Tìm hiểu về Advanced Encryption Standard

I. Khái niệm

Trong mật mã học, Advance Encryption Standard (AES) là một thuật toán mã hóa khối. Giống như DES, AES được áp dụng trên phạm vi toàn thế giới và đã được nghiên cứu rất kỹ lưỡng. Thuật toán được thiết kế bởi hai nhà mật mã học người Bỉ :Joan Daeman và Vincent Rijmen. Thuật toán được đặt tên là “Rijndael” khi tham gia cuộc thi thiết kế AES.

II. Phát triển

Thuật toán được dựa trên bản thiết kế Square có trước đó của Daeman và Rijmen; còn Square lại được thiết kế dựa trên Shark.

Khác với DES sử dụng mạng Feistel, Rijndael sử dụng mạng thay thế hoán vị. AES có thể dễ dàng thực hiện với tốc độ cao bằng phần mềm hoặc phần cứng và không đòi hỏi nhiều bộ nhớ. Do AES là một tiêu chuẩn mã hóa mới, nó đang được sử dụng đại trà.

III. Mô tả

\* Đặc điểm kỹ thuật

AES là một thuật toán mã hóa khối đối xứng với độ dài khóa là 128 bít (một chữ số nhị phân có giá trị 0 hoặc 1), 192 bít và 256 bít tương ứng gọi là AES-128, AES-192 và AES-256. AES-128 sử dụng 10 vòng (round), AES-192 sử dụng 12 vòng và AES-256 sử dụng 14 vòng.

Vòng lặp chính của AES thực hiện các hàm sau: SubBytes(), ShiftRows(), MixColumns() và AddRoundKey(). Ba hàm đầu của một vòng AES được thiết kế để ngăn chặn phân tích mã bằng phương thức “mập mờ“ (confusion) và phương thức “khuếch tán“ (diffusion), còn hàm thứ tư mới thực sự được thiết kế để mã hóa dữ liệu. Trong đó “khuếch tán“ có nghĩa là các kiểu mẫu trong bản rõ (Dữ liệu đầu vào của phép mã hóa hoặc dữ liệu đầu ra của phép giải mã) được phân tán trong các bản mã (Dữ liệu đầu ra của phép mã hóa hoặc dữ liệu đầu vào của phép giải mã), “mập mờ“ nghĩa là mối quan hệ giữa bản rõ và bản mã bị che khuất. Một cách đơn giản hơn để xem thứ tự hàm AES là: Trộn từng byte (SubBytes), trộn từng hàng (ShiftRows), trộn từng cột (MixColumns) và mã hóa (AddRoundKey).

\* Đặc tả thuật toán

Đối với thuật toán AES, độ dài của khối đầu vào, khối đầu ra và trạng thái là 128 bít, số các cột (các từ có độ dài 32 bít) tạo nên trạng thái là Nb = 4.

Trong thuật toán AES, độ dài khóa mã K có thể là 128, 192 hay 256 bít. Độ dài khóa được biểu diễn bằng Nk = 4, 6 hoặc 8 thể hiện số lượng các từ 32 bít (số cột) của khóa mã.

Đối với thuật toán AES, số vòng được thay đổi trong quá trình thực hiện thuật toán phụ thuộc vào kích cỡ khóa. Số vòng này được ký hiệu là Nr. Nr = 10 khi Nk = 4, Nr = 12 khi Nk = 6 và Nr = 14 khi Nk = 8.

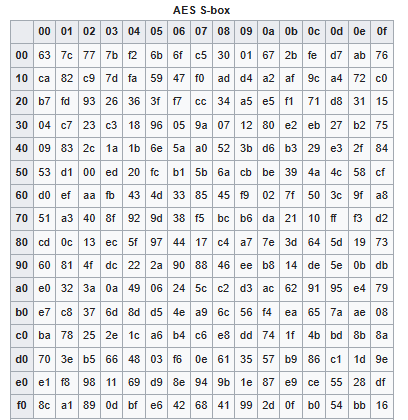
|  |  |  |  |
| --- | --- | --- | --- |
|  | Độ dài khóa (Nk từ) | Độ dài khối (Nb từ) | Số vòng (Nr) |
| AES-128 | 4 | 4 | 10 |
| AES-196 | 6 | 4 | 12 |
| AES-256 | 8 | 4 | 14 |

Đối với phép mã hóa và phép giải mã, thuật toán AES sử dụng một hàm vòng gồm bốn phép biến đổi byte như sau: phép thay thế byte (một nhóm gồm 8 bít) sử dụng một bảng thay thế (S-Box), phép dịch chuyển hàng của mảng trạng thái theo các offset (số lượng byte) khác nhau, phép trộn dữ liệu trong mỗi cột của mảng trạng thái, phép cộng khóa vòng và trạng thái.

