

Advanced Encryption Standard

De Wikipedia, la enciclopedia libre

Advanced Encryption Standard (**AES**), también conocido como **Rijndael** (pronunciado "Rain Doll" en inglés), es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos. El AES fue anunciado por el Instituto Nacional de Estándares y Tecnología (NIST) como FIPS PUB 197 de los Estados Unidos (FIPS 197) el 26 de noviembre de 2001 después de un proceso de estandarización que duró 5 años. Se transformó en un estándar efectivo el 26 de mayo de 2002. Desde 2006, el AES es uno de los algoritmos más populares usados en criptografía simétrica.

El cifrado fue desarrollado por dos criptólogos belgas, Joan Daemen y Vincent Rijmen, ambos estudiantes de la *Katholieke Universiteit Leuven*, y enviado al proceso de selección AES bajo el nombre "Rijndael".

Índice

- 1 Historia
- 2 Desarrollo
- 3 Descripción del cifrado
 - 3.1 Pseudocódigo
 - 3.2 Etapa SubBytes- Substitución de bits
 - 3.3 Etapa ShiftRows-Desplazar filas
 - 3.4 Etapa MixColumns- Mezclar columnas
 - 3.5 Etapa AddRoundKey- Cálculo de las subclaves
 - 3.6 Optimización del cifrado
- 4 Seguridad
 - 4.1 Ataques de canal auxiliar
 - 4.2 Implementaciones
- 5 Notas
- 6 Referencias
- 7 Enlaces externos

Historia

En 1997, el Instituto Nacional de Normas y Tecnología (NIST) decidió realizar un concurso para escoger un nuevo algoritmo de cifrado capaz de proteger información sensible durante el siglo XXI. Este algoritmo se denominó **Advanced Encryption Standard** (AES).

El 2 de enero de 1997 el NIST anunció su intención de desarrollar AES, con la ayuda de la industria y de la comunidad criptográfica. El 12 de septiembre de ese año se hizo la convocatoria formal. En esta convocatoria se indicaban varias condiciones para los algoritmos que se presentaran:

- Ser de dominio público, disponible para todo el mundo.
- Ser un algoritmo de cifrado simétrico y soportar bloques de, como mínimo, 128 bits.

- Las claves de cifrado podrían ser de 128, 192 y 256 bits.
- Ser implementable tanto en hardware como en software.

El 20 de agosto de 1998 el NIST anunció los 15 algoritmos admitidos en la primera conferencia AES:

- **CAST-256** (*Entrust Technologies, Inc.*)
- **CRYPTON** (*Future Systems, Inc.*)
- **DEAL** (*Richard Outerbridge, Lars Knudsen*)
- **DFC** (*CNRS – Centre National pour la Recherche Scientifique – Ecole Normale Supérieure*)
- **E2** (*NTT – Nippon Telegraph and Telephone Corporation*)
- **FROG** (*TecApro International, S.A.*)
- **HPC** (*Rich Schroepel*)
- **LOKI97** (*Lawrie Brown, Josef Pieprzyk, Jennifer Seberry*)
- **MAGENTA** (*Deutsche Telekom AG*)
- **MARS** (*IBM*)
- **RC6** (*RSA Laboratories*)
- **RIJNDAEL** (*John Daemen, Vincent Rijmen*)
- **SAFER+** (*Cylink Corporation*)
- **SERPENT** (*Ross Anderson, Eli Biham, Lars Knudsen*)
- **TWOFISH** (*Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson*)

La segunda conferencia AES tuvo lugar en marzo de 1999 donde se discutieron los análisis a los que fueron sometidos los candidatos por la comunidad criptográfica internacional. Se admitieron comentarios hasta el 15 de abril. El NIST decidió en agosto de 1999 cuales serían los 5 finalistas:

- **MARS**
- **RC6**
- **RIJNDAEL**
- **SERPENT**
- **TWOFISH**

Estos algoritmos fueron sometidos a una segunda revisión, más exhaustiva, que duró hasta el 15 de mayo de 2000. Durante este periodo el NIST admitió análisis de los algoritmos finalistas.

