Topic 2.1 - Symmetric Encryption

Eric Casanovas

Universitat d'Andorra

16th February, 2022

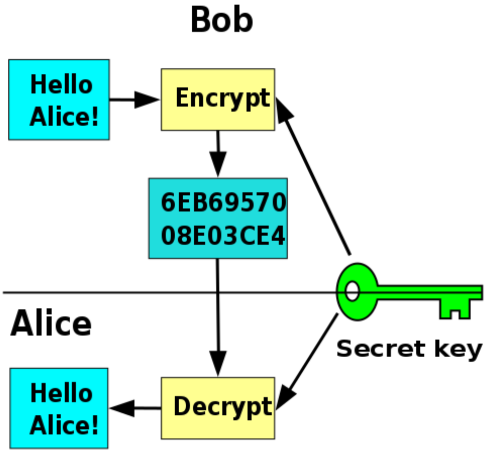
1. Basis of Symmetric key
2. Stream Ciphers
3. RC4
4. Block Ciphers
5. DES
6. AES

1. Basis of Symmetric key
2. Stream Ciphers
3. RC4
4. Block Ciphers
5. DES
6. AES

Basis I

- The key used to encrypt and decrypt are **the same**
- Then who knows the key, knows the message
- The key should be known by sender and receiver
- 2 approaches:
  - Stream ciphers: digits encrypted 1 at a time
  - Block ciphers: encryption of fixed group of bits, called blocks

Basis II


Bob
Hello Alice!
Encrypt
6EB69570
08E03CE4
Secret key
Alice
Hello Alice!
Decrypt

1 Basis of Symmetric key

2 Stream Ciphers

3 RC4

4 Block Ciphers

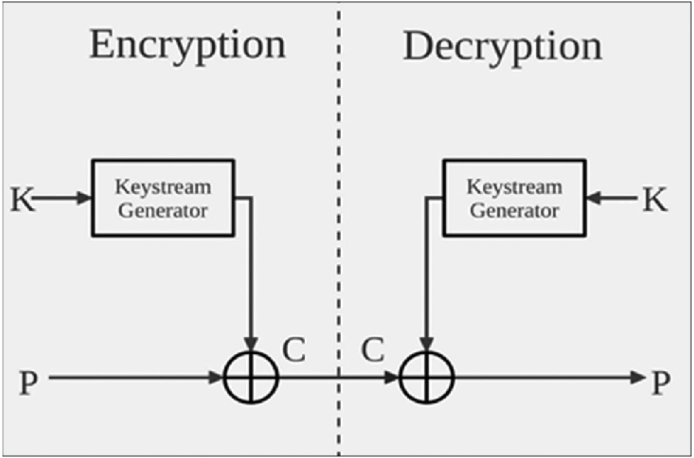
5 DES

6 AES

Stream Ciphers I

- Plaintext digits are combined with a pseudorandom cipher digit stream (keystream)
- Each plaintext digit is encrypted one at a time with the corresponding digit of the keystream, to give a digit of the ciphertext stream
- A digit is typically a bit and the combining operation an XOR
- Examples:
  - **Vernam's cipher**
  - **RC4**
  - Rabbit
  - Salsa20
  - CHACHA

Stream Ciphers II



Vernams cipher I

- Gilbert Vernam invented it in 1917 while working at AT&T
- The key is the same length as the message
- The key is XORed with the message.

Vernams cipher II

- $Enc(K, M) = M \bigoplus K = C$
- $Dec(K, C) = C \bigoplus K = M \bigoplus K \bigoplus K = M$

$\bigoplus$	0	1
0	0	1
1	1	0

```
  M   00101101          C   10001111
  K   10100010          K   10100010
⊕ ------------        ⊕ ------------
  C   10001111          M   00101101
```

- CARE: $M \bigoplus C = M \bigoplus M \bigoplus K = K$

Vernams cipher III

- Does it remind you in something seen so far?
- Do you think that this algorithm is secure?

