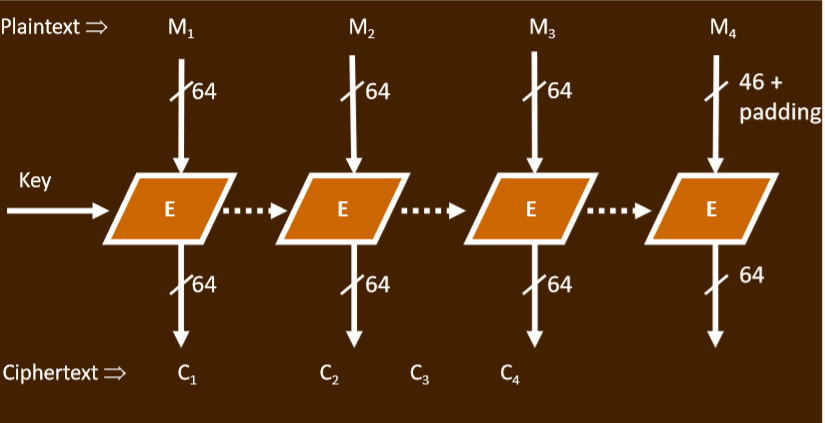
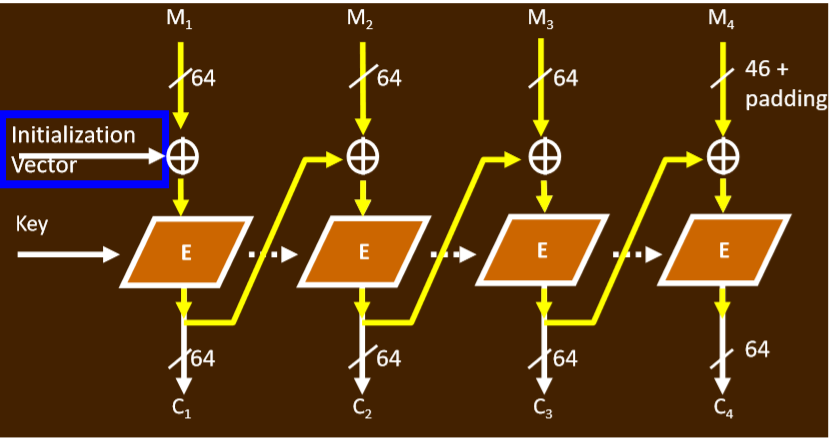
Electronic Code Book (ECB)

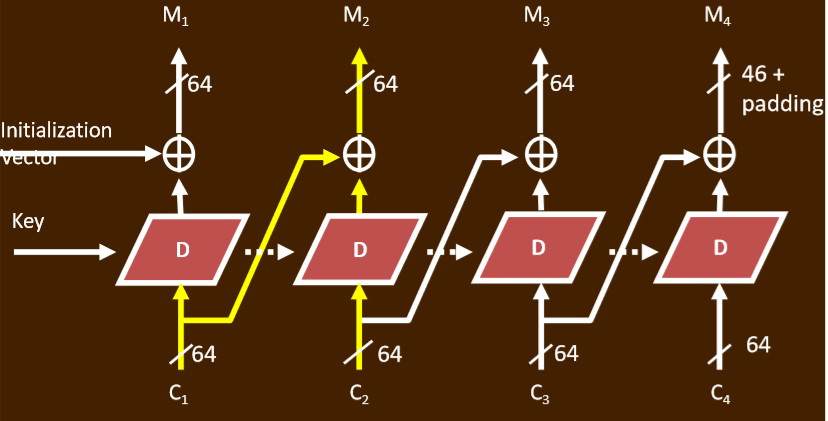


Decryption is opposite direction. Each block is independently encrypted/decrypted. Errors do not propagate

Parallel processing: Yes(both). Encrypting 2 same plaintext gives the same plaintext. => detect 2 blocks are identical. Ciphertext can be manipulated profitably since the hacker can isolate the information they want to change

Cipher Block Chaining (CBC)

