Course Instructor: Dr. Dibyendu Roy — Winter 2022-2023
Scribed by : Pallikonda Sai Teja — Lecture (Week 07)
Student ID : 202011052

1 Advanced Encryption Standard AES

1.1 Shift Rows

Shift Rows : $\{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$

$$\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{vmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{vmatrix} \rightarrow \begin{vmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{11} & S_{12} & S_{13} & S_{10} \\ S_{22} & S_{23} & S_{20} & S_{21} \\ S_{33} & S_{30} & S_{31} & S_{32} \end{vmatrix}$$

1.2 Mix Column

Mix Column : $\{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$

$(S_{ij})_{4x4} \rightarrow (S_{ij}{}^{l})_{4x4}$
Consider the column c $\in \{0, 1, 2, 3\}$

$$\text{column} = \begin{vmatrix} S_{0c} \\ S_{1c} \\ S_{2c} \\ S_{3c} \end{vmatrix}$$

$S_{ic} = (a_7a_6a_5a_4a_3a_2a_1a_0)$
Polynomial = $a_0 + a_1x + a_2x^2 + ... + a_7x^7$

for i= 0 to 3
    t_i = Binary to Polynomial(S_{ic})
$u_0 = [(x * t_0) + (x+1) * t_1 + t_2 + t_3] \bmod(x^8 + x^4 + x^3 + x + 1)$
$u_1 = [(x * t_1) + (x+1) * t_2 + t_3 + t_0] \bmod(x^8 + x^4 + x^3 + x + 1)$
$u_2 = [(x * t_2) + (x+1) * t_3 + t_0 + t_1] \bmod(x^8 + x^4 + x^3 + x + 1)$
$u_3 = [(x * t_3) + (x+1) * t_0 + t_1 + t_2] \bmod(x^8 + x^4 + x^3 + x + 1)$
$(S_{ij}{}^{l})$ = Polynomial to Binary(u_i)

$$\begin{vmatrix} S_{0c} \\ S_{1c} \\ S_{2c} \\ S_{3c} \end{vmatrix} \rightarrow \text{Mix Column} \rightarrow \begin{vmatrix} S_{0c}{}^{l} \\ S_{1c}{}^{l} \\ S_{2c}{}^{l} \\ S_{3c}{}^{l} \end{vmatrix}$$

$$\begin{vmatrix} x & x+1 & 1 & 1 \\ 1 & x & x+1 & 1 \\ 1 & 1 & x & x+1 \\ x+1 & 1 & 1 & x \end{vmatrix} * \begin{vmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{vmatrix} \bmod(x^8 + x^4 + x^3 + x + 1) = (S_{ij}{}^{l})_{4x4}$$

1.3 Key Scheduling Algorithm of AES-128

Input : 128 bit key.
Output : 11 Round Keys, each of length 128 bit.

Key = [Key[15], Key[14],.., Key[0]] $\rightarrow$ 16 bytes
We will prepare 44 words which are denoted by w[0], w[1], ..., w[43].

- *ROTWORD*$(B_0, B_1, B_2, B_3) = (B_1, B_2, B_3, B_0)$
- *SUBWORD*$(B_0, B_1, B_2, B_3) = (B_0{}^l, B_1{}^l, B_2{}^l, B_3{}^l)$

where $B_i{}^l$ = SUBBYTE(B_i) $\forall$ i $\in \{0, 1, 2, 3\}$

- 10 Round Constants (Word)

Rcon[1] = 01000000
Rcon[2] = 02000000
Rcon[3] = 04000000
Rcon[4] = 08000000
Rcon[5] = 10000000
Rcon[6] = 20000000
Rcon[7] = 40000000
Rcon[8] = 80000000
Rcon[9] = 1B000000
Rcon[10] = 36000000

```
for i = 0 to 3
    w[i] = (Key[4i], Key[4i+1], Key[4i+2], Key[4i+3])
for i = 4 to 43
    temp = w[i-1]
    if i ≡ 0 mod4
        temp = SUBWORD(ROTWORD(temp)) ⊕ Rcon[i/4]
    w[i] = w[i-4] ⊕ temp
return (w[0], w[1],..., w[43])
```

Key_1 = w[0] || w[1] || w[2] || w[3]
Key_2 = w[4] || w[5] || w[6] || w[7]
.
.
Key_{10} = w[40] || w[41] || w[42] || w[43]

1.4 Sub Bytes

Subbyte(A) = ?
A = X || Y. Here X is the row number and Y is the column number in Subbyte table. And from subByte table we can find SubByte(A).

1.5 Inverse Sub-Bytes

SubByte(A) = B
Find A given B.
So, Firstly find B inside the table by exhaustive search.
X $\rightarrow$ Row Number.
Y $\rightarrow$ Column Number.

Output = X || Y
A = X || Y

1.6 Inverse Shift Row

$$\begin{vmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{vmatrix} \rightarrow \begin{vmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{13} & S_{10} & S_{11} & S_{12} \\ S_{22} & S_{23} & S_{20} & S_{21} \\ S_{31} & S_{32} & S_{33} & S_{30} \end{vmatrix}$$

1.7 Mix Column

$\mathrm{Mixcolumn}(S_{ij}) = (S_{ij})_{4x4} = M * S$
$\mathrm{Mixcolumn}(\mathrm{Mixcolumn}(\mathrm{Mixcolumn}(\mathrm{Mixcolumn}(S)))) = M^4 * S = I * S$ $\boxed{M^4 = I}$