Vernams cipher IV

- Does it remind you in something seen so far?
- Yes, it's a One Time Pad (OTP)
- Do you think that this algorithm is secure?
- Yes, it is secure and was demonstrated in 1949 by Claude E. Shannon (one of the fathers of the modern cryptography. But only secure if, and only if:
  - The key is truly random
  - Only sender and receiver knows the key
  - The key is not reused
  - The key is destroyed after use
- So why it is not used today?

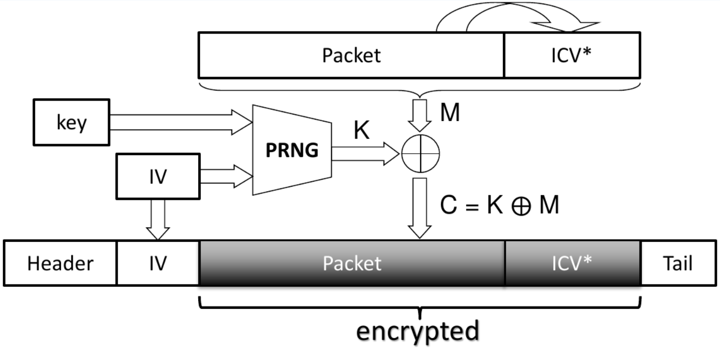
Vernams cipher V

- So why it is not used today?
- The key (keystream) is as long as the message
- Very difficult to distribute and manage the keys
- To send a message of 1GB you need to share a 1GB key

Stream Ciphers security I

- Instead of a key of the same length of the message, a stream cipher makes use of a key of e.g. 64, 128, 256 bits.
- Based on this key, it generates a pseudorandom keystream (e.g. using IV)
- The keystream is now pseudorandom and so is not truly random
- So, the proof of security associated with the OTP no longer holds
- However, we can consider "secure" if keystream:
  - is as random as possible
  - has large period
  - NOT reused

Stream Ciphers security II



- A PRNG can be a hash function
- Do not reuse IVs to increase randomness
- ICV: Integrity Check Value (for data integrity and proof of decryption)

Stream Ciphers security III

IN SUM:

- They are secure?
- They have not been studied in depth to conclude that they are secure
- Block ciphers are more studied, better studied and prefered today

1. Basis of Symmetric key
2. Stream Ciphers
3. RC4
4. Block Ciphers
5. DES
6. AES

What is RC4 I

- Stream cipher designed in 1987 by Ron Rivest
- 2 parts:
  - KSA: Key Scheduling Algorithm
  - PRGA: PseudoRandom Generator Algorithm
- Multiple vulnerabilities have been discovered
- Keys from 40-128 bits
- No specified way to use IV
- Many real systems using RC4 have been attacked
- Used in:
  - Secure Socket Layer (SSL)/ Transport Layer Security (TLS) protocols
  - IEEE 802.11 wireless LAN standard
  - WEP

How it works? I

- 4 steps:
  - Initialize the arrays
  - Run KSA on them
  - Run PRGA on KSA output to generate keystream
  - XOR data with keystream

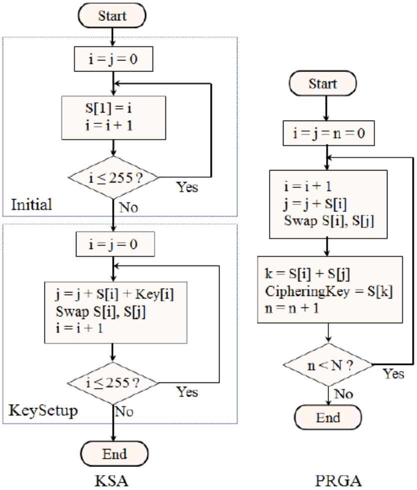
How it works? II

How RC4 works? (Spanish)

OR

Page 48 of the book *Network Security Essentials: Applications and Standards (Fourth edition)*

