CENG 434 Kriptoloji – 4. Ders

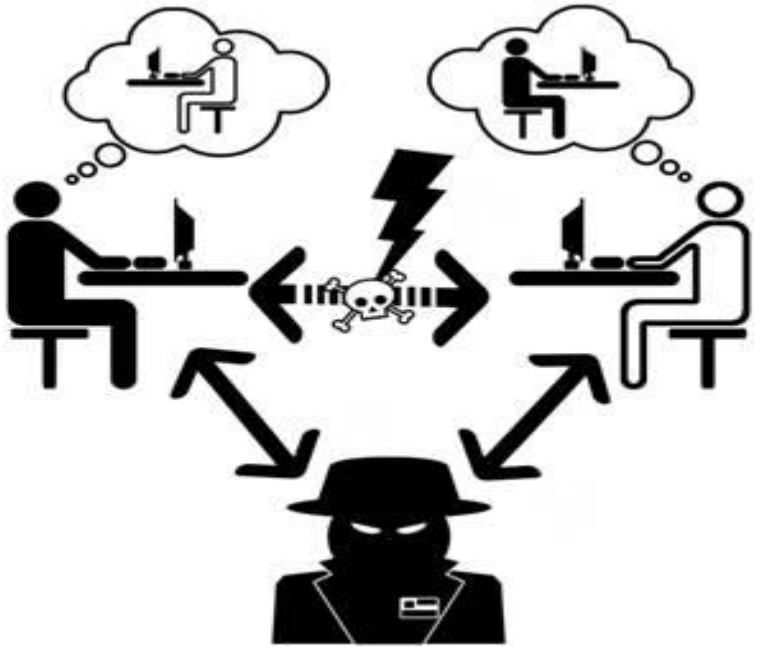
Alper UĞUR

CENG 507 :
KRİPTOGRAFİK ALGORİTMALAR VE SİSTEMLER

CENG 434:
KRİPTOLOJİ

Güvenlik Hizmetleri (Security Services)

- Gizlilik (Confidentiality)
- Bütünlük (Integrity)
- Kimlik doğrulama (Authentication)
- Ulaşılabilirlik (Availability)
- Rededememe (Non-repudiation)



Güvenlik

- Koşulsuz güvenlik
  - One-time pad
- Hesaplamaya bağlı güvenlik
  - Harcadığın emeğe/paraya değmeli
  - Elde ettiğin bilgiye değmeli

Kriptanalizde birkaç adım

- Kaba kuvvet (brute force)

- pin: ****
- Saldırı: 0000 … 9999

Kriptanalizde birkaç adım

- Sıklık analizi

Table 1. Turkish Unigram Frequencies and Replacing Values in Homophonic Cipher

A	%11,92	12	I	%5,114	5	R	%6,722	7
B	%2,844	3	İ	%8,6	9	S	%3,014	3
C	%0,963	1	J	%0,034	1	Ş	%1,78	2
Ç	%1,156	1	K	%4,683	5	T	%3,314	3
D	%4,706	5	L	%5,922	6	U	%3,235	3
E	%8,912	9	M	%3,752	4	Ü	%1,854	2
F	%0,461	1	N	%7,487	7	V	%0,959	1
G	%1,253	1	O	%2,476	2	Y	%3,336	3
Ğ	%1,125	1	Ö	%0,777	1	Z	%1,5	2
H	%1,212	1	P	%0,886	1			

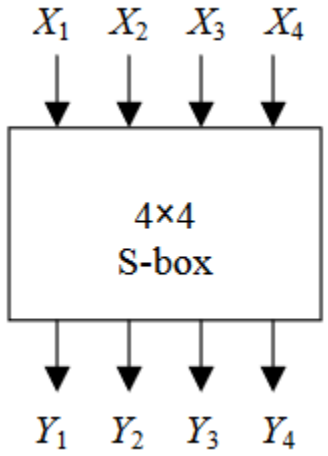
Kaynak: S.I. Serengil, M. Akin. "Attacking Turkish Tests Encrypted by Homophonic Cipher" Proceedings of the 10th WSEAS International Conference on E.H.W.O., pp. 123–126, UK, 2011.

Kriptanalizde birkaç adım

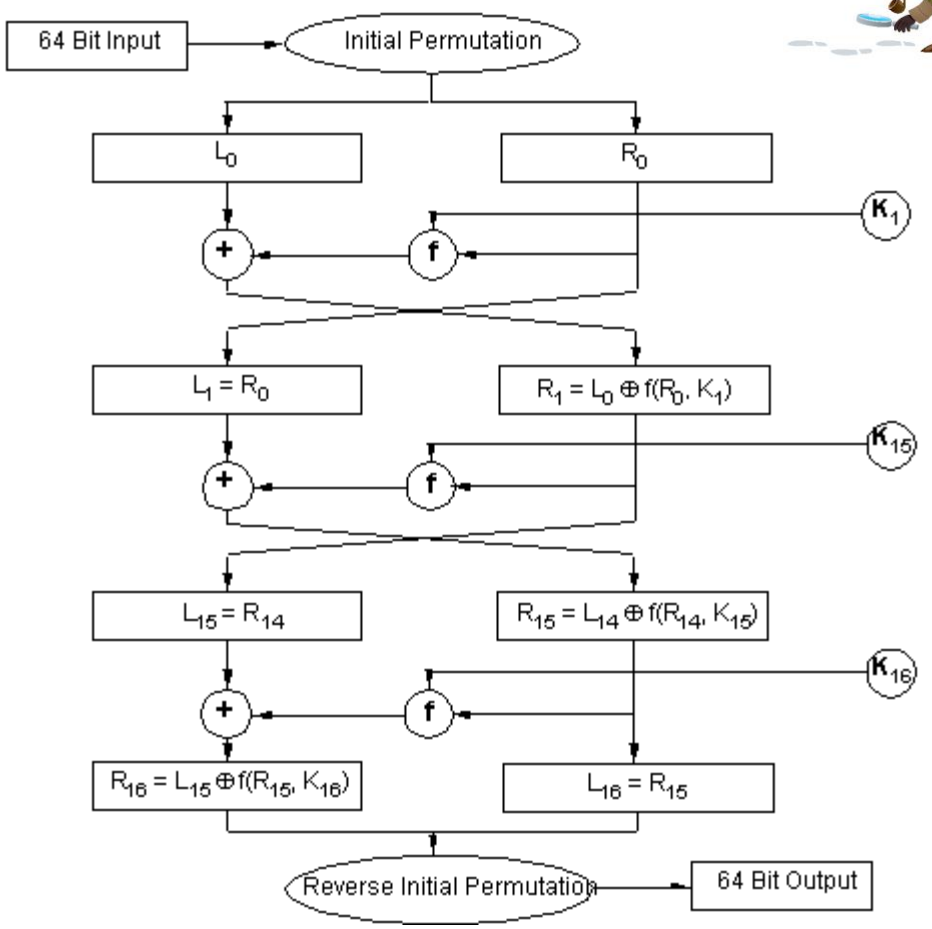
- Doğrusal (linear) kriptanaliz

$$X_{i_1} \oplus X_{i_2} \oplus \ldots \oplus X_{i_u} \oplus Y_{j_1} \oplus Y_{j_2} \oplus \ldots \oplus Y_{j_v} = 0 \qquad (1)$$

where X_i represents the i-th bit of the input $X = [X_1, X_2, \ldots]$ and Y_j represents the j-th bit of the output $Y = [Y_1, Y_2, \ldots]$. This equation is representing the exclusive-OR "sum" of u input bits and v output bits.



$$X_2 \oplus X_3 \oplus Y_1 \oplus Y_3 \oplus Y_4 = 0$$



Kaynak: A Tutorial on Linear and Differential Cryptanalysis by Howard M. Heys

Kriptanalizde birkaç adım

- Doğrusal (linear) kriptanaliz

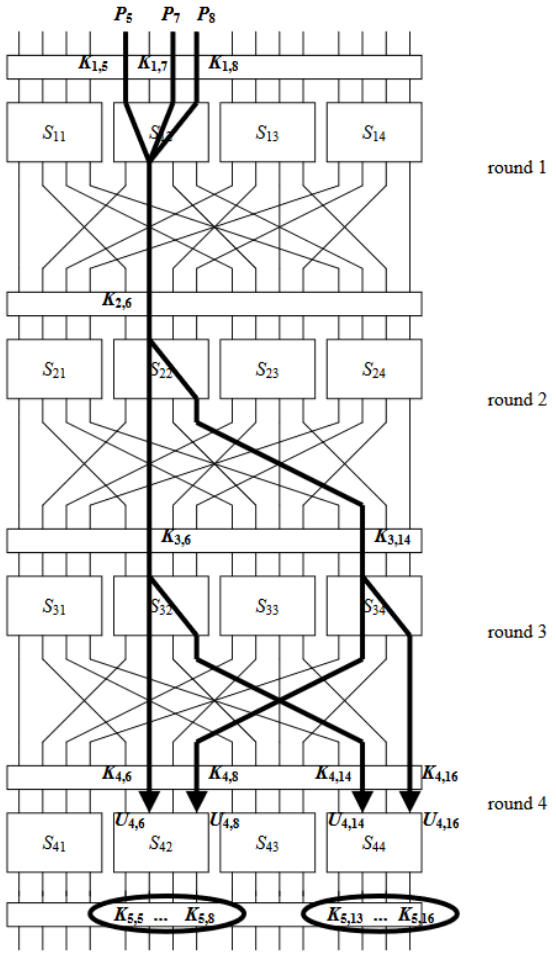


Figure 3. Sample Linear Approximation

Kaynak: A Tutorial on Linear and Differential Cryptanalysis by Howard M. Heys

Kriptanalizde birkaç adım

- Fark (differential)

input $X = [X_1\ X_2\ \ldots\ X_n]$ and output $Y = [Y_1\ Y_2\ \ldots\ Y_n]$.

