Client Server - Security
Jan Newmarch jan@ise.canberra.edu.au

1. Security

- Security of the organisation affects 100 % programs you write
- Many more opportunities for security problems in client/server applications

2. Security threats

- Unauthorised access
- Disclosure of information
- Modification of information
- Denial of service

3. Orange Book

- D - minimal
- C1 - discretionary access control
- C2 - controlled access protection
- B1 - labelled security protection
- B2 - structured protection
- B3 - security domains
- A1 - verified design

4. Orange Book

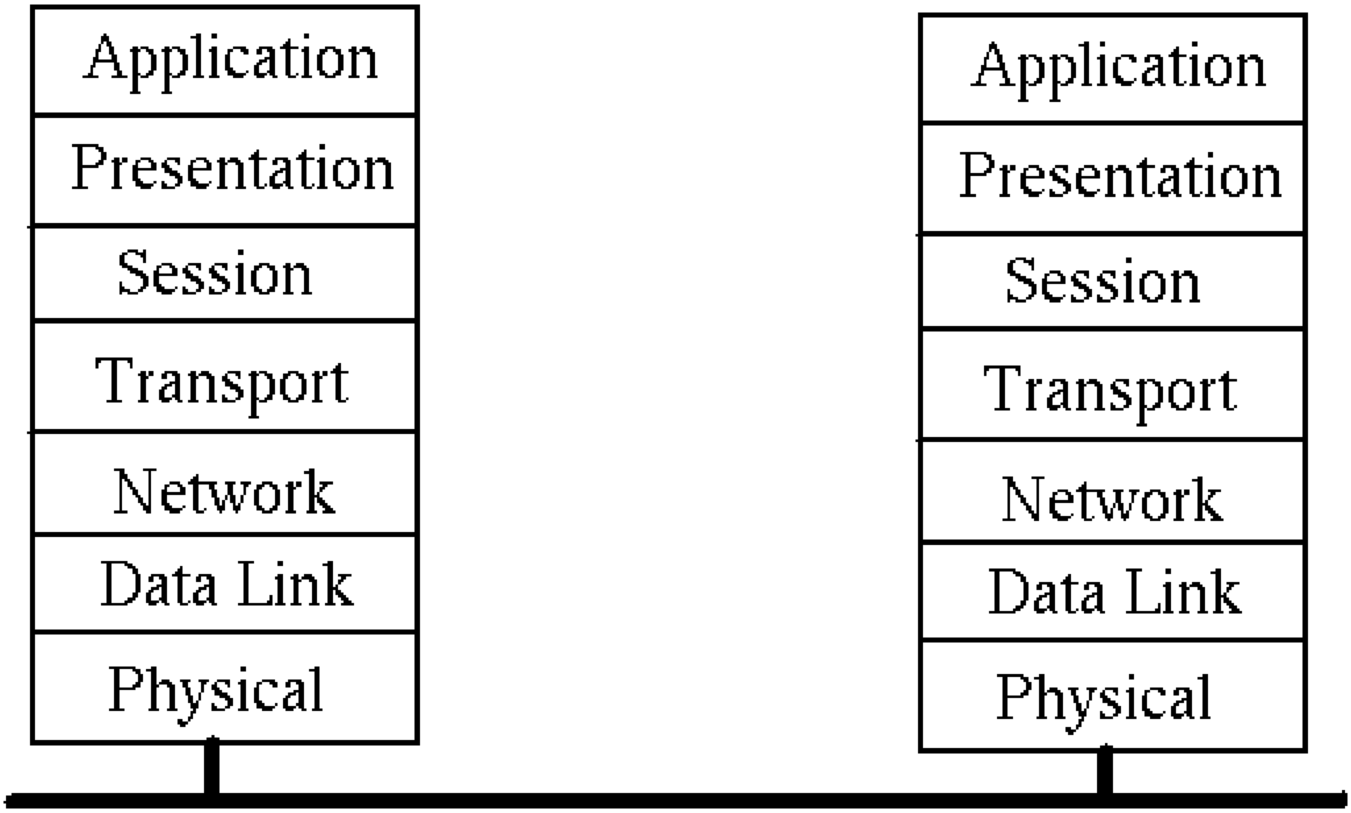
- C1
  - separation of users and data
  - indentification and authentication
  - integrity base
  - tested controls
- C2
  - finer grained access control
  - login procedures
  - auditing of security events
- B1
  - security policy model
  - labelling (e.g. TOP SECRET)
  - mandatory access control