How it works? III



1. Basis of Symmetric key
2. Stream Ciphers
3. RC4
4. Block Ciphers
5. DES
6. AES

Concepts

- Padding
- IVs

Padding

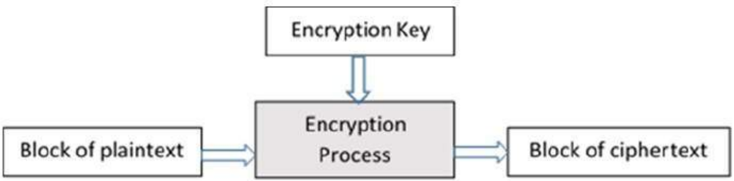
- If the last message block is not complete, a bit string is appended such that:
  - It has the minimum possible length
  - It doesn't introduce any ambiguity in decryption

IVs

- They are block of bits used to randomize the encryption
- Thanks to it we can produce distinct ciphertexts even if the same plaintext is encrypted multiple times
- IV usually does not need to be secret
- Not recommended to reuse IV with the same key

Block Ciphers I

- Process plaintext in blocks of **n** bits
- Produces a block of ciphertext of equal size for each plaintext block
- Examples:
  - DES: k = 64 bits
  - 3DES: k = 64 bits
  - AES: k = 128, 192, 256 bits

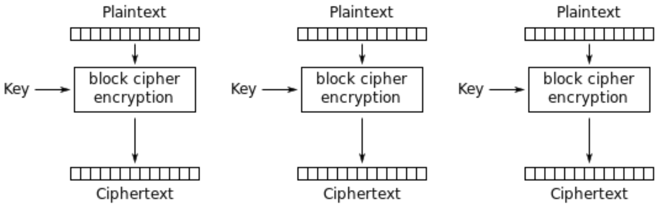


Block Ciphers II

- Stream ciphers can be used for encryption only
- However, block ciphers can be used for:
  - Stream ciphers
  - PRNG
  - Hash functions
  - MACs (Message Authentication Codes)
- Block ciphers also have modes of operation:
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC)
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
  - Counter Mode (CTR)

ECB I

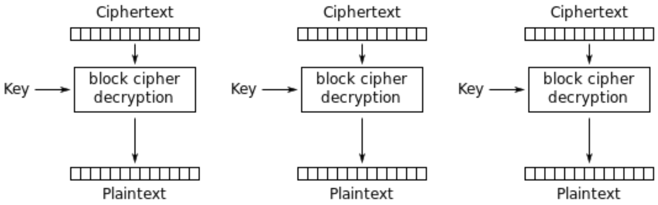
- ECB Encryption:
  $c_i = Enc(k, m_i)$



Electronic Codebook (ECB) mode encryption

ECB II

- ECB Decryption:
  $m_i = Dec(k, c_i)$



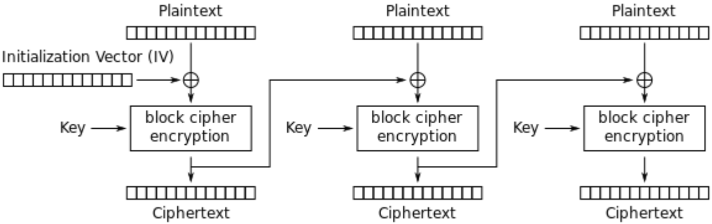
Electronic Codebook (ECB) mode decryption

ECB III

- Required padding
- $m_1 = m_2 \rightarrow c_1 = c_2$
- Not recommended for use in cryptographic protocols
- Encryption -> Parallelizable
- Decryption -> Parallelizable

CBC I

- CBC Encryption:
  $c_0 = iv$
  $c_i = Enc(k, m_i \bigoplus c_{i-1})$



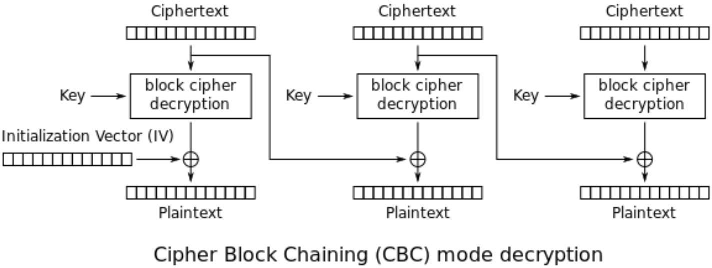
Cipher Block Chaining (CBC) mode encryption

