Slayt1

Passive Attack örneği: Mesajı gizlice okumak

Active Attack örneği: Mesajı gizlice okuyup değiştirmek

Properties of Security Services

•Authentication-assurance that the communicating entity is the one claimed

•Access Control-prevention of the unauthorized use of a resource

•Data Confidentiality–protection of data from unauthorized disclosure

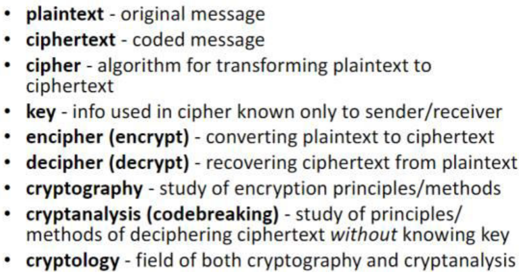
•Data Integrity-assurance that data received is as sent by an authorized entity

•Non-Repudiation-protection against denial by one of the parties in a communication

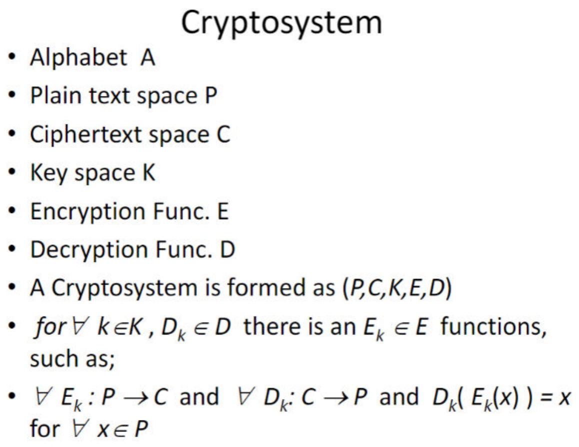
Slayt2

Some Attack types on Network

- Disclosure (ifşaat)
- Traffic Analysis
- Masquerade (Gerçeği gizleme)
- Content Modification
- Sequence Modification (sırayı değiştirme)
- Timing Modification
- Repudation (inkarcılık)



- **plaintext** - original message
- **ciphertext** - coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering ciphertext from plaintext
- **cryptography** - study of encryption principles/methods
- **cryptanalysis (codebreaking)** - study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - field of both cryptography and cryptanalysis

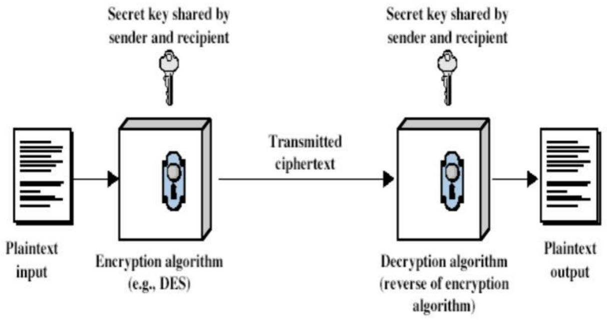


Cryptosystem

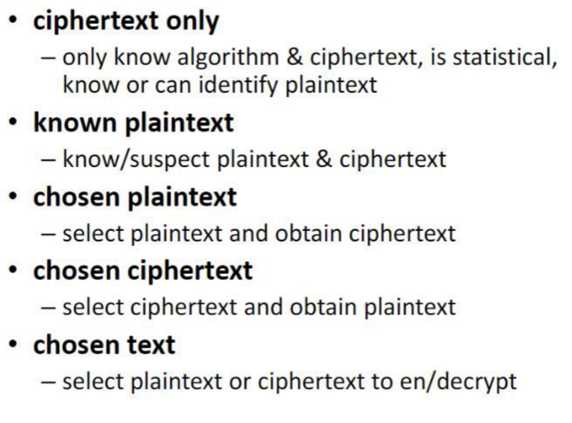
- Alphabet A
- Plain text space P
- Ciphertext space C
- Key space K
- Encryption Func. E
- Decryption Func. D
- A Cryptosystem is formed as (*P,C,K,E,D*)
- $for \forall\ k \in K$, $D_k \in D$ there is an $E_k \in E$ functions, such as;
- $\forall\ E_k : P \rightarrow C$ and $\forall\ D_k : C \rightarrow P$ and $D_k(\ E_k(x)\ ) = x$ for $\forall\ x \in P$

Symmetric Encryption (conventional Encryption, private-key encryption, single key encryption)

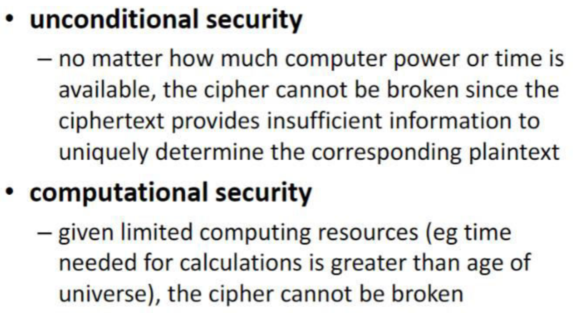
Sender and recipient share a common key.


Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Cryptanalytic attacks



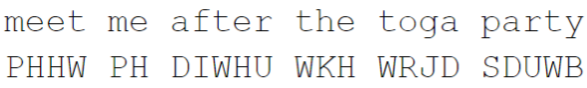
- **ciphertext only**
  - only know algorithm & ciphertext, is statistical, know or can identify plaintext
- **known plaintext**
  - know/suspect plaintext & ciphertext
- **chosen plaintext**
  - select plaintext and obtain ciphertext
- **chosen ciphertext**
  - select ciphertext and obtain plaintext
- **chosen text**
  - select plaintext or ciphertext to en/decrypt



- **unconditional security**
  - no matter how much computer power or time is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext
- **computational security**
  - given limited computing resources (eg time needed for calculations is greater than age of universe), the cipher cannot be broken

Substitution Ciphers: Letters of plaintext are replaced by other letters

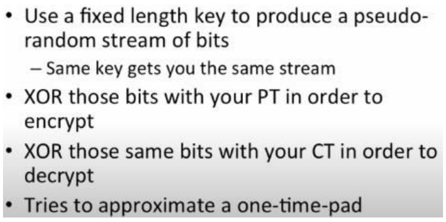
Earliest known substitution cipher is Caesar cipher



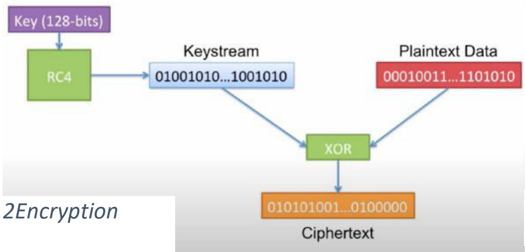
```
meet me after the toga party
PHHW PH DIWHU WKH WRJD SDUWB
```

Symmetric Encryption Types

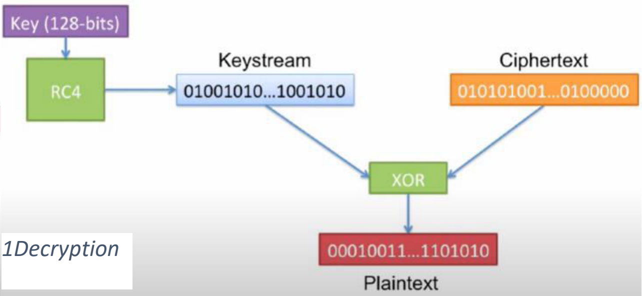
1) Stream Cipher Encryption



- Use a fixed length key to produce a pseudo-random stream of bits
  - Same key gets you the same stream
- XOR those bits with your PT in order to encrypt
- XOR those same bits with your CT in order to decrypt
- Tries to approximate a one-time-pad

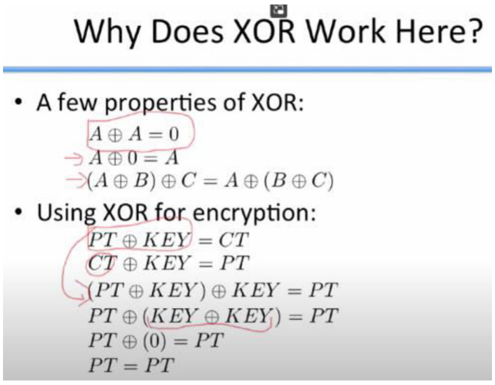


2Encryption



1Decryption

Keystream is pseudo-random



Why Does XOR Work Here?