- Quá trình mã hóa

+ Bước SubBytes

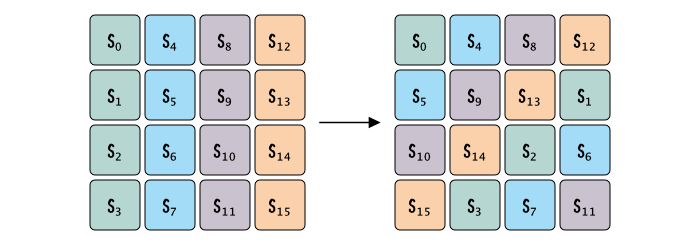
Ở bước này các byte sẽ được thế thông qua bảng tra S-box. Bước này chính là bước làm nên sự phi tuyến tính của thuật toán. Hộp S-box được tạo ra từ một phép biến đổi khả nghịch trong trường hữu hạn GF(28)có tính chất phi tuyến. Để chống lại các cuộc tấn công dựa trên các đặc tính đại số, hộp S-Box này được tạo nên bằng cách liên kết hợp phép nghịch đảo với một phép biến đổi affine khả nghịch. Hộp S-Box cũng được chọn để tránh các điểm bất động (fixed point).



Ví dụ về một S-box có kích thước 16x16. Với các khối dữ liệu đầu vào 8 bit (4 bit mỗi ký tự) cũng như các khối dữ liệu đầu ra cũng sẽ là 8 bit. Giả sử dữ liệu đầu vào 8 bit của chúng ta là 69 ( tương ứng với hàng 6 cột 9). Từ đó chúng ta có thể rút ra được dữ liệu đầu ra 8 bit sau khi đi qua S-box sẽ là f9.

+ Bước ShiftRow

Đúng như cái tên của nó: ShiftRow. Ở bước này các hàng được dịch theo một số bước nhất định. Trong AES, hàng đầu sẽ được giữ nguyên. Mỗi byte của hàng thứ 2 được dịch sang trái một vị trí, hàng thứ 3 sẽ được dịch 2 vòng và hàng thứ 4 sẽ được dịch 3 vòng. Vì vậy các byte ở mỗi cột khối đầu ra sẽ bao gồm đầy đủ các byte ở 4 cột khối đầu vào. Với các trường hợp độ dài khối lớn hơn, số vị trí dịch chuyển cũng sẽ lớn hơn.



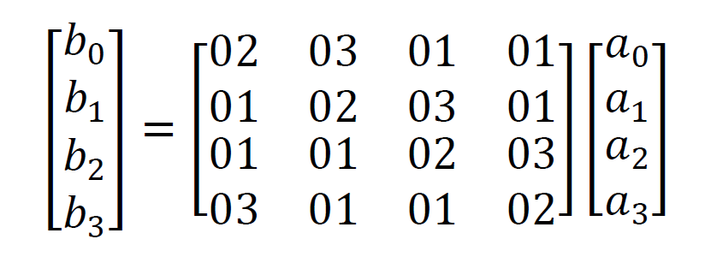
Ví dụ về cách bước ShiftRow hoạt động. Ban đầu chúng ta sẽ có một khối có kích thước 4 x 4. Áp dụng lý thuyết ở trên thì với hàng đầu tiên ( chứa các byte được đánh dấu là S0, S4, S8 và S12 ) sẽ được giữ nguyên. Ở hàng thứ hai các khối sẽ phải dịch sang trái một vị trí ( S5 thế chỗ S1, S9 thế chỗ S5, S13 thế chỗ S9 và S1 thế chỗ S13). Tương tự với hàng thứ ba các khối sẽ dịch sang trái 2 vị trí (S2 thế chỗ S10 và ngược lại ) và ở hàng thứ 4 sẽ là 3 vị trí ( S15 thế chỗ S3 ).

