Imperial College London

Computer Networks and Distributed Systems

Part 2.4 – Security

Course 527 – Spring Term 2015-2016

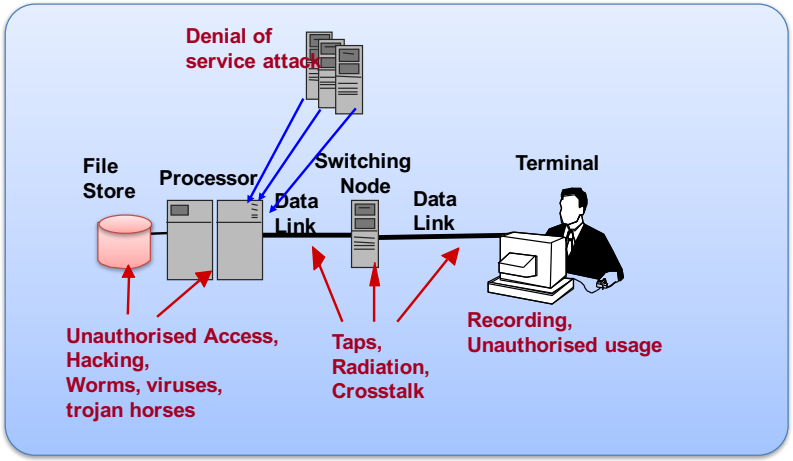
Emil Lupu and Daniele Sgandurra

e.c.lupu@imperial.ac.uk, d.sgandurra@imperial.ac.uk

Cyber Security in the News

Contents

Threats



Information or Resources

- Theft / copying, disclosure
- Modification, corruption or fabrication
- Destruction

Services

- Unauthorised utilisation of resources
- Disruption of service
- Denial of access to authorised users

Users

- Abuse of privilege by legitimate user
- Masquerading – impersonation of the identity of another authorised user.

Large scale networks (and hence distributed systems) cannot be made physically secure.

Types of Attacks

Passive Attack

- Observe information in network without interference
- Message content – break confidentiality
- Message traffic analysis – frequency, length, source, destination
  Could have military significance

Active Attack

- Modify message contents or message stream
- Delete, delay, reorder, replay, insert valid or invalid messages
- Masquerade as authorised user
- Denial of service by flooding servers with valid requests
- Passwords gained through passive attack can be used for active attack

Other Security Threats

Unattended Terminal

- Passer-by can gain access to resources accessible by user
- Solution – forced log off after timeout

Man in the Middle attack

- Relays all interactions between client and server pretending to be the client to the server, and the server to the client.
- Substitutes his own encryption keys for that of the server.

Malware (viruses, trojan horses, worms)

Compromise Detection

- Intrusion Detection
- Anomaly Detection

Security Goals

Confidentiality: prevent disclosure of information to unauthorised users + prevent analysis of traffic characteristics.

Integrity: prevent modification of information by unauthorised users – includes no duplication, replays, insertions or reordering.

Availability: prevent denial of service e.g. by disruption

Require:

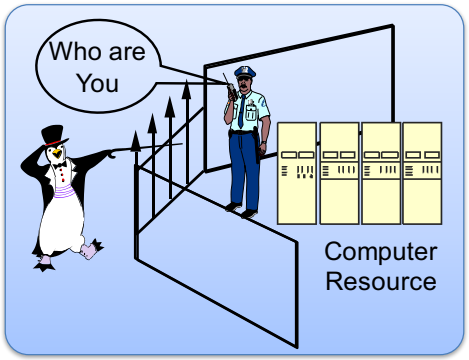
- Identification: establishing the identity of the subject
- Authentication: validity of the identity of sender or server.
- Access Control: control over who has access to services or resources within the system

Identification

Establishing Identity of Subject

Identification can use:

- Name or ID provided by user
- Workstation address
- Magnetic card
- Smart card



Recommendations:

- Use name/initials not numbers
- Same ID for ALL systems & email
- Individual accountability -> unique ID for each user
  Identify users in logs and audit trails
- No sharing of ID

Authentication

Is the Verification of subject Identity

- Personal identification number (PIN) for magnetic or smart cards
- Passwords
- Biometrics eg. retina or fingerprint scanning
- Maximum number of authentication attempts (e.g.3)
- Logging and investigation of all authentication failures

Challenge-Response

- Authentication is often based on challenge-response protocols where the authenticated party needs to prove knowledge/possession of a secret

What is required

Vulnerability Analysis: identify potential weak elements within system – What is critical to the organisation?

Threat Assessment: likelihood of a threat occurring which exploits the vulnerability detected

Risk Analysis: analyse the potential consequences of problems arising from security breach + estimate cost of a successful attack e.g. loss of revenue.

Prevention techniques: what can be done to prevent security breaches and what are their cost?

Cost benefit analysis: do the consequences of security breaches justify the cost of protection?

If security controls cause too much inconvenience or loss in performance they will be bypassed.

Recovery: may be less costly than prevention??