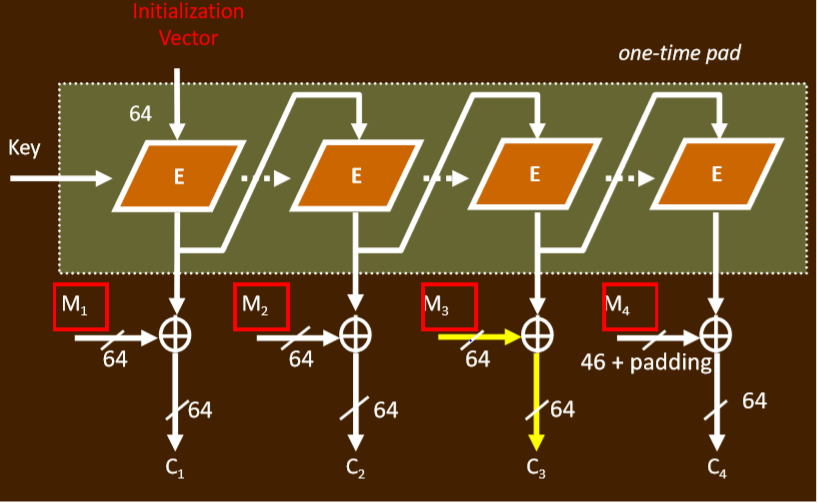


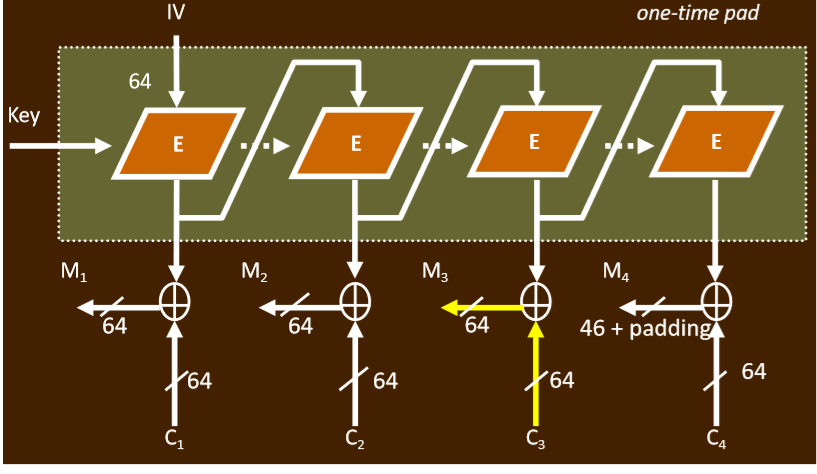


Parallel processing: No for encryption, yes for decryption. Errors propagate: yes(encryption), a little(decryption). Info leak? Same plaintext blocks produce different ciphertext. Manipulated? Yes, because errors do not propagate.

IV changes ciphertext whenever IV changes so that it makes harder for hacker to guess ciphertext patterns and manipulate profitably.

Output Feedback Mode (OFB)