$$\Delta X = [\Delta X_1\ \Delta X_2\ \ldots \Delta X_n] \qquad \Delta Y = [\Delta Y_1\ \Delta Y_2\ \ldots \Delta Y_n]$$

$$\Delta X_i = X_i' \oplus X_i''$$

s_1	0	1	1	0	0	1	1	0	1	0	0	1	0	0	1	0
s_2	1	1	1	0	0	1	1	0	0	0	0	1	0	0	1	0
Δ	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Kaynak: A Tutorial on Linear and Differential Cryptanalysis by Howard M. Heys

Simetrik Şifreleme

- Şifreleme – Kodlama = Anahtar

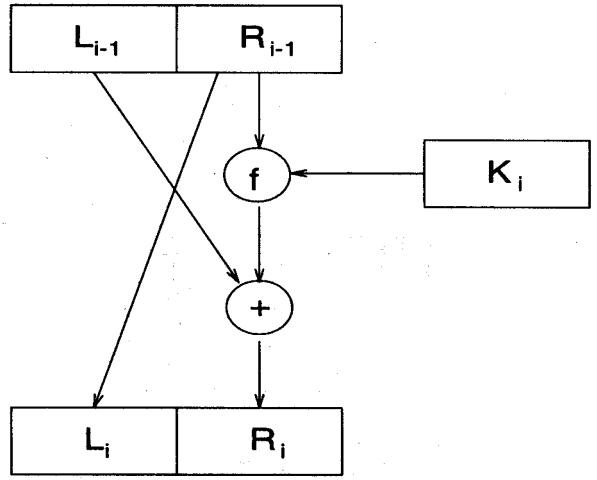
- Simetrik şifreleme gizli anahtar tek

  $C = E_K(P)$

  $P = D_K(C)$

DES (Data Encryption Standard)

- LUCIFER Project



- Blok şifreleme
- 64bitlik bloklar
- 56bitlik anahtar
- 16 çevrim (round)
- Feistel Network
  - Enc: $L_i = R_{i-1}$, $R_i = L_{i-1}$ sOR $f(R_{i-1}, K_i)$
  - Dec: $R_i = L_{i+1}$, $L_i = R_{i-1}$ sOR $f(L_{i+1}, K_i)$

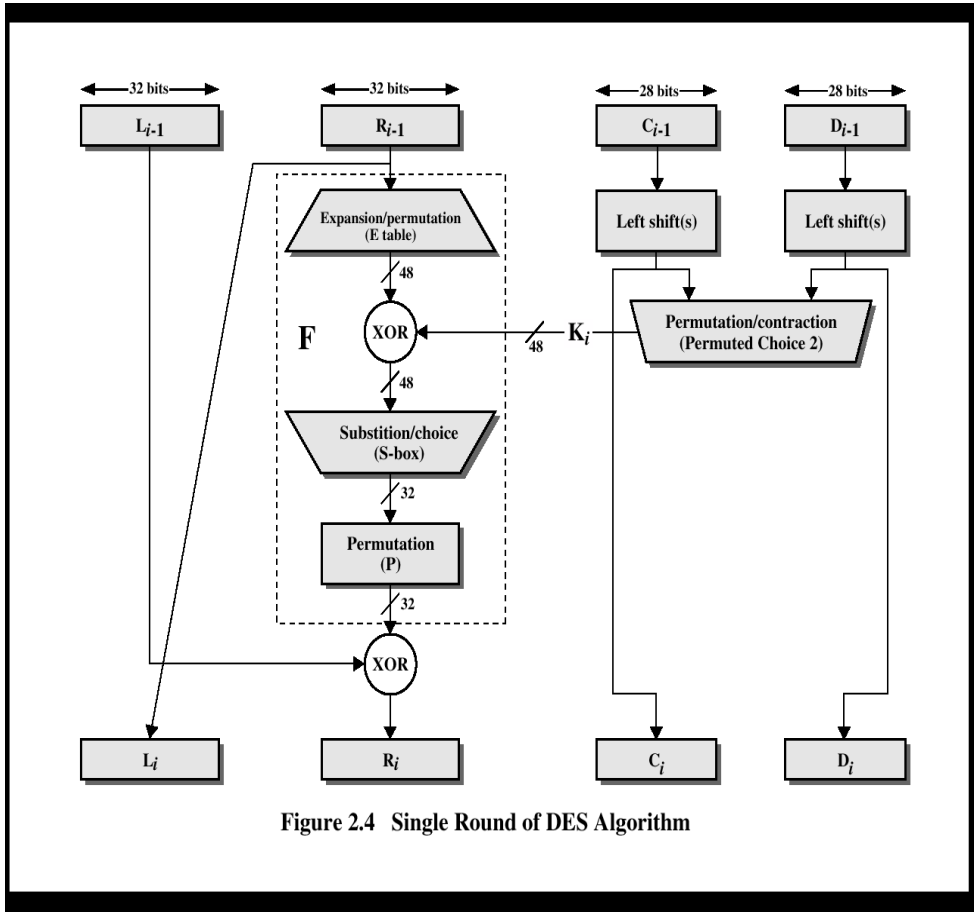


Figure 2.4 Single Round of DES Algorithm

DES güvenlik

- $C = E_K(P)$
- $P = D_K(C)$

zayıf anahtarlar

$E_K(E_K(P))=P$

K= 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
K= F E F E F E F E F E F E F E F E
K= 1 F 1 F 1 F 1 F 0 E 0 E 0 E 0 E
K= E 0 E 0 E 0 E 0 F 1 F 1 F 1 F 1

yarı-zayıf anahtarlar

$E_{K2}(E_{K1}(P))=P$

K1= 0 1 F E 0 1 F E 0 1 F E 0 1 F E
K2= F E 0 1 F E 0 1 F E 0 1 F E 0 1

K1= 0 1 1 F 0 1 1 F 0 1 0 E 0 1 0 E
K2= 1 F 0 1 1 F 0 1 0 E 0 1 0 E 0 1

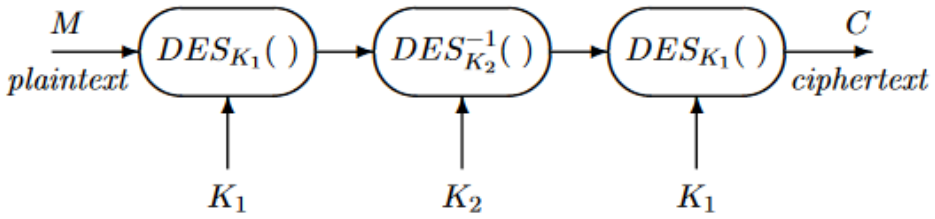
D.W. Davies and W.L. Price, Security for Computer Networks. John Wiley and Sons, New-York,1984.

