CS458/CS558 Introduction to Computer Security

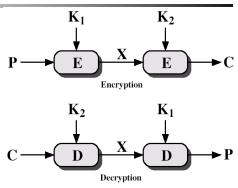
Double-DES?

- Could use 2 DES encrypts on each block
  $C = E(K2, E(K1, P))$
- Issue of reduction to single stage
  - Would it be possible to find a key K3 such that
    $E(K2, E(K1, P)) = E(K3, P)$
  - Answer: NO - proved in 1992

Double-DES?

- Meet-in-the-middle attack
  - Works whenever use a cipher twice
  - Based on the observation:

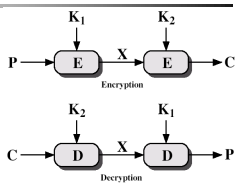
If $C = E(K2, E(K1, P))$,
then $X = E(K1,P) = D(K2,C)$



Double-DES?

- Meet-in-the-middle attack
  - Works whenever use a cipher twice
  - Based on the observation:

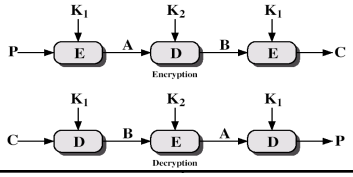
If $C = E(K2, E(K1, P))$,
then $X = E(K1,P) = D(K2,C)$



- Encrypt P for all 2^{56} keys. Store the result in a table and then sort the table.
- Decrypt C with 2^{56} keys and check the result against the table for a match.
- If a match occurs, then test the two resulting keys against some more known plaintext-ciphertext pairs.

Triple-DES with Two-Keys

- An obvious counter to the meet-in-the-middle attack is to use three stages of encryption with 3 different keys → requiring key length 168-bits
- Can use 2 keys with E-D-E sequence
  - $C = E(K1, D(K2, E(K1,P)))$
  - The only advantage of the use of decryption for the second stage is: if $K1=K2$ then can work with single DES



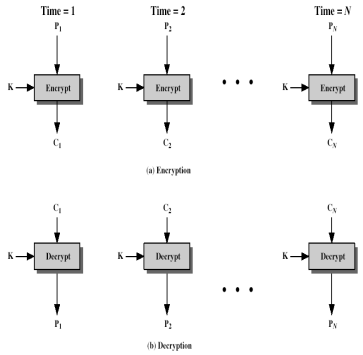
Triple-DES with Three-Keys

- No current known practical attacks
  - Brute-force: 2^{112}
  - Differential cryptanalysis: > 10^{52} plaintext-ciphertex pairs
- Can use Triple-DES with 3 Keys to avoid these
  - $C = E(K3, D(K2, E(K1, P)))$
- Has been adopted by some Internet applications, eg PGP, S/MIME

Section 6 Block Cipher Modes of Operation

Electronic Codebook Mode (ECB)

- The message is divided into blocks.
- Plaintext is handled one block at a time and each block is encrypted using the same key.

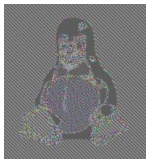


Advantages and Limitations of ECB

- Each block is encrypted independently of the other blocks
- Ideal for a short amount of data, e.g. transmit a DES key securely.
- For lengthy mesg, the ECB mode may not be secure.
  - The same n-bit block of plaintext, if it appears more than once in the mesg., always produces the same ciphertext - does not hide data patterns well.

Example: Disadvantage of ECB

- A pixel-map version of the image on the left was encrypted with ECB mode and with other modes.

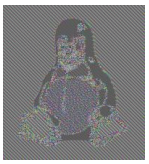


Encrypted using ECB mode

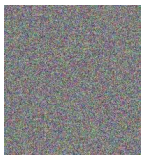
Example: Disadvantage of ECB

- A pixel-map version of the image on the left was encrypted with ECB mode and with other modes.

Original



Encrypted using ECB mode



Encrypted using other modes

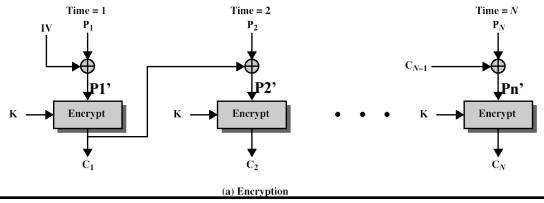
Advantages and Limitations of ECB

- Weakness is due to the encrypted message blocks being independent
  - Would like a technique in which the same plaintext block, if repeated, produces different ciphertext block.