Tools

Access control

- Network
- Operating System

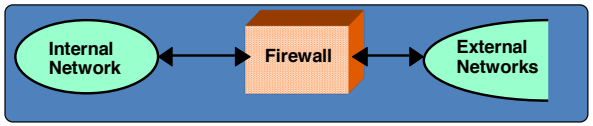
Cryptography

Key Management and PKI

FIREWALLS

Firewalls

A security gateway between internal and external networks

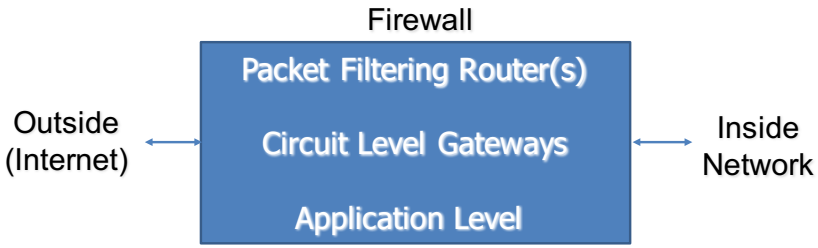


Based on packet filters – user defined filtering rules for both incoming and outgoing messages.

Filtering criteria

- Address – only permit access from selected sites or hosts e.g. remote sites of the same organisation, or collaborators. Could be based on combinations of IP port address
- Message type – permit incoming Email & HTML (Web) messages but prevent Telnet or FTP

Firewall Components



Packet Filtering Routers (a.k.a. Chokes, Screening Filters) - Restrict freeflow of packets between networks.

Gateways - Applications (proxies) that provide higher-level processing e.g. authenticating users, "cleaning data", redirecting data, logging, accounting Gateway apps normally run on so-called BASTION HOSTS.

End-to-End encryption between Firewalls is used to create VPNs

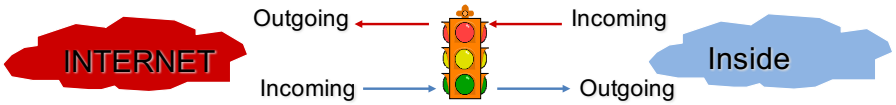
Packet Filtering

Drop Packets based on Source address and/or Destination addresses and/or ports and/or packet header field values

-> Packet Filtering Rules.

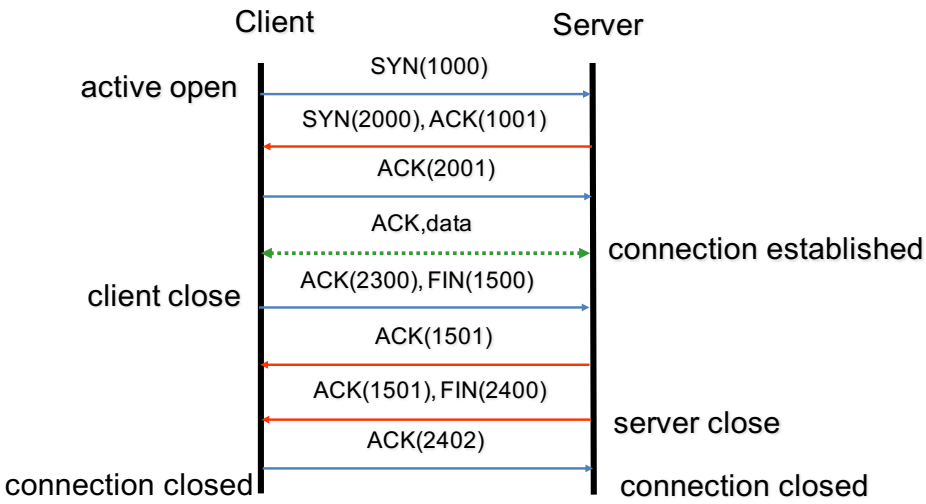
Rule	Dir	Action	Inside Addr	Inside Port	Outside Addr	Outside Port	Description
1.	In	Block	*	*	9.9.9.0	*	Don't let these guys in!
2.	In	Allow	*	*	6.6.6.6	*	We trust this host
3.	*	Allow	1.1.1.7	300	5.5.5.5	300	Very specific access
4.	Out	Allow	1.1.1.1	*	*	*	Allow this inside host access
5.	Out	Allow	1.1.1.0	*	4.4.4.3	80	Allow access to this service
6.	Block	*	*	*	*		Block anything else

Packet Filter Placement



- Packet filtering can be performed on incoming or outgoing packets at any interface of the router.

