I. Mô tả thuật toán

1. Giới thiệu chung

AES - Advanced Encryption Standard là một chuẩn mã hóa dữ liệu tiên tiến, với bản chất là thuật toán mã hóa khối đối xứng đã được chính phủ Mỹ Hoa Kỳ áp dụng làm tiêu chuẩn mã hóa.

Mã hóa khối đối xứng:

* Khối: Thuật toán áp dụng xử lý đơn vị dữ liệu là các khối (data blocks). Cụ thể, các khối dữ liệu được xử lý có độ dài 128 bits, con số này là cố định.
* Đối xứng: Cả bên gửi và bên nhận sử dụng chung một khóa cho cả 2 giai đoạn mã hóa và giải mã.

2. Các ký hiệu và định nghĩa

2.1. Đầu vào và đầu ra của thuật toán

Là các đoạn dữ liệu độ dài 128 bits, được nhắc đến như các block data. Những block này được mã hóa / giải mã bằng khóa có độ dài 128/192/256 bit, ứng với các chuẩn AES128/192/256. Tuy nhiên block size luôn luôn là cố định.

2.2. Các Byte

Đơn vị xử lý cơ bản của thuật AES là các byte. Đầu vào, ra và khóa của thuật toán được xử lý như những chuỗi các byte.

Mỗi byte này sẽ được biểu diễn như sau: {b7, b6, b5, b4, b3, b2, b1, b0}. Chúng được hiểu như những phần tử hữu hạn trong trường Galois (Finite Field), biểu diễn bởi đa thức:

b7x7 + b6x6 + b5x5 + b4x4 + b3x3 + b2x2+ b1x + b0 (2.2.1)

Ex: {00101011} ứng với đa thức: x5 + x3 + x + 1

Ngoài ra, byte còn có thể được biểu diễn dưới dạng hexa. Ví dụ: {01100011} ứng với phần tử {63}.

2.3. Các trạng thái

Các giai đoạn xử lý bên trong của thuật toán AES được thực hiện trên mảng 2 chiều các byte, được gọi là State (Trạng thái). Các mảng này có kích thước 4x4, được tạo nên như sau:

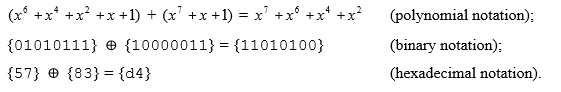
A close up of a clock

Description generated with high confidence

3. Các phép toán liên quan

3.1. Phép cộng

Phép cộng các phần tử trong trường Galois được thực hiện bởi việc “cộng” các hệ số tương ứng của đa thức, cụ thể là XOR các bit tương ứng với nhau. Phép trừ cũng được thực hiện tương tự. Ví dụ như sau:



3.2. Phép nhân

Phép nhân trong trường Galois được thực hiện bởi việc nhân đa thức thông thường, theo sau bởi phép chia lấy dư (modulo) cho đa thức không giảm bậc 8. Trong khuôn khổ thuật AES, đa thức này là:



Hoặc {01} {1b} trong khuôn khổ biểu diễn hexadecimal.

Ví dụ, {57} • {83} = {c1} vì:

A screenshot of a cell phone

Description generated with very high confidence

Và lại có

A close up of a logo

Description generated with very high confidence

Việc chia cho đa thức m(x) đảm bảo bậc của kết quả sẽ luôn nhỏ hơn 8, từ đó có thể biểu diễn thành byte.

3.3. Nhân với x

Đa thức 2.2.1 sau khi nhân với x ta sẽ thu được kết quả

A picture containing furniture

Description generated with high confidence

Dễ thấy kết quả này có bậc lớn hơn 8, ta cần phải giản ước nó. Nếu b7 = 0 thì kết quả đã ở dạng rút gọn. Nhưng nếu b7 = 1 thì giản ước bằng cách trừ đi đa thức m(x).

Tóm lại việc nhân đa thức với x (vì x vốn là bậc 1, nên tương ứng với {00000010} hoặc {02}) có thể thực hiện bằng phép dịch trái 1 bit, tùy điều kiện mà triển khai thêm phép XOR với {1b} (đa thức m(x)). Công đoạn này được gọi là xtime().

Từ cơ sở này, phép nhân với bậc mũ bất kì có thể được cài đặt bằng việc lặp lại các xtime() và thêm vào các kết quả trung gian. Ví dụ như {57} • {13} = {fe} vì:

{57} • {02} = xtime({57}) = {ae}

{57} • {04} = xtime({ae}) = {47}

{57} • {08} = xtime({47}) = {8e}

{57} • {10} = xtime({8e}) = {07}

Do đó:

A screenshot of a cell phone

Description generated with very high confidence

Cụ thể, ta tách thừa số thứ 2 thành các bội số của 2, nhân với thừa số thứ nhất được các kết quả trung gian rồi cộng (XOR) với nhau. Bằng cách này phép nhân với số bất kì có thể được cài đặt một cách dễ dàng.

