CBC Mode of Operation

T-79.5502 Advanced Course on Cryptology. 2.5.2006

Maarit Hietalahti

Contents

- Cipher Block Chaining (Textbook 7.8.2)
- A Common Misconception
- CBC plaintext padding schemes
- Vaudenay's attack
- A confidentiality limitation in CBC (Knudsen)
- A Side Channel Attack on a TLS Application (Textbook 12.5.4)
- An attack on Needham-Schroeder Symmetric Key Authentication protocol
- Conclusions

Cipher Block Chaining (Textbook 7.8.2)

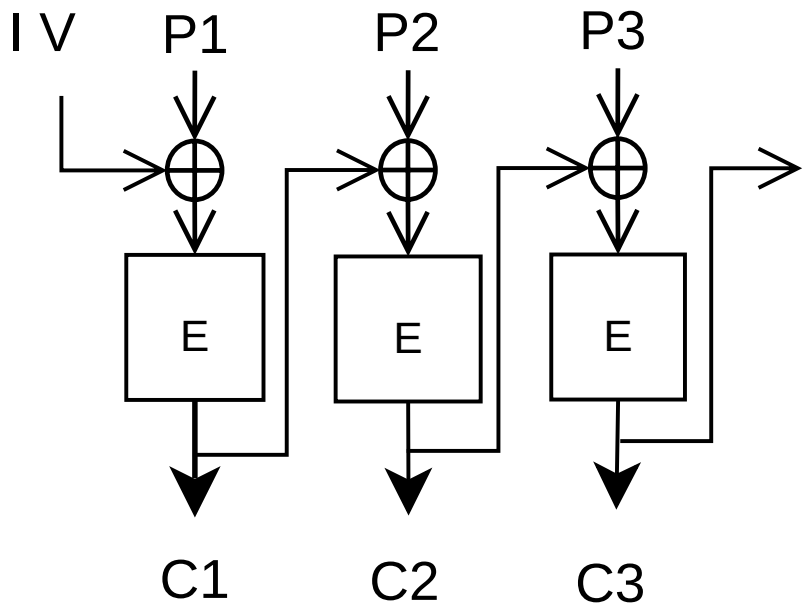

Figure 1: Encryption in CBC Mode of Operation

Encrypt: $C_0 \leftarrow IV, C_i \leftarrow \mathcal{E}(P_i \oplus C_{i-1}), i = 1, 2, \ldots, m$

Decrypt: $C_0 \leftarrow IV, P_i \leftarrow \mathcal{D}(C_i) \oplus C_{i-1}, i = 1, 2, \ldots, m$

A Common Misconception

- In CBC, data blocks are chained together, therefore in some block cipher specifications CBC mode is used for data integrity protection.
- Wrong! CBC does not provide data integrity protection
- RC5-CBC-PAD mode and IP Encapsulating Security Payload (ESP) in IPSec, for example, have a padding scheme that should not be used for data integrity
- Vaudenay’s attack on CBC padding byte scheme when it is used as a "data integrity" checking method

CBC plaintext padding schemes

RC5-CBC-PAD

- Plaintext is divided into blocks of 8 bytes (64 bits)
- The final plaintext block must be padded: the final a plaintext bytes $0 \leq a \leq 7$ are followed by $8 - a$ padding bytes, valued $8 - a$

  for example: $messagebyte_1 \| messagebyte_2 \| '06' \| '06' \| '06' \| '06' \| '06' \| '06'$

ESP

- X padding bytes $1 \leq X \leq 255$

$$'01' \| '02' \| '03' \| \cdots \| 'X'$$

Vaudenay's attack

- A key holder becomes a decryption oracle:
- Malice sends two ciphertext blocks r, C_i to the oracle
- r is random, $C_i = \mathcal{E}(P \oplus C_{i-1})$; Malice is interested in finding out P.
- The corresponding decryption will be $P \oplus C_{i-1} \oplus r$
- Decryption oracle answers YES for a "valid padding"
- Most likely, '01' is the "padding byte" (with probability 2^{-8}), other possibilities are very small and therefore neglected
- Malice discovers the final byte of P:
  $LSB_8(P) = LSB_8(r) \oplus' 01' \oplus LSB_8(C_{i-1})$. (Note: a mistake in the textbook!)