- Bước MixColumns

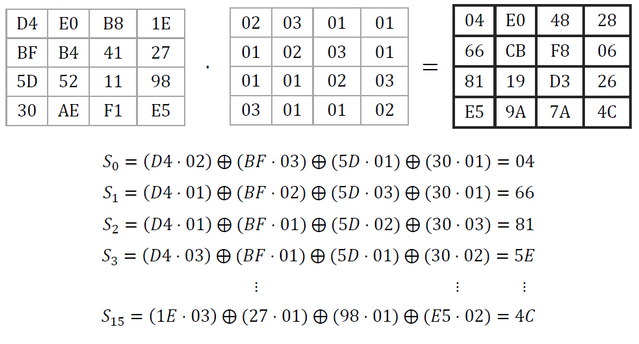
Bốn byte trong từng cột được kết hợp lại theo một phép biến đổi tuyến tính khả nghịch. Mỗi khối 4 byte đầu vào sẽ cho một khối 4 byte ở đầu ra với tính chất mỗi byte ở đầu vào đều ảnh hưởng tới cả 4 byte đầu ra. Cùng với bước ShiftRow, MixColumns đã tạo ra tính chất khuếch tán cho thuật toán. Mỗi cột được xem như một đa thức trong trường hữu hạn và được nhân với đa thức:

c(x) = 3x3 + x2 + x + 2 (modulo x4 + 1)

Vì thế, bước này có thể được xem là một phép nhân ma trận trong trường hữu hạn.



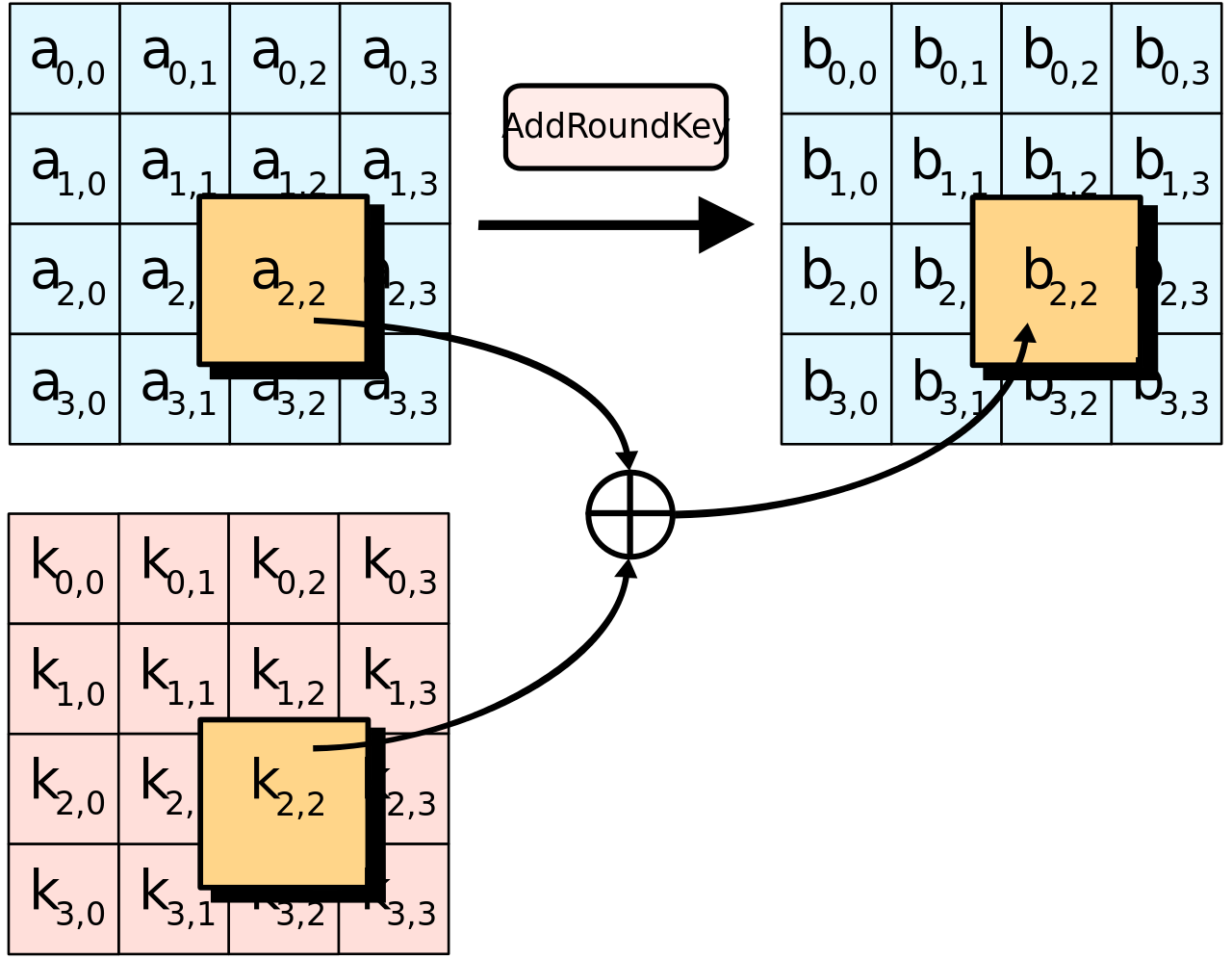
Ví dụ thể hiện phương thức hoạt động tổng quát của bước MixColumns. Ma trận ở chính giữa hình là hình thức diễn giải cụ thể của trường GF(28) dưới dạng ma trận.