DES güvenlik

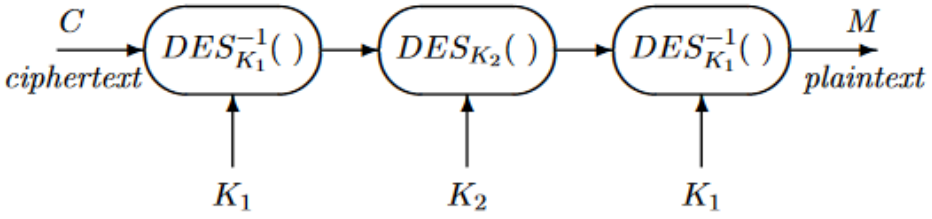
3DES

$C = E_{K3}(D_{K2}(E_{K1}(P)))$

$$C = DES_{K_1}\left\{DES_{K_2}^{-1}[DES_{K_1}(M)]\right\} \quad \text{(triple DES encryption)}$$
$$M = DES_{K_1}^{-1}\left\{DES_{K_2}\left[DES_{K_1}^{-1}(C)\right]\right\} \quad \text{(triple DES decryption)}$$



Triple DES encryption (2 keys)



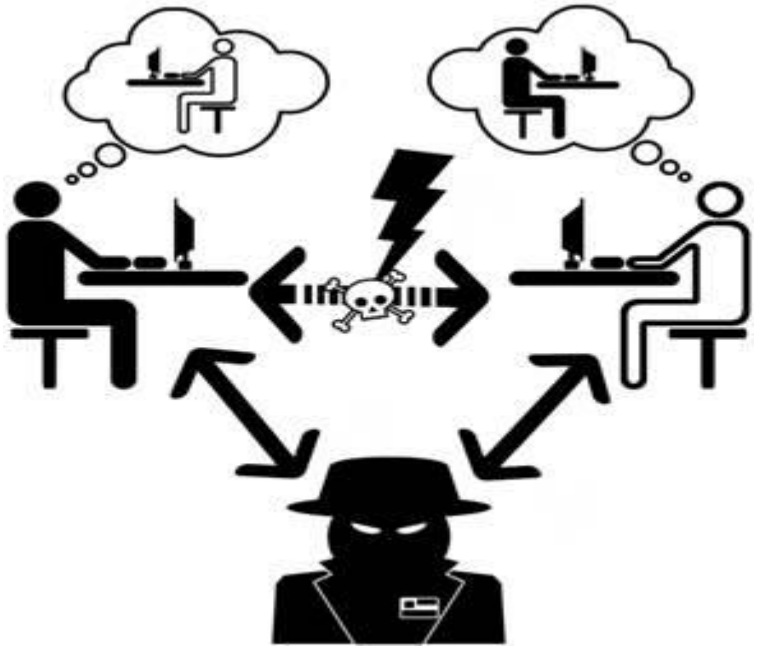
Triple DES decryption (2 keys)

Simetrik Şifreleme Algoritmaları

- DES
- 3DES
- AES
- Rijndael
- RC4
- Blowfish
- …

Güvenlik Hizmetleri (Security Services)

- Gizlilik (Confidentiality)
- Bütünlük (Integrity)
- Kimlik doğrulama (Authentication)
- Ulaşılabilirlik (Availability)
- Rededememe (Non-repudiation)



Kimlik doğrulama (Authentication)

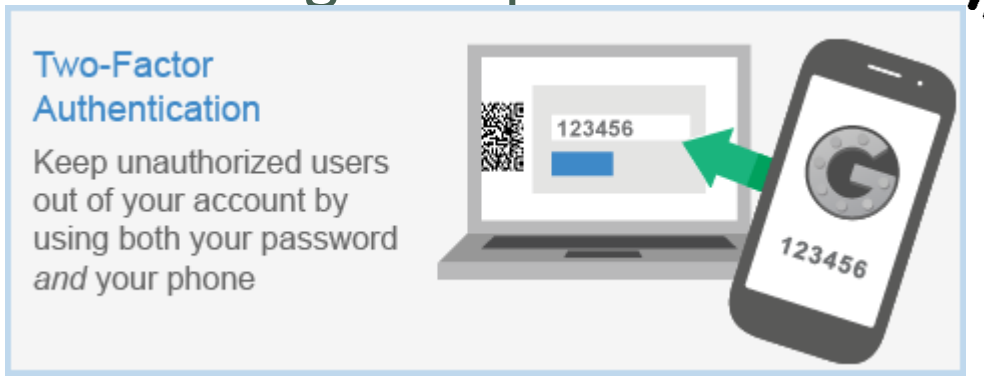
- Varlığın iddia ettiği kimliğini doğrulamak (Who are you,really?)
- Varlığın orijinalliğini doğrulamak (authentic document)

GANDALF ?

Kimlik doğrulama (Authentication)

- Elde edilen (You have)
- Sahip olunan (You own)
- Her ikisi (Both)

- Challenge-Response



"This site wants a two-factor authentication. A retina scan and a urine sample."

Simetrik şifreleme

- Kimlik doğrulama
  - Sadece anahtar sahipleri
  - $C = E_K(P)$
- Mesajın orijinalliği?
- Her anahtarı olan içeriği değiştirip gönderebilir
- Kendisinin oluşturduğunu reddedebilir

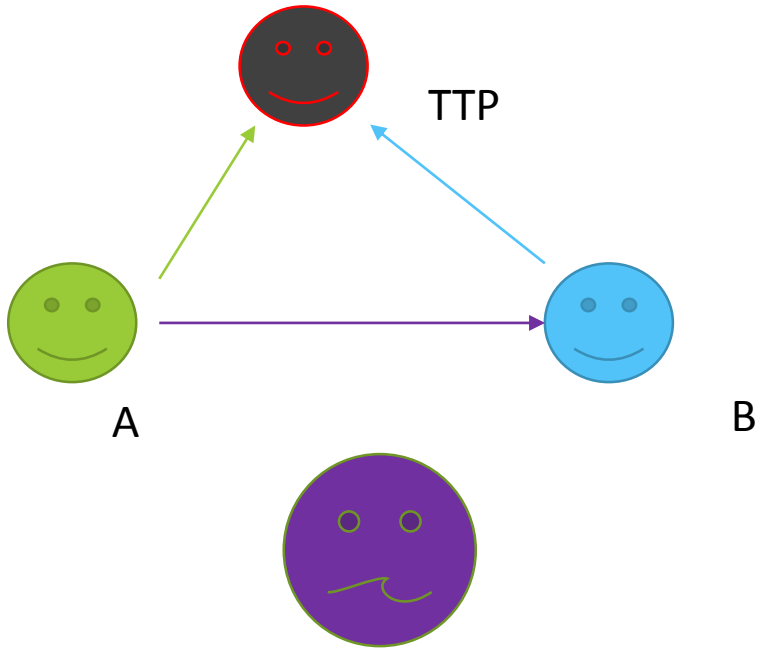
Simetrik şifreleme

- Güvenilir Üçüncü Taraf (Trusted Third Party)
- Mesajın orijinalliği?
- K1: (A,B) K2: (A,C) K3: (B,C)