Durante los días 13 y 14 de abril de 2000 tuvo lugar la tercera conferencia AES donde se discutieron los últimos análisis de los algoritmos finalistas. En ella estuvieron presentes los desarrolladores de los algoritmos finalistas.

El 15 de mayo de 2000 finalizó el periodo público de análisis. El NIST estudió toda la información disponible para decidir cual sería el algoritmo ganador. El 2 de octubre de 2000 se votó cual sería el algoritmo que finalmente ganaría el concurso. El resultado fue el siguiente:

- **MARS:** 13 votos
- **RC6:** 23 votos
- **RIJNDAEL:** 86 votos
- **SERPENT:** 59 votos
- **TWOFISH:** 31 votos

El algoritmo Rijndael ganó el concurso y en noviembre de 2001 se publicó FIPS 197 donde se asumía oficialmente.

Desarrollo

Rijndael fue un refinamiento de un diseño anterior de Daemen y Rijmen, Square; Square fue a su vez un desarrollo de Shark.

Al contrario que su predecesor DES, Rijndael es una red de sustitución-permutación, no una red de Feistel. AES es rápido tanto en software como en hardware, es relativamente fácil de implementar, y requiere poca memoria. Como nuevo estándar de cifrado, se está utilizando actualmente a gran escala.

Descripción del cifrado

Estrictamente hablando, AES no es precisamente Rijndael (aunque en la práctica se los llama de manera indistinta) ya que Rijndael permite un mayor rango de tamaño de bloques y longitud de claves; AES tiene un tamaño de bloque fijo de 128 bits y tamaños de llave de 128, 192 o 256 bits, mientras que Rijndael puede ser especificado por una clave que sea múltiplo de 32 bits, con un mínimo de 128 bits y un máximo de 256 bits.

La mayoría de los cálculos del algoritmo AES se hacen en un campo finito determinado.

AES opera en una matriz de 4×4 bytes, llamada *state* (algunas versiones de Rijndael con un tamaño de bloque mayor tienen columnas adicionales en el state).

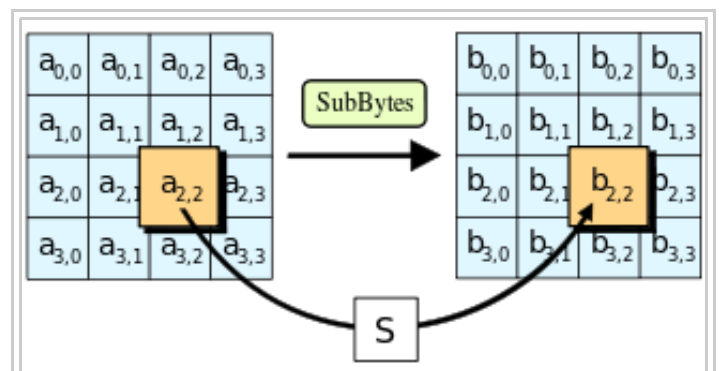
Pseudocódigo

- Expansión de la clave usando el esquema de claves de Rijndael.
- Etapa inicial:

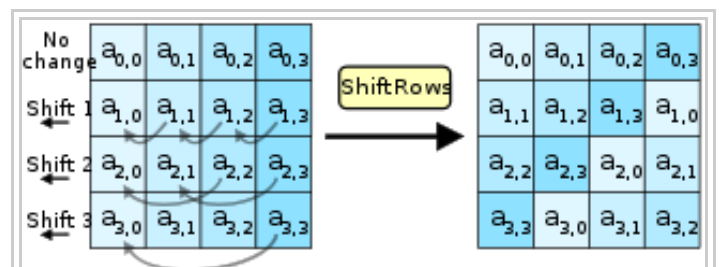
1. AddRoundKey

- Rondas:

1. SubBytes — en este paso se realiza una sustitución no lineal donde cada byte es reemplazado con otro de acuerdo a una tabla de búsqueda.
2. ShiftRows — en este paso se realiza una



En la fase de SubBytes, cada byte en el state es reemplazado con su entrada en una tabla de búsqueda fija de 8 bits, S ; $b_{ij} = S(a_{ij})$.