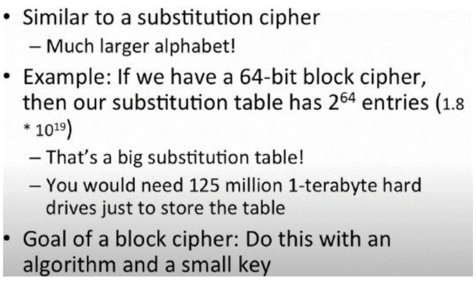
- A few properties of XOR:

$$A \oplus A = 0$$
$$A \oplus 0 = A$$
$$(A \oplus B) \oplus C = A \oplus (B \oplus C)$$

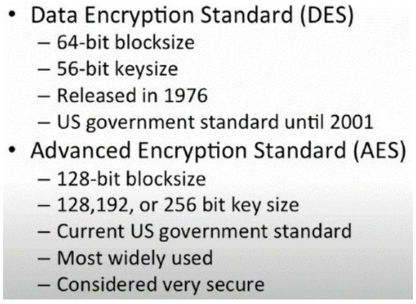
- Using XOR for encryption:

$$PT \oplus KEY = CT$$
$$CT \oplus KEY = PT$$
$$(PT \oplus KEY) \oplus KEY = PT$$
$$PT \oplus (\underbrace{KEY \oplus KEY}) = PT$$
$$PT \oplus (0) = PT$$
$$PT = PT$$

2) Block Cipher Encryption



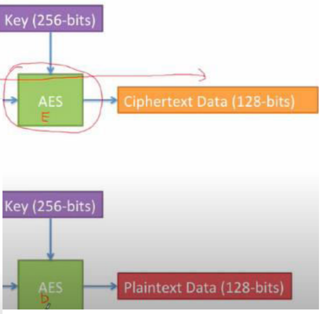
- Similar to a substitution cipher
  - Much larger alphabet!
- Example: If we have a 64-bit block cipher, then our substitution table has 2^{64} entries ($1.8 * 10^{19}$)
  - That's a big substitution table!
  - You would need 125 million 1-terabyte hard drives just to store the table
- Goal of a block cipher: Do this with an algorithm and a small key

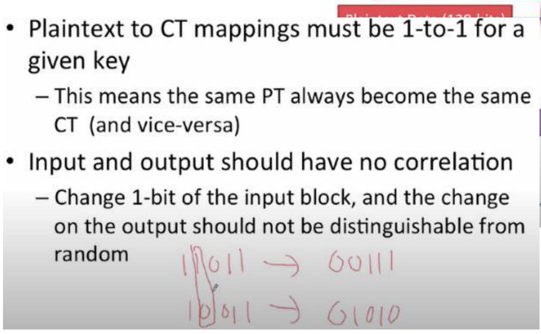


- Data Encryption Standard (DES)
  - 64-bit blocksize
  - 56-bit keysize
  - Released in 1976
  - US government standard until 2001
- Advanced Encryption Standard (AES)
  - 128-bit blocksize
  - 128,192, or 256 bit key size
  - Current US government standard
  - Most widely used
  - Considered very secure

Algorithm has two modes (encrypt, decrypt) and you have to specify which one are you doing

- Encryption:





- Plaintext to CT mappings must be 1-to-1 for a given key
  - This means the same PT always become the same CT (and vice-versa)
- Input and output should have no correlation
  - Change 1-bit of the input block, and the change on the output should not be distinguishable from random

11011 → 00111

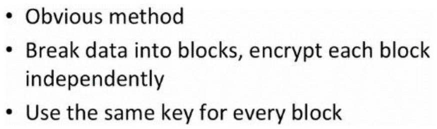
10011 → 01010

Block size and key size increases: more secure,

but slower. Encrypts a fixed length block at a time.

Operating Modes of Block Cipher

1) Electronic Code Book (ECB)

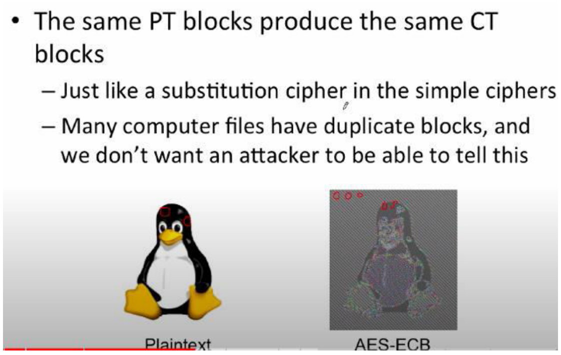


- Obvious method
- Break data into blocks, encrypt each block independently
- Use the same key for every block

Parallel encryption of blocks of bits is possible,

thus, it is a faster way of encryption.