TCP Session



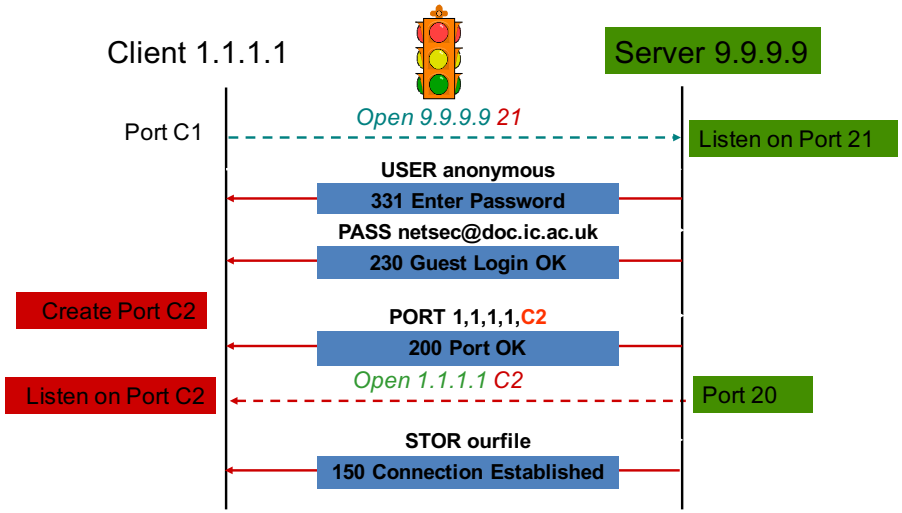
Filtering Rules Revisited

Rule	Dir	Action	Inside Addr	Inside Port	Outside Addr	Outside Port	Description
1.	Out	Allow	*	*	*	25	Mail anywhere (SMTP=25)

- In TCP, an initial open request packet does not have the ACK bit set, subsequent packets do

Rule	Dir	Action	Src Addr	Src Port	Dest Addr	Dest Port	TCP Flags	Description
1.	Out	Allow	*	*	*	25		Mail out
2.	In	Allow	*	25	*	*	ACK	Only replies allowed

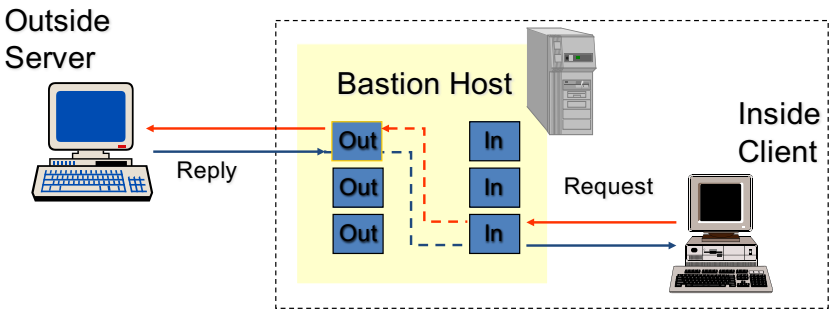
Filtering FTP (Dynamic Callbacks)



FTP Rules

Rule	Dir	Action	Src Addr	Src Port	Dest Addr	Dest Port	TCP Flags	Description
1.	Out	Allow	1.1.1.1	*	9.9.9.9	21		FTP outgoing
2.	In	Allow	9.9.9.9	21	1.1.1.1	*	ACK	Only replies allowed
3.	In	Allow	9.9.9.9	20	1.1.1.1	>1023		
4.	Out	Allow	1.1.1.1	*	9.9.9.9	20	ACK	

Circuit-Level Gateways



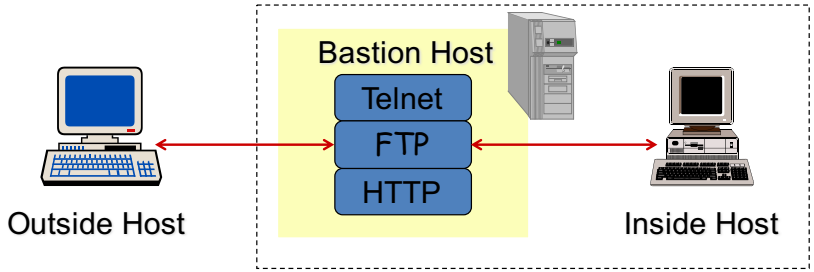
Relay's connections and maintains connection state. Some can also authenticate users.

Can drop connections based on destination, incorrect connection packets, time, volume, etc.

Normally used for Inside-to-Outside connections. Client normally recompiled to connect to CL-Gateway port

Good for auditing / accounting.

Application-Level Gateways



Like CL-GW's but application-specific (app-knowledgeable)

Can block/filter/report based on app-level msg content. Can scan for data leaks, viruses, etc. Can rewrite data.

Can be configured to limit application features

More processing overhead than CL-GW's

Bastion Host

Runs application-level and circuit-level gateways

Can run other servers too

Performs auditing / accounting.