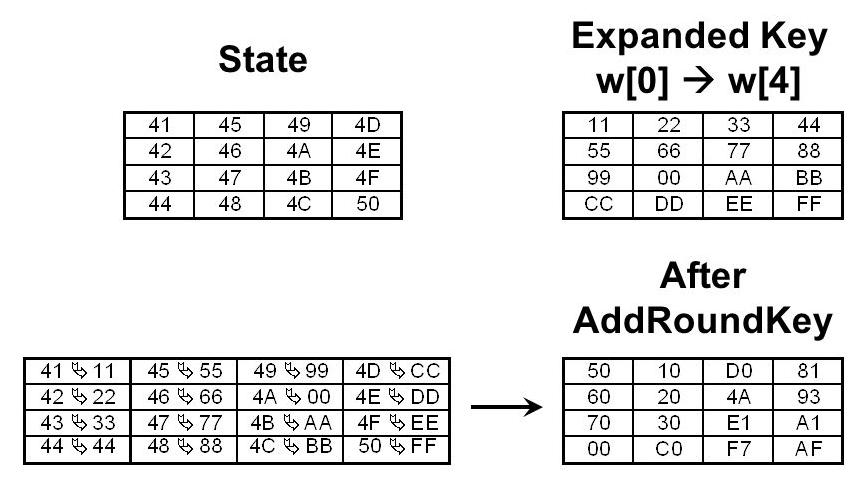
Ví dụ chi tiết về phương thức hoạt động của bước MixColumns. Bằng cách thực hiện phép nhân ma trận bình thường, mà ở trong ví dụ này là lấy cột một của ma trận ngoài cùng bên trái ( gồm D4, BF, 5D, 30) nhân theo đúng thứ tự với hàng một của ma trận biến đổi ở chính giữa ( gồm 02, 03, 01, 01) để cho ra giá trị của S’1,1 ở phía ma trận ngoài cùng bên phải là 04. Ví dụ này thể hiện cho chúng ta thấy tính chất khuếch tán của bước MixColumns khi mà tính chất 4 byte đầu vào ( cột một ma trận ngoài cùng ) đều sẽ ảnh hưởng tới việc cho ra giá trị của S’1,1 ở phía ma trận ngoài cùng bên phải.