Cipher Block Chaining (CBC)

- The input to the encryption algorithm is the XOR (⊕) of the current plaintext block and the preceding ciphertext block. ($A \oplus A = 0, 0 \oplus A = A$)
- Each ciphertext block is dependent on all plaintext blocks processed up to that point.

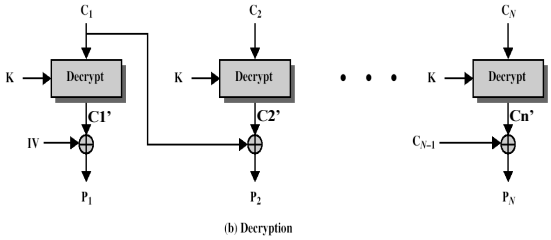
$C_j = E(K,[C_{j-1} \oplus P_j]), C_0 = IV$



Cipher Block Chaining (CBC)

- For decryption, each cipher block is passed through the decryption alg.. The result is XORed with the preceding ciphertext block to produce the plaintext.

$P_j = C_{j-1} \oplus D(K,C_j), C_0 = IV$



Cipher Block Chaining (CBC)

- How to prove that the decryption process is correct?

Encryption: $C_j = E(K,[C_{j-1} \oplus P_j]), C_0 = IV$

Decryption: $P_j = C_{j-1} \oplus D(K,C_j), C_0 = IV$

Cipher Block Chaining (CBC)

- How to prove that the decryption process is correct?

Encryption: $C_j = E(K,[C_{j-1} \oplus P_j]), C_0 = IV$

Decryption: $P_j = C_{j-1} \oplus D(K,C_j), C_0 = IV$

Proof:

$A \oplus A = 0$

$0 \oplus A = A$

$$\begin{aligned} P_j &= C_{j-1} \oplus D(K,C_j) \\ &= C_{j-1} \oplus D(K, E(K,[C_{j-1} \oplus P_j])) \\ &= C_{j-1} \oplus C_{j-1} \oplus P_j \\ &= 0 \oplus P_j \\ &= P_j \end{aligned}$$

Message Padding

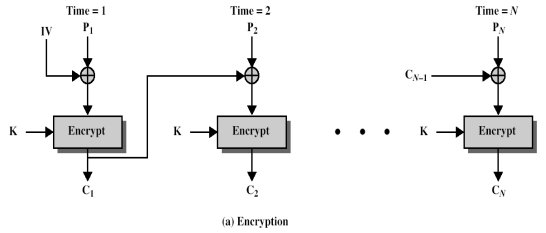
- At end of message must handle a possible last block, which is not as large as block size of cipher

Message Padding

- At end of message must handle a possible last block, which is not as large as block size of cipher
  - Pad either with known non-data value (eg nulls)
  - Or pad last block along with count of pad size
    - eg. [b1 b2 b3 0 0 0 0 5] - 3 data bytes, 5 bytes pad+count

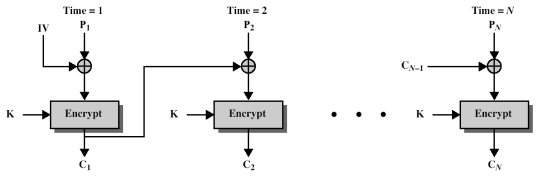
Advantages and Limitations of CBC

- Advantage:
  - Avalanche effect: A ciphertext block depends on all blocks before it - any change to one block affects all the following ciphertext blocks



Advantages and Limitations of CBC

- Disadvantage:
  - Encryption is sequential (i.e., cannot be parallelized)
  - Need Initialization Vector (IV)
    - Which must be known to sender & receiver
    - IV must either be a fixed value or be sent encrypted in ECB mode before rest of message

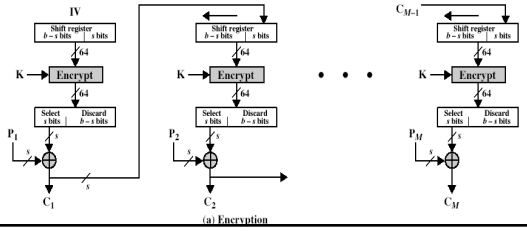


Cipher FeedBack Mode (CFB)

- When the data unit is smaller than the block size (e.g. data is only available a bit/byte at a time).
- Convert DES into a stream cipher and can be used to encrypt any number of bits
  - Property of stream cipher: the ciphertext is of the same length as the plaintext.
  - Eliminates the need to pad a mesg.