Problem of ECB:



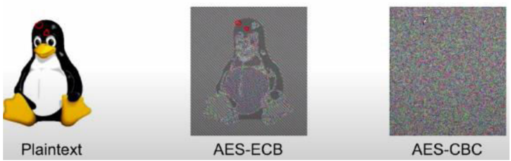
- The same PT blocks produce the same CT blocks
  - – Just like a substitution cipher in the simple ciphers
  - – Many computer files have duplicate blocks, and we don't want an attacker to be able to tell this

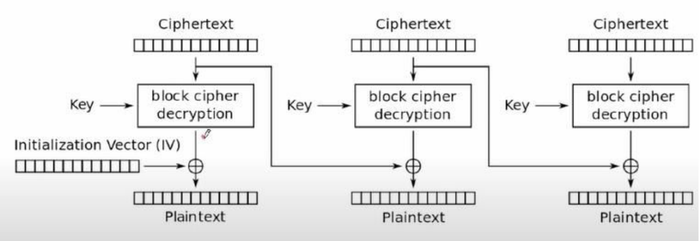
2) Cipher Block Chaining (CBC)

Each block is dependent on the previous one. **This fixes many of the problem with ECB**.

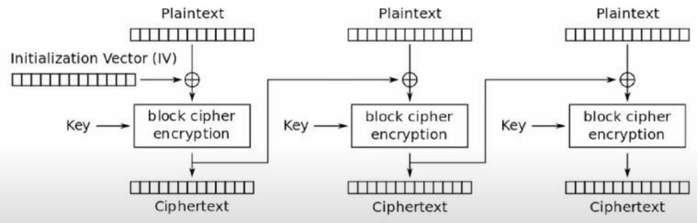
Each plaintext is XORed with previous ciphertext

Initialization vector is a randomly chosen number, not a secret.



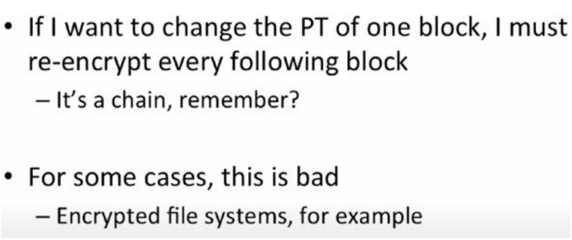


Cipher Block Chaining (CBC) mode decryption



Cipher Block Chaining (CBC) mode encryption

Problem of CBC



- If I want to change the PT of one block, I must re-encrypt every following block
  – It's a chain, remember?
- For some cases, this is bad
  – Encrypted file systems, for example

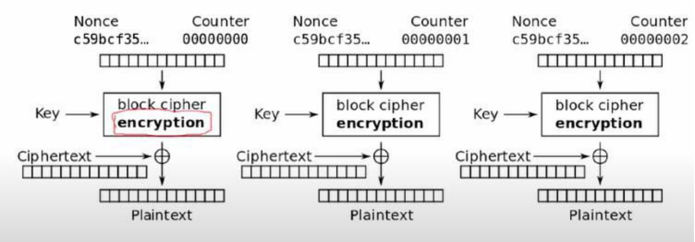
3) Counter (CTR)

Simulates a stream cipher

Each block is encrypted independently, but it involves an incrementing nonce

Nonce: a randomly chosen number, not a secret (same as initialization vector of CBC).

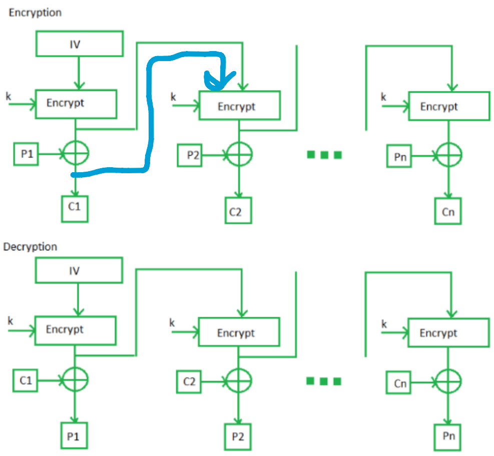
Encrypt nonce with key and XOR it with plaintext. Increment nonce each time.



