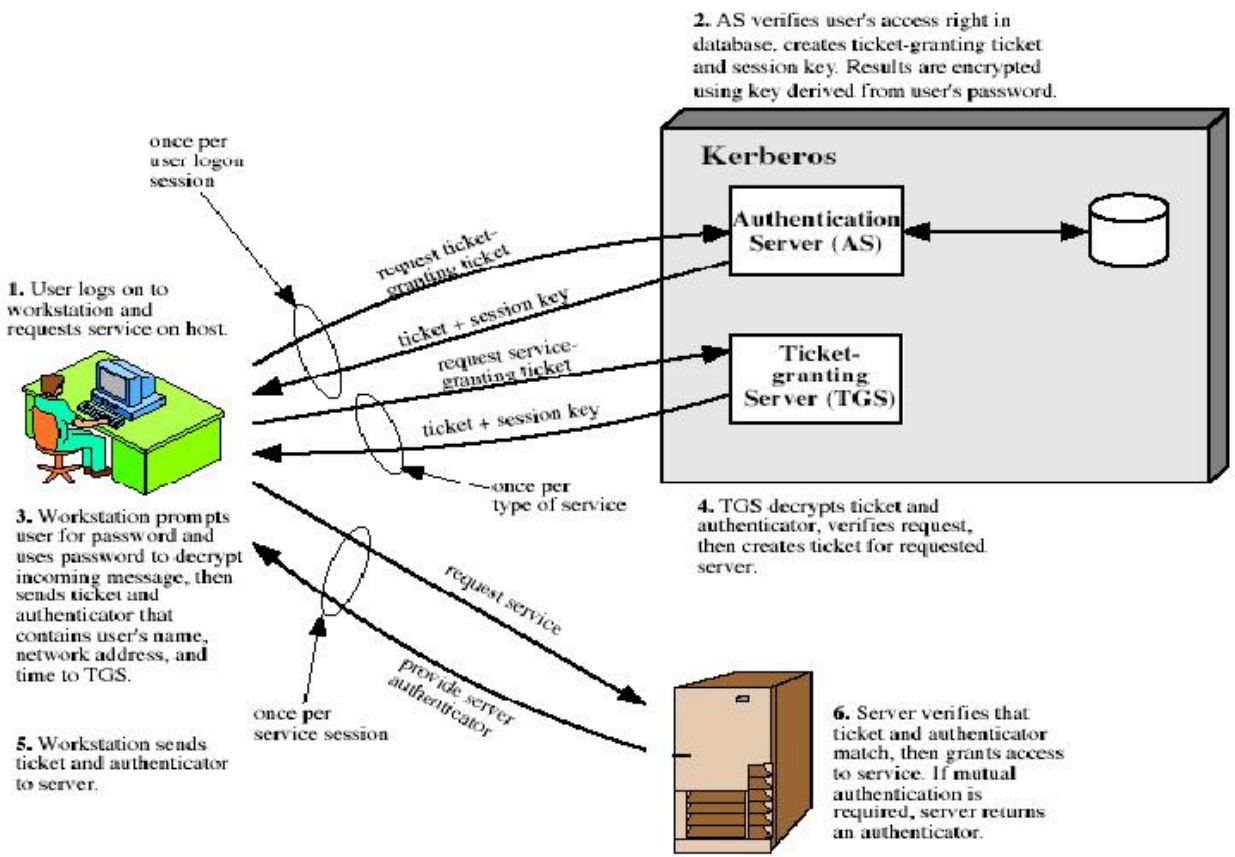
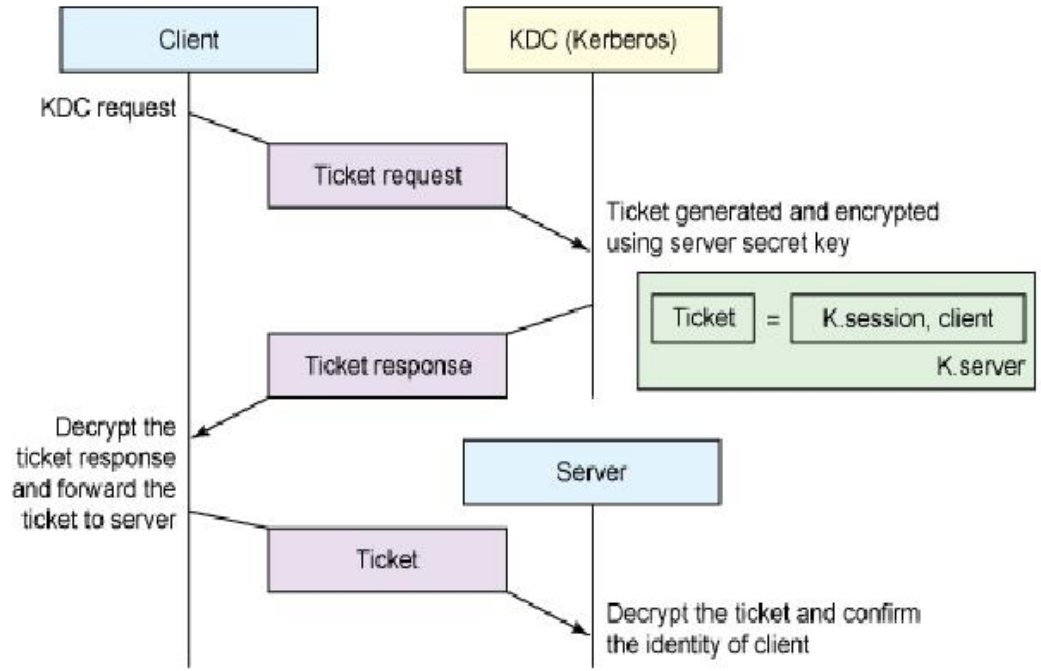
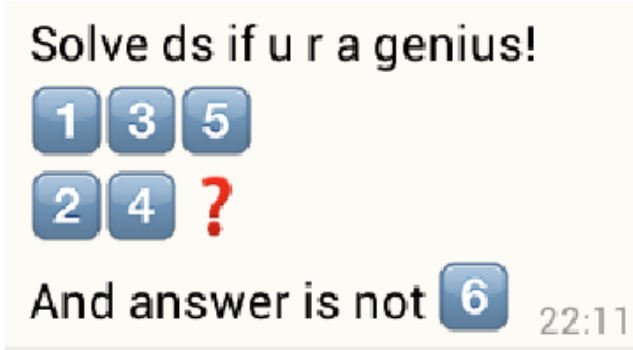
KERBEROS


