[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy
Scribed by: Aniruddh Modi (202151022)

Winter 2023-2024
Lecture 8 9 (Week 5)

1 Lecture 8

1.1 AES-128

1.1.1 Round Functions

AES 128 consists of 10 rounds, of which first 9 are identical. The initial 9 rounds consist of following functions :

1. Sub bytes
2. Shift rows
3. Mix Columns

The last round involves :

1. Sub bytes
2. Shift rows

All these functions are $\{0,1\}^{128} \to \{0,1\}^{128}$.

1.1.1.1 Sub byte function $Subbytes : \{0,1\}^{128} \to \{0,1\}^{128}$.
$X = x_0x_1x_2x_3...x_{15}$ such that $|x_i| = 8$ bits.

$$\begin{bmatrix} x_0 & x_4 & . & . \\ x_1 & . & . & , \\ x_2 & . & . & , \\ x_3 & . & . & x_{15} \end{bmatrix} \to \begin{bmatrix} s_{00} & s_{01} & . & . \\ s_{10} & . & . & , \\ s_{20} & . & . & , \\ s_{30} & . & . & s_{33} \end{bmatrix}$$

In the subbyte function, we do following :

1. $c_7c_6...c_1 \leftarrow (01100011)$ which is $(63)_{10}$
2. Then, we calculate $S(s_{ij}) = (a_7a_6..a_0)$
3. **for** $i = 0$ to 7 **do**
   $b_i = (a_i + a_{(i+4) \mod 8} + a_{(i+5) \mod 8} + a_{(i+6) \mod 8} + a_{(i+7) \mod 8 + D)} \mod 2$
   end for
4. This will give us $S^{'}_{ij} = b$

We define $S : \{0,1\}^8 \to \{0,1\}^8$ as follows :

$S(X) = \begin{cases} 0, & X = 0 \\ Y, & \text{otherwise} \end{cases}$

Let $X = a_0a_1a_2a_3...a_{15}$. We define $P(X) = a_0 + a_1x^+a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 \in \mathbb{F}_2[x]$. Now, Let $\mathbb{F}_2[x] \,/ < G(x) >$ be a field over a primitive polynomial $G(X) = x^8+x^4+x^3+x^1+1$.

Now, we will find the multiplicative inverse of $P(X)$ under $G(X)$.

$$P(x) \cdot Q(x) \equiv 1 \mod G(X)$$

$$P(x) \cdot Q(x) + h(X) \cdot G(X) = 1$$

We will use extended euclidean algorithm to solve this. Lets take an example :

Example:

$x = (01010011) \to P(x) = x^6 + x^4 + x + 1$

$g(x) = x^8 + x^4 + x^3 + x + 1$

$$\begin{array}{r|llllllll}
 & x^2 & & +1 & & & & & \\
\hline
x^6 + x^4 + x + 1 & x^8 & +0x^6 & +x^4 & +x^3 & +0x^2 & +x & +1 & \\
(-) & x^8 & +x^6 & +0x^4 & +x^3 & +x^2 & & & \\
\hline
 & x^6 & +x^4 & +x^2 & +x & +1 & & & \\
 & x^6 & +x^4 & & +x & +1 & & & \\
\hline
 & & & & x^2 & & & &
\end{array}$$

$$\begin{array}{r|llll}
 & x^4 & +x^2 & & \\
\hline
x^2 & x^6 & +x^4 & +x & +1 \\
 & x^6 & & & \\
\hline
 & & x^4 & +x & +1 \\
 & & x^4 & & \\
\hline
 & & x & +1 &
\end{array}$$

$$\begin{array}{r|lll}
 & x & +1 & \\
\hline
x + 1 & x^2 & & \\
 & x^2 & +x & \\
\hline
 & & x & \\
 & & x & +1 \\
\hline
 & & & 1
\end{array}$$

$$1 = q(x)p(x) + h(x)g(x)$$
$$1 = (x+1)(x+1) + x^2$$
$$1 = x^2 + (x+1)[(x^6+x^4+x+1) + (x^2)(x^4+x^2)]$$
$$1 = x^2 + (x+1)[(x^6+x^4+x+1) + [(x^8+x^4+x^3+x+1) + (x^2+1)(x^6+x^4+x+1)]]$$
$$1 = (x+1)(x^6+x^4+x+1) + [1 + (x+1)(x^4+x^2)]x^2$$
$$1 = (x+1)(x^6+x^4+x+1) + [1 + x^2 + x^3 + x^4 + x^5]x^2$$
$$1 = (x+1)(x^6+x^4+x+1) + (x^5+x^4+x^3+x^2+1)[(x^8+x^4+x^3+x+1) + (x^2)(x^6+x^4+x+1)]$$
$$1 = (x^5+x^4+x^3+x^2+1)(x^8+x^4+x^3+x+1) + [(x+1) + (x^5+x^4+x^3+x^2+1)(x^2+1)](x^6+x^4+x+1)$$
$$1 = (x^7+x^6+x^3+x)(x^6+x^4+x+1)$$

$$\therefore g(x) = x^7 + x^6 + x^3 + x \rightarrow 11001010(a_7a_6a_5a_4a_3a_2a_1a_0)$$

$Subbytes(01010011) = (11101101)$ Now, lets take four MSBs of input (0101) and 4 LSBs of the input (0011) and then we can easily compute inverse using the table lookup.