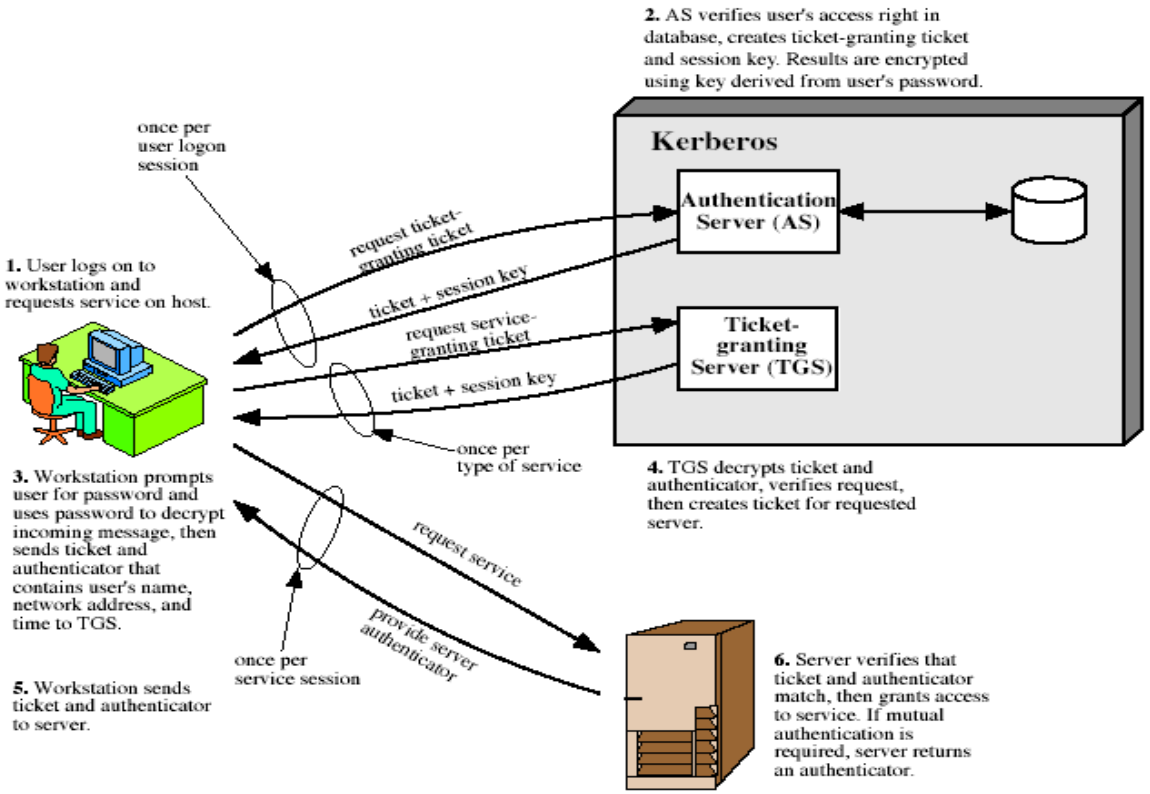
- $E_{K1}(M)$; M, $E_{K1}(M)$

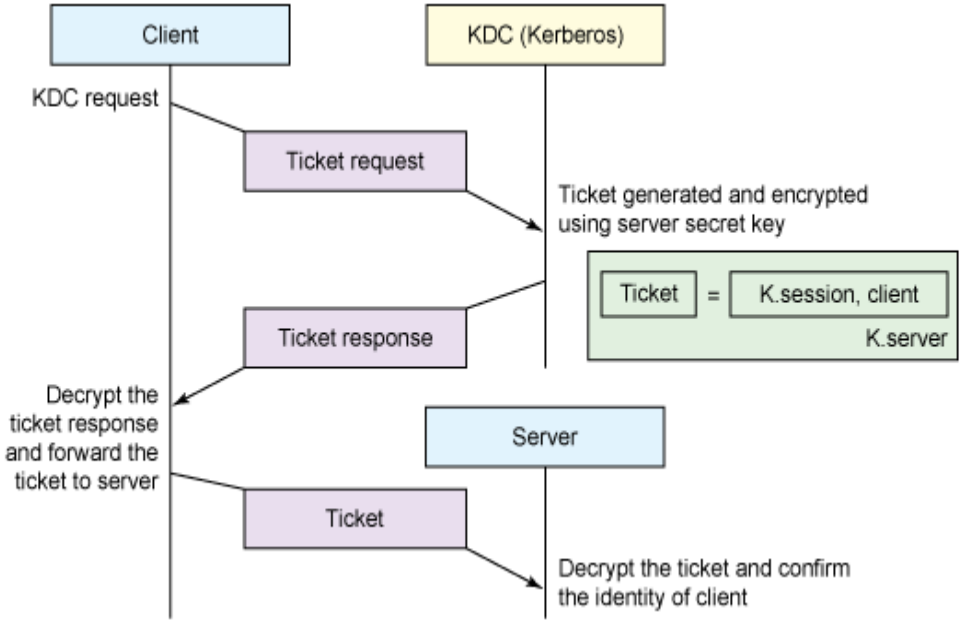
- M, $E_{K1}(M)$; M, $E_{K2}(E_{K1}(M))$

- M, $E_{K2}(M)$



KERBEROS


2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.
Kerberos
Authentication Server (AS)
Ticket-granting Server (TGS)
once per user logon session
request ticket-granting ticket
ticket + session key
request service-granting ticket
ticket + session key
once per type of service
1. User logs on to workstation and requests service on host.
3. Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user's name, network address, and time to TGS.
4. TGS decrypts ticket and authenticator, verifies request, then creates ticket for requested server.
request service
provide server authenticator
once per service session
5. Workstation sends ticket and authenticator to server.
6. Server verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.

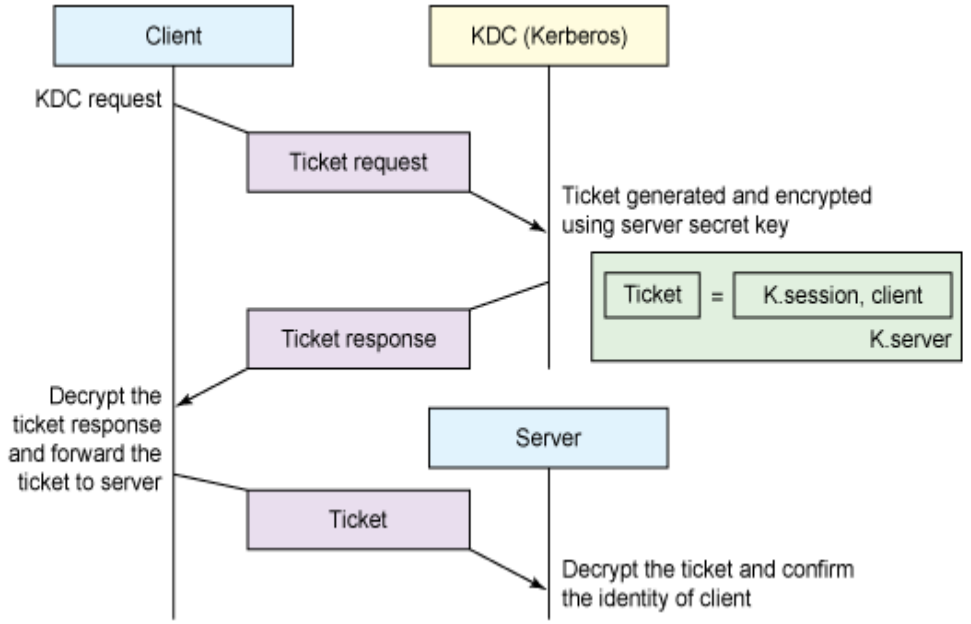

Client
KDC (Kerberos)
KDC request
Ticket request
Ticket generated and encrypted using server secret key
Ticket = K.session, client
K.server
Ticket response
Decrypt the ticket response and forward the ticket to server
Server
Ticket
Decrypt the ticket and confirm the identity of client

Kerberos 4 Overview

- a basic third-party authentication scheme
- have an *Authentication Server (AS)*
  - users initially negotiate with AS to identify themselves
  - AS provides a non-corruptible *authentication credential (ticket granting ticket TGT)*
- have a *Ticket Granting server (TGS)*
  - users subsequently request access to other services from TGS on basis of users TGT

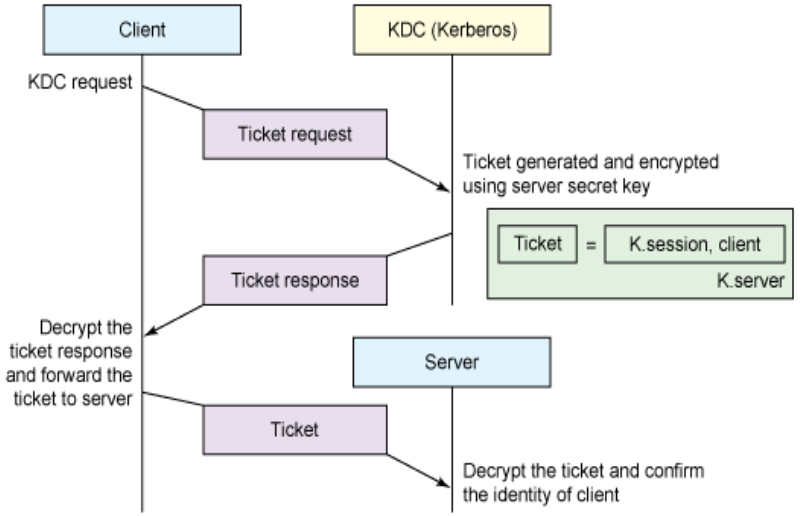
A Simple Authentication Dialogue

- (1) C -> AS : ID_C || P_C || ID_V
  - C = client
  - AS = authentication server
  - ID_C = identifier of user on C
  - P_C = password of user on C
  - ID_V = identifier of server V
  - C asks user for the password
  - AS checks that user supplied the right password



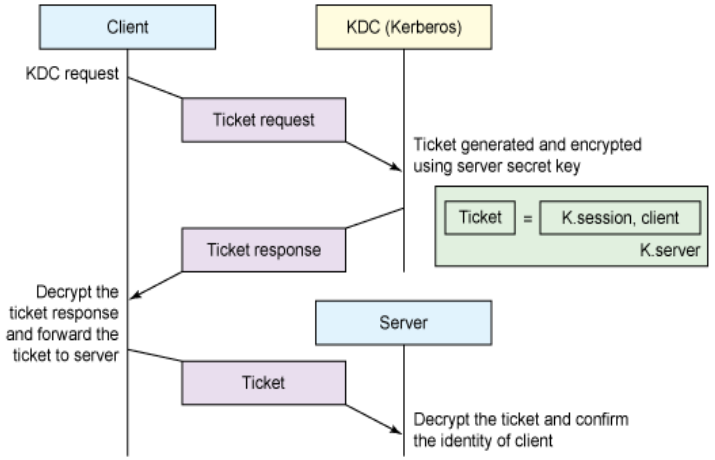
Message 2

- (2) AS -> C : Ticket
- Ticket = $E_{K(V)}$ [ID_C || AD_C || ID_V]
  - K(V) = secret encryption key shared by AS and V
  - AD_C = network address of C
  - Ticket cannot be altered by C or an adversary



Message 3

- (3) C -> V: ID_C | | Ticket
  - Server V decrypts the ticket and checks various fields
  - AD_C in the ticket *binds* the ticket to the network address of C
  - However this authentication scheme has problems