Counter (CTR) mode decryption

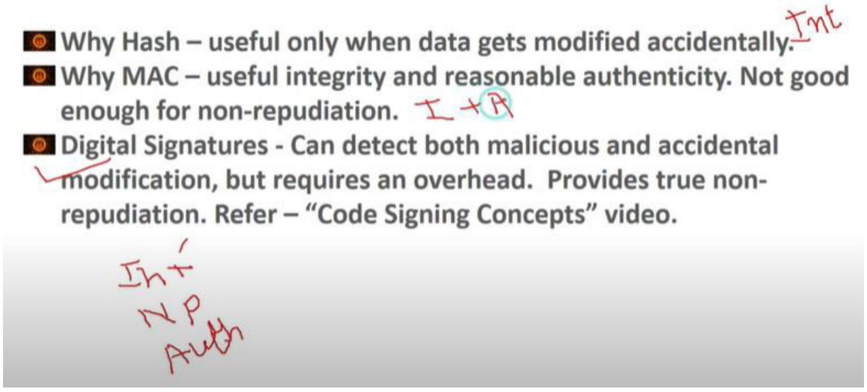
Using encryption mode again!

4) Output Feedback Mode



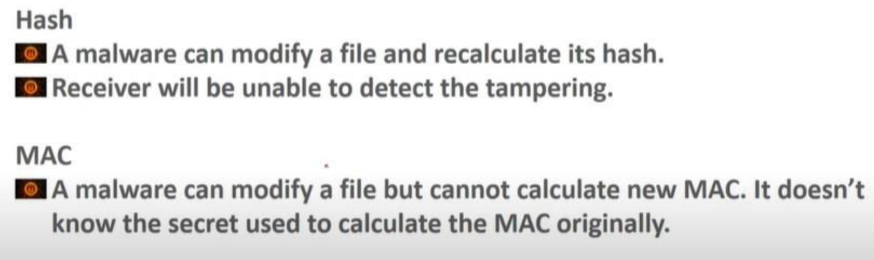
Cipher feedback mode'un output feedback mode'dan tek farkı xor'dan sonraki basamak sonraki adıma gidiyo (mavi ile çizdim)

Hash vs Mac



- Why Hash – useful only when data gets modified accidentally. Int
- Why MAC – useful integrity and reasonable authenticity. Not good enough for non-repudiation. I + A
- Digital Signatures - Can detect both malicious and accidental modification, but requires an overhead. Provides true non-repudiation. Refer – "Code Signing Concepts" video.

Int
NP
Auth



Hash

- A malware can modify a file and recalculate its hash.
- Receiver will be unable to detect the tampering.

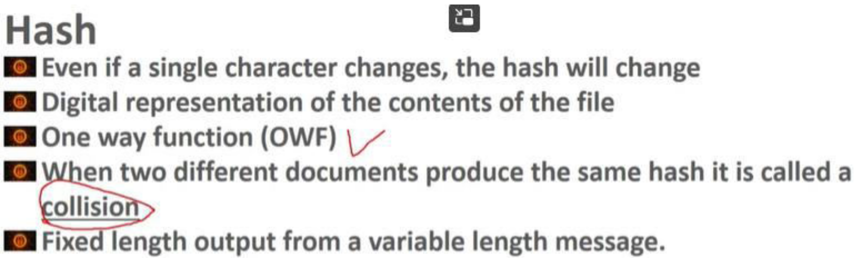
MAC

- A malware can modify a file but cannot calculate new MAC. It doesn't know the secret used to calculate the MAC originally.

Hash: One Way Func

Mac: Key based auth code

Mac is resistant to man in the middle attack, hash is not.



Hash

- Even if a single character changes, the hash will change
- Digital representation of the contents of the file
- One way function (OWF) ✓
- When two different documents produce the same hash it is called a collision
- Fixed length output from a variable length message.

Mac&Hash (MHASH)

It is a message integrity and authenticity method which combines hashing and MAC(message authentication code). It uses two keys to generate authentication code and appends it to the message. [m | h(k2 | (h(k1 | m))]

Diffie-Hellman Key Exchange (Simetrik şifrelemede anahtarı paylaşmak için)

It is used for changing keys in cryptography. Let's think of a scenario where Alice and Bob wish to swap keys

