AUTHENTICATION SYSTEMS

Password, Key Distribution and Certification

Authentication Systems

- Password – based
- Address – based
- Cryptographic protocols
- Key Distribution Centres
- Certification Authorities

Password - Based

- Legacy systems – dumb terminals
- Vulnerable to passive attack/cloning
- Problems with distributed resources
- Password guessing attacks
  - on–line
  - off-line/dictionary

Verifying Passwords

- Passwords replicated on every resource
- Authentication storage node supplies information on request
- Authentication facilitator confirms or rejects password
- Above two methods require authentication

Hashing Passwords v Encryption

- Hashed passwords vulnerable to dictionary attack (unless chosen randomly)
- Encryption vulnerable to node (i.e. encryption key) compromise
- May combine (i.e. hash followed by encryption)

Address – Based Authentication

- Identity confirmed by network address
- Option 1- Machine B has NA list of ≡ machines: any account on A is ≡ to same named account on B (e.g. /etc/hosts.equiv files on UNIX)
- Option 2- Machine B has list (address, remote name, local name): remote name on address is granted same rights as local name (e.g. .rhosts files on UNIX)

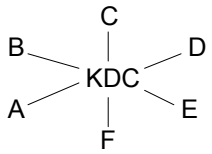
Attacks on Network Based Authentication

- Network machine compromised – all remote resources available to local accounts are compromised
- Address impersonation leads to compromise of network resources

Cryptographic Authentication Protocols

- Password → cryptographic key
  - by hash function (e.g. DES key)
  - by decrypting higher grade key (e.g. RSA)
  - may use seed value of random number generator in routine to find primes for RSA
  - alternatively use stage number of prime checking as password

Key Distribution Centres (KDCs)



-KDC knows keys for all nodes

-For A/B secure communications
-KDC authenticates A
-Provides key K_{AB}
-Passes K_{AB} encrypted under K_A to A and encrypted under K_B to B
-KDC may ask A to forward ticket to B

KDC Issues

- Centralises all network security information
- Single point of failure
- Could have performance limitations

Certification Authorities

- Holds public keys for nodes
- Public key information for nodes (e.g. A, e_A , n_A) signed by CA private key
- All nodes have CA public key to check authenticity of certificate
- Initially nodes know only CA public key

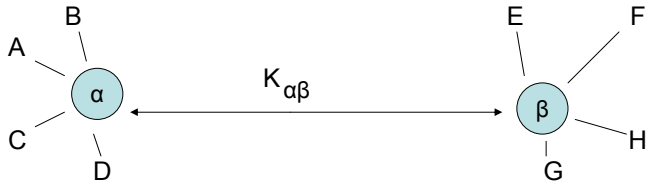
CAs v KDCs

- CA may be off-line
- CA less complex than KDC
- CA crash impacts only new users or problems with expired or revoked certificates
- certificates do not need high security storage
- CA may be compromised without compromising existing communications

Life of Certificate

- Expiration date on certificate
- Updated Certificate Revocation List updated regularly

KDC Domains



A requires secure comms with E

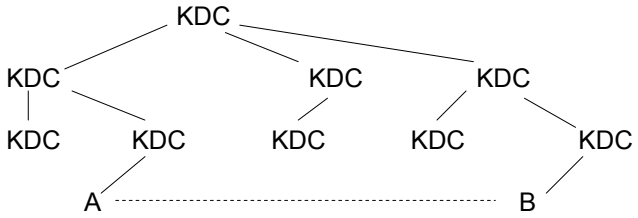
A requests α to facilitate contact with β

α gives A and β $K_{A\beta}$ encrypted under K_A and $K_{\alpha\beta}$ respectively

A requests β to facilitate secure comms with E

β provides K_{AE} to A and E encrypted under $K_{A\beta}$ and K_E respectively

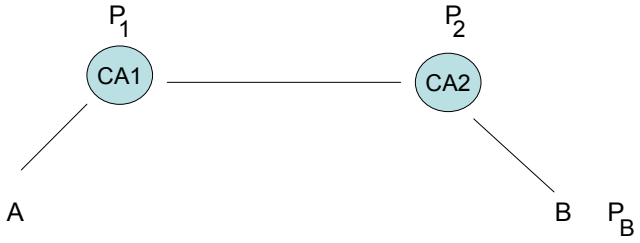
KDC Hierarchy



KDC chain established for AB

Details of chain sent with key K_{AB}

Multiple CA Domains



CA1 has public key P_1 , CA2 has public key P_2

A receives P_2 signed by CA1

A receives P_B signed by CA2