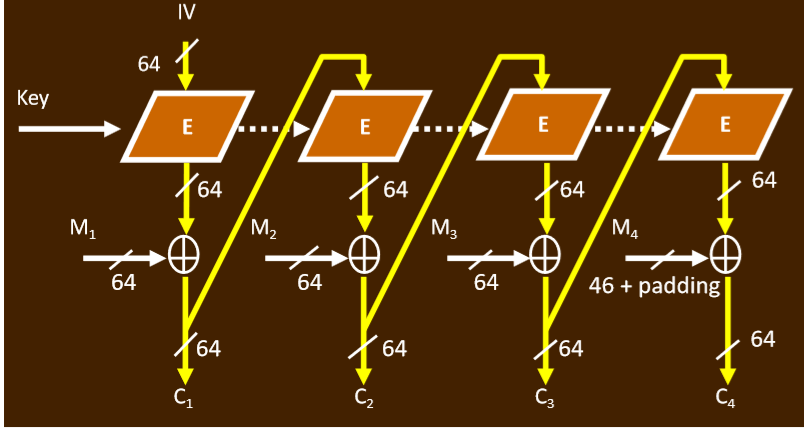


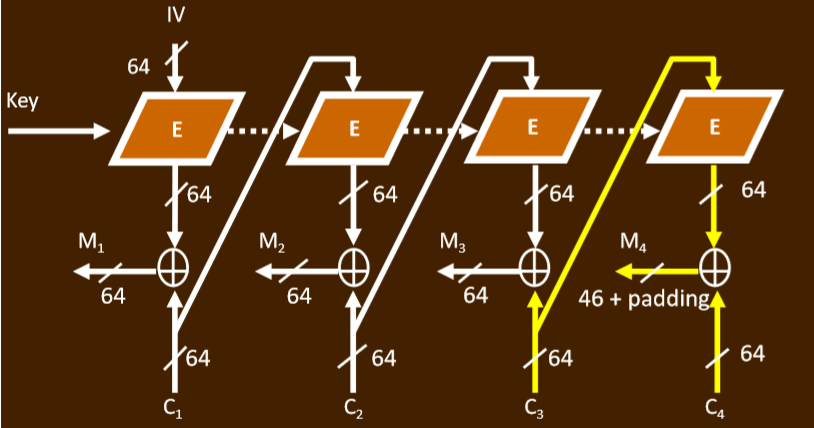
Stream cipher

Parallel processing: if the key is encrypted before the availability of the plaintext block, encrypt/decrypt Yes. (generating pad: NO, blocks: yes)

Errors propagate: just the particular block that is errored. Info leak? Yes and the hacker can send any arbitrary profitable valid messages => IV MUST be one-time pad (OTP).

Cipher Feedback Mode(CFB)



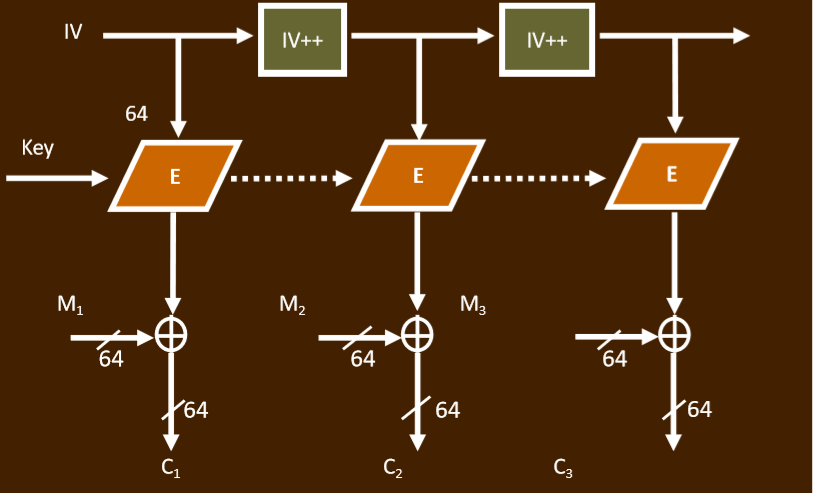


Stream cipher

Parallel processing: No( Encryption), yes (Decryption). Info leak: same plaintext, different ciphertext

Error propagate: Yes (Encryption), only the current ciphertext block and the one next to it (Decryption)

Counter Mode (CTR)



Stream cipher

Info leak: same plaintext, different cipher text due to different IV

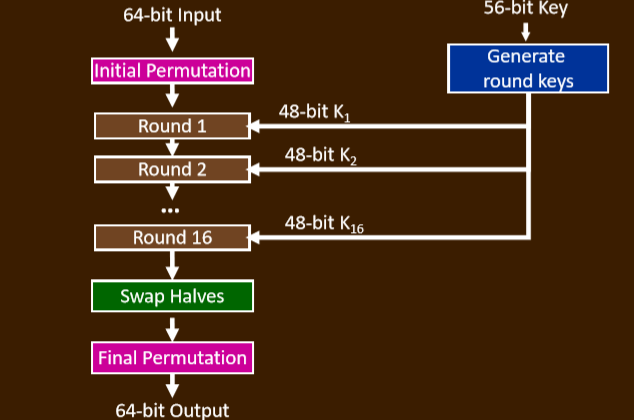
Parallel processing: YES (for both generating pad and XOR)

