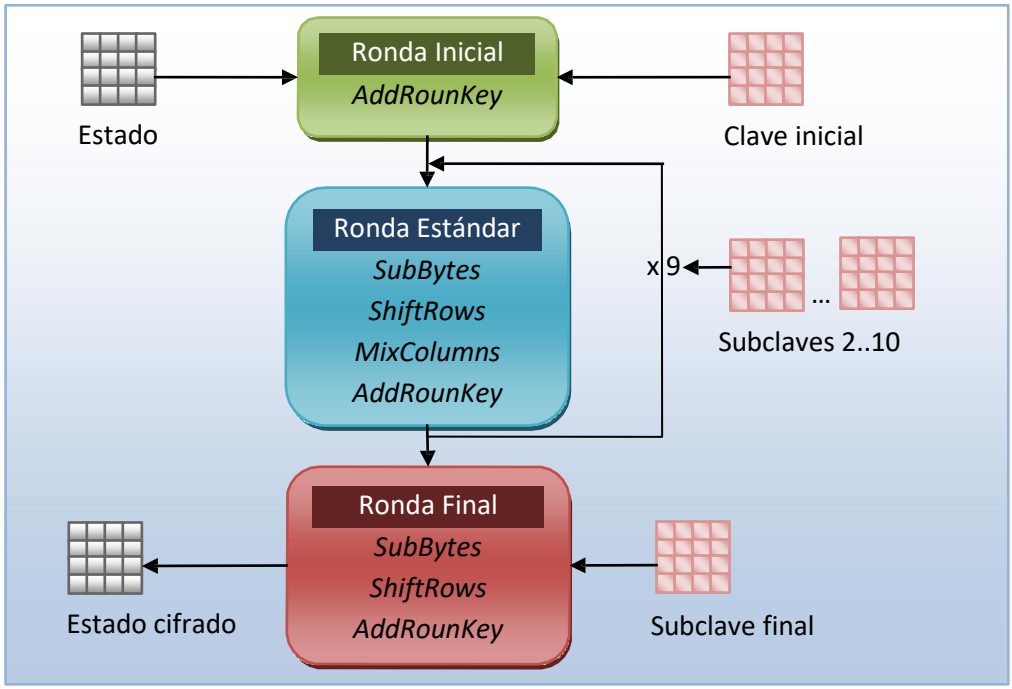
El proceso de cifrado del algoritmo consiste en aplicar a cada estado un conjunto de operaciones agrupadas en lo que se denominan rondas, el algoritmo realiza 11 rondas, donde en cada ronda se aplica una subclave diferente.
Las 11 rondas se pueden clasificar en 3 tipos:

- 1 ronda inicial (se aplica la subclave inicial).
- 9 rondas estándar (se aplican las 9 subclaves siguientes, una en cada ronda).
- 1 ronda final (se aplica la última subclave).

Las operaciones que realiza el algoritmo dentro de las rondas se reducen a 4 operaciones básicas:

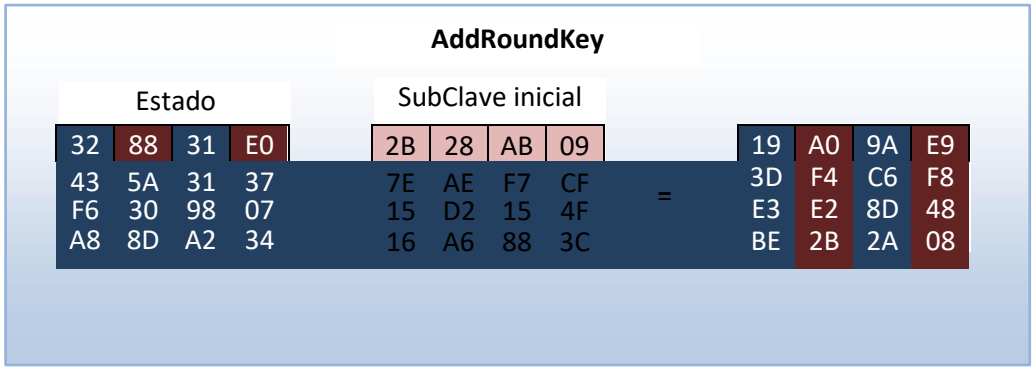
- SubBytes.
- ShiftRows.
- MixColumns.
- AddRoundKey.