5. Orange Book

- B2
  - formal security policy model
  - device labels
  - DAC and MAC (Message Authentication Code) (fancy checksums)
  - covert channel control
  - more extensive testing

- B3
  - reference monitor mediates all access
  - security administrator support
  - system recovery required
  - auditing expanded
- A1
  - formal design and verification

6. ISO Protocol



7. ISO Security - Functions and Levels

- Peer entity authentication (3, 4, 7)
- Data origin authentication (3, 4, 7)
- Access control service (3, 4, 7)
- Connection confidentiality (1, 2, 3, 4, 6, 7)
- Connectionless confidentiality (1, 2, 3, 4, 6, 7)
- Selective field confidentiality (6, 7)
- Traffic flow confidentiality (1, 3, 7)
- Connection integrity with recovery (4, 7)
- Connection integrity without recovery (4, 7)
- Connection integrity selective field (7)
- Connectionless integrity selective field (7)
- Non-repudiation at origin (7)
- Non-repudiation of receipt (7)

8. ISO Security - Mechanisms

- Peer entity authentication
  - encryption
  - digital signature
  - authentication exchange
- Data origin authentication
  - encryption
  - digital signature
- Access control service
  - access control lists
  - passwords
  - capabilities lists
  - labels

9. ISO Security - Mechanisms

- Connection confidentiality
  - ecryption
  - routing control
- Connectionless confidelity
  - encryption
  - routing control
- Selective field confidelity
  - encryption
- Traffic flow confidelity
  - encryption
  - traffic padding
  - routing control

10. ISO Security - Mechanisms

- Connection integrity with recovery
  - encryption
  - data integrity
- Connection integrity without recovery
  - encryption
  - data integrity
- Connection integrity selective field
  - encryption
  - data integrity
- Connectionless integrity
  - encryption
  - digital signature
  - data integrity

11. ISO Security - Mechanisms

- Connectionless integrity selective field

  - encryption
  - digital signature
  - data integrity
- Non-repudiation at origin
  - digital signature
  - data integrity
  - notarisation
- Non-repudiation of receipt
  - digital signature
  - data integrity
  - notarisation

12. Security Policies

- Anything not explicitly permitted is prohibited
- Anything not explicitly prohibited is permitted

13. Security Functions Needed

- Identification
- Authentication - proof of identity
- Authorisation - privileges belonging to identity
- Delegation and administration
- Recording
- Monitoring

14. Software Vulnerability

- Any piece of software has bugs
- The more complex the software, the more bugs
- Software dealing with security will have security bugs