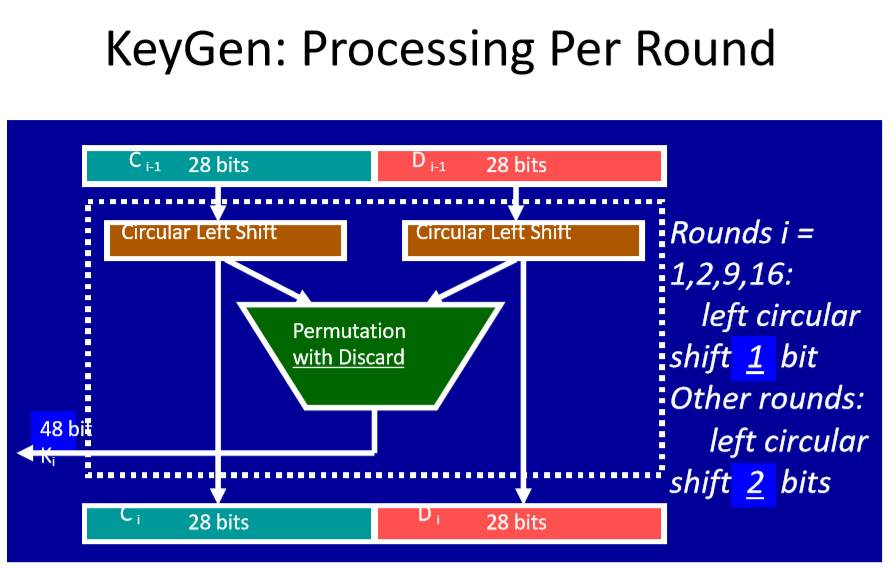
Error propagate: NO. (only that block is affected)

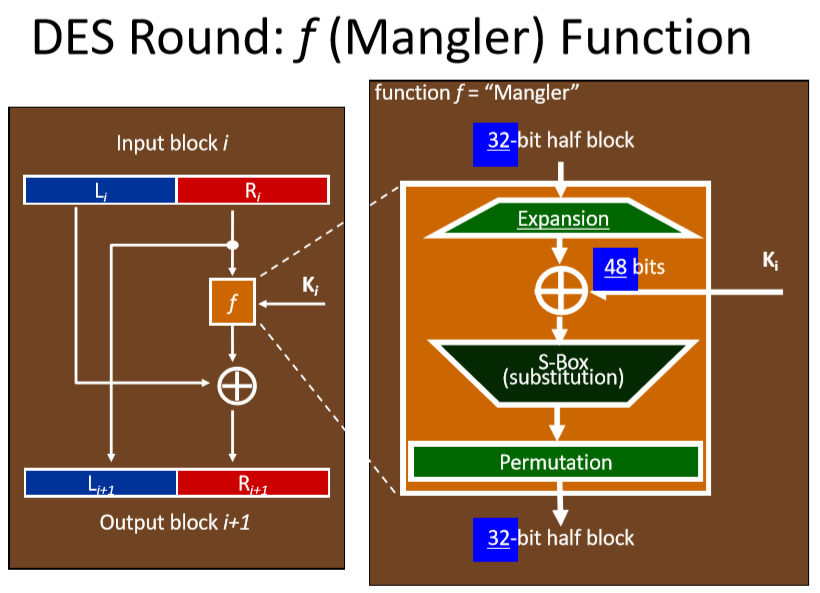
Allow decryption for ciphertext at any location.

