Block Ciphers Modes

Ramki Thurimella
Only Encryption

- I.e. messages could be modified
- Should not assume that nonsensical messages do no harm
- Always must be combined with authentication
Padding

- Must be reversible
- Simply adding enough 0s at the end until the length of the padded message is a multiple of the block size is NOT a good idea. This is not reversible
- Use one of the following two methods instead:
  - First always append 0111 1111 and as many 0s as necessary
  - Let $l(P) + n$ be the shortest multiple of the block length for some $n \geq 1$. Pad $P$ with $n$ bytes where each byte contains $n$ in binary.
$C_i \leftarrow E(K,P_i)$ for $i = 1,2,\ldots,k$

- **Serious Weakness:**
  - Repetition of plaintext between different blocks translates to repetition of ciphertext

- **For e.g.:**
  - In Unicode, every other byte is a 0
  - Many files have repeated 0s

- **Fix**
  - Use chaining, i.e. use the ciphertext from the previous block in computing the ciphertext for the current block, or
  - Use a nonce along with $K$ and $P_i$ when encrypting
Cipher Block Chaining (CBC)

• Mathematically
  \[ C_i \leftarrow E(K, P_i \oplus C_{i-1}) \text{ for } i=1,2,...,k \]

• What should \( C_0 \) be?

• \( C_0 \) is called the *initialization vector (IV)*. Some choices:
  – Fixed value
  – Counter
  – Random
  – Nonce-derived
IV: how about fixed value?

- If two *messages* have the same plaintext, their corresponding ciphertext would be same
- Not recommended
Counter IV

- Create a message counter starting at, say 0, and use this as the IV
- If plaintext for two messages starts similarly, then XOR with counter-based IV could cancel out the changes and produce the same ciphertext
- Not recommended
Random IV

- $C_0$ random block value
- $C_0$ like other $C_i$ is NOT a secret
- Ciphertext is one block longer than the plaintext as $C_0$ needs to be communicated to the receiver
- This is a major disadvantage as plaintext messages typically tend to be one block long
Nonce-Generated IV

- *Nonce* a unique number (stands for number used once)
- This could be constructed, for example, from the message number
- Nonce should be as long as one block
- Encrypt the nonce to generate IV
- IV does not need to be sent separately if the receiver has the message number, key, and the encryption method
Output Feedback Mode (OFB)

• Generate a pseudorandom key stream of length equal to the length of the plaintext P

• Mathematically

\[ K_0 = IV \]
\[ K_i \leftarrow E(K, K_{i-1} \quad C_{i-1}) \text{ for } i=1,2,\ldots,k \]
\[ C_i \leftarrow P_i \oplus K_i \]
OFB (cont.)

- Random IV, or
- Can be generated from a nonce as in CFB
- Advantages
  - Encryption and decryption the same
  - No need for padding
- Disadvantage
  - If the same IV is used for two messages, the will be encrypted with the same key stream
  - So what?
  - Eve captures \( C_i \) and \( C'_i \) and computes their XOR
  - This is \( P_i \oplus K_i \oplus P'_i \oplus K_i = P_i \oplus P'_i \)
  - Knowing XOR of two plaintext message, and partial information about one of them, information about the other can be derived
OFB (cont.)

- Second disadvantage
  - If one key block repeats, then the entire stream blocks up until that point repeat
  - A collision is expected to happen after encrypting $2^{64}$ blocks of data assuming a block size of 128 bits
Counter (CTR) mode

- Operates in stream cipher mode
- Mathematically
  \[ K_i \leftarrow E(K, \text{Nonce} || i) \text{ for } i=1,2,\ldots,k \]
  \[ C_i \leftarrow P_i \oplus K_i \]
- Simple
- Nonce and \( i \) should fit in one block
- Nonce should never be repeated
Exercise

- Assuming
  - 48-bit message number
  - 16-bit nonce
  - 64 bits for $I$

- How many different messages can be encrypted?

- How long can be each message?
CTR (cont.)

Advantages

- Identical encryption and decryption
- Key stream computation can be parallelized
- The security of CTR mode is directly linked to the security of the block cipher
Final recommendation

- CBC with random IV
  - Though there is overhead of increased ciphertext length
  - slow