A continuación se muestra un diagrama de como se aplican las operaciones y claves en cada una de las rondas:



2.5.1 Ronda inicial

La ronda inicial aplica solamente la operación **AddRoundKey** que no es más que un XOR byte a byte entre el bloque a cifrar y la clave inicial.



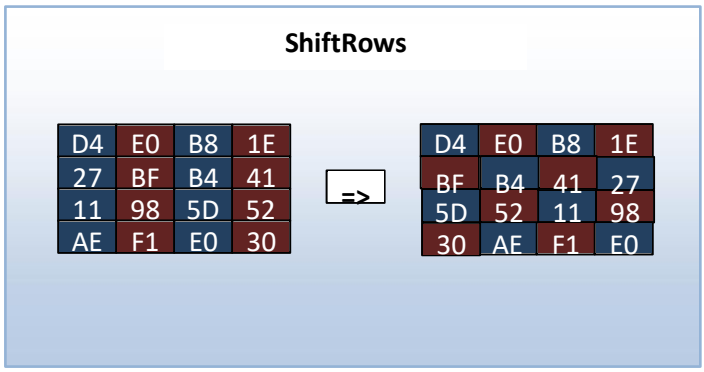
2.5.2 Rondas estándar

Luego se realizan 9 rondas estándar donde cada ronda consiste en las siguientes operaciones:

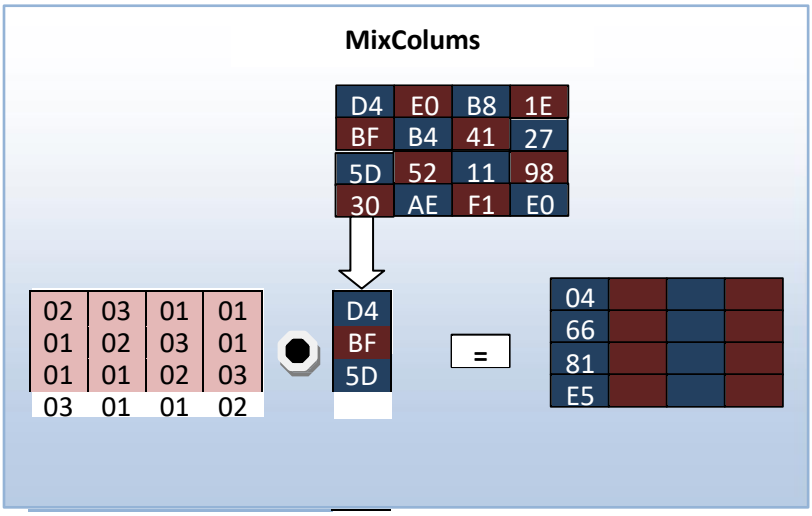
SubBytes: Cada byte del estado se reemplaza por otro valor de acuerdo a la tabla de sustitución de bytes S-Box ya vista en el cálculo de las subclaves.

ShiftRows: En cada fila del estado, a excepción de la primera, se rotan

circularmente hacia la izquierda los bytes, en la segunda fila se rotan una posición, en la tercera dos posiciones y en la cuarta tres posiciones.



MixColumns: A cada columna del estado se le aplica una transformación lineal, esto es multiplicarlo por una matriz predeterminada en el campo GF.



AddRoundKey: Se aplica la misma operación que en la ronda inicial pero utilizando otra subclave.

2.5.3 Ronda final

Por último la ronda final consiste en las operaciones de:

SubBytes: igual al de la ronda estándar.

ShiftRows: igual al de la ronda estándar.

AddRoundKey: igual al de la ronda inicial y estándar pero aplicando la última subclave.

2.6 Descifrado AES

El proceso de descifrado aplica las mismas operaciones que el cifrado pero de forma inversa utilizando las mismas subclaves generadas en orden inverso, además se utiliza una matriz distinta en la operación MixColumns de manera de obtener la inversa de la transformación lineal aplicada en el proceso de cifrado.