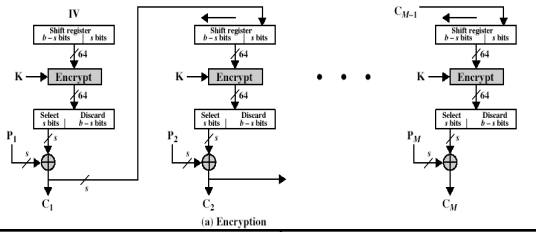
Cipher FeedBack Mode (CFB): Encryption

- E.g. The unit of transmission is s bits.
  - 64-bit shift register is initially set to some initialization vector (IV).
  - The leftmost s bits of the output of the encryption function is XORed with the first unit of plaintext P1 to produce C1.



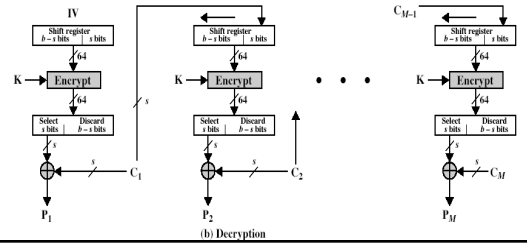
Cipher FeedBack Mode (CFB): Encryption

- E.g. the unit of transmission is s bits.
  - The contents of the shift register are shifted left by s bits and C1 is placed in the rightmost s bits of the shift register.
  - Continue this process until all plaintext units have been encrypted.



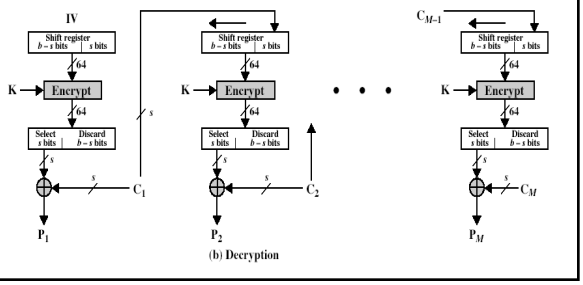
Cipher FeedBack Mode (CFB): Decryption

- E.g. the unit of transmission is s bits.
  - Same scheme as encryption, except that the received ciphertext unit is XORed with the output of the encryption function to produce the plaintext unit.



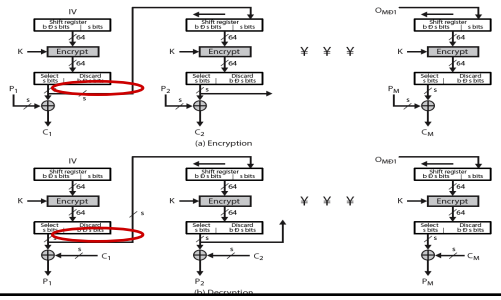
Cipher FeedBack Mode (CFB): Decryption

- Question: how to prove that the decryption is correct?



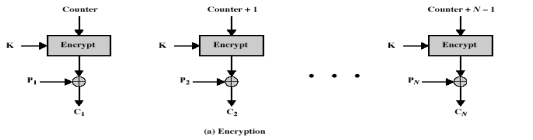
Output FeedBack (OFB)

- In OFB, the output of the encryption function is fed back to the shift register, while in CFB, the ciphertext unit is fed back to the register.



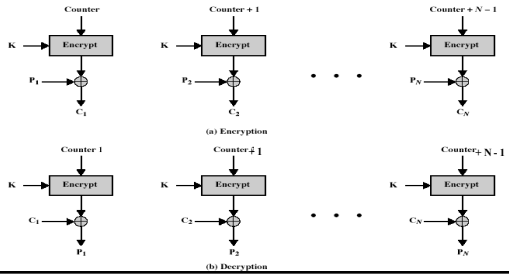
Counter (CTR)

- The counter is initialized to some value and then incremented by 1 for each subsequent block.
- The counter can be any simple function which produces a sequence which is guaranteed not to repeat for a long time



Counter (CTR)

- The counter is initialized to some value and then incremented by 1 for each subsequent block.
- The counter can be any simple function which produces a sequence which is guaranteed not to repeat for a long time



Advantages of CTR

- Efficiency
  - Encryption/decryption can be done in parallel on multiple blocks of plaintext or ciphertext
  - The execution of the encryption algorithm does not depend on the plaintext and ciphertext - can preprocess in advance.
  - Uses: high-speed network encryptions
- Random access to encrypted data blocks
- Provable security (good as other modes)