CBC II

- CBC Decryption:

  $c_0 = iv$

  $m_i = Dec(c_i) \bigoplus c_{i-1}$

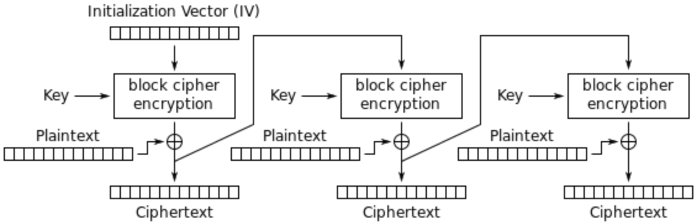


CBC III

- Required padding
- Required IV
- Each ciphertext requires the last processed plaintext
- Encryption -> Not Parallelizable
- Decryption -> Parallelizable

CFB I

- CFB Encryption:
  $c_0 = iv$
  $c_i = Enc(k, c_{i-1}) \bigoplus m_i$



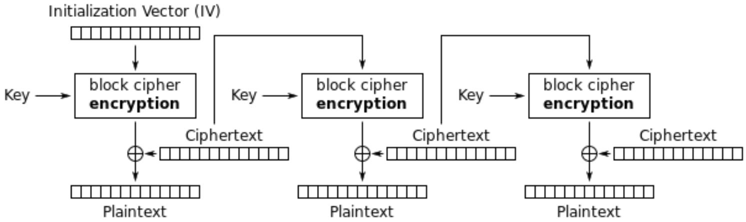
Cipher Feedback (CFB) mode encryption

CFB II

- CFB Decryption:

  $c_0 = iv$

  $m_i = \mathbf{Enc}(c_{i-1}) \bigoplus c_i$



Cipher Feedback (CFB) mode decryption

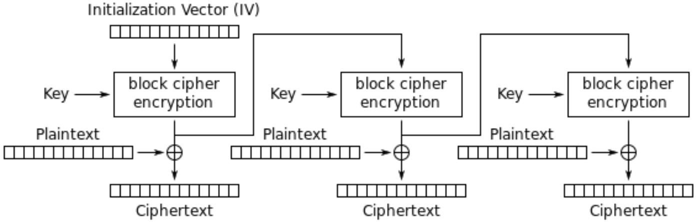
CFB III

- Not necessary padding
- Required IV
- Similar CBC
- Encryption and Decryption uses the same function
- Encryption -> Not Parallelizable
- Decryption -> Parallelizable

OFB I

- OFB Encryption:

$$r_0 = iv$$
$$r_i = Enc(k, r_{i-1})$$
$$c_i = m_i \bigoplus r_i$$



Output Feedback (OFB) mode encryption

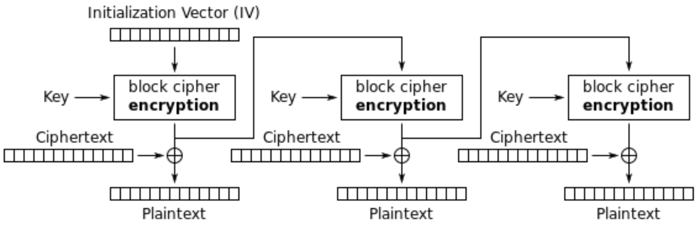
OFB II

- OFB Decryption:

$r_0 = iv$

$r_i = Enc(k, r_{i-1})$

$m_i = c_i \bigoplus r_i$



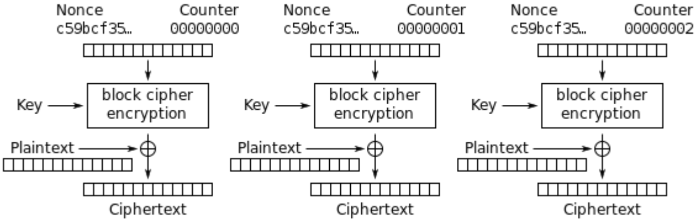
Output Feedback (OFB) mode decryption

OFB III

- Not necessary padding
- Required IV
- Encryption and Decryption uses the same function
- Encryption -> Not Parallelizable
- Decryption -> Not Parallelizable