15. Security Locations

- Host security
- Network security

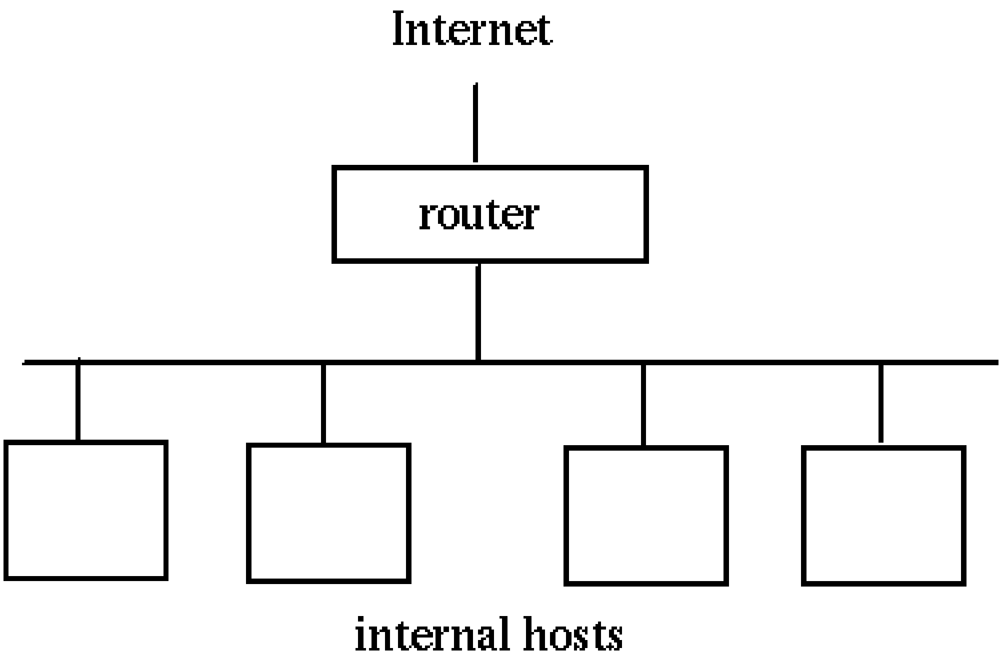
16. Host Security

- Passwords
- Access files (.rhosts)
- Physical access

17. Network Security

- Isolate network
- Protect behind firewalls
- Use proxy servers

18. Routers



19. Firewalls

- A firewall is a router that controls access as well as routes
- Some packets are passed, some are discarded
- The firewall controls all access between the network and internet
- A "filter" uses rules to decide what passes and what doesn’t

20. Filter Mechanisms

- Ethernet packets contain ethernet header information, plus IP data
- IP packets contain IP header information, plus TCP/UDP data
- So on, upto particular protocol data
- The filter can examine the IP header and the TCP/UDP header

21. IP Header Information

An IP packet has a header containing the following information

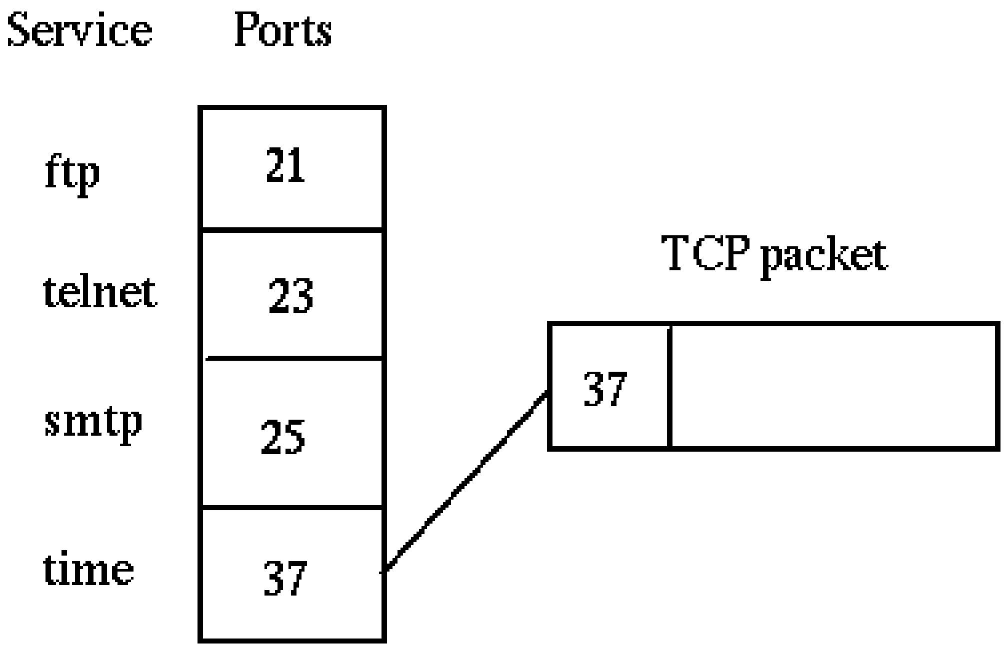
- IP source address
- IP destination address
- protocol type (e.g TCP)
- options (little used)
- packet body

22. Filtering by IP Header

- Block outward access by discarding all packets with IP destination address external to organisation
- Block inward access by discarding all packets with IP source address external to organisation
- Fragmentation of IP packets makes it harder to filter on packet data information

23. TCP and UDP Ports

Services are identified by numbers known as ports



24. TCP Header Information

A TCP packet has a header containing the following information

- TCP source port
- TCP destination port
- ACK bit
- data

25. Filtering by TCP Header

- First packet opening connection has ACK bit unset, later ones have it set
- Block inward TCP calls by discarding packets with ACK bit unset and external IP source
- All later packets from same source are discarded by recipient as garbage because there was no start packet

26. UDP Header Information

A UDP packet has a header containing the following information

- UDP source port
- UDP destination port
- data

27. Filtering by UDP Header

- Much harder as there is no ACK bit, so can't tell when an external source is trying to connect
- No sequence order or delivery guarantee, so harder to use content

28. Remote Procedure Call

- Portmapper is on port 111
- RPC servers use a random port
- RPC servers register port and RCP identifier with portmapper
- RPC clients ask portmapper for port of RCP server
- Clients then establish connection directly with server
- Clients may ask portmapper to forward RPC requests directly to server