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

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

«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: $7089^2 = 50253921 \rightarrow s_3 = 2539, R_3 = 0.2539$

$s_0 = 5197$
s_1: $5197^2 = 27008809 \rightarrow s_1 = 0088, R_1 = 0.0088$
s_2: $0088^2 = 00007744 \rightarrow s_2 = 0077, R_2 = 0.0077$
s_3: $0077^2 = 00005929 \rightarrow s_3 = 0059, R_3 = 0.0059$

$s_i = 6500$
s_{i+1}: $6500^2 = 42250000 \rightarrow s_{i+1} = 2500, R_{i+1} = 0.0088$
s_{i+2}: $2500^2 = 06250000 \rightarrow s_{i+2} = 2500, R_{i+1} = 0.0088$

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

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

- 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,..}

The algorithm is

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

Where n>0

Lehmer PRNG

Lehmer Algoritması (Doğrusal uyumlu üreteç tabanlı)

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

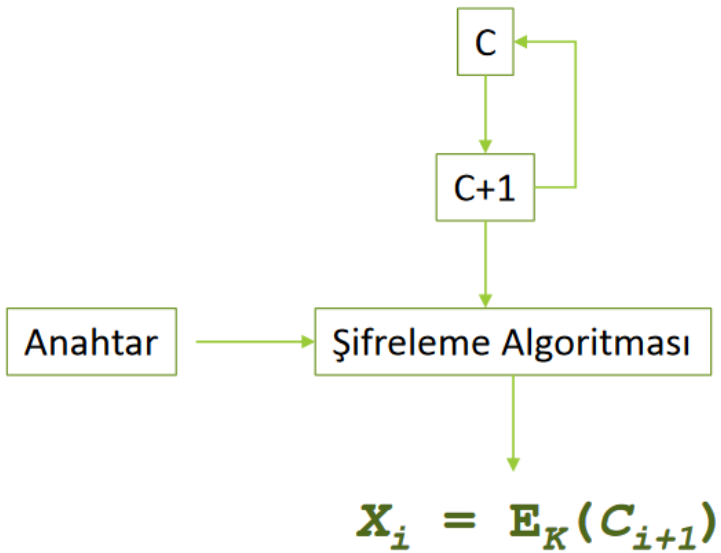
M 2^{p-1} p CPU bitleri (32 bit, 64 bit, etc.)

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

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?
THAT'S THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.