Kerberos: An Authentication Service

- Building blocks of Kerberos:
  - DES
  - Cipher-block-chaining
  - Needham-Schroeder Protocol
- Components of Kerberos
- Design and Implementation

Data Encryption Standard

- operates on 64-bit blocks of plaintext
- steps:
  - initial permutation
  - split key into two parts, R and L
  - 16 rounds of identical operation (function f), which combines data and key
  - after the sixteenth round, R and L are concatenated, and a final permutation

Permutations and Substitution in DES

- permutation and shift tables for DES
- compression and expansion permutation
- S-box:
  - 48 bits are divided into 8 6-bit blocks
  - each 6-bit block is operated by a different s-box;
  - s-box are 4x16 arrays, 6-bit acts as index
    - b0+b5 index into rows, b1…b4 index into columes
  - Security of DES depends on S-box

Encrypting Longer Texts

- what are the alternatives?
- Cipher block chaining:
  - the first block is encrypted using the key
  - the second block is xored with the first block's encryption, then is encrypted using the key, and so on...
  - "feedback register"
- Decryption of ciper block chaining
- Initialization vector in CBC mode

Authentication in a Distributed Environment

- Example: access control in a distributed file system environment
  - early NFS-type solution: trust the message's word
  - another solution: trust the machine's word
    - problem: what do you do about the root on that machine?
  - third solution: require the user to prove his or her identity for every service
- Requirements on authentication services

The Needham-Schroeder Protocol

- Purpose:
  - Alice and Bob establishing each other's identity

A trusted party Trent and the protocol:

- 1. Alice -> Trent: (A, B, R_a)
- 2. Trent -> Alice: E_a (R_a, B, K, E_b(K, A))
- 3. Alice verifies R_a, then -> Bob: E_b(K, A)
- 4. Bob -> Alice: E_k(R_b)
- 5. Alice -> Bob: E_k(R_b-1)
- 6. Bob verifies R_b-1

Kerberos: A Third-Party Authentication Service

- keeps a database of clients and their private keys
  - clients can be users or services
  - the private key is an encrypted password for users
  - clients "register" with kerberos, at which time the private keys are generated
- levels of protection:
  - authentication at connection establishment only
  - authentication for every message

Involved Parties in Kerberos

- User
- Authentication Server
- Ticket-granting server
- the server whose services are requested by the user
- Interactions among the parties

Kerberos Credentials

- Naming:
  - primary name, instances and realm
- ticket: a ticket is good for one client and one server, K_s(s, c, addr, timestamp, life, K_s,c)
  - lifetime of tickets important
- authenticator: K_s,c(c, addr, timestamp)

Kerberos Steps

- Getting initial ticket (or ticket to the ticket-grant service):
  - user type in user name
  - user -> kerberos: (c, ticket_granting_service)
  - kerberos -> user: K_c(K_c,tgs, k_tgs(T_c,tgs))
  - user type in password, extract K_c,tgs, and the ticket
  - erase password and DES key from memory

Getting Server Tickets

- Client->tgs: (s,K_tgs(T_c,tgs),K_c,tgs(c,addr,timestamp))
- tgs->client: K_c,tgs(K_s(T_c,s),K_c,s)

Requesting Service

- client -> server: K_c,s(c,addr,timestamp)+K_s(T_c,s)
- assume that clocks are synchronized to some degree
- server->client: K_c,s(timestamp+1)

Kerberos Database Operations

- Master database server and slave servers
- changes to the database can be performed only on the server
- slaves hold read-only copies of the database
- KDBM uses kerberos to authenticate the administrators
- periodic updates for database consistency

Using Kerberos

- library, adminstrator, tools
- library functions: krb_mk_req, krb_rd_req
- Example: Kerberos in NFS
  - to avoid overhead, kerberos is only used at mount time to establish user+workstation to a particular principal in NFS