29. Spoofing Header Information

- Source address not guaranteed
- Easier to spoof UDP than TCP

30. Proxy servers

Client queries local server that can

- filter request
- answer locally
- forward request to remote server

31. Security of Common Applications: Telnet

- Outbound telnet service has
  - local client, remote server
  - outgoing packets from client to server
  - incoming packets from server to client
- Inbound telnet service has
  - local server, remote client
  - incoming packets from client to server
  - outgoing packets from server to client

32. Telnet: Outgoing Packets, From Client

- Packets contain keystrokes
- IP source address: local
- IP destination address: remote
- IP packet type: TCP
- TCP source port > 1023
- TCP destination port: 23

33. Telnet: Incoming Packets, From Server

- Packets contain server responses
- IP source address: remote
- IP destination address: local
- IP packet type: TCP
- TCP source port: 23
- TCP destination port > 1023, same as source for outgoing packet

34. Telnet: Incoming Packets, From Client

- Contains user keystrokes
- IP source address: remote
- IP destination address: local
- IP type: TCP
- TCP source port > 1023
- TCP destination port: 23

35. Telnet: Outgoing Packets, From Server

- Contain server responses
- IP source address: local
- IP destination address: remote
- IP packet type: TCP
- TCP source port: 23
- TCP destination port > 1023, same as source port for incoming packet

36. Telnet Filtering

- Known port 23
- Easier to filter TCP than UDP
- Simple filtering rules
- Easy to block/allow telnet

37. FTP Filtering

- Problem is to *allow* ftp from local client.
- In establising connection from local client, no problem as connection is from local to remote. (The remote server may block your connection though.)
- In data transfer, remote server must open data channel to your client. This may get blocked by ACK bit unset at client end.
- Resolve this by forcing client to initiate data transfer on a new port, not client

38. X11 Filtering

- The problem is with a local X server and a user wishing to use an application external.
- The remote client application must open the X connection, and this may be blocked by the ACK bit.

39. RPC Filtering

- Only fixed port is that of portmapper, 111
- All initial RPC calls are addressed to this
- Portmapper should be blocked to external access because it is a security risk by itself
- External process can still scan all ports looking for RPC servers
- Most common RPC services use UDP, so can only block by blocking all UDP.
- Use TCP for RPC if you need to use RPC through firewall

40. Lessons for Client/Server

- Verify IP origin
- Use logging
- Care with user configurable files
- Care with passwords
- Use handshaking
- Keep security layer simple
- Encrypt sensitive data
- Design protocol to allow security checks
- Ensure your protocol is allowed by security manager if crossing firewalls
- More likely to find UDP blocked than TCP

41. Use Logging

- Log all activities, to provide an audit trail
- Inspect the audit trail
- Failed accesses are the most interesting

42. IP verification

- Using the IP origin, find the IP name
- Using the IP name, find the IP address(es)
- Verify that the origin address is in these

43. User Configurable Files

- Ensure that users can't compromise security
  - accidentally
  - deliberately

44. Passwords

- Ordinary passwords should not be sent across insecure network in plain text
- rlogin sends them in plain text
- telnet uuencodes them before sending (7 bit representation of 8 bit data), but this is not secure
- Receiving program should not accept encrypted passwords, in case they are just duplicates
- One-shot passwords are secure either in plain text or encrypted

45. Handshaking

- Make the client and server negotiate indentities

46. Simple Security Layer

- If the security is too complex, then it will have exploitable bugs

47. Encrypt Data

- Sensitive data should not be sent in plain text
- Variety of encryption algorithms, with different costs and value

48. Protocol Design

- Design protocol to allow validation and verification of every transaction, if security is an issue