CTR I

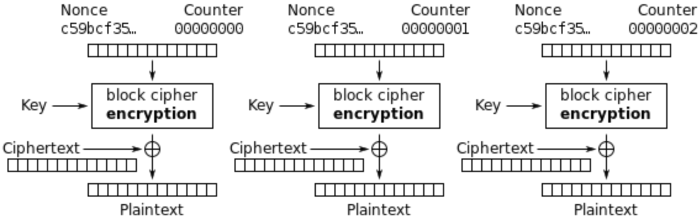
- CTR Encryption:
  $r_i = Enc(k, iv + i)$
  $c_i = m_i \bigoplus r_i$



Counter (CTR) mode encryption

CTR II

- CTR Decryption:
  $r_i = Enc(k, iv + i)$
  $m_i = c_i \bigoplus r_i$

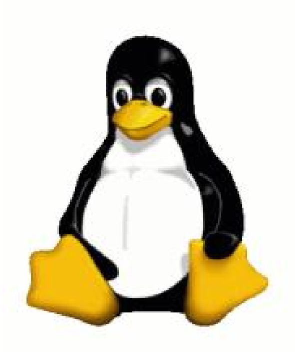


Counter (CTR) mode decryption

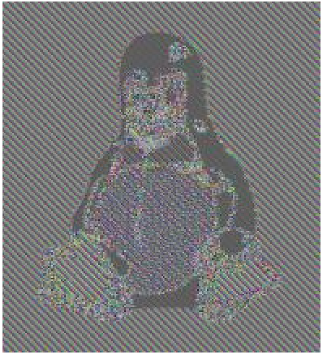
CTR III

- Not necessary padding
- Required IV
- Public nonce and "counter" to increase randomness
- Encryption and Decryption uses the same function
- Encryption -> Parallelizable
- Decryption -> Parallelizable

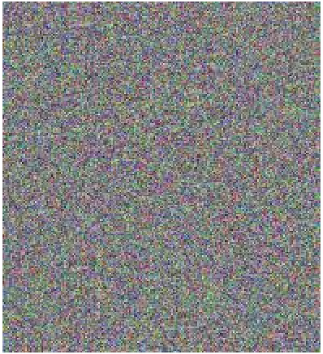
Operation Mode Comparison I



(a) Our god Tux



(b) Tux in ECB



(c) Tux in other modes

Operation Mode Comparison II

- CFB, OFB and CTR require only the encryption box (even for decryption)
- CTR and ECB are parallelizable
- Only ECB and CBC require padding of the last incomplete block

1. Basis of Symmetric key
2. Stream Ciphers
3. RC4
4. Block Ciphers
5. DES
6. AES

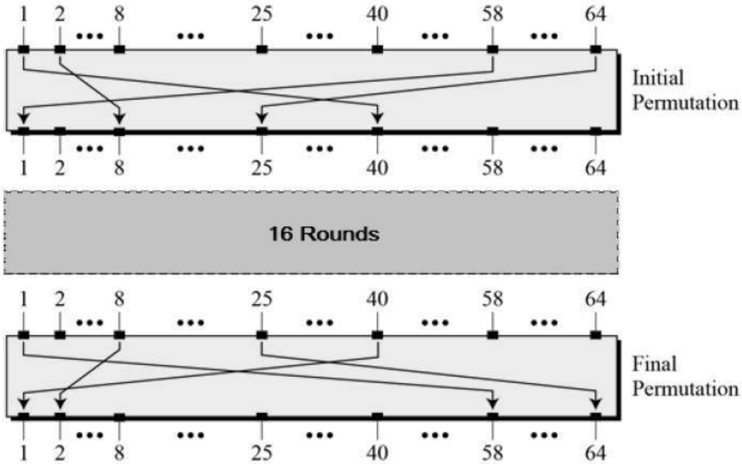
DES

- DES or Data Encryption Standard
- Symmetric key algorithm to encrypt data
- Developed in the 1970s at IBM
- In 1998 Electronic Frontier Foundation built a computer that broke DES in 3 days
- Block cipher of 64 bits (b4 bits per block)
- 56 bits key
- We can use any mode of operation