Capítulo 3

Arquitecturas multicore y herramientas paralelas

3.1 Evolución hacia las arquitecturas multicore

Desde prácticamente sus inicios, las computadoras han sido más rápidas cada año, hoy en día este crecimiento se ve interrumpido debido a problemas térmicos y de consumo en los procesadores, por este motivo los fabricantes decidieron integrar múltiples procesadores en el mismo circuito integrado dando lugar a las arquitecturas multicore existentes en el mercado actual, de esta manera se puede aprovechar el paralelismo que estas arquitecturas proveen para, en la medida que se pueda, acelerar el computo haciendo más rápidas las aplicaciones.

3.1.1 Maquinas multiprocesadores, multicores y clusters

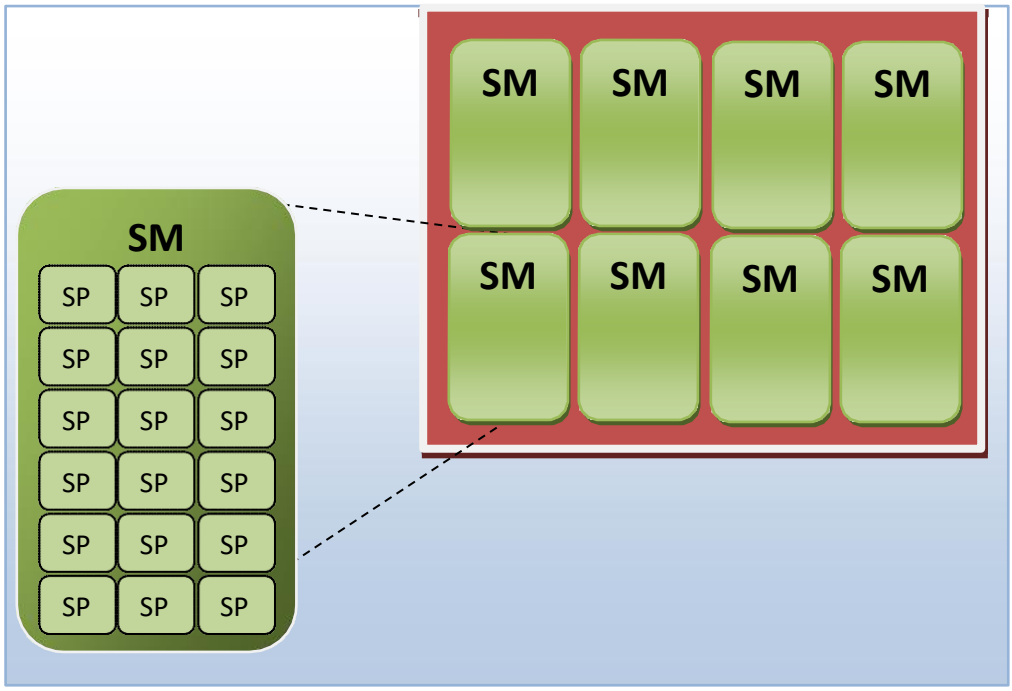
Casi desde el primer momento que se utilizaron procesadores existían maquinas con más de un procesador (ILLIAC IV o CRAY-1), aunque no en el mismo circuito integrado; el costo de tener una máquina de estas características era muy elevado, además del gran tamaño que tenían, el crecimiento de la tecnología en el ámbito de las redes permitió que se pudieran conectar maquinas en red pudiendo verlas como una maquina multiprocesador, a esto de lo llamó clúster. El clúster tiene la ventaja de poder escalar en el número de procesadores y en un principio tenía un costo mucho menor a las maquinas multiprocesador, por lo tanto se convirtió en una de las maquinas paralelas más usadas en el ámbito académico-científico. La evolución de los clusters dio lugar a arquitecturas multicluster (varios clusters conectados en red) y arquitecturas actuales más complejas como son Grid y Cloud.

Hoy en día los costos de tener una maquina con más de un procesador no son tan altos y son accesibles, los procesadores además están dentro de un mismo circuito integrado reduciendo el tamaño que tenían este tipo de máquinas en un principio; si además consideramos tenerlas conectadas en red nos da la posibilidad de tener clusters con una gran cantidad de procesadores.

