Double Encryption, Meet in the middle attacks

Kaden Pirmohamed

Summary

Today's discussion emphasized the significance of triple encryption over double encryption and highlighted why double encryption, despite being used with modern standards like AES, fails to provide the expected level of security.

KEY

- C_i: Blocks of encrypted/Cipher text
- X_i: Blocks of plain text
- IV: Initial Value
- $E_k(X_i)$: Encryption of X_i
- $D_k(C_i)$: Decryption of Cipher text

Modes of Operation

1. Electronic Code Book (ECB)

Encryption

Formula: $C_1 = E_k(X_1)$

Decryption

Formula: $X_1 = D_k(C_1)$

2. Cipher Block Chaining (CBC)

Encryption

Formulas:

$$IV = C_0$$
$$C_i = E_k(X_i \oplus C_{i-1}) \quad \rightarrow \quad C_1 = E_k(X_1 \oplus C_0) \quad \rightarrow \quad C_2 = E_k(X_2 \oplus C_1) \quad ...$$

Decryption

Formula:

$$X_i = D_k(C_i) \oplus C_{i-1}$$

3. Cipher Feedback (CFB)

Encryption

Formulas:

$$IV = C_0$$
$$C_i = E_k(C_{i-1}) \oplus X_i \quad \rightarrow \quad C_1 = E_k(C_0) \oplus X_1$$

Decryption

Formula:

$$X_i = E_k(C_{i-1}) \oplus C_i$$

4. Output Feedback (OFB)

Encryption

Formulas:

$$\begin{aligned} IV &= O_0 \\ O_i &= E_k(O_{i-1}) \\ C_i &= X_i \oplus O_i \end{aligned}$$

Decryption

Formula:

$$X_i = C_{i-1} \oplus O_i$$

5. Counter (CTR)

Encryption

Formulas:

$$\begin{aligned} IV &= CTR_0 \\ CTR_i &= CTR_{i-1} + 1 \mod 2^m \\ CTR_i &= CTR_{i-1} + i \\ C_i &= E_k(CTR_i) \oplus X_i \end{aligned}$$

Decryption

Formula:

$$X_i = E_k(CTR_i) \oplus C_i$$

Notes

Unlike classical ciphers, in AES and SAES, double encryption isn't identical to single encryption.

To enhance AES security, double encryption is employed. Alice and Bob select two keys k_1 and k_2 to encrypt $E_{k_2}(E_{k_1}(p))$ (with p being plaintext). Eve must find both keys to break this encryption.

Brute-forcing both keys entails trying all pairs of k_1 and k_2, resulting in 2^{32} bits total, a formidable task even with modern computing. However, double encryption is vulnerable to a meet-in-the-middle attack:

1. Let the ciphertext be $C = E_{k_2}(E_{k_1}(P))$
2. Decrypt both sides: $D_{k_2}(C) = E_{k_1}(P)$
3. Construct decryption and encryption tables ($D_k(C)$ and $E_k(p)$)
4. Identify matching entries, potential k_1, k_2 pairs
5. Repeat steps 1-4 with new plaintext and ciphertext

Typically, there's a single pair of k_1 and k_2 that works both times, taking only four times as long as single encryption. With certain techniques, it may only take twice as long.

For increased security without succumbing to meet-in-the-middle attacks, triple encryption is the solution:

- Utilize two keys k_1 and k_2: $C = E_{k_1}(E_{k_2}(E_{k_1}(P)))$

- Decrypt both sides: $D_{k_1}(C) = E_{k_2}(E_{k_1}(P))$
- Decrypt both sides again: $D_{k_2}(D_{k_1}(C)) = E_{k_1}(P)$

Noticeably, there's no feasible way to isolate a single key on either side of the equation, thereby achieving the additional security provided by double encryption without concerns about a meet-in-the-middle attack.