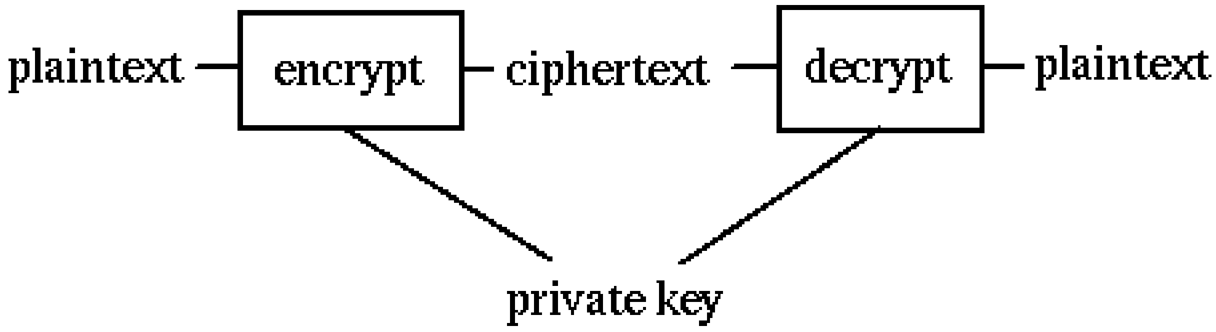
49. Encryption

- Protect data from prying
- Digitally sign data
  - proof of origin
  - proof of integrity
- Source is plaintext
- After encryption, have ciphertext

50. Ciphers

- Substitution ciphers are easy (e.g 'a' replaced by 'b', 'b' replaced by 'c', etc)
- Substitution ciphers are relatively easy to crack
- Vernam ciphers (one-time pad) can be totally secure, but are impractical

51. Private Key

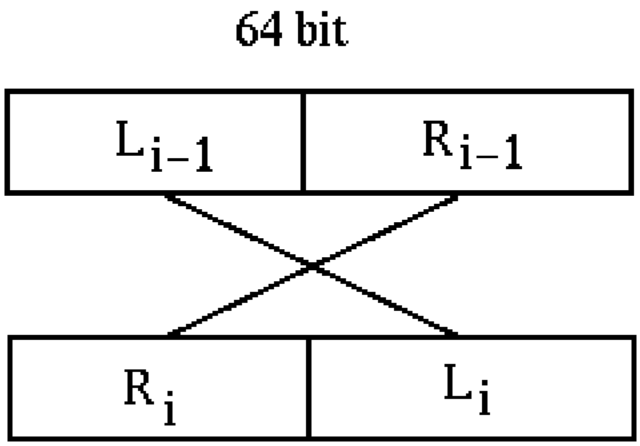


- Both encryption and decryption use same key
- Key must be kept secret from others
- Key must be known by at least two people

52. DES

- US Govt standard since 1977
- Uses 56-bit key
- "Strong" algorithm
- Strong algorithms are forbidden exports from the US
- The US Govt can probably crack it
- Triple-DES more secure. Uses DES 3 times with 2 keys

53. DES Algorithm



- There are 19 iterations of basic algorithm
- Li = Ri-1 and Ri = Li-1 bitwise-and f(Ri-1, Ki)
- Ki calculated by
  - Split 56 bit key into 2x28 bits
  - left rotate each by iteration index

54. DES Algorithm

- f() is calculated in 4 steps
  - Ri-1 expanded to 48 bit E
  - E and Ki exclusively or-ed
  - 48 bit result split into 8x6 bit chunks
  - Each 6 bit chunk has substitution cipher applied, resulting in 4 bit chunk
  - 8x4 bit outputs permuted

55. Private Key Problems

- Distribution
- If each key known to exactly 2 people, then n people require n(n-1)/2 keys, which gets large quickly
- Can't send someone a secret message (e.g. the key) unless you can already send them a secret message

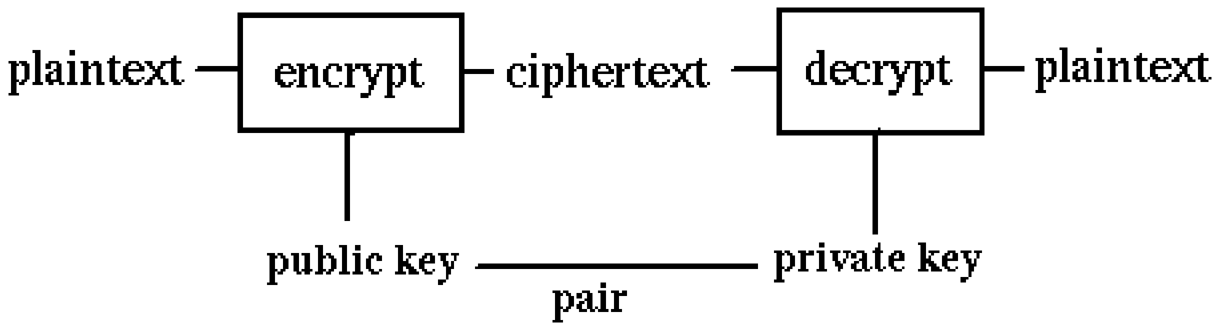
56. Key Distribution Centre

- KFC holds n keys, one for each of n individuals
- If A, B want to talk, both ask KDC for a new key
- KDC invents new key, encrypts it separately for A, B who each decrypt it
- If KDC is compromised, all keys are compromised