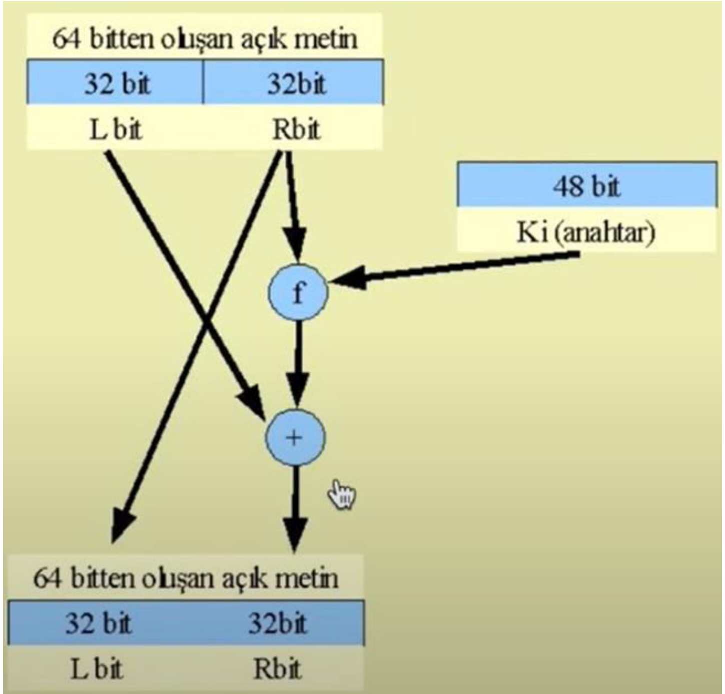
Select two prime numbers, n = 3 and q = 17; these are public to anyone
Alice's private key is p1 = 15
Bob's private key is p2 = 13

Alice sends the number n^p1 mod q = 3^15 mod 17 = 6 to Bob
Bob sends the number n^p1 mod q = 3^13 mod 17 = 12 to Alice

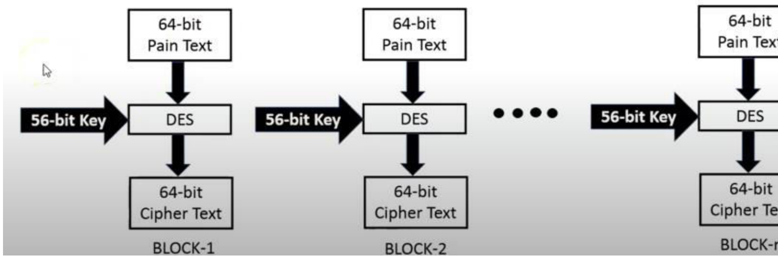
Alice generates the new key with 12^15 mod 17 = 10
Bob generates the new key with 6^13 mod 17 = 10

Since only Alice and Bob know their private key, nobody else can find the new generated key other than them.

DES (Data Encryption Standart)



Symmetric, uses block cipher method for encryption and decryption.