Should run a "Trusted"/Secure OS

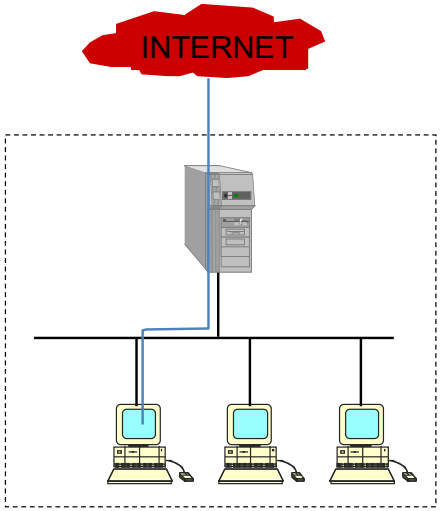
Administer via a dedicated terminal

Minimal OS

- Remove inessential applications, utilities, services, e.g cc, awk, sed, ld, X11
- Set file permissions, turn on file quotas, process limits etc
- No regular user accounts
- No NFS mounts
- Make filesystems read-only if possible

Firewall with One Bastion Host

- Earliest example of Firewall.
- Bastion host acts as filter and gateway. Also runs proxies for permitted services
- Note: Bastion Host must not automatically forward packets.
- On some Bastion Host Firewalls, users were allowed to login directly onto bastion host (bad!)



Firewall with one Packet Filter

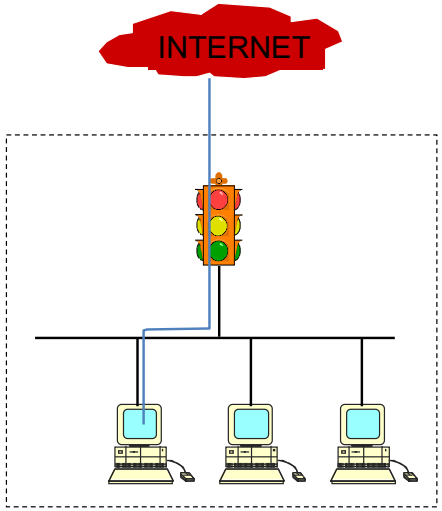
Block all packets for "services" that are not used.

Block all packets that explicitly set IP source routing or other unusual options.

Block access to/from particular sites, networks.

Allow incoming connections to predetermined services, block others.

Optionally allow internal hosts to initiate outgoing TCP connections.



Screened Host Architecture

PACKET FILTER

Block packets for services that we do not wish to cross firewall, and packets with IP source routing.

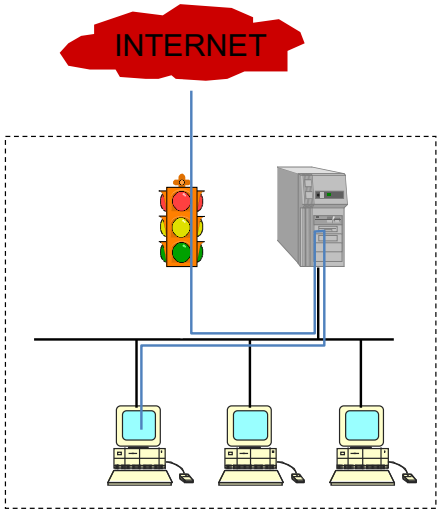
Block packets for internal network

Only pass packets for which the Source or Destination IP address is the Bastion Host

BASTION HOST

Run application-level gateways, e.g. web server, mail server.

Run circuit-level gateways to allow users on the inside to access servers on the internet



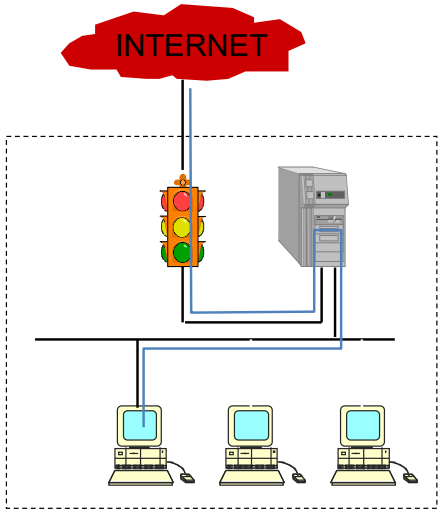
Screened Host (Dual-Homed)Architecture

DUAL-HOMED BASTION HOST

What if the packet filter is compromised?

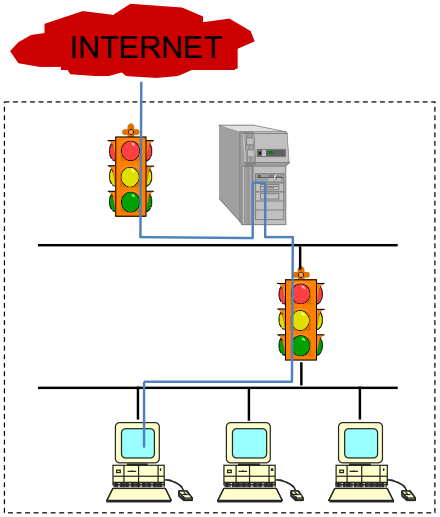
For greater security we can use a dual-homed bastion-host, i.e. a host that has 2 network-interfaces.

Now any intruder who gets past the packet filter will also need to get past the bastion host.