1.8 Inverse Mix Column

$\mathrm{Mixcolumn}(\mathrm{Mixcolumn}(\mathrm{Mixcolumn}(S^l))) = S$

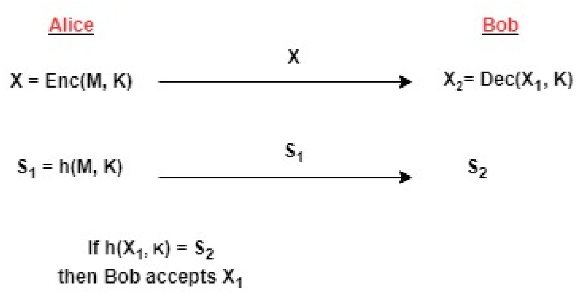
2 Modes of Operation

1. ECB (Electronic Code book Mode)
2. CFB (Cipher Feedback Mode)
3. CBC (Cipher Block Chaining Mode)
4. OFC (Output Feedback Mode)
5. Counter Mode
6. CCM (Counter with cipher block chaining mode)

3 Hash Function

$h(X) = Y$ $h : A \rightarrow B$

1. If X is altered to X^l then $h(X^l)$ will be completely different from $h(X)$.
2. Given Y it is practically infeasible to find X such that $h(X) = Y$.
3. Given X and $Y = h(X)$ it is practically infeasible to find X^l such that $h(X) = h(X^l)$



3.1 Definition

A Hash function is a four tuple (P, S, K, H) where the following conditions are satisfied

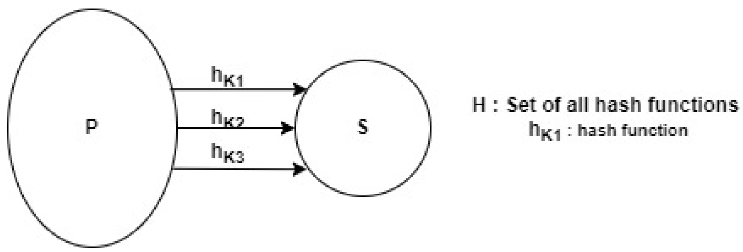
1. P is the set of all possible messages.

2. S is the set of all possible message digests or authentication tags.

3. K is the key space

4. For each $\mathrm{K_i} \in \mathrm{K}$ there is a hash function $\mathrm{h_{K_i}} \in \mathrm{H}$ such that
   $\mathrm{h_{K_i}} : \mathrm{P} \rightarrow \mathrm{S}$

Here $|\mathrm{P}| \geq |\mathrm{S}|$
More interestingly $|\mathrm{P}| \geq 2{*}|\mathrm{S}|$



• If key is involved in the computation of hashed valve then the hash function is known as keyed hash function.
• If key is not required to compute the hashed valve then the hash function is known as unkeyed hash function.

4 Problems

4.1 Problem 01

$\mathrm{h} : \mathrm{P} \rightarrow \mathrm{S}$
Given $\mathrm{y} \in \mathrm{S}$ Find $\mathrm{x} \in \mathrm{P}$
such that $\mathrm{h(x)} = \mathrm{y}$
This problem is known as pre-image finding problem.
For an hash function h if you cannot find pre-image in a feasible time then h is known as pre-image resistant hash function.
• Finding pre-image is computationally hard for pre-image resistant hash function.

4.2 Problem 02

$\mathrm{h} : \mathrm{P} \rightarrow \mathrm{S}$
Given $\mathrm{x} \in \mathrm{P}$ and h(x) find $\mathrm{x}^l \in \mathrm{P}$ such that $\mathrm{x}^l \neq \mathrm{x}$ and $\mathrm{h}(\mathrm{x}^l) = \mathrm{h}(\mathrm{x})$.
This problem is known as second pre-image finding problem.
If finding second pre-image is computationally hard for h then h is known as second pre-image resistant hash function.

4.3 Problem 03

h : P $\rightarrow$ S
Find x, $x^l \in$ P such that x $\neq x^l$ and h(x) = h(x^l)
This problem is known as collision finding problem.
If finding collision is computationally hard then h is known as collision resistant hash function.

Ideal Hash Function

Let h : P $\rightarrow$ S be an hash function.
h will be called ideal hash function if given x $\in$ P to find h(x) either you have to apply h on x or you have to look into the table corresponding to h (hash table).

5 Pre Image Finding Problem Algorithm

h : X $\rightarrow$ Y
|X| = n and |Y| = m
Y = $\{y_1, y_2, ..., y_m\}$
$X_0 \subseteq X$, $|X_0| = Q$.
    for each x $\in X_0$
        compute $y_x = h(x)$
        if $y_x = y$
        return x
E_i = event $h(x_i) = y$, $1 \leq i \leq Q$.
$Pr[E_i] = 1/M$
$Pr[E_i{}^c] = 1 - 1/M$
$Pr[E_1 \cup E_2 \cup ... \cup E_Q]$
$= 1 - Pr[E_1{}^c \cap E_2{}^c \cap ... \cap E_Q{}^c]$ $\boxed{\text{Since every event is independent of other}}$
$= 1 - Pr[E_1{}^c]*Pr[E_2{}^c]*...*Pr[E_Q{}^c]$ $\boxed{\text{If A \& B are independent } Pr[A \cap B] = Pr[A] * Pr[B]}$
$= 1 - \Pi(Pr[E_i{}^c])$
$= 1 - [1 - \frac{1}{M}]^Q$
$= 1 - [1 - (\frac{Q}{1})\frac{1}{M} + (\frac{Q}{2})\frac{1}{M^2} + (\frac{Q}{3})\frac{1}{M^3} + ...]$
$= 1 - (1 - \frac{Q}{M})$ $\boxed{\text{Since } M >> 1}$
$= 1 - 1 + \frac{Q}{M}$
$= \frac{Q}{M}$
Complexity $= \frac{1}{Probability} = \frac{1}{\frac{Q}{M}}$
$\therefore$ Complexity $= \frac{M}{Q} = O(M)$