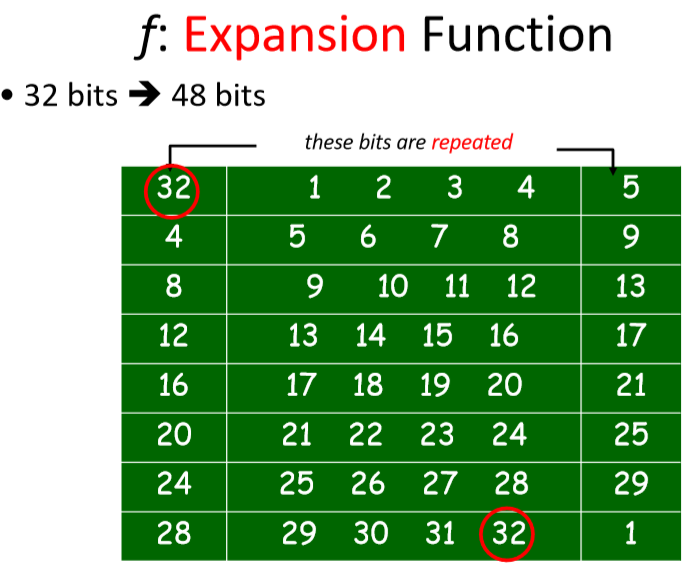
Data Encryption Standard (DES)

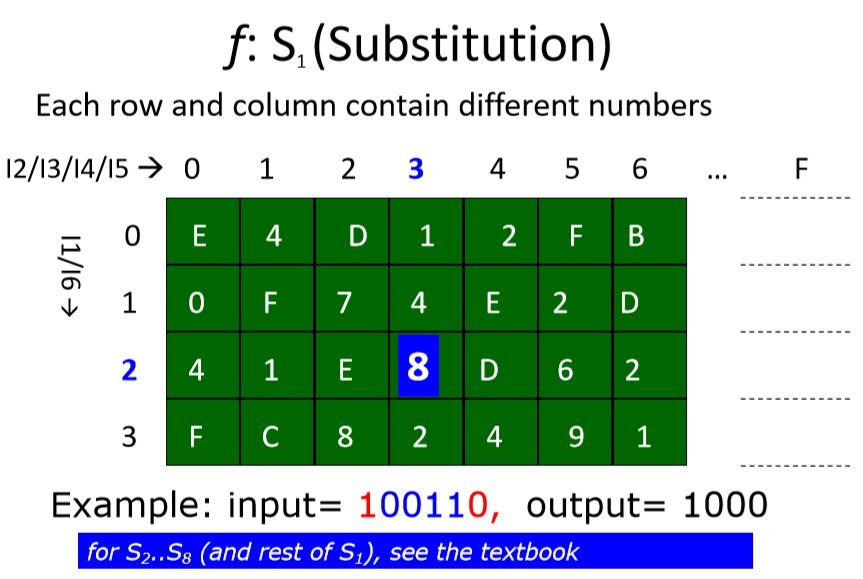
final permutation = inverse(Initial permutation)