Kerberos v4 / Kerberos v5

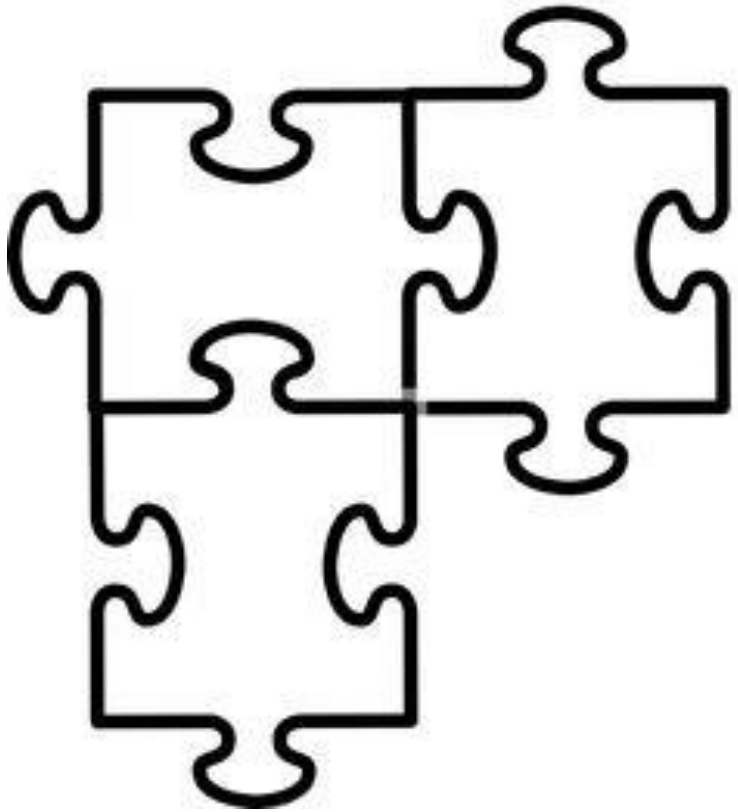
- Şifreleme
- V4: DES V5: Sınırlama yok
- Ağ protokolü
- V4: IP V5: Çoklu IP adresi
- Bilet süresi
- V4: 21 saat V5: (başlangıç-bitiş zamanı)

- Encryption system: V4 requires DES, V5 can use any
- Internet protocol: V4 requires IP, V5 multiple IPs
- Ticket lifetime: 21 hours in V4 , V5 tickets include explicit start and end time

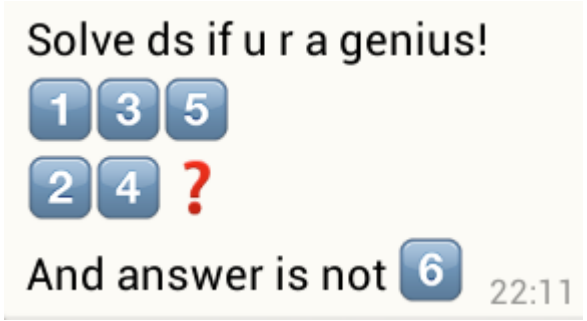
Authentication

- s.509
- RADIUS

Ara - 15dk



Rasgele Sayı Üreteçleri



- Math.rnd(seed);

- Rasgelelilik (Randomness)
  - Tek bir sayıdan bahsetmek yerine, bir dizi sayı söz konusu

  - Düzgün dağılım (Uniform distribution)

  sayıların dağılımı, ortay çıkma sıklıkları
  - Bağımsızlık (Independence)

  Dizideki hiçbir sayı diğerlerinden çıkarım yapılarak tahmin edilemez

Sözde Rasgele Sayı Üretimi Pseudo-random number generators (PRNGs)

- Tablo tabanlı
- Donanım üreteçleri
- Yazılım (algoritma tabanlı) üreteçler

- table look-up generators
- hardware generators
- algorithmic (software) generators

Sözde Rasgele Sayı Üretimi Pseudo-random number generators (PRNGs)

«Eskimiş yöntemler»

- Kareortası yöntemi
  1. Başlangıç tohumu (4 basamaklı tamsayı)
  2. Karesini al
  3. Ortasındaki 4 basamaklı sayıyı al
  4. Bu sayıyı yeni Başlangıç tohumu olarak ata
  5. Sayıyı 10.000'e böl.
  6. Sonuç rasgele sayın olacak
  7. Yeni üretmek için 2'ye geri dön.

$s_0 = 5497$
s_1: $5497^2 = 30217009 \rightarrow s_1 = 2170, R_1 = 0.2170$
s_2: $2170^2 = 04708900 \rightarrow s_2 = 7089, R_2 = 0.7089$
s_3: $7089^2 = 50253921 \rightarrow s_3 = 2539, R_3 = 0.2539$

Midsquare method:

1. Start with an initial seed (e.g. a 4-digit integer).
2. Square the number.
3. Take the middle 4 digits.
4. This value becomes the new seed. Divide the number by 10,000. This becomes the random number. Go to 2.

Sözde Rasgele Sayı Üretimi Pseudo-random number generators (PRNGs)

«Eskimiş yöntemler»

- Kareortası yöntemi
  1. Başlangıç tohumu (4 basamaklı tamsayı)
  2. Karesini al
  3. Ortasındaki 4 basamaklı sayıyı al
  4. Bu sayıyı yeni Başlangıç tohumu olarak ata
  5. Sayıyı 10.000'e böl.
  6. Sonuç rasgele sayın olacak
  7. Yeni üretmek için 2'ye geri dön.

$s_0 = 5197$

s_1: $5197^2 = 27\underline{0088}09 \rightarrow s_1 = 0088$, $R_1 = 0.0088$

s_2: $0088^2 = 00\underline{0077}44 \rightarrow s_2 = 0077$, $R_2 = 0.0077$

s_3: $0077^2 = 00\underline{0059}29 \rightarrow s_3 = 0059$, $R_3 = 0.0059$

$s_i = 6500$

s_{i+1}: $6500^2 = 42\underline{2500}00 \rightarrow s_{i+1} = 2500$, $R_{i+1} = 0.0088$

s_{i+2}: $2500^2 = 06\underline{2500}00 \rightarrow s_{i+2} = 2500$, $R_{i+1} = 0.0088$

Midsquare method:

1. Start with an initial seed (e.g. a 4-digit integer).
2. Square the number.
3. Take the middle 4 digits.
4. This value becomes the new seed. Divide the number by 10,000. This becomes the random number. Go to 2.

Doğrusal uyumlu üreteçler (Linear congruential generator)

4 tamsayı

- m mod $m > 0$
- a çarpan (katsayı) 0 , $0 < a < m$
- c artım (eklenen) 0, $0 < c < m$
- X_0 başlangıç değeri 0, $0 < X_0 < m$

Algoritma: n>0 olmak üzere

$X_n + 1 = (aX_n + c) \bmod m$

4 integer

- m the modulus $m > 0$
- a the multiplier 0 , $0 < a < m$
- c the increment 0, $0 < c < m$
- X_0 the starting value 0, $0 < X_0 < m$

The algorithm is

$X_n + 1 = (aX_n + c) \bmod m$

Where n>0

Doğrusal uyumlu üreteçler (Linear congruential generator)

4 tamsayı

- m mod $m > 0$
- a çarpan (katsayı) 0 , $0 < a < m$
- c artım (eklenen) 0, $0 < c < m$
- X_0 başlangıç değeri 0, $0 < X_0 < m$

Algoritma: n>0 olmak üzere

$X_{n+1} = (aX_n + c) \bmod m$

4 integer

- m the modulus $m > 0$
- a the multiplier 0 , $0 < a < m$
- c the increment 0, $0 < c < m$
- X_0 the starting value 0, $0 < X_0 < m$

The algorithm is

$X_{n+1} = (aX_n + c) \bmod m$

Where n>0

- a=1, c=1 ?
- a=7 c=0 m= 32 X_0 =1 {7,17,23,1,7,..}
- a= 5 {5,25,29,17,21,9,13,1,5,..}

Lehmer PRNG

Lehmer Algoritması (Doğrusal uyumlu üreteç tabanlı)

$X_{i+1} = (aX_i + c) \bmod m$, with $0 \leq X_i \leq m$

M 2^{p-1} p CPU bitleri (32 bit, 64 bit, etc.)