Screened Subnet Architecture

2nd inner packet filter prevents attack from compromised bastion host

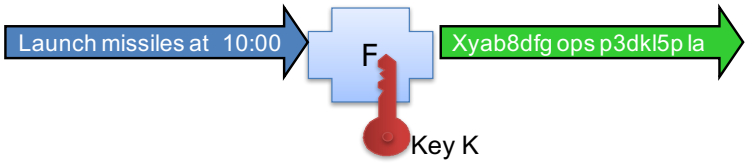


INNER PACKET FILTER

- Block packets for services that we do not wish to cross firewall, and packets with IP source routing.
- Block packets addressed for outer filter
- Only pass packets for which the Source or Destination IP address is the Bastion Host and for which the ports are for defined proxies on the gateway.
- Block everything else.

CRYPTOGRAPHY

Cryptography



Encryption is a transformation of information based on:

Substitution e.g. table look up

Transposition e.g. exchange bytes 1 & 3, 2 & 4 etc.

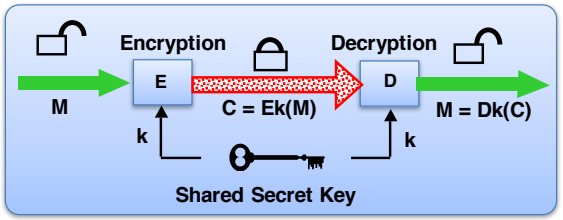
Combine with a key which specifies what substitution or transposition to use -> Encryption *Function* $E + Key\ k$

Decryption:

Inverse of encryption to obtain original information
Computation time required to decrypt without the key makes it impractical but not impossible.

Secret Key (Symmetric) Cryptography

Single Secret Key



Basis for Data Encryption Standard (DES)

- Same algorithm applied for Encryption and Decryption i.e. E = D
- 56 bit key applied to blocks of 64 bits of data – easily cracked

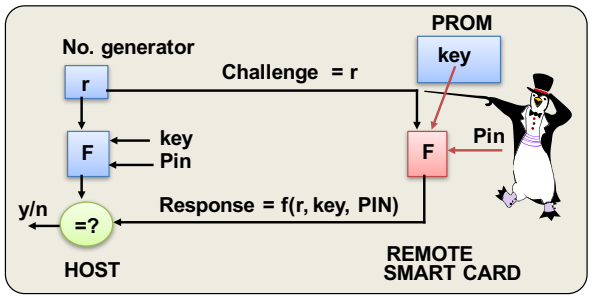
Advanced Encryption Standard (AES) selected in March 2001 after public competition: Rijndael from Belgium 128, 192 or 256 bit keys

Problems of key management

Remote Access Authentication

Smart Cards

- Embedded microprocessor + storage
- Preprogrammed with Key in PROM
- Require possession + knowledge of PIN



Challenge / Response Protocol

Host generates random number r which is sent to Smart Card. Card uses one way function F, seeded with cryptographic key K and PIN number to calculate response which is sent back to host. Host uses same function to check response.

Not susceptible to replay

Key Management

Secret key required for each partner or even session
-> Key distribution problem

Hierarchical Key Usage

- Distribute master key by courier e.g. monthly
- Use master key to encrypt new keys sent out e.g. daily
- Use current day key to encrypt session key.

Use session key distribution server e.g. Kerberos

Public Key Encryption

Rivest, Shamir, Adleman (RSA) Algorithm

Use two keys:

- Public Key Kpub sent out over network and stored with name servers – used by sender for encryption
- Secret Key Ksec used by recipient for decryption
- Cannot deduce secret key from public key

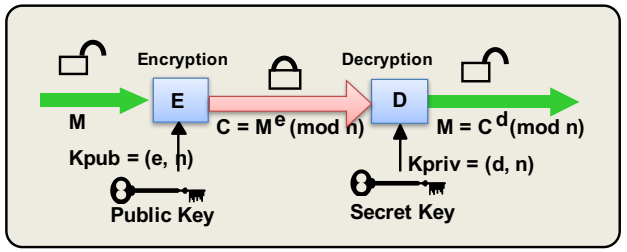
DKpriv(EKpub(M)) = M

i.e. decryption of encrypted message yields the original message – symmetric encryption & decryption

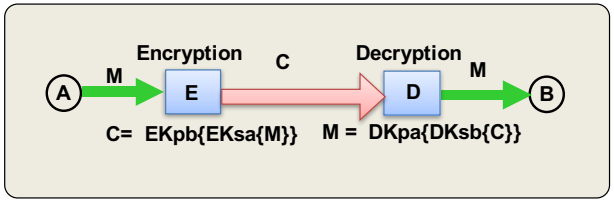
Public Key Encryption

Simple Key management - no secret keys to distribute
Computation intensive -> poor performance

Use public key system for distribution of secret session keys, and session keys for encrypting messages



Public Key Signatures



Use public key encryption where:

DKpub(EKpriv(M)) = M and DKpriv(EKpub(M)) = M

- A uses a secret key to encrypt a message
- B receives the message and decrypts it with A's public key.
- If the decryption is successful it must have been encrypted by A, i.e. "signed" by A ➔ authenticates sender.