En el paso ShiftRows, los bytes en cada fila del state son rotados de manera cíclica hacia la izquierda. El número de lugares que cada byte es rotado difiere para cada fila.

transposición donde cada fila del «state» es rotada de manera cíclica un número determinado de veces.

3. **MixColumns** — operación de mezclado que opera en las columnas del «state», combinando los cuatro bytes en cada columna usando una transformación lineal.
4. **AddRoundKey** — cada byte del «state» es combinado con la clave «round»; cada clave «round» se deriva de la clave de cifrado usando una iteración de la clave.

■ **Etapla final:**

1. **SubBytes**
2. **ShiftRows**
3. **AddRoundKey**

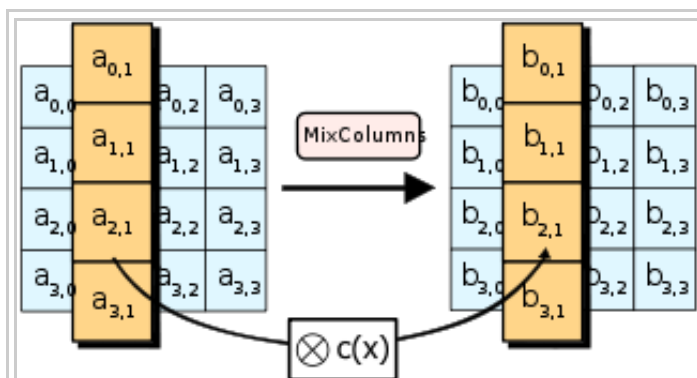
Etapla SubBytes- Substitución de bits

En la etapa **SubBytes**, cada byte en la matriz es actualizado usando la caja-S de Rijndael de 8 bits. Esta operación provee la no linealidad en el cifrado. La caja-S utilizada proviene de la función inversa alrededor del $\text{GF}(2^8)$, conocido por tener grandes propiedades de no linealidad. Para evitar ataques basados en simples propiedades algebraicas, la caja-S se construye por la combinación de la función inversa con una transformación afin inversible. La caja-S también se elige para evitar puntos estables (y es por lo tanto un derangement), y también cualesquiera puntos estables opuestos.

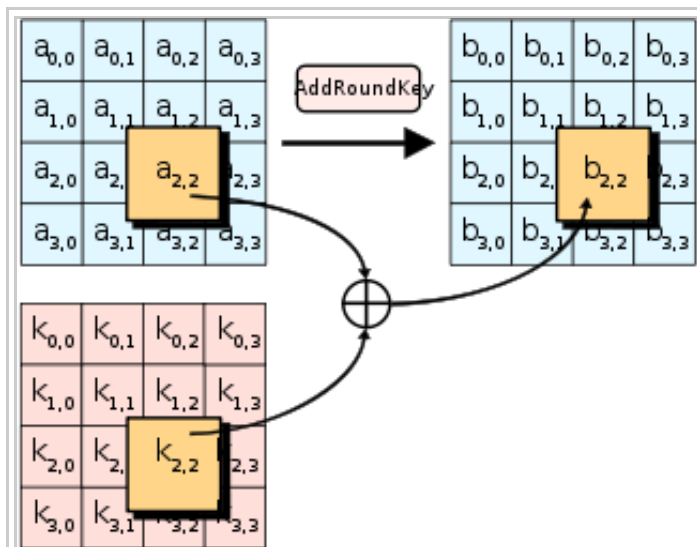
La caja-S es descrita en mayor profundidad en el artículo caja-S de Rijndael.