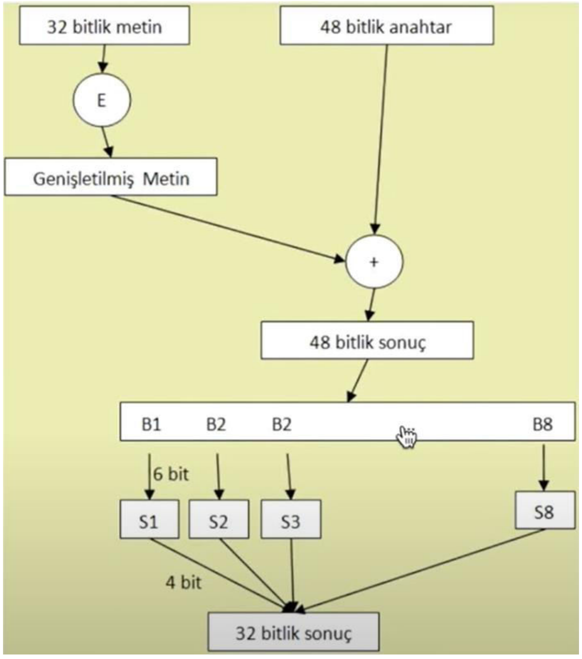


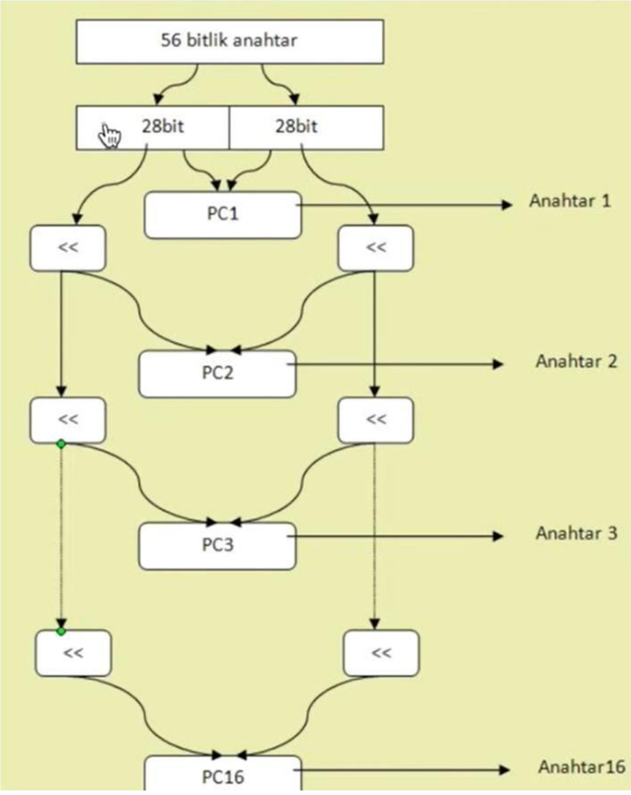
DES'in bir adımı (her 64 bitlik blok için 16 kez tekrarlanıyor.)

Anahtar 64 bit, 8'I parity bit. Kalan 56 bit key generatorden geçiyor, her adım için farklı 48-bit anahtar kullanılmış oluyor.

XOR

F fonksiyonunun içi (E expansion , s substation, expansionun tersini yapıyor.)



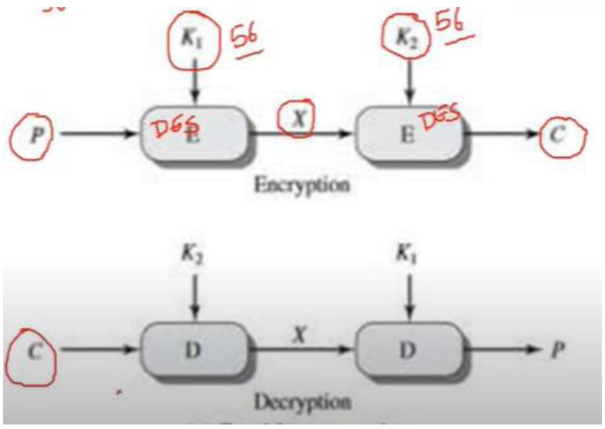


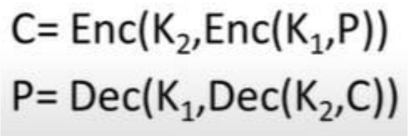
24 bit soldan, 24 bit sağdan

Decryption: ciphertexti aynı algoritmadan aynı anahtarla geçirince decrypt ediyor, anahtarlar tersten kullanılıyor (16,15,14…)

Double DES

DES is vulnerable to brute force attacks since key is 56 bit (2^56 tries)





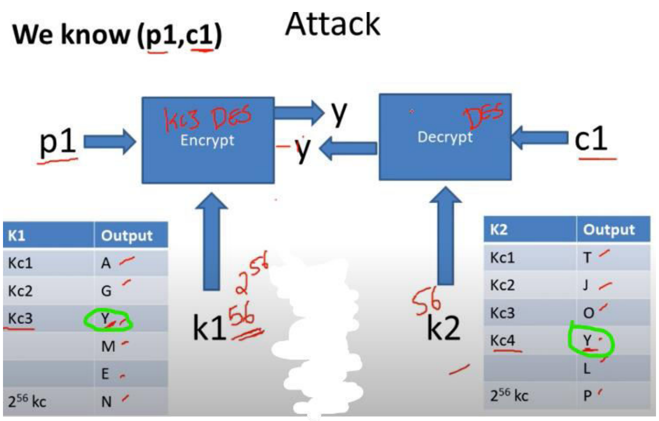
$$C= Enc(K_2,Enc(K_1,P))$$

$$P= Dec(K_1,Dec(K_2,C))$$

Double DES needs 2^(56*2) = 2^112 tries.

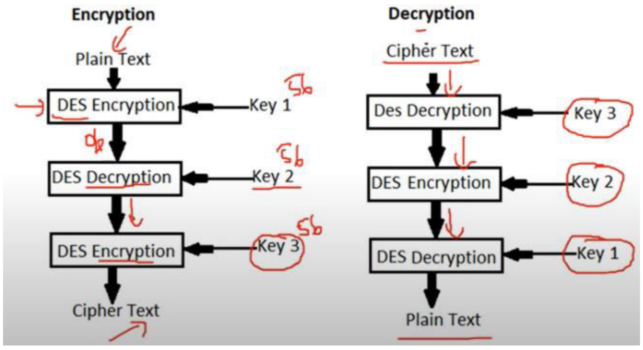
Problem: Meet in the middle attack

If the attacker knows one pair of plaintext and ciphertext, he can try k1-pt combinations on encryption algorithm and try k2-c combinations on decryption algorithm. When there is a match on X, it means cipher is cracked. So still 2^56 tries.