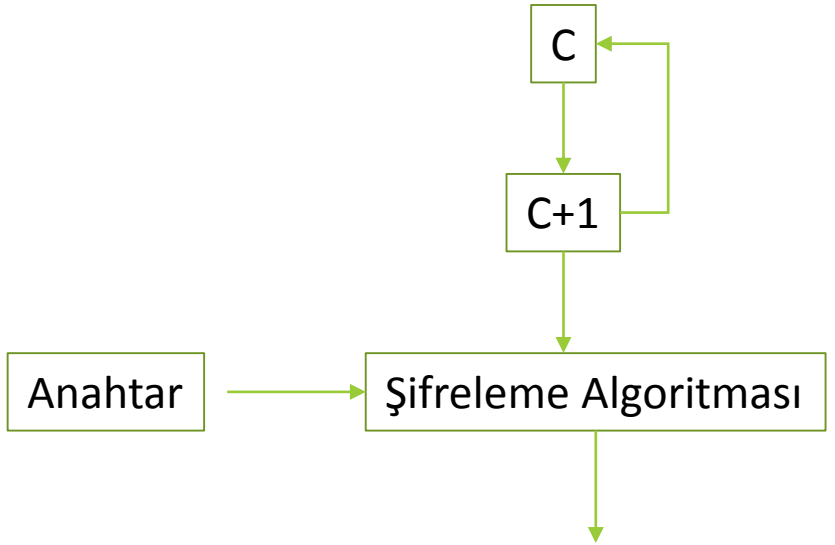
$m = 31$, $a = 7$, $c = 0$ $X_0 = 19$ {9, 1, 7, 18, 2, 14, 5, 4, 28, 10, 8, 25, 20, 16}

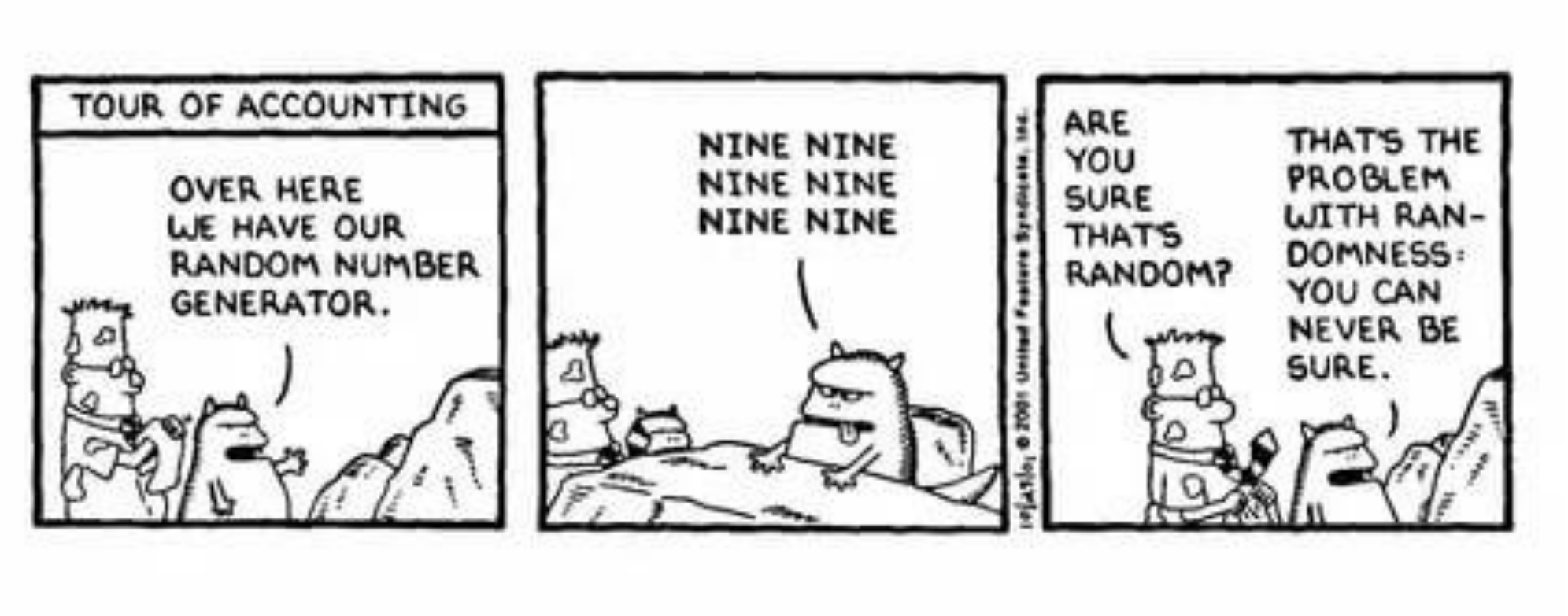
Doğrusal uyumlu üreteçler (Linear congruential generator)

Lagged Fibonacci generator (LFG)

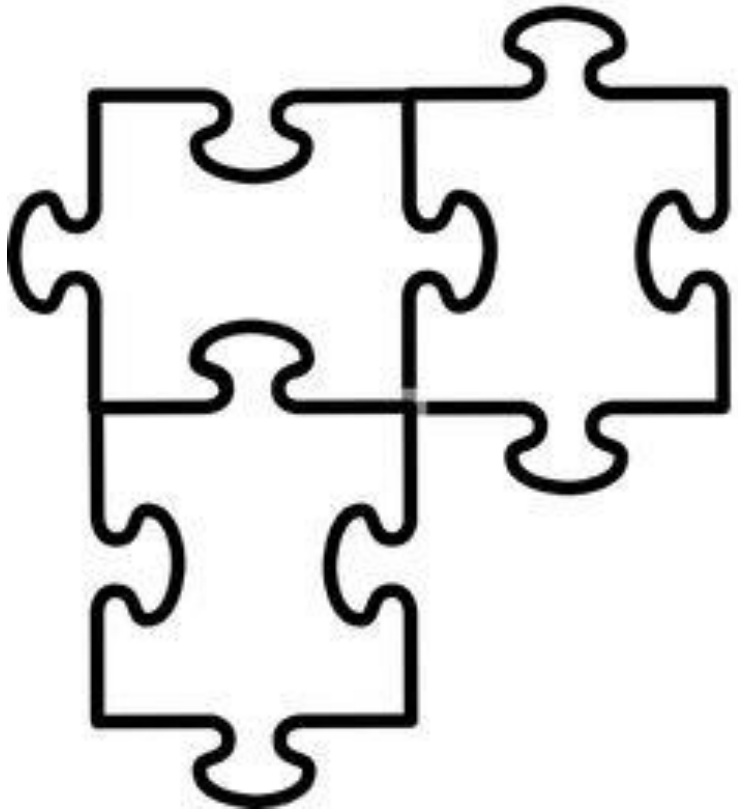
Blum Shub Shub

Kriptografik Üreteçler




TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
NINE NINE NINE NINE NINE NINE
ARE YOU SURE THATS RANDOM?
THATS THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.

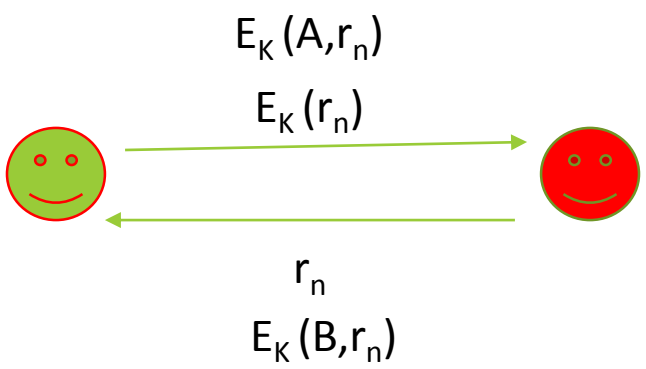
Ara - 10dk

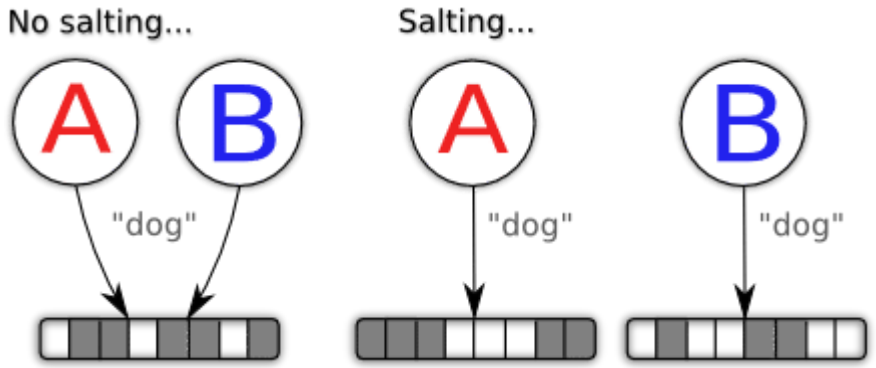


Rasgele Sayı Üreteçleri

- Kimlik doğrulama
- Meydan okuma-Cevap
- Protokol güvenliği
- «Tuz'la da kokmasın»