- Bước AddRoundKey

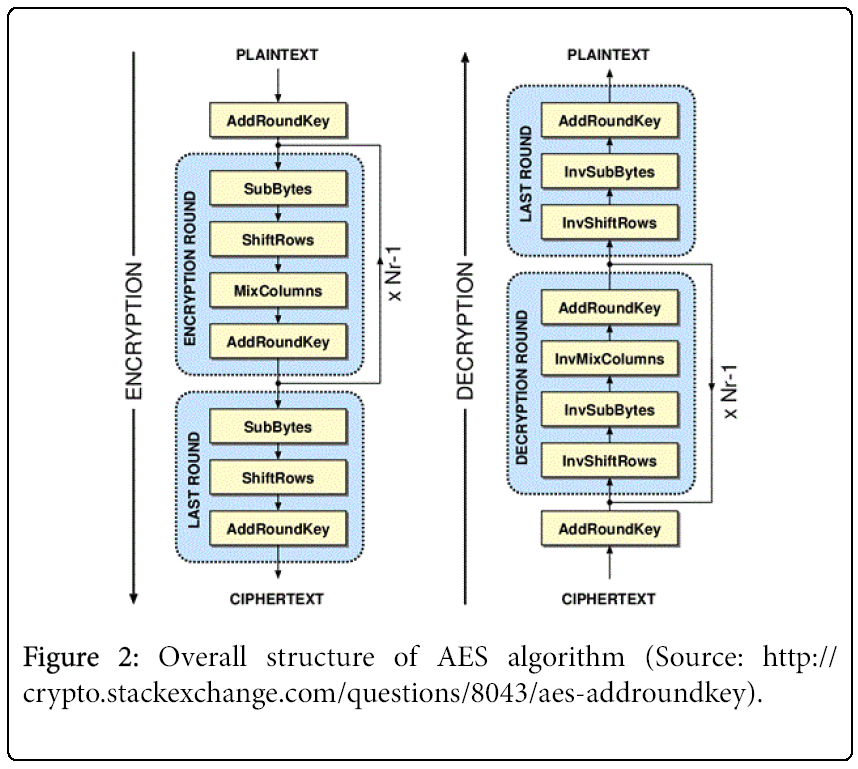
Tại bước này, các khóa con sẽ được kết hợp với các khối tương ứng. Khóa con trong mỗi chu trình được tạo ra từ khóa chính với quá trình tạo kháo con Rijndael. Mỗi khóa con đều có độ dài tương tự các khối. Quá trình kết hợp được thực hiện bằng cách XOR từng bit của khóa con với khối dữ liệu.



Ví dụ tổng quát về phương thức thực hiện của bước AddRoundKey.



Ví dụ chi tiết về phương thức thực hiện của bước AddRoundKey. Ở đây chúng ta có một ma trận các khối chứa dữ liệu 1 byte ở góc trên bên trái và một ma trận khoác con có kích thước tương tự ở góc trên bên phải. Ta lấy trường hợp của khối S4,4 có giá trị 50, sau khi chuyển giá trị từ Hex về Binary thì giá trị sẽ trở thành 1010000. Tương tự với khối K4,4 thì giá trị sau khi được chuyển từ Hex về Binary sẽ là 11111111. XOR hai giá trị này lại với nhau và chúng ta sẽ có một giá trị nhị phân mới là 10101111 tương ứng với AF ở dạng Hex ( kết quả ở khối S’4,4 ở góc dưới bên phải hình )



Sơ đồ hoạt động của toàn bộ quá trình mã hóa AES. Như có thể thấy ở trên hình, các bước được liệt kê ở phía trên được lặp lại ở mỗi vòng mã hóa ( mỗi Round ). Bên trái chính là quá trình encrypt và bên phải chính là quá trình decrypt. Ở quá trình encrypt ta có thể thấy một bước AddRoundKey được khởi động ngay trước khi các vòng mã hóa được bắt đầu. Chính ma trận key được sử dụng ở trong bước này sẽ là nền tảng để sinh ra ma trận key mới được sử dụng cho các vòng mã hóa tiếp theo sẽ xảy ra. Quá trình này được gọi là KeyExpansion.

IV. Sự khác biệt giữa AES và RSA

AES được sử dụng để bảo vệ các dữ liệu ở trạng thái nghỉ (lưu trữ trong kho). Ứng dụng của AES bao gồm tự mã hóa ổ đĩa, mã hóa lưu trữ, mã hóa cơ sở dữ liệu... Trong khi đó, RSA được sử dụng chủ yếu để kết nối các website trong trình duyệt và trong nhiều lĩnh vực khác.