Can be decrypted by anyone with A's public key

-> Encrypt with B's public key – only B can decrypt.

No verification of time or prevention of replay:

X could capture message and repeat it e.g. if message was from point of sale terminal in Company X to A's Bank to transfer £1000 to X's account.

A can pretend secret key Ksec_a was stolen and deny sending message.

Need to store complete encrypted message for later verification.

What is needed?

Message Digest

A one way function on a message -> a code which can be used to check its integrity e.g. checksum or hash function.

Detects message modification but does not provide secrecy.

Sender computes digest, appends to message and receiver re-computes to check

- Given message m it is easy to compute H(m)
- Given H(m) it is impossible to compute m
- No 2 messages can generate same H(m)

Use cryptographic hashing or encrypt digest.

Hashing is faster than encryption eg MD5 or Secure Hashing Algorithm (SHA)

Pad message to multiple of 512 bits & mangle blocks of 512 bits to produce 128 bit (MD5) or 160 bit code (SHA).

Certification Problem

Alice has generated a new electronic work of art in the form of a large 50MB file. She wants to be able to prove that she is the originator of the work so requires a proof from an origin authority (OA).

Using public key cryptography and message digests show

- 1) what information Alice sends to the OA
- 2) the response from the OA.

Certification Solution

Alice -> OA: $Alice, K_{privA}\{Alice, H(file)\}$

Alice sends a digest of the file to obtain the digital signature, encrypted with her secret key. OA uses Alice's public key to decrypt and prove it came from Alice

OA-> Alice : $K_{secOA}\{Alice, H(file), t\}$

Digital signature includes Alice's ID and time stamp of when it was generated. This can be checked with the OA's public key.

Digital Signature

Needed for legally binding documents.
E.g. Joe sends message to stockbroker to buy 10,000 shares in HAL Plc which goes bust next day.

Joe may deny sending message or stockbroker may change number to 100,000 to offload other shares he had bought for himself.

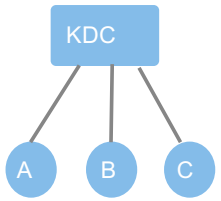
- Verify author, date and time of signature
  -> non-repudiation of origin
- Verify contents at time of signature
  -> receiver or someone else cannot modify contents or forge message claiming it came from sender.

- Signature must be verifiable by third party to resolve dispute.
- Signature must be unforgeable
- Must be practical to retain copy of digital signature in storage.

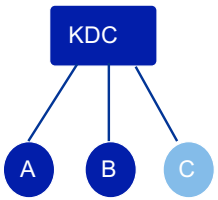
Need Notarisation Service (Arbiter) to prevent non-repudiation by sender or subsequent claims of loss of secret keys.

SYMMETRIC KEY DISTRIBUTION AND AUTHENTICATION

Key Distribution Centres (KDC)



- How does A establish a secure session key with B?
- Assumption: KDC knows secret key of all entities.



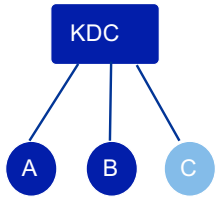
A -> KDC: $E_A\{\text{request } K_S, B\}$

KDC -> A: $E_A\{K_S\}, E_B\{K_S\}$

A->B: $E_B\{K_S\}$

Observations:

- B does not know to whom he is talking
- Timestamps and/or nonces need to be added.
- Variations can include: B distributing the key to A or the KDC sending both parties the session key
- Revocation can be done by the KDC easily.



Needham-Schroeder (or any of the variants)

Key distribution

Authentication

A -> KDC: A,B, Req, N_A

KDC -> A: $E_A\{Ks, Req, N_A, E_B\{K_S,A\}\}$

A->B: $E_B\{K_S,A\}$

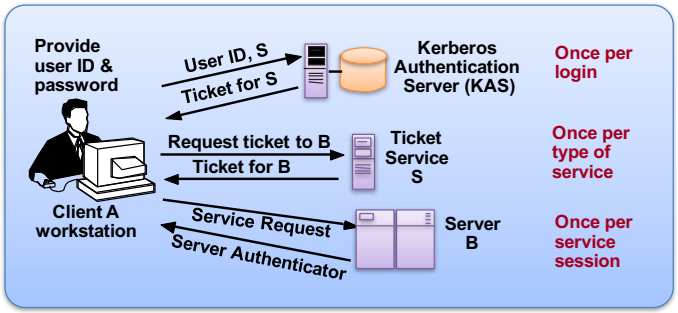
B->A: $E_{Ks}\{N_B\}$

A->B: $E_{Ks}\{N_B-1\}$

Kerberos Authentication Service (V4)

Separate authentication and ticket granting service

Users and Service providers must register with Kerberos



Kerberos Authentication Service (V4)

KAS stores identities, server private keys and encrypted user passwords used to generate private user keys

- Must be secure + master/slave redundancy

Ticket service can be replicated – does not hold secret data.