Authentication
Challenge-Response
Protocol security
Salting passwords



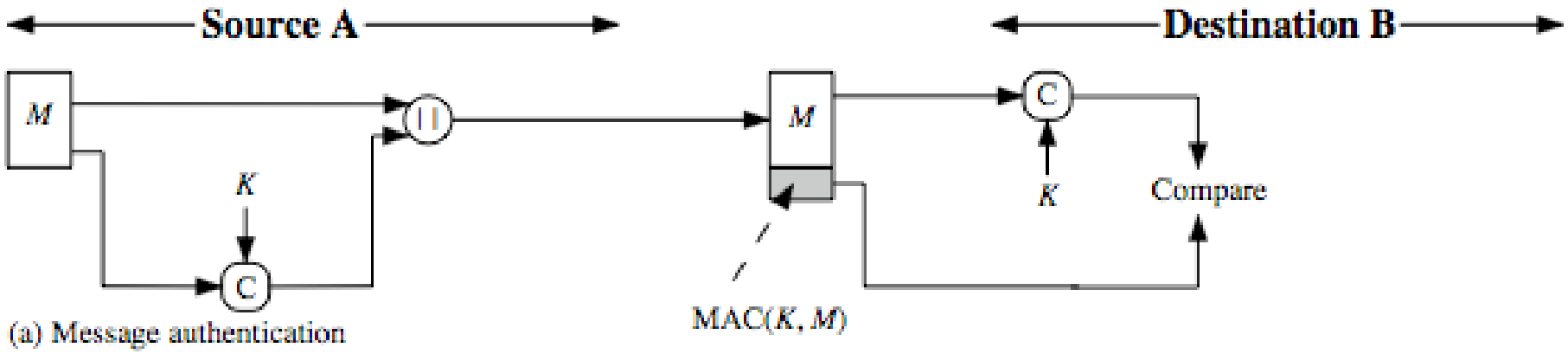


Rasgele Sayı Üreteçleri

Bütünlük(Integrity)

- MAC (Message Authentication Code)
- MAC = C(K,M)
- Mesaj özeti HASH

a small fixed-sized block of data
generated from message + secret key
MAC = C(K,M)
appended to message when sent

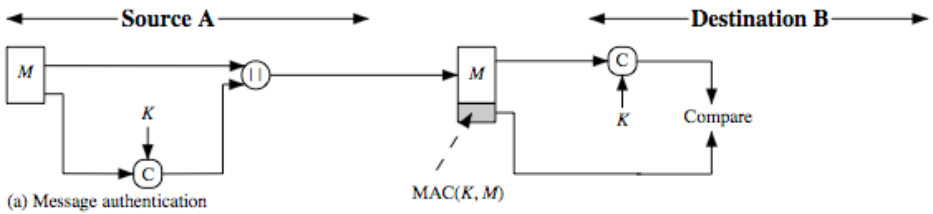


Bütünlük(Integrity)

- Aynı özete sahip başka bir mesaj bulunamamalı
- Düzgün dağılım
- Çığ etkisi
- Tersinir olmayan bir fonksiyon olmalı
- MAC = C(K,M) C^{-1} (MAC) = ~~K,M~~

- N byte -> 256bit

- knowing a message and MAC, is infeasible to find another message with same MAC
- MACs should be uniformly distributed
- MAC should depend equally on all bits of the message



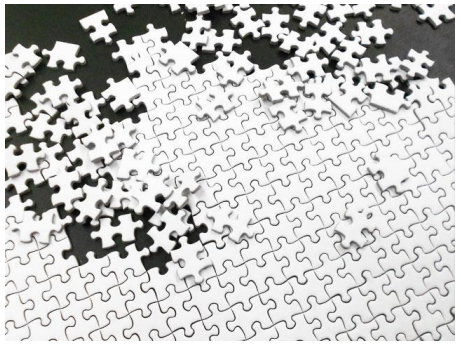
Anahtar Yönetimi (Key Management)

- Anahtarın saklanması (Key Storage)
- Anahtarların değişimi (Key Eschange)
- Anahtarların yenilenmesi (Key Renewal)
- Anahtarların iptali (Key Revocation)

CENG 507 :
KRİPTOGRAFİK ALGORİTMALAR VE SİSTEMLER

CENG 434:
KRİPTOLOJİ

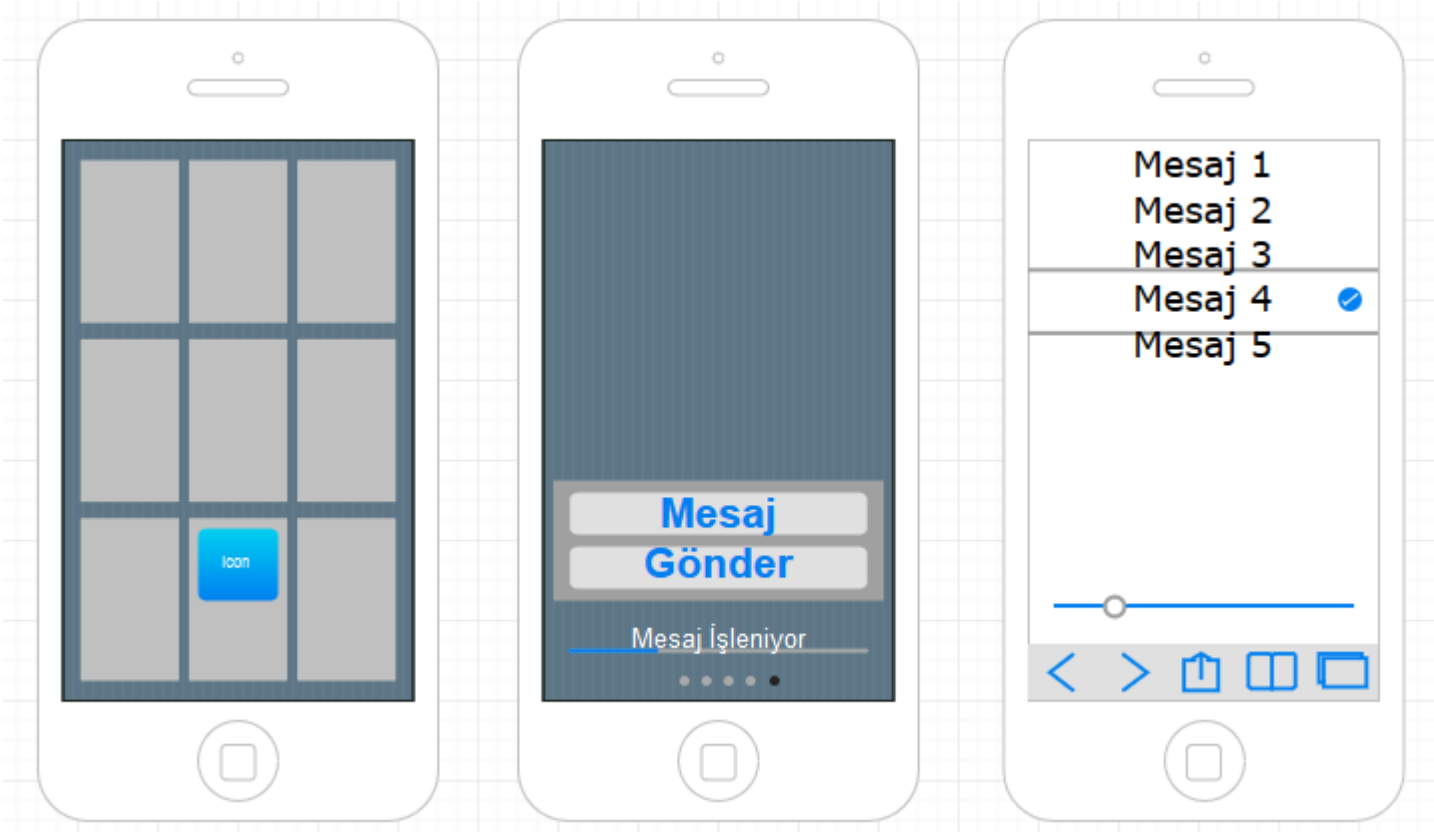
Araştırma ve Proje detayları için EDS'yi takip edin.