Etapla ShiftRows-Desplazar filas

El paso **ShiftRows** opera en las filas del state; rota de manera cíclica los bytes en cada fila por un determinado offset. En AES, la primera fila queda en la misma posición. Cada byte de la segunda fila es rotado una posición a la izquierda. De manera similar, la tercera y cuarta filas son rotadas por los offsets de dos y tres respectivamente. De esta manera, cada columna del state resultante del paso **ShiftRows** está compuesta por bytes de cada columna del state inicial. (variantes de Rijndael con mayor tamaño de bloque tienen offsets distintos).



En el paso **MixColumns**, cada columna del state es multiplicada por un polinomio constante $c(x)$.



En el paso **AddRoundKey**, cada byte del state se combina con un byte de la subclave usando la operación XOR (\oplus).

Etapla MixColumns- Mezclar columnas

En el paso `MixColumns`, los cuatro bytes de cada columna del state se combinan usando una transformación lineal inversible. La función `MixColumns` toma cuatro bytes como entrada y devuelve cuatro bytes, donde cada byte de entrada influye todas las salidas de cuatro bytes. Junto con `ShiftRows`, `MixColumns` implica difusión en el cifrado. Cada columna se trata como un polinomio $\text{GF}(2^8)$ y luego se multiplica el módulo $x^4 + 1$ con un polinomio fijo $c(x)$. El paso `MixColumns` puede verse como una multiplicación matricial en el campo finito de Rijndael.

Etapla AddRoundKey- Cálculo de las subclaves

En el paso `AddRoundKey`, la subclave se combina con el state. En cada ronda se obtiene una subclave de la clave principal, usando la iteración de la clave; cada subclave es del mismo tamaño que el state. La subclave se agrega combinando cada byte del state con el correspondiente byte de la subclave usando XOR.

Optimización del cifrado

En sistemas de 32 bits o de mayor tamaño de palabra, es posible acelerar la ejecución de este algoritmo mediante la conversión de las transformaciones `SubBytes`, `ShiftRows` y `MixColumn` en tablas. Se tienen cuatro tablas de 256 entradas de 32 bits que utilizan un total de 4 kilobytes (4096 bytes) de memoria, un Kb cada tabla. De esta manera, una ronda del algoritmo consiste en 16 búsquedas en una tabla seguida de 16 operaciones XOR de 32 bits en el paso `AddRoundKey`. Si el tamaño de 4 kilobytes de la tabla es demasiado grande para una plataforma determinada, la operación de búsqueda en la tabla se puede realizar mediante una sola tabla de 256 entradas de 32 bits mediante el uso de rotaciones circulares.

Seguridad

Hasta 2005, no se ha encontrado ningún ataque exitoso contra el AES. La Agencia de Seguridad Nacional de los Estados Unidos (NSA) revisó todos los finalistas candidatos al AES, incluyendo el Rijndael, y declaró que todos ellos eran suficientemente seguros para su empleo en información no clasificada del gobierno de los Estados Unidos. En junio de 2003, el gobierno de los Estados Unidos anunció que el AES podía ser usado para información clasificada:

*"The design and strength of all key lengths of the AES algorithm (i.e., 128, 192 and 256) are sufficient to protect classified information up to the SECRET level. TOP SECRET information will require use of either the 192 or 256 key lengths. The implementation of AES in products intended to protect national security systems and/or information must be reviewed and certified by NSA prior to their acquisition and use."*¹

Este hecho marca la primera vez que el público ha tenido acceso a un cifrador aprobado por la NSA para información super secreta (TOP SECRET). Es interesante notar que muchos productos públicos usan llaves de 128 bits por defecto; es posible que la NSA sospeche de una debilidad fundamental en llaves de este tamaño, *[cita requerida]* o simplemente prefieren tener un margen de seguridad para documentos super secretos (que deberían conservar la seguridad durante décadas en el futuro).