Use time stamps to detect replay.

- Lifetime of tickets issued by ticket server must be less than the lifetime of ticket originally issued by Kerberos.
- Servers accept tickets within limited window of timestamp to prevent replay or masquerading.
- Require clock synchronisation (within minutes)

Tickets and Authentication

Ticket for Server X

- Secure transfer of authenticated identity of user to server, plus optional authorisation data that can be used for access control.

Tax = Kx {Aname, Arealm, Aaddr, Kax, TS, TL, Xname}

- TS = ticket timestamp
- TL = ticket lifetime
- Can be reused
- Generated by KAS for ticket service, or by ticket service for Server B

Authenticators

Authenticator for Server X

- Proves identity of user presenting ticket is same as that to whom ticket was issued.

Xax = Kax{Aname, Arealm, Appl. data, TS}

- Can only be used once when session with B is established
- Generated by client A

Kerberos Interactions

A logs in and provides userid & password. Workstation uses password to generate Ka then discards password.

A -> KAS: Options, Aname, Arealm, Sname, TS

- KAS generates session key Kas & ticket Tas for ticket server S
- KAS uses A's password to generate Key Ka

KAS -> A: Ka{Kas,TS2, TL, Tas}

- Only client A can decrypt this to obtain ticket Tas
- Use Kas to create authenticator Xas for S

A -> S: Bname, Tas, Xas

- Decrypt Tas to get session key Kas. Check authenticator & ticket.
- Generate session key Kab and ticket Tab

S -> A: Kas{ Kab, Bname, TS, Tab}

- Decrypt to get ticket and session key for B. Create authenticator for B

A -> B: Tab, Xab

- Decrypt ticket to get session key, check authenticator, generate response by adding 1 to TS in authenticator. This authenticates Server B to A.

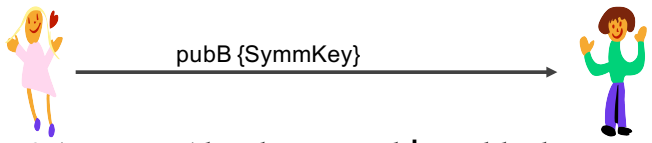
B -> A: Kab{TS +1}.

- Only server B can generate this response.

CERTIFICATES AND PUBLIC KEY INFRASTRUCTURE

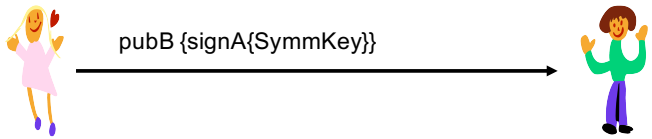
Public Key Dist. of Symmetric Keys

Take 1



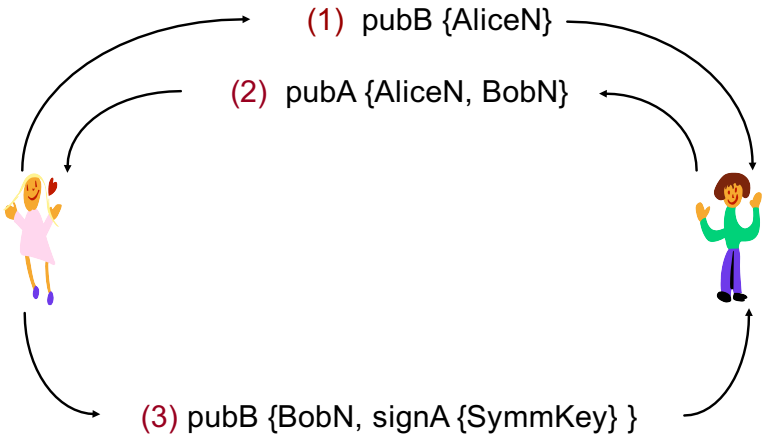
- Assumes Alice knows Bob's public key
- Bob has no idea he is speaking with Alice

Take 2



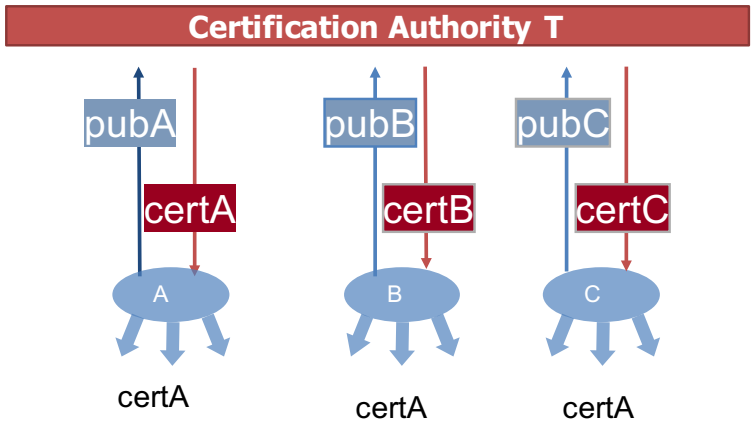
- Assumes both Alice and Bob know each other's public key
- No idea of freshness

Take 3



assumes Alice and Bob know each other's public key – How?