How DES works I

1. Initial permutation
2. 16 rounds of of Feistel functions
3. Final permutation (inverse of the initial)

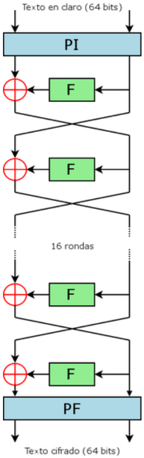
How DES works II



How DES works III

- In the 16 rounds:

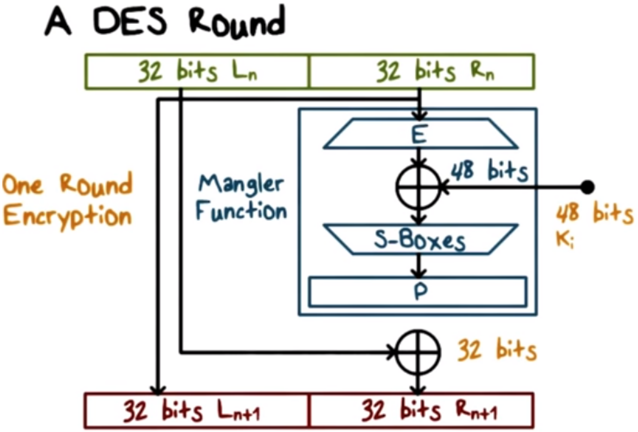
How DES works IV



How DES works V

- The feistel (F) function:

How DES works VI



How DES works VII

- In order to decrypt you have to do the process in reverse order!

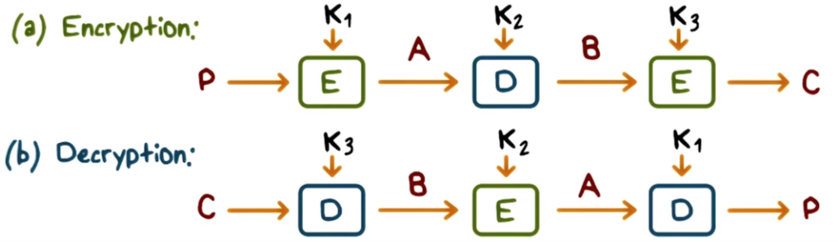
How DES works?

DES Security

- DES is not used today
- Many attacks to the algorithm:
  - Brute Force Attack (trying every combination 2^{64} possible keys)
  - Differential Cryptanalysis (To break the 16 rounds, 2^{49} chosen texts)
  - Linear Cryptanalysis (2^{43} known plaintexts)
- For these reasons DES was imporoved in 1998 to 3DES
- *Differential Cryptanalysis: It studies how differences in information input can affect the resultant difference at the output*
- *Linear Cryptanalysis: It studies the probabilistic linear relations between parity bits of the plaintext, the ciphertext, and the secret key*

3-DES

- Using 3 times DES:
- 3 keys: 3 different (k = 168 bits) or k1 = k3 (k = 112 bits)



1 Basis of Symmetric key

2 Stream Ciphers

3 RC4

4 Block Ciphers

5 DES

6 AES

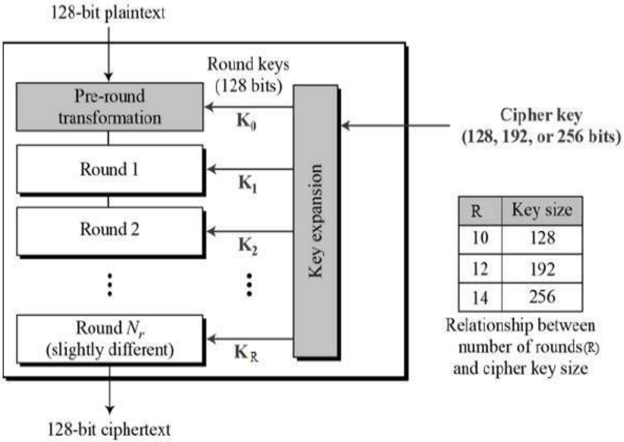
AES

- AES or Advanced Encryption Standard
- Symmetric key algorithm to encrypt data
- Developed in 2001 by NIST (National Institute of Standards Technology)
- Block cipher of 128 bits (128 blocks per bit)
- 128/192/256 bits key
- we can use any mode of operation

