Week 6: Mode of Operation

**Recommendation for Block Cipher Modes of Operation**

NIST 800-38A, 2001

Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, NIST-800-38D, 2007
Modes of Operation – ECB Mode(1/2)

- **Electronic Code Book Mode**
  - Break a message into a sequence of plaintext blocks
  - Each plaintext block is encrypted (or decrypted) independently
  - The same plaintext block always produces the same ciphertext block
  - May not be secure; e.g., a highly structured message
  - Typically used for secure transmission of single values (e.g., encryption key)
Modes of Operation – ECB Mode(2/2)

(Step 1) A & B agreed to use E() and K each other beforehand

Alice

P_1

K → E → C_1

P_2

E → C_2

... → ...

P_n

E → C_n

Bob

C_1

K → D → P_1

C_2

D → P_2

... → ...

C_n

D → P_n

(Step 2) A wants to send a block of P to B and divides P into equal block P_i

A & B agreed to use E() and K each other beforehand

A wants to send a block of P to B and divides P into equal block P_i
Modes of Operation – CBC Mode (1/2)

Cipher Block Chaining Mode

- Each ciphertext block is affected by previous blocks
- No fixed relationship between the plaintext block and its input to the encryption function
- The same plaintext block, if repeated, produces different ciphertext blocks
- IV (Initializing Vector) must be known to both ends
- Most widely used for block encryption

\[ C_1 = E_K(P_1 \oplus IV) \]
\[ P_1 = IV \oplus D_K(C_1) \]
\[ C_2 = E_K(P_2 \oplus C_1) \]
\[ P_2 = C_1 \oplus D_K(C_2) \]
\[ C_3 = E_K(P_3 \oplus C_2) \]
\[ P_3 = C_2 \oplus D_K(C_3) \]
\[ C_4 = E_K(P_4 \oplus C_3) \]
\[ P_4 = C_3 \oplus D_K(C_4) \]
Modes of Operation – CBC Mode (2/2)

Alice

Bob
Modes of Operation – Ctr Mode

Parallel Operation
Unique Counter
Variation of CBC
Stream Cipher

- **Overview**
  - Originate from *one-time pad*
  - *bit-by-bit Exor* with pt and key stream \( (c_i = m_i \oplus z_i) \)
  - Encryption = Decryption  -->  Symmetric
  - Use **LFSR** (Linear Feedback Shift Register)
  - (external) Synchronous or self-synchronous

- **Properties**
  - *Faster and Low Complexity in H/W*  -->  *Lightweight !*
  - Security measure : Period of key stream, LC(Linear Complexity), Statistical properties
  - Vast amounts of theoretical knowledge
  - Proprietary and Confidential for Military
How LFSR works

- Notation: \(< L, C[D]\) > where connection polynomial
  
  \[ C[D] = 1 + c_1 D + c_2 D^2 + \ldots + c_L D^L \in \mathbb{Z}_2[D] \]

- If \(c_L = 1\), \{i.e., \(\text{deg}\{C[D]\} = L\}\}, \(C[D]\) is called a nonsingular polynomial

- If initial vector \(\sigma_0\) is \([s_{L-1}, \ldots, s_1, s_0]\), \(s_i = \{0, 1\}\), output sequence \(s = s_0, s_1, \ldots\) is uniquely determined by the recursion
  
  \[ s_j = (c_1 s_{j-1} + c_2 s_{j-2} + \ldots + c_L s_{j-L}) \mod 2, \ j \geq L \]

- (Ex) \(<4, 1 + D + D^4>\), \(\sigma_0 = [0, 1, 1, 0] \Rightarrow c_1 = 1, c_4 = 1, s_4 = s_3 + s_0\)

<table>
<thead>
<tr>
<th>(t)</th>
<th>(D_3)</th>
<th>(D_2)</th>
<th>(D_1)</th>
<th>(D_0)</th>
<th>(t)</th>
<th>(D_3)</th>
<th>(D_2)</th>
<th>(D_1)</th>
<th>(D_0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>13</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Output seq. = 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1
Synchronous Stream Cipher (1/2)

- \( f \): next state ft, \( \sigma_{i+1} = f(\sigma_i, k) \), \( \sigma_0 \): initial value
- \( g \): keystream generating ft (e.g., LFSR or Block Cipher)
- \( z_i = g(\sigma_i, k) \), \( k \): key
- \( h \): output ft, \( c_i = h(z_i, m_i) \), \( m_i \): pt, \( z_i \): key stream, \( c_i \): ct

**Encryption**

\[
\begin{align*}
\sigma_i & \rightarrow f \rightarrow \sigma_{i+1} \\
& \leftarrow k \\
& \leftarrow g \\
m_i & \rightarrow h \\
& \rightarrow c_i \\
& \leftarrow z_i
\end{align*}
\]

**Decryption**

\[
\begin{align*}
\sigma_i & \rightarrow f \rightarrow \sigma_{i+1} \\
& \leftarrow k \\
& \leftarrow g \\
z_i & \rightarrow h^{-1} \\
& \leftarrow c_i \\
c_i & \rightarrow m_i
\end{align*}
\]
Synchronous Stream Cipher(2/2)

• Keystream is independent of pt and ct
• Properties
  – Synchronization requirement
  – No error propagation
  – Active attack
    • Insertion, deletion or replay will lose synchronization
    • Change selected ciphertext digits ➔ Need to have integrity check mechanisms
Self-Sync. Stream Cipher (1/2)

- \( \sigma_i = (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1}) \), \( \sigma_0 = (c_{t}, c_{t+1}, \ldots, c_{t}) \): initial value
- \( g \): keystream generating ft, (e.g., LFSR or Block Cipher)
- \( z_i = g(\sigma_i, k), k \): key
- \( h \): output ft, \( c_i = h(z_i, m_i), m_i \): pt, \( z_i \): keystream, \( c_i \): ct

![Diagram showing encryption and decryption processes.](image)
Self-Sync. Stream Cipher (2/2)

• Keystream is dependent of pt and ct
• Properties
  – Self-Synchronization
  – Limited error propagation
  – Active attack
    • Difficult to detect insertion, deletion, or replay
    • Easy to find passive modification
  – More diffusion ➔ more resistant against attacks based on plaintext redundancy
OFB Mode = Sync. Stream cipher

The structure is similar to that of CFB, but:
- CFB: Ciphertext is fed back to the shift register
- OFB: Output of E is fed back to the shift register

For security reason, only the full feedback (j = block size) mode is used.

- No error propagation
- More vulnerable to a message stream modification attack
- May useful for secure transmission over noisy channel (e.g., satellite communication)
CFB Mode = Self-sync stream cipher

 Cipher Feedback Mode

- A way of using a block cipher as a stream cipher
- A shift register of block size maintains the current state of the cipher operation, initially set to some IV
- The value of the shift register is encrypted using key $K$ and the leftmost $j$ bits of the output is XORed with $j$-bit plaintext $P_i$ to produce $j$-bit ciphertext $C_i$
- The value of the shift register is shifted left by $j$ bits and the $C_i$ is fed back to the rightmost $j$ bits of the shift register
- Typically $j = 8, 16, 32, 64$ ...
- Decryption function $D_K$ is never used
Mode of Operation - CCM
Mode of operation - summary

• Use of mode
  – ECB: key management, useless for file encryption
  – CBC: File encryption, useful for MAC
  – $m$-bit CFB: self-sync, impossible to use channel with low BER
  – $m$-bit OFB: external-sync. $m=1,8$ or $n$
  – Ctr: secret ctr, parallel computation
  – CCM: authenticated encryption = ctr + CBC
  – Performance Degradation/ Cost Tradeoff