57. Session Keys

- Diffie-Hellman exponential key exchange allows a secret key to be generated by public exchange of information between two people
- A and B exchange two integers n, m with 1 < n < m, m large
- Each decides on a secret number a and b
- A transmits the number n^a (mod m)
- B transmits the number n^b (mod m)
- Each can then calculate (n^a)^b = (n^b)^a = n^(a*b) (mod m)
- This is the private key, for use in, say, DES
- It is hard to calculate this from the public information
- This does not authenticate the other user

58. Public Key



- 2 keys needed, one to encrypt one to decrypt
- Both keys generated at the same time
- Both keys needed
- One key can't easily be derived from the other

59. Public Key Encryption

- A wants to send a private mesage to B
- A has B's public key (so do lots of other people)
- A encrypts message using B's public key
- B is only person who can decrypt the message using his private key

60. Digital Signature

- A wants to send a message to B
- B wants to verify it came untampered from A
- A encrypts message using A's private key
- B decrypts it using A's public key
- If B gets garbage
  - message didn't come from A; or
  - message was changed in transit
- Signature may be detached from text, and sent separately

61. Secure Verified Data

- A signs the message using A's private key
- A encrypts the cipher using B's public key
- Result can only be decrypted using B's private key (secure)
- Can then be verified by A's public key

62. Session Keys

- Public key algorithme are usually slower than private key
- Get a secure channel using public key
- Use this to exchange a private key
- Switch to faster private key encryption

63. RSA Public Key Algorithm

- Choose two very large primes (simple e.g., P=47, Q=71) and set N = P*Q = 3337 and M = (P-1)*(Q-1) = 3220
- Choose E relatively prime to M, e.g E=79
- Set D = E^-1 (mod M) = 79^-1 (mod 3220) = 1019
- Public key is (N, E) = (3337, 79)
- Private key is (N, D) = (3337, 1019)
- To encrypt n, C = cipher = n^E (mod N) = n^79 mode 3337
- To decrypt C, n = C^D mod N

64. Pretty Good Privacy

- PGP s/w based on RSA algorithm
- Some versions can't be used outside of the US, but some can
- Key management
- Encrypt and decrypt documents (files)
- Digitally sign documents, with optional separate signature

65. PGP Keys

- Keys are stored in
  - pubring.pgp
  - secring.pgp
- pubring contains the keys you collect of others
- secring contains your secret keys
- secring is password protected

66. PGP Encryption

- Encrypt file by

```
pgp -e file userid
```

Encrypted file is old file with .pgp extension

- Sign a file with your private key

```
pgp -s file
```

creates file with .pgp extension

- Encrypt and sign by

```
pgp -se file userid
```

67. PGP Decryption

- Decrypt a file by

```
pgp file.pgp
```

- Verify signature by

```
pgp file.pgp userid
```

- Decrypt and verify by

```
pgp file.pgp userid
```

68. PGP Key Management

- To generate a public/private key pair

```
pgp -kg
```

You will be prompted for a "pass phrase" as a password protection on the keyring files

- To extract a public key

```
pgp -kx
```

69. PGP Use in Programs

- No library available
- Use external program call e.g. in C

```
system("pgp -e file");
```

- Available on DOS from D:
- Available
  on Unix already installed. Source is in /pub/local/src/pgp...gz

70. Web of Trust

- How do you get the public key of a person?
- You can get it directly from them
- You can get it from someone you trust
- A public key can be signed by any number of people
- If you completely trust the person who signed it, then you can accept the key
- If you partially trust two or more people who have signed it, then you can accept the key
- Otherwise, you have no confidence in the key
- AUUG will sign keys of people by getting it directly from them
- Because I trust both AUUG people and the method they are using for authenticating keys, I will trust AUUG-signed keys

71. References

- W.R. Cheswick and S. M. Bellovin, *Firewalls and Internet Security* Addison-Wesley, 1994, ISBN 0-201-63357-4
- S. Garfinkel, *PGP - Pretty Good Privacy* O'Reilly and Associates, 1995, ISBN 1-56592-098-8