Certificates



certA= {idA, pubA, expiresT} signT

$$certA = K_T^{-1}(A, K_A, T_{exp})$$

Certificates

A certificate is a statement that binds an identity to a cryptographic key.

The certificate is only as good as the steps taken by the certification authority to verify the identity and to protect the binding.

The public key of the certification authority is assumed to be known by the communicating parties.

What is the identifier? Does it need to be unique?

Multiple certification authorities?

X 509 Certificates

Associate public key with user or service

Signed by a certification authority (CA)

Not forgeable – can be stored in a directory

Issuer can revoke specific certificates – hold revocation list.

Certificate Fields:

Version of certificate format
Serial number – unique within CA

Signature algorithm

Issuer name + unique identifier

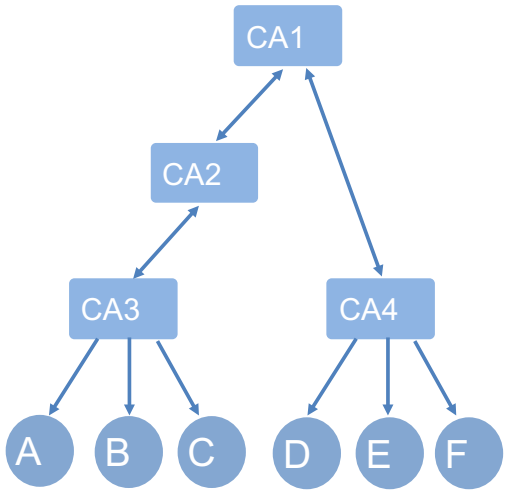
Period of validity

Subject name + unique identifier – to whom the certificate was issued

Extension fields for additional info about subject or issuer eg public key to be used to verify signature

Signature – hash code of all other fields encrypted with issuer's private key.

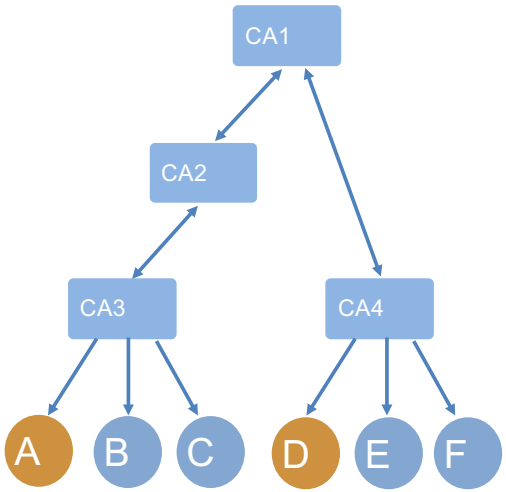
X.509 - Certification authority Hierarchy



Each pair (parent-child) of CAs create a certificate for each other, e.g. CA1 for CA2 and vice-versa.

Certificates can be used to build a certification path.

How can A verify D's certificate



A needs to verify signCA4(pubD)

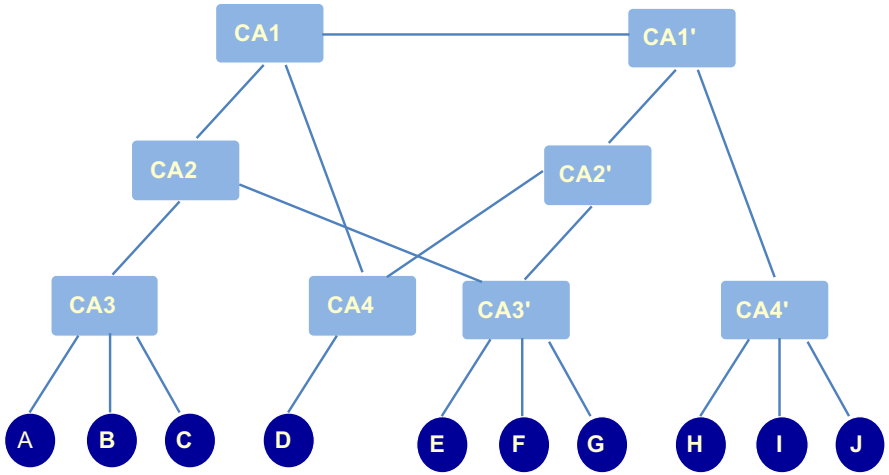
Gets from CA Hierarchy:

1) signCA3 (pubCA2)
   signCA2 (pubCA1)
   signCA1 (pubCA4)

The certificates allow A to:

1) verify pubCA2
2) verify pubCA1
3) verify pubCA4
4) verify pubD

PKI Topologies



Summary

Physical security impossible in Distributed Systems

Networks inherently less secure than centralised computer systems as they are more susceptible to wire tapping etc.

Cannot trust distributed workstations

Dependence on distributed systems and value of information they manipulate make security an essential aspect