3.1.2 Graphics Processing Units

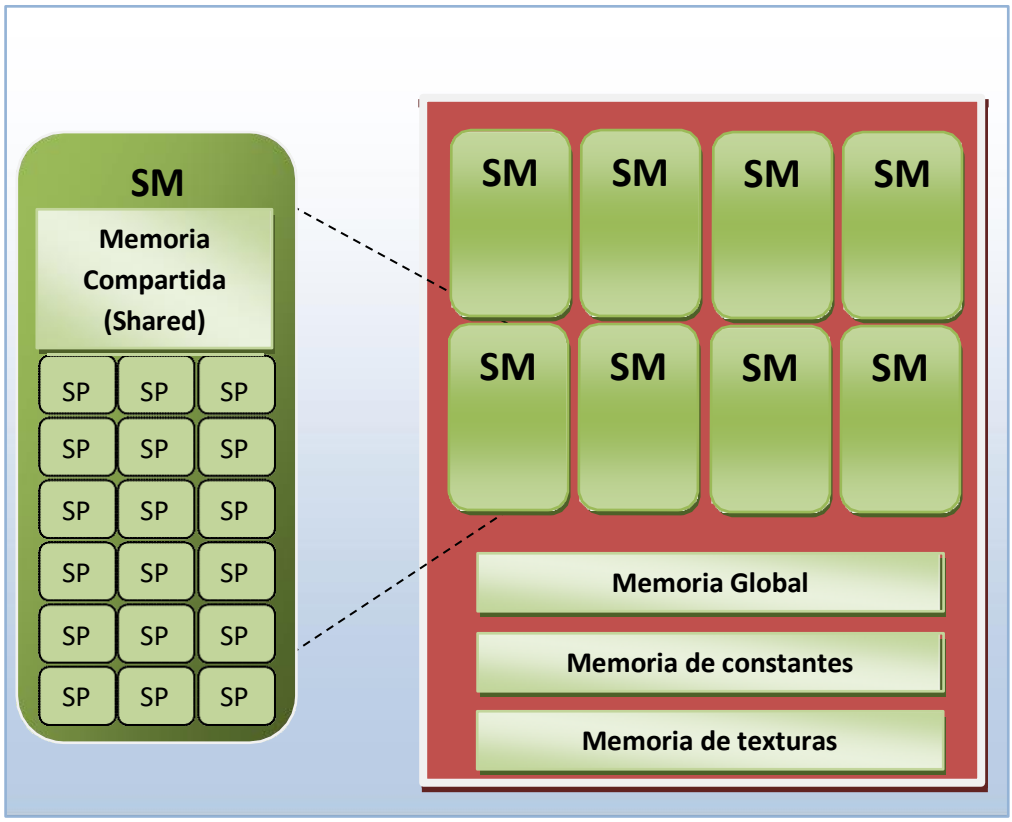
En los últimos años se le dio importancia a las GPU (Graphics Processing Units), estas son una arquitectura multicore (o también llamadas manycores) con una gran cantidad de procesadores simples dedicados a procesamiento gráfico, pero se logró utilizarlas con mucho éxito en aplicaciones de propósito general logrando un muy buen rendimiento, por este motivo también se las conoce como GPGPU (General Purpose GPU).

Las GPU están compuestas por un conjunto de Streaming Multiprocessors (SMs), cada uno posee cores simples, denominados Streaming processors (SP). Cada SM es capaz de ejecutar simultáneamente una gran cantidad de hilos (el límite depende de la arquitectura), lo cual permite que los SP estén siempre realizando trabajo útil, aun cuando parte de dichos hilos están esperando por accesos a memoria.



Las GPU poseen distintos tipos de memoria:

- Una memoria global que comparten todos los SMs y que su acceso es costoso por lo tanto hay que minimizar la cantidad de accesos.
- Una memoria de constantes de solo lectura y una memoria de texturas también compartidas por todos los SMs.
- Una memoria compartida o shared de acceso rápido ubicada en cada SM y de acceso solo restringido a este.
- Registros internos a cada SM.



3.2 Herramientas paralelas

SharpAESCrypt

Es una implementacion en C# de “AESCrypt file format”, una libreria multiplataforma de Open Source creada para llevar la encripcion AES a diferentes lenguajes de programacion y distintas plataformas, entre ellas Android, Windows, Linux, en versions para lenguajes como son Python, C#, C++, C entre otros, SharpAESCrypt es el corazon de la aplicacion **WAES**, e implementa lectura asincronica para mayor velocidad de encripción o desencripcion.