4. Đặc tả thuật toán

Bảng đặc tả các tham số của AES:

A screenshot of a cell phone

Description generated with very high confidence

Số liệu trong bảng ứng với 3 chuẩn AES - 128, 192 và 256. Cả 3 chuẩn đều có chung kích thước data block xử lý (Nb), chỉ khác ở Nk và Nr. Thuật AES tiến hành theo các vòng, số lượng vòng tùy thuộc vào chuẩn tương ứng. Hàm vòng được tạo nên bởi các chuyển đổi byte: Byte substitution, Shift Rows, Mix Columns và Add Round Keys. Các phép này đều tác động lên đối tượng là từng byte của State.

4.1. Quá trình mã hóa (Cipher)

Đầu quá trình mã hóa, dữ liệu đầu vào được copy vào mảng trạng thái theo mô tả trong sec 2.3. Quá trình mã hóa trải qua Nr bước tất cả, với Nr phụ thuộc vào từng chuẩn AES. Trong đó, vòng đầu và vòng cuối khác với những vòng còn lại.

Pseudo Code của The Cipher như sau:

A screenshot of a cell phone

Description generated with very high confidence

2 tham số đầu tiên của Cipher là mảng trạng thái đầu vào và đầu ra của quá trình mã hóa. Tham số cuối cùng là mảng các Roundkey w[] với tổng số Nb \* (Nr + 1) phần tử được tạo ra từ Key schedule, sẽ được mô tả sau. Các function được dùng trong Cipher là SubBytes(), ShiftRows(), MixColumns(), AddRoundKey() đều tác động lên đối tượng đó là state.

4.1.1. SubBytes() function

Chuyển hóa SubBytes() là một phép thay thế thực hiện trên từng byte của state. Mỗi byte này sẽ được thay thế bằng 1 byte khác, phụ thuộc vào chính giá trị của byte được thay thế đó. Byte thay thế được lấy ra từ một bảng được gọi là Sbox.

A screenshot of a cell phone

Description generated with high confidence

Thông qua SubBytes(), mỗi byte của state sẽ được thay thế bằng 1 byte lấy từ Sbox này. Quá trình được mô tả trực quan như sau:



Cách thay thế: Nếu byte cần được thay thế của state có giá trị hexa là {m n} thì sẽ được thay thế với giá trị của phần tử nằm ở hàng m cột n. Ví dụ: Nếu s0,1 có giá trị {c9} thì s’0,1 sẽ là giá trị ở hàng c cột 9: {dd}.

4.1.2. ShiftRows() Transformation

Chuyển hóa của ShiftRows chỉ là xoay vòng các byte tương ứng trong cùng một hàng của state, được mô tả như sau:



Chỉ có các byte của dòng đầu tiên là được giữ nguyên vị trí.

4.1.3. MixColumns() Transformation

Chuyển hóa MixColumn áp dụng trên State theo từng cột, coi mỗi cột là các hệ số của một đa thức bậc 3. Đa thức này sẽ được nhân modulo x4 + 1 với một đa thức cố định a(x):



Kết quả này có thể được viết dưới dạng phép nhân ma trận. Giả sử:



Thì ta sẽ có:

A close up of a clock

Description generated with high confidenceVới 0 ≤ c < Nb

Kết quả của 4 byte hàng dọc sẽ là:

A screenshot of a cell phone

Description generated with very high confidence

Như ta đã biết phép nhân với một số bất kì trong trường Galois có thể thực hiện bằng việc thêm các kết quả trung gian và lặp lại các xtime(). Vậy function này cũng có thể cài đặt một cách dễ dàng.

Mô hình diễn tả biến đổi MixColumns():



4.1.4. AddRoundKey() Transformation

Trong biến đổi AddRoundKey(), một Round Key được cộng vào State bằng phép XOR đơn giản, trong đó mỗi RoundKey độ dài Nb word sinh ra từ Key Schedule (giới thiệu sau):



[wi] là round key, round là thứ tự vòng mã hóa có giá trị 0 ≤ round ≤ Nr.

Hình dưới diễn tả quá trình này, trong đó l = round \* Nb.

A close up of a clock

Description generated with high confidence

AddRoundKey() XOR mỗi cột của State với một word từ key schedule

4.2. Key Expansion

Thuật toán AES sử dụng khóa K (Khóa mã hóa) để thực hiện quá trình sinh khóa (Key Expansion). Quá trình này sinh ra tổng cộng Nb(Nr + 1) word: Thuật toán cần 4 word đầu, sau đó mỗi round trong tổng cộng Nr round lại cần Nb word dữ liệu khóa. Kết quả của quá trình này là một mảng các phần tử kích thước 4 byte, các [wi], với i nằm trong khoảng 0 ≤ i ≤ Nb(Nr + 1).