El método más común de ataque hacia un cifrador por bloques consiste en intentar varios ataques sobre versiones del cifrador con un número menor de rondas. El AES tiene 10 rondas para llaves de 128 bits, 12 rondas para llaves de 192 bits, y 14 rondas para llaves de 256 bits. Hasta 2005, los mejores ataques conocidos son sobre versiones reducidas a 7 rondas para llaves de 128 bits, 8 rondas para llaves de 192 bits, y 9 rondas para llaves de 256 bits (Ferguson et al, 2000).

Algunos criptógrafos muestran preocupación sobre la seguridad del AES. Ellos sienten que el margen entre el número de rondas especificado en el cifrador y los mejores ataques conocidos es muy pequeño. El riesgo es que se puede encontrar alguna manera de mejorar los ataques y de ser así, el cifrado podría ser roto. En el contexto criptográfico se considera "roto" un algoritmo si existe algún ataque más rápido que una búsqueda exhaustiva (ataque por fuerza bruta). De modo que un ataque contra el AES de llave de 128 bits que requiera 'sólo' 2^{120} operaciones sería considerado como un ataque que "rompe" el AES aun tomando en cuenta que por ahora sería un ataque irrealizable. Hasta el momento, tales preocupaciones pueden ser ignoradas. El ataque de fuerza bruta más largamente publicitado y conocido ha sido contra una clave de 64 bits RC5 por distributed.net.

Otra preocupación es la estructura matemática de AES. A diferencia de la mayoría de cifradores de bloques, AES tiene una descripción matemática muy ordenada.^{2 3} Esto no ha llevado todavía a ningún ataque, pero algunos investigadores están preocupados que futuros ataques quizá encuentren una manera de explotar esta estructura.

En 2002, un ataque teórico, denominado "ataque XSL", fue anunciado por Nicolas Courtois y Josef Pieprzyk, mostrando una potencial debilidad en el algoritmo AES. Varios expertos criptográficos han encontrado problemas en las matemáticas que hay por debajo del ataque propuesto, sugiriendo que los autores quizá hayan cometido un error en sus estimaciones. Si esta línea de ataque puede ser tomada contra AES, es una cuestión todavía abierta. Hasta el momento, el *ataque XSL* contra AES parece especulativo; es improbable que nadie pudiera llevar a cabo en la práctica este ataque.

Ataques de canal auxiliar

Los ataques de canal auxiliar no atacan al cifrador en sí, sino a las implementaciones del cifrador en sistemas que revelan datos inadvertidamente.

En abril de 2005, Daniel J. Bernstein anunció un ataque temporizado de cache⁴ que solía romper un servidor a medida que usaba el cifrado AES para OpenSSL. Este servidor fue diseñado para dar la mayor cantidad de información acerca de los tiempos de ejecución como fuera posible, y el ataque requería cerca de 200 millones de ficheros de texto en claro. Se dice que el ataque no es práctico en implementaciones del mundo real;⁵ Bruce Schneier llamó a esta investigación un "*bonito ataque de tiempos*".⁶

En octubre de 2005, Adi Shamir y otros dos investigadores presentaron un artículo demostrando varios ataques de tiempos de cache⁷ contra AES. Uno de los ataques obtuvo una clave de AES entera después de tan sólo 800 escrituras, en 65 milisegundos. Este ataque requiere que el atacante pueda ejecutar programas en el mismo sistema que realiza el cifrado de AES.

Implementaciones

- Una calculadora de AES que muestra valores intermedios en Javascript⁸
- Implementación de AES por Brian Gladman con licencia BSD⁹
- Implementación de AES de dominio público de Pablo Barreto escrita en C

- Implementación de AES de dominio público de D.J. Bernstein¹⁰
- Código fuente con licencia GPL del algoritmo optimizado de Rijndael en C¹¹
- Biblioteca GPL Nettle que también incluye una implementación de AES¹²
- Evolsystem: ejemplo de algoritmo de cifrado AES - Rijndael¹³
- Rijndael Inspector: programa hecho en Flash para cifrar y descifrar utilizando AES-128¹⁴


Notas

- Tamaños de bloque de 128, 160, 192, 224 y 256 bits son soportados por el algoritmo Rijndael, pero sólo bloques de 128 bits de tamaño son especificados en el AES.