1.1.1.2 Shift Row

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

1.1.1.3 Mix Column

$$S^{'} = \begin{bmatrix} x & x+1 & 1 & 1 \\ 1 & x & x+1 & 1 \\ 1 & 1 & x & x+1 \\ x+1 & 1 & 1 & x \end{bmatrix} S \ modG(x)$$

1.1.2 Key Scheduling Algorithm

There are 11 Rounds $k_1, k_2, ..., k_{11}$ where length of each round key is 128 bit. $K = key[0], key[1], ..., key[15]$.

We should remember that

1. $ROTWORD(B_0, B_1, B_2, B_3) = (B_1, B_2, B_3, B_0)$, where length of B_i is 8 bits.
2. $SUBWORD(B_0, B_1, B_2, B_3) = (B_0^{'}, B_1^{'}, B_2^{'}, B_3^{'})$.

3. Some Fixed Constants:

$$
\begin{aligned}
&(a) \quad \text{Rcon}[1] = 01000000\\
&(b) \quad \text{Rcon}[2] = 02000000\\
&(c) \quad \text{Rcon}[3] = 04000000\\
&(d) \quad \text{Rcon}[4] = 08000000\\
&(e) \quad \text{Rcon}[5] = 10000000\\
&(f) \quad \text{Rcon}[6] = 20000000\\
&(g) \quad \text{Rcon}[7] = 40000000\\
&(h) \quad \text{Rcon}[8] = 80000000\\
&(i) \quad \text{Rcon}[9] = 01\text{B}00000\\
&(j) \quad \text{Rcon}[10] = 36000000
\end{aligned}
$$

1.1.2.1 Algorithm

function KeyExpansion(byte $key[16]$, word $w[44]$)
  word $temp$
  for $i = 0$ to 3 **do**
    $w[i] = (key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])$
  end for
  for $i = 4$ to 43 **do**
    $temp = w[i]$
    if $i \mod 4 = 0$ **then**
      $temp = \text{SUBWORD}\{\text{ROTWORD}(temp) \oplus Rcon[i/4]\}$
    end if
    $w[i] = w[i-4] \oplus temp$
  end for
  return $(w[0], w[1], \ldots, w[43])$
end function

Length of $w[i]$ is 32 bit.

$$
\begin{aligned}
k_1 &= w[0] \| w[1] \| w[2] \| w[3]\\
k_2 &= w[4] \| w[5] \| w[6] \| w[7]\\
k_3 &= w[8] \| w[9] \| w[10] \| w[11]\\
&\vdots\\
k_{43} &= w[40] \| w[41] \| w[42] \| w[43]
\end{aligned}
$$

1.1.2.2 Properties of AES Operations:

1. SubByte and ShiftRow Invertibility: Both SubByte and ShiftRow operations in AES are invertible.

2. MixColumn Invertibility: The MixColumn matrix operation in AES must be invertible under modulo $x^8 + x^4 + x^3 + x + 1$.

1.1.3

Modes of Operations:

1.1.3.1 ECB:

- ECB stands for Electronic Code Book.
- **Input:** Key K, n-bit Plaintext $(x_1, x_2, \ldots, x_t)$
- **Encryption:**
  1. $Enc(x_i, K) = C_i$, where $1 \leq i \leq t$
- **Decryption:**
  1. $Dec(C_i, K) = x_i$, where $1 \leq i \leq t$

1.1.3.2 CBC:

- CBC stands for Cipher Block Chaining.
- **Input:** Key K, n-bit Plaintext $(x_1, x_2, \ldots, x_t)$
- **Encryption:**
  1. $C_0 = IV$ (Public parameter)
  2. $C_j = Enc(C_{j-1} \oplus x_j, K)$, where $1 \leq j \leq t$
- **Decryption:**
  1. $C_0 = IV$
  2. $x_j = Dec(C_j, K) \oplus C_{j-1}$, where $1 \leq i \leq t$

1.2 Stream Cipher

Stream ciphers encrypt plaintext bitwise. Let $M = m_0 m_1 \ldots m_l$ where $m_i \in \{0, 1\}$ be the plaintext and $K = k_0 k_1 \ldots k_l$ where $k_i \in \{0, 1\}$ be the key.

The ciphertext C is obtained by bitwise XOR (exclusive OR) of the plaintext and the key:

$$C = M \oplus K = (m_0 \oplus k_0)(m_1 \oplus k_1) \ldots (m_l \oplus k_l)$$

The encryption and decryption operations are given by: **Encryption:** $C_i = M_i \oplus K_i$ **Decryption:** $M = C \oplus K$

1. $P(M = m_1 | C = C_{h1}) = P(M = m_1)$
2. $C = M \oplus K$
   $$C_1 = m_1 \oplus K, \quad C_2 = m_2 \oplus K$$
   then $C_1 \oplus C_2 = (m_1 \oplus K) \cdot (m_2 \oplus K) = m_1 \oplus m_2$
3. $K = k_0 k_{l-1} \ldots k_l$, $r = n - l$ $M = m_0 \ldots m_n$ $k_1 = K \| \ldots \| k_{r-l}$ $C = M \oplus K$ $C' = C_0 \ldots C_{r-1}$

1.2.1 Important Points:

1. The length of the key (K) should be greater than or equal to the length of the message (M).
2. You cannot use the same key to encrypt different messages.
3. The length of the key (K) should be greater than or equal to the length of the message (M).