K1	Output
Kc1	A
Kc2	G
Kc3	Y
	M
	E
2^{56} kc	N

K2	Output
Kc1	T
Kc2	J
Kc3	O
Kc4	Y
	L
2^{56} kc	P



ciphertext = Enc(K 3, (Dec(K 2, (Enc(K 1, plaintext)))

Plaintext = Dec(K1,Enc(K2, Dec(K3, Ciphertext)))

Triple DES is secure but slow (3 times slower than DES)

RSA (Açık Anahtarlı bir şifreleme yöntemi)

Yeterince büyük iki asal sayı seçilir (p & q)

n = pq

φ(n) = (p-1)(q-1) (totient fonksiyonu)

φ(n) ile aralarında asal ve 1<e<φ(n) bir e sayısı bulunur. Bu e sayısı = public key

d = 1 mod φ(n) eşitliğini sağlayan bir d sayısı hesaplanır. Bu sayının hesaplanmasında extended Euclid algoritması kullanılır. d private key olarak saklanır.

c = m^e mod n

m = m^d mod n

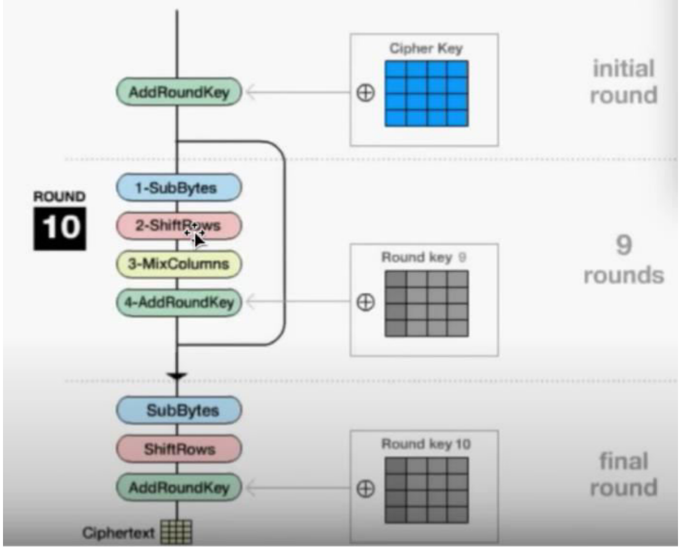
e ve n public.

d is private

AES (Advanced Encryption Standart) (bunu hiç anlamadım)

Security >= triple des, but significantly more efficient. Symmetric block cipher.

Block length = 128 bits, supports key length of 128,192,256

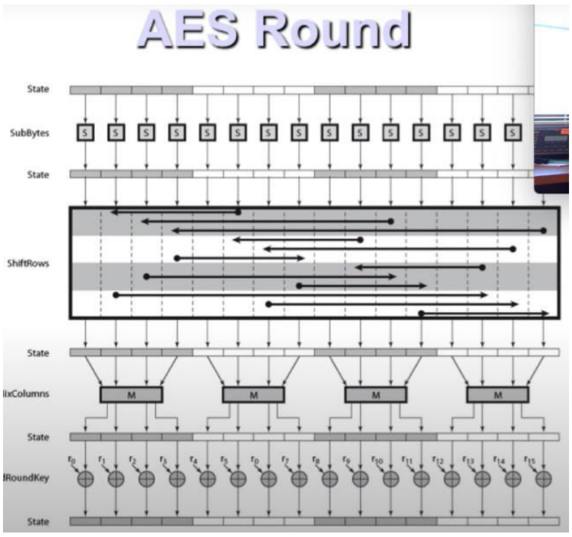


Substitute bytes -> substitution box kullanarak byteların yerini değiştiriyo

Shift rows-> ilk satır 0, ikinci satır 1, üçüncü satır 2… kez sola shift ediliyo. (cyclic şekilde)

Mix columns -> columnları tek tek belirli bir matrisle çarpıyo

Add round key -> her columna round keyle XOR'lanıyo



Avalanche Effect: kriptografik algoritmaların, tipik olarak blok şifrelerin ve kriptografik özet fonksiyonlarının arzu edilen özelliğidir, burada bir girdi biraz değiştirilirse çıktı önemli ölçüde değişir.