Referencias

- ↑ http://www.cnss.gov/Assets/pdf/cnssp_15_fs.pdf
 - ↑ «A simple algebraic representation of Rijndael (http://web.archive.org/web/http://www.macfergus.com/pub/rdalgeq.html)».
 - ↑ «Sean Murphy (http://www.isg.rhul.ac.uk/~sean)».
 - ↑ «D. J. Bernstein / Papers (http://cr.yp.to/papers.html#cachetiming)».
 - ↑ http://groups.google.com/groups?selm=42620794%40news.cadence.com
 - ↑ «Schneier on Security: AES Timing Attack (http://www.schneier.com/blog/archives/2005/05/aes_timing_atta_1.html)».
 - ↑ http://www.wisdom.weizmann.ac.il/~tromer/papers/cache.pdf
 - ↑ «JavaScript AES Example (http://people.eku.edu/styere/Encrypt/JS-AES.html)».
 - ↑ http://fp.gladman.plus.com/cryptography_technology/rijndael
 - ↑ «Poly1305-AES: a state-of-the-art message-authentication code (http://cr.yp.to/mac.html)».
 - ↑ http://www.cr0.net:8040/code/crypto/aes
 - ↑ «Nettle - a low-level crypto library (http://www.lysator.liu.se/~nisse/nettle)».
 - ↑ «Evolsystem AES Rijndael (Technology Preview) (http://web.archive.org/web/http://www.evolsystem.cl/algoritmo/)».
 - ↑ «Rijndael Inspector (http://www.formaestudio.com/rijndaelinspector/)».
- Nicolas Courtois, Josef Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations". pp267–287, ASIACRYPT 2002.
 - Joan Daemen and Vincent Rijmen, "The Design of Rijndael: AES - The Advanced Encryption Standard." Springer-Verlag, 2002. ISBN 3-540-42580-2.
 - Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Michael Stay, David Wagner and Doug Whiting: Improved Cryptanalysis of Rijndael. FSE 2000, pp213–230

Enlaces externos

-  Wikilibros alberga un libro o manual sobre **Seguridad informática**.
- código de referencia (http://embeddedsw.net/Cipher_Reference_Home.html)
- Literature survey on AES (http://www.iaik.tu-graz.ac.at/research/krypto/AES/)
- The archive of the old official AES website (http://csrc.nist.gov/encryption/aes/)
- FIPS PUB 197: the official AES Standard (http://www.csrc.nist.gov/publications/fips/fips197/fips-197.pdf) (PDF file)
- John Savard's description of the AES algorithm (http://www.quadibloc.com/crypto/co040401.htm)

- Animación explicando el funcionamiento del algoritmo Rijndael - por E. Zabala (<http://www.formaestudio.com/rijndaelinspector/>)
- ¿Fue una buena idea usar AES256 con el archivo INSURANCE de Wikileaks? (<http://fernando-acero.livejournal.com/78069.html>) Por Fernando Acero Martín, experto en criptografía, del Ejército del Aire Español.

Obtenido de «http://es.wikipedia.org/w/index.php?title=Advanced_Encryption_Standard&oldid=68979237»

Categoría: Cifrado por bloques

- Esta página fue modificada por última vez el 13 ago 2013, a las 04:43.
- El texto está disponible bajo la Licencia Creative Commons Atribución Compartir Igual 3.0; podrían ser aplicables cláusulas adicionales. Léanse los términos de uso para más información.
Wikipedia® es una marca registrada de la Fundación Wikimedia, Inc., una organización sin ánimo de lucro.