Ngoài ra, trong khi AES sử dụng private key thì RSA lại sử dụng public key. Hiệu suất hoạt động của RSA cũng chậm hơn so với AES. Chính vì vậy, giải pháp tốt nhất để bảo mật việc truyền dữ liệu từ xa là kết hợp cả mã hóa RSA và AES.

V. Khả năng bảo mật

Vào thời điểm năm 2006, dạng tấn công lên AES duy nhất thành công là tấn công kênh bên ( Side Channel Attack ). Vào tháng 6 năm 2003, chính phủ Hoa Kỳ tuyên bố AES có thể được sử dụng cho các thông tin mật.

“ Thiết kế và độ dài khóa của thuật toán AES (128,192 và 256 bit) là đủ an toàn để bảo vệ các thông tin được xếp vào loại TỐI MẬT (secret). Các thông tin TUYỆT MẬT (top secret) sẽ phải dùng khóa 192 bit hoặc 256 bit. Các phiên bản thực hiện AES nhằm mục đích bảo vệ hệ thống an ninh hay thông tin quốc gia phải được NSA kiểm tra và chứng nhận trước khi sử dụng.” – ([Wayback Machine (archive.org)](https://web.archive.org/web/20070927035010/http:/www.cnss.gov/Assets/pdf/cnssp_15_fs.pdf))

Điều này đánh dấu lần đầu tiên công chúng có quyền tiếp xúc với một thuật toán mật mã mà NSA phê chuẩn cho thông tin TUYỆT MẬT. Nhiều phần mềm thương mại hiện nay đang sử dụng khóa mặc định có độ dài 128 bit.

Phương pháp thường dùng nhất để tấn công các dạng mã khóa khối như AES là thử các kiểu tấn công có chu trình thu gọn. Đối với khóa 128 bit, 192 bit và 256 bit chúng ta có các chu trình tương ứng là 10, 12 và 14. Tại thời điểm năm 2006, những lần tấn công thành công nhất lần lượt là 7 chu trình với 128 bit, 8 chu trình với 192 bit và 9 chu trình với 256 bit.

Một số nhà khoa học trong lĩnh vực mật mã cũng bày tỏ quan ngại về vấn đề an ninh của AES. Họ cho rằng ranh giới giữa số chu trình của thuật toán và số chu trình bị phá vỡ quá nhỏ. Nếu các kỹ thuật tấn công được cải thiện thì AES có thể sẽ bị phá vỡ (bằng bất cứ phương pháp nào nhanh hơn tấn công kiểu duyệt toàn bộ). Vì thế một cuộc tấn công cần thực hiện 2120 cũng được coi là thành công mặc dù tấn công này chưa thể thực hiện trong thực tế. Tại thời điểm hiện nay, nguy cơ này không thực sự nguy hiểm và có thể bỏ qua. Cuộc tấn công theo kiểu duyệt toàn bộ quy mô nhất từng được thực hiện là do distributed.net thực hiện lên hệ thống 64 bit RC5 vào năm 2002 (Theo định luật Moore thì nó tương đương với việc tấn công vào hệ thống 66 bit hiện nay).

Một vấn đề khác nữa là cấu trúc toán học của AES khá đơn giản so với các thuật toán mã hóa khác. Tuy điều này chưa dẫn đến mối nguy hiểm nào nhưng một số nhà nghiên cứu sợ rằng có người sẽ lợi dụng được cấu trúc này trong tương lai.

Vào năm 2002, Nicolas Courtois và Josef Pieprzyk đã phát hiện được một cách tấn công trên lý thuyết là XSL và chỉ ra được điểm yếu tiềm tàng của AES. Tuy nhiên một vài chuyên gia về mật mã học khác cũng chỉ ra một số vấn đề trong cơ sở toán học của phương thức tấn công này và cho rằng tác giả đã có sai lầm trong tính toán. Việc tấn công dạng này có thực sự trở thành hiện thực hay không vẫn còn đang bỏ ngỏ và cho tới nay thì phương thức tấn công XSL vẫn chỉ là suy đoán.

- Về phương thức tấn công kênh bên (Side channel attack)

Side channel attack không tấn công trực tiếp vào thuật toán mã hóa mà thay vào đó, tấn công lên các hệ thống thực hiện thuật toán có cơ sở hàm làm lộ dữ liệu.