Vaudenay's attack continued

- If the decryption procedure detects a padding error, the oracle may answer NO explicitly, or give no answer at all
- If the procedure terminates, the oracle is called a *bomb oracle*
- Vaudenay's attack can be applied to several cryptographic protocols used in many real-world applications, as long as a YES/NO answer is available (even if encrypted)
- Malice can change r and retry. Often the oracle can be maintained to be a non-explosive one, and so it answers further questions.
- Then, a whole plaintext block can be extracted in $8 \times 2^8 = 2048$ oracle calls.

A confidentiality limitation in CBC (Knudsen)

- When two ciphertext blocks are equal: $C_i = C'_j$, we have
  $C_{i-1} \oplus C'_{j-1} = P_i \oplus P'_j$
- Plaintext usually contains redundancy, which helps in recovering the plaintexts from this equation
- Random IVs for each encryption session makes the probability of two equal ciphertexts less likely and therefore this attack less feasible

A Side Channel Attack on a TLS Application

(Canvel et al., Textbook 12.5.4)

- Vaudenay's attack with an email server acting as a decryption oracle
- The link between client and server is encrypted using a strong session key as a result of a TLS protocol run.
- The session encryption uses a strong block cipher (triple DES) in the CBC mode of operation
- A YES/NO answer is extracted by a timing analysis
- Target: the email password encrypted in C

A Side Channel Attack continued

- Malice sends r, C to server (pretending to be the owner of the password)
- Server performs CBC decryption and checks the validity of the padding.
- If the padding is correct, server will further check data integrity by recalculating a MAC. If not, there is no need to calculate the MAC.
- The data integrity calculation will fail with an overwhelming probability. Hence, an encrypted error message will be sent back to the client machine
- "Invalid MAC" with probability 2^{-8} implies a "valid padding"
- A sufficiently large r results in the recalculation of a lengthy CBC MAC
- Server's response time can differ in terms of a few milliseconds
- Error handling procedure cannot act as a bomb oracle, therefore the oracle is reliable and the attack can proceed
- A possible fix: The server should take a random elapse of sleep before sending an error message

An attack on Needham-Schroeder

Or the insufficiency of Encryption-Decryption Approach for Authentication (Textbook 17.2.1.2)

- Encryption: any encryption algorithm that is not designed to also provide a data integrity protection, for example AES, with the CBC mode of operation
- The attack will not use any weakness in the algorithms quality of confidentiality service
- The first two steps of Needham-Schroeder Symmetric Key Authentication protocol:
  1. Alice $\rightarrow$ Trent: $Alice, Bob, N_A$
  2. Trent $\rightarrow$ Alice: $\{N_A, K, Bob, Y\}_{K_{AT}}$; where $Y = \{Alice, K, T\}_{K_{BT}}$
- Let $P_1, P_2, \ldots, P_m$ denote the plaintext message blocks of $\{N_A, K, Bob, Y\}_{K_{AT}}$
- K should be no smaller than one block, N_A also sufficiently large

- $\Rightarrow P_2$ contains the session key, or part of it
- Malice knows the size of the session key, plaintext size and implementation details (they should not be a secret)
- Let $IV, C_1, C_2, \ldots, C_m$ denote the ciphertext blocks corresponding to the plaintext blocks $P_1, P_2, \ldots, P_m$
- Let $IV', C'_1, C'_2, \ldots, C'_m$ denote the ciphertext blocks of a previous run of the same protocol between the same principals
- Malice intercepts the second step of the protocol and replaces the blocks flowing from Trent to Alice:
- 2. Trent $\rightarrow$ Malice("Alice"): $IV, C_1, C_2, \ldots, C_m$
- 2. Malice("Trent") $\rightarrow$ Alice: $IV, C_1, C'_2, \ldots, C'_m$
- Alice will get N_A in good order, since it is located in the first two blocks
- The key is previously unknown to Alice, and random, therefore she doesn't notice the change in the second block

- The tail of the plaintext message is the same as previously, and encrypted correctly in the CBC mode of operation (the ciphertext doesn't have to be the same)
- The new key will now be $\hat{K} = D_{K_{AT}}(C_2') \oplus C_1 = K' \oplus C_1' \oplus C_1$
- K' is the old session key (or part of it) which Malice may have already required $\Rightarrow$ lack of forward secrecy!
- Despite of the freshness of the nonce, the rest of the message is not fresh!

Conclusions

- CBC mode should not be used for data integrity protection
- Oracle services can be generally available
- Error messages in cryptographic protocols need to be handled with care
- It is better to use message authentication techniques based on one-way transformations rather than encryption-decryption techniques