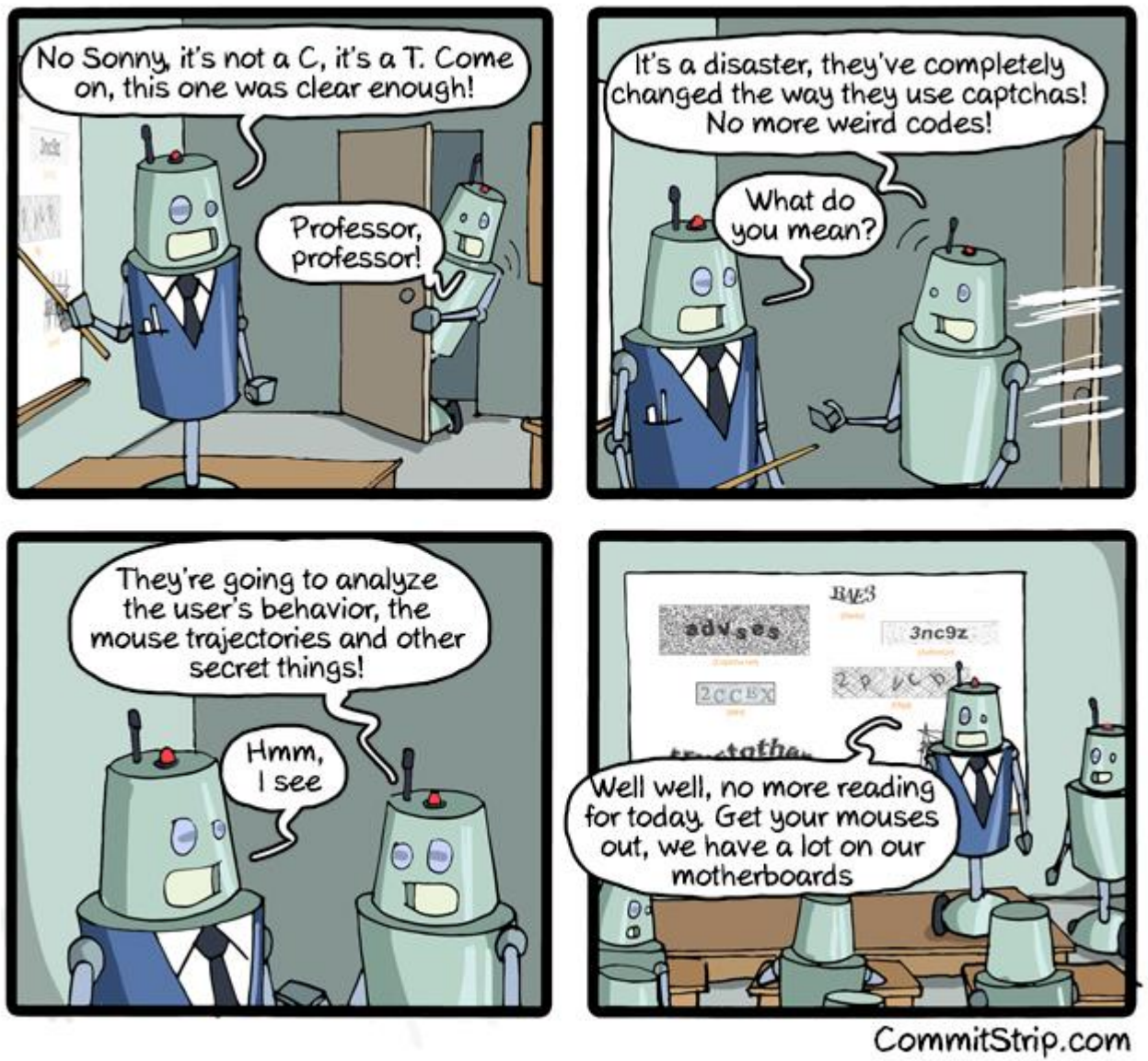
Proje

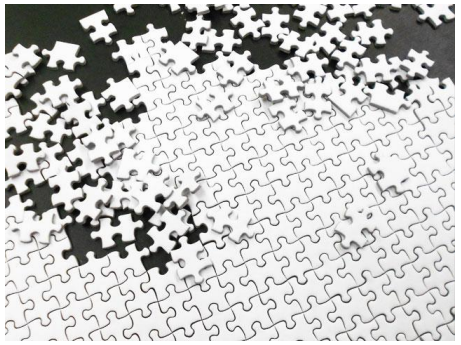
- Kullanıcı girişi
- Bir metin
- $Kullanıcı_A$
- $Kullanıcı_B$

- Tasarımı
- Uygulama
- Test senaryosu



ü
6
R

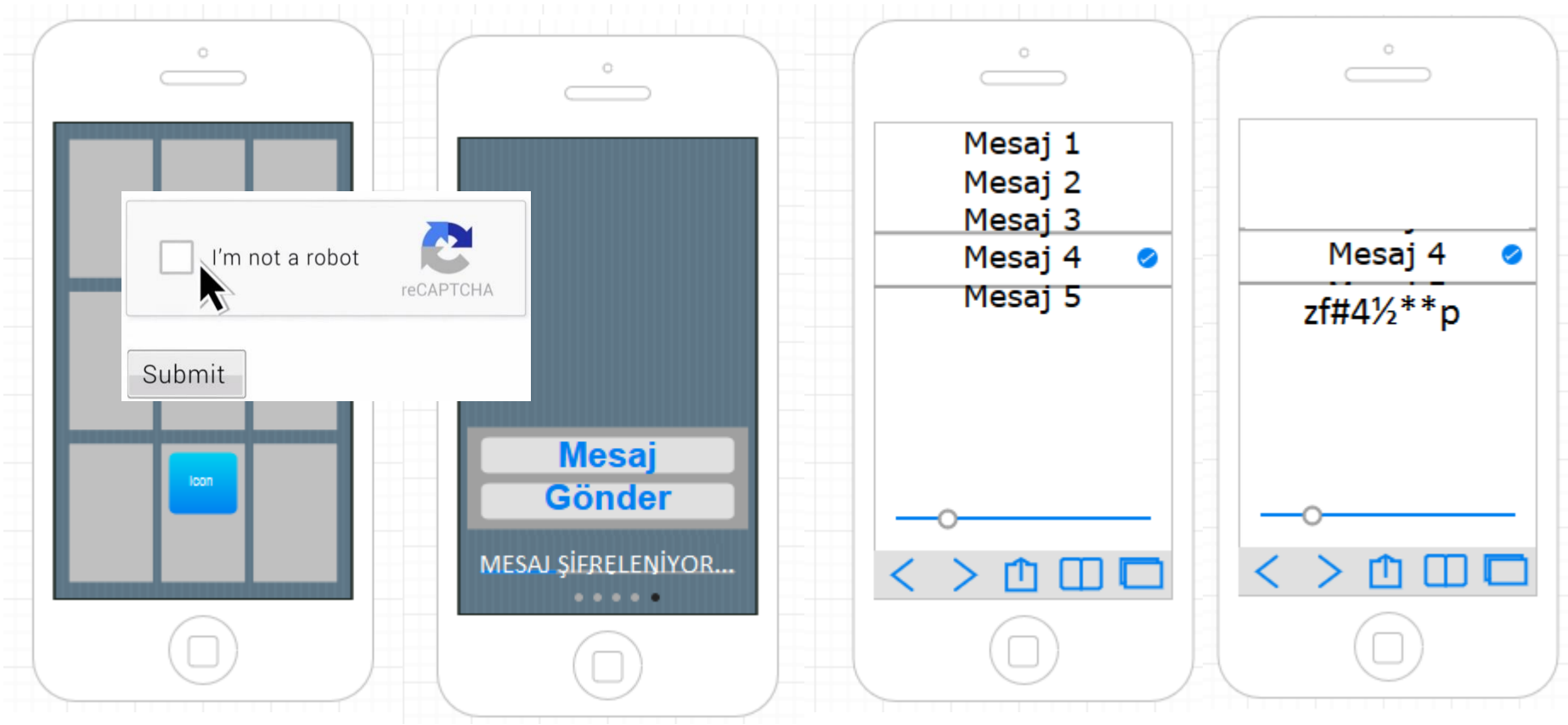

No Sonny, it's not a C, it's a T. Come on, this one was clear enough!
Professor, professor!
It's a disaster, they've completely changed the way they use captchas! No more weird codes!
What do you mean?
They're going to analyze the user's behavior, the mouse trajectories and other secret things!
Hmm, I see
Well well, no more reading for today. Get your mouses out, we have a lot on our motherboards
CommitStrip.com

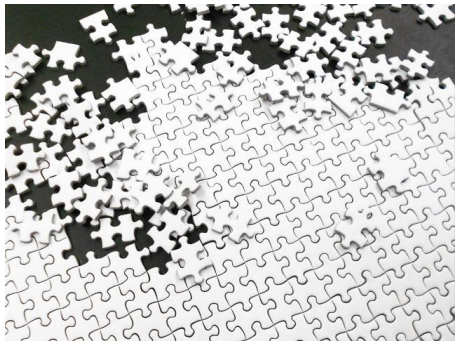


Proje

- Kullanıcı Girişi
- 3 Hatalı Giriş (bekle)
- 5 Hatalı Giriş (kilitle)
- Parola değiştirme
- Anahtar saklama
- Anahtar değişimi

- Tasarımı
- Uygulama
- Test senaryosu





Araştırma + Sunum

- Bireysel
- Simetrik Şifreleme Algoritmaları
  - Standartlar
  - Analiz
  - Karşılaştırma