Tháng 4 năm 2005, Daniel J.Berstein công bố một phương thức tấn công lên hệ thống mã hóa AES trong OpenSL gọi là Cache Timing Attack. Một máy chủ được thiết kế để đưa ra tối đa thông tin về thời gian có thể thu được và cuộc tấn công cần tới 200 triệu bản rõ lựa chọn. Một số người cho rằng phương thức tấn công này không thể thực hiện được trên Internet với khoảng cách vài điểm mạng.

Tháng 10 năm 2005, Adi Shamir và 2 nhà nghiên cứu khác có một bài nghiên cứu minh họa một vài phương thức khác. Trong đó có một phương thức tấn công có thể lấy được khóa AES với 800 lần ghi trong 65ms. Phương thức tấn công này yêu cầu kẻ tấn công có khả năng chạy chương trình trên chính hệ thống thực hiện việc mã hóa.

VI. Ứng dụng

Thuật toán AES cho phép thực hiện hiệu quả bằng cả phần mềm và phần cứng. Thông thường với những ứng dụng không yêu cầu cao về hiệu năng và tốc độ thì AES được thực hiện ở dạng phần mềm. Với việc thực hiện trên phần mềm, thuật toán AES có thể được viết bằng nhiều ngôn ngữ lập trình như Assembler, C/C++, Visual Basic, Java, C#... và có thể vận hành trên nhiều hệ điều hành như Windows, Linux/Unix, Solaris.... Khi thực hiện trên phần cứng, thuật toán AES hỗ trợ thực hiện hai dòng: dòng thiết bị thứ nhất dựa vào một hệ vi xử lý phụ kết hợp với hệ vi xử lý chính của máy tính, dòng thiết bị thứ hai thường được thiết kế ở dạng thẻ thông minh (smart card) hoặc các thiết bị cắm qua cổng USB (Universal Serial Bus). Trong Thông tư số 01/2011/TT-BTTTT ngày 04/01/2011 của Bộ Thông tin và Truyền thông Công bố Danh mục tiêu chuẩn kỹ thuật về ứng dụng công nghệ thông tin trong cơ quan nhà nước quy định Khuyến nghị áp dụng tiêu chuẩn AES và được xếp vào nhóm Tiêu chuẩn về an toàn thông tin.

Bên cạnh ứng dụng trong việc đảm đảm an toàn cho các tài liệu chính phủ, AES cũng được áp dụng nhiều trong các lĩnh vực khác nhau. Cụ thể:

- Ứng dụng cho tất cả người dùng phổ thông trong quá trình giải mã dữ liệu bằng cách truy cập vào trang web AES Encryption, nhập dữ liệu và áp mã khoá. Tuy nhiên, đây chỉ là phương pháp áp dụng cho các tác vụ thông thường và tính bảo mật thường không cao.

- Mã hoá thông tin trên phần mềm với các ngôn ngữ lập trình như C/C++, [Java](https://bizflycloud.vn/tin-tuc/tu-hoc-java-tong-quan-va-huong-dan-cai-dat-java-jdk-16-moi-2021-20210427172617043.htm) hay Assembler. AES hỗ trợ rất nhiều cho các hệ điều hành như Linux hay Windows.

- AES áp dụng cho các thiết bị phần cứng bao gồm dòng thiết bị dựa trên hoạt động của hệ vi xử lý và dòng thiết bị cắm qua cổng USB hoặc thẻ thông minh Smart Card.

- Ứng dụng trong truyền thông tin qua Internet thông qua kết nối HTTPS: Dữ liệu sẽ được mã hoá và giải mã thông qua thuật toán AES, thông tin được bảo mật tốt hơn khi so sánh với kết nối HTTP. Bên cạnh đó, wifi hiện nay cũng được ứng dụng thuật toán AES, khi kết hợp với giao thức WPA2, giao tiếp này trở nên an toàn, hiệu quả hơn nhiều và ngăn chặn tấn công trung gian. Bên cạnh đó, AES cũng được sử dụng để mã hoá wifi trên router, kết hợp với giao thức phổ biến WPA2 được gọi là AES/WPA2. AES còn được sử dụng nhằm hỗ trợ mã hoá SSL.