How AES works? I

- Algorithm description:
  1. KeyExpansion
  2. Pre-round transformation: AddRoundKey
  3. 9, 11 or 13 rounds (depending on key length 128, 192 or 256)
     1. SubBytes
     2. ShiftRows
     3. MixColumns
     4. AddRoundKey
  4. Final round:
     1. SubBytes
     2. ShiftRows
     3. AddRoundKey

How AES works? II

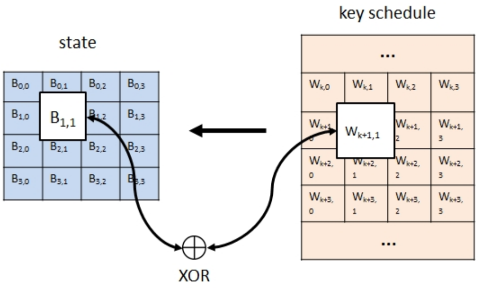


R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

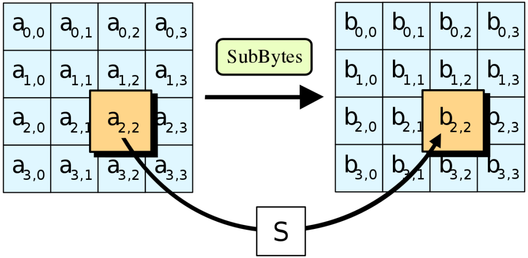
How AES works? III

- AddRoundKey



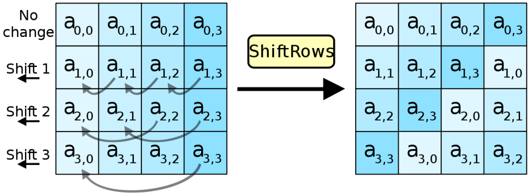
How AES works? IV

- SubBytes



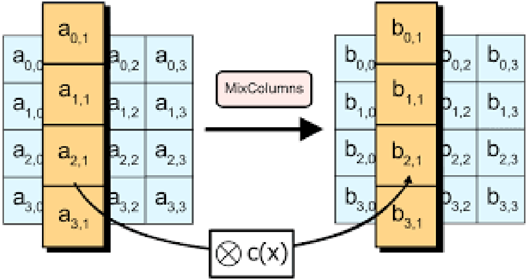
How AES works? V

- ShiftRows

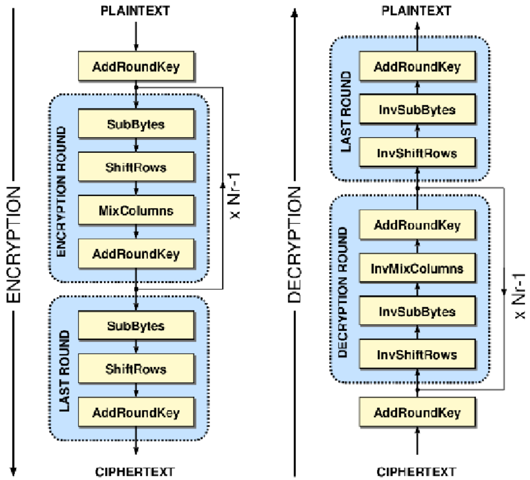


How AES works? VI

- MixColumns



How AES works? VII



How AES works? VIII

How AES works?

AES Security

- There are Side-channel attacks
- There are no known practical cryptanalytic attacks
- Today we can consider AES secure
- *side-channel attack: Is an attack based on information gained from the implementation of a computer system, rather than weaknesses in the implemented algorithm itself. This information can be timing information, power consumption, electromagnetic leaks or even sound*

The END!