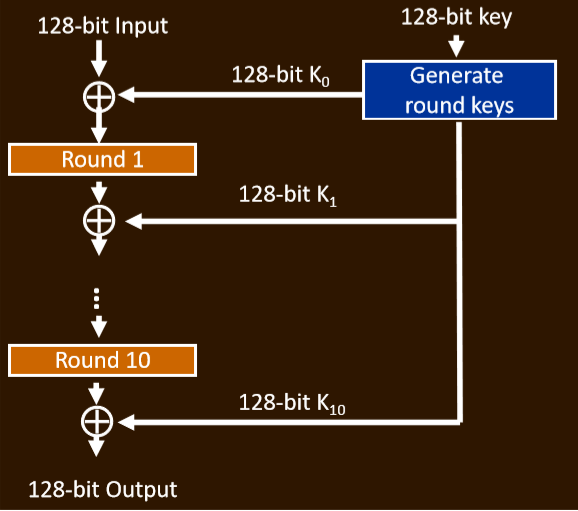


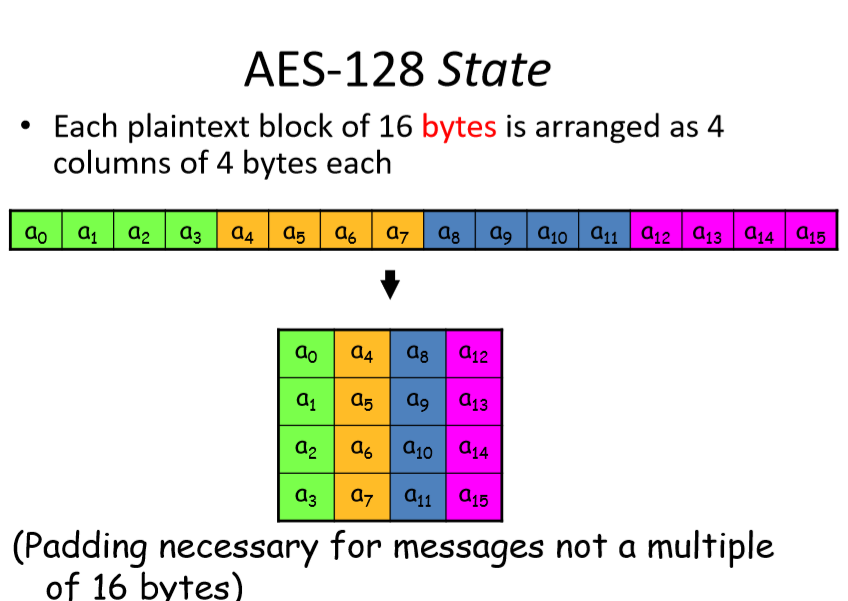


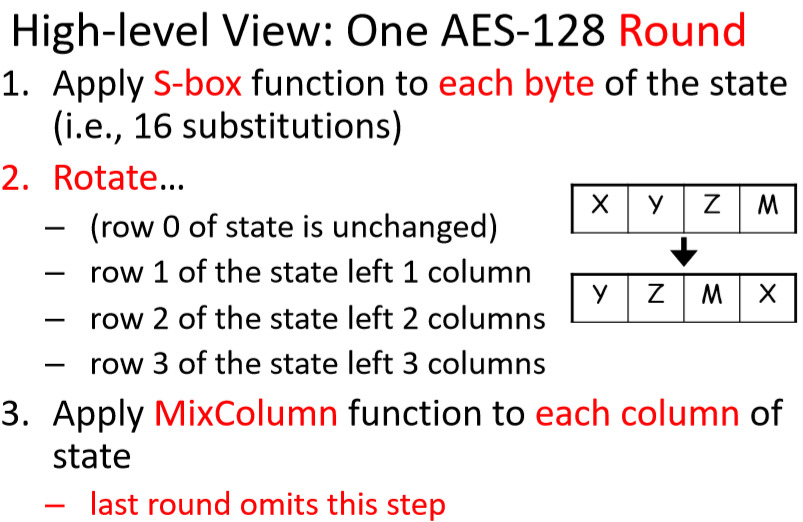




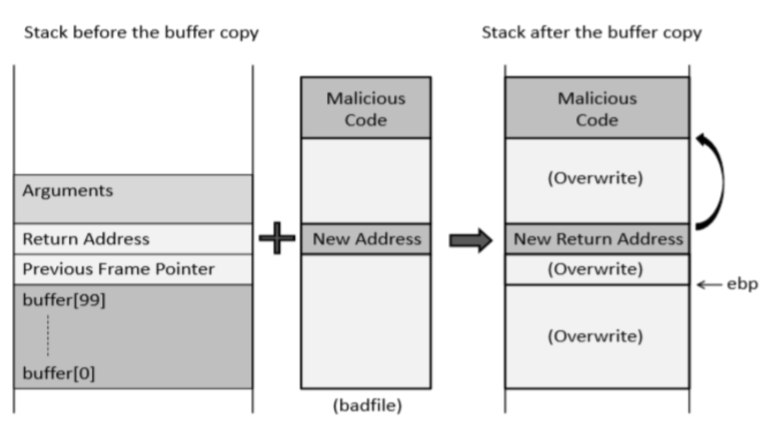
AES:







Buffer overflow attack:



$ebp - $esp + 0x4 = new return address

$ebp: the address of the frame pointer

$esp: the address of stack pointer

Prevented by:

Address Space Layout Space Layout Randomization (ASLR) is primarily used to protect against buffer overflow attacks by randomizing the locations of different memory registers such as $esp and $ebp

enable “Stack Guard” protection mechanism prevents changes to active return addresses by either detecting the change of the return address before the function returns or by completely preventing the write to the return address