Security

Outline

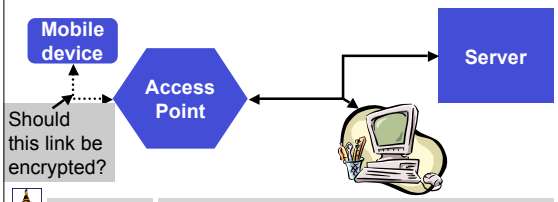
- Encryption Algorithms
- Authentication Protocols
- Message Integrity Protocols
- Key Distribution
- Firewalls

Risk analysis

- Important to understand threat and perform risk analysis
  - No system is "secure", systems usually trade security for performance, ease of use etc.
  - If information is worth **x** and it costs **y** to break into system and if (**x** < **y**), then not worth encryption
  - Wasteful to build a system that is more secure than is necessary
    - Network data is transient (unlike stored data)

End-to-end argument

- End-to-end argument is appropriate for building a secure system
  - Perform security at lower levels if simple and does not impact performance
  - Higher levels usually know best regarding data integrity requirements

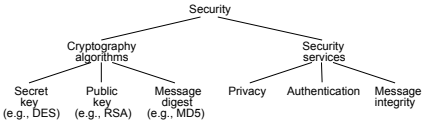


Security Attacks

- Social engineering attacks
  - Preys on people gullibility (good nature), hardest to defend
    - E.g. I once got an unlisted number from a telephone operator because I sounded desperate (I was, but that was not the point)
    - E.g. Anna kour*va virus
    - E.g. If I walk in with coupla heavy looking boxes into the elevator to go to Boyd 5th floor (at night) would you let me in? You can go into "secure" companies by looking like you "belong" there
- Denial of service attacks
  - Network flooding, Distributed DOS, holding resources, viruses

Overview

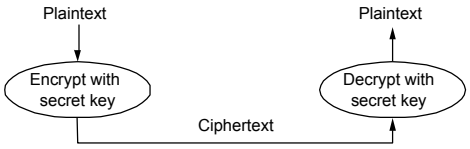
- Cryptography functions
  - Secret key (e.g., DES)
  - Public key (e.g., RSA)
  - Message digest (e.g., MD5)
- Security services
  - Privacy: preventing unauthorized release of information
  - Authentication: verifying identity of the remote participant
  - Integrity: making sure message has not been altered



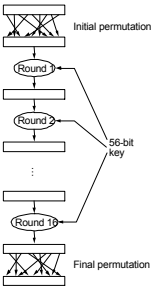
Encryption methods

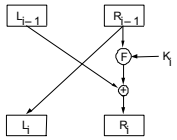
- Symmetric cryptography
  - Sender and receiver know the secret key (apriori)
    - Fast encryption, but key exchange should happen outside the system
- Asymmetric cryptography
  - Each person maintains two keys, public and private
    - M ≡ PrivateKey(PublicKey(M))
    - M ≡ PublicKey (PrivateKey(M))
  - Public part is available to anyone, private part is only known to the sender
  - E.g. Pretty Good Privacy (PGP), RSA

Secret Key (DES)

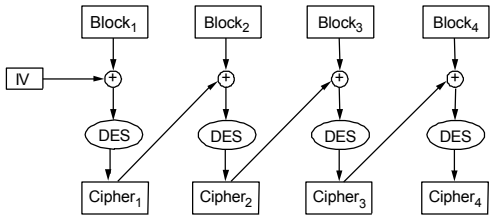


- 64-bit key (56-bits + 8-bit parity)
- 16 rounds
- Each Round





- Repeat for larger messages



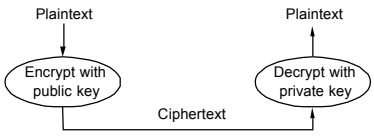
RSA

- Named after Rivest, Shamir and Adleman
  - Only receiver receives message:
    - Encode message using receivers public key
  - Only sender could've sent the message
    - Encode message using sender's private key
  - Only sender could've sent the message and only receiver can read the message
    - Encode message using receivers public key and then encode using our private key

Strength

- Strength of crypto system depends on the strengths of the keys
- Computers get faster – keys have to become harder to keep up
- If it takes more effort to break a code than is worth, it is okay
  - Transferring money from my bank to my credit card and Citibank transferring billions of dollars with another bank should not have the same key strength

Public Key (RSA)



- Encryption & Decryption

$c = m^e \bmod n$

$m = c^d \bmod n$

RSA (cont)

- Choose two large prime numbers p and q (each 256 bits)
- Multiply p and q together to get n
- Choose the encryption key e, such that e and (p - 1) x (q - 1) are relatively prime.
- Two numbers are relatively prime if they have no common factor greater than one
- Compute decryption key d such that
  - $d = e^{-1} \bmod ((p - 1) \times (q - 1))$
- Construct public key as (e, n)
- Construct public key as (d, n)
- Discard (do not disclose) original primes p and q

My Public Key

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGPfreeware 7.0.3 for non-commercial use <http://www.pgp.com>

mQGiBDqtLPwRBADnG0+9IkDvI8t/3wdL3CSO4DytEH0NjrNwAYYIaewp3MklsxkP
p6iVblwiiCH4T4Nqkaru+kaEQ1hSTa7E/F9yQCWN5J0u1U7mtgTKFyt7VG0txAVx
tV7TuyxNogJkpm2BqoKqqUdCdbm+GurX/G2ynbINjEOvhcy0i1ttxgyDrwCg/8HZ
tM0i06VVNcR/QCmA+JdHGwMEAIjXLVV97huEtpuWDiq4J53ecV3HXQm6XoUZq4Sc
n+nsvXe4UD+6ldub/riOqBy22fBBAKhUsM3lGFgr7h19X3RGdw/yBVox+BLajpW+
F+ddjJAVSFeTvNanhnXL9a3nwCThb4aEUTdD61kgoUWJl2BnsK1DUSo2X6AsZYo+
GknOA/92dUNYUzspPLkXvPjOo+uJErZA4aN+UYsJwD3AlYugVLkc3nQBQySO4bAR
XitjnN0DA6Kz/j6e+cqReCyEuBnPtaY/Nn/dAn1lgUlJ/EtKQ9J4krI3+RxRmlpY
UtWyTaakV/QCXkB/yB9i6iAfsCprlcRSpmZAGuNXr+pHTHB0ILQmU3VyZW5kYXIg
Q2hhbmRyYSA8c3VyZW5kYXJAY3MudWdhLmVkdT6JAFgEEBECABgFAjqtLPwICwMJ
CAcCAQoCGQEFGwMAAAAACgkQlU7dFVWfeisqTACfXxU9a1mbouW2nbWdx6MHatQ6
TOgAoM9W1PBRW8Iz3BIgcnSsZ2UPNJHDuQINBDqtLPwQCAD2Qle3CH8IF3Kiutap
QvMF6PlTETlPtvFuuUs4INoBp1ajFOmPQFXz0AfGy0OplK33TGSGSfgMg7116RfU
odNQ+PVZX9x2Uk89PY3bzpnhV5JZzf24rnRPxfx2vIPFRzBhznzJZv8V+bv9kV7H
AarTW56NoKVyOtQa8L9GAFgr5fSI/VhOSdvNILSd5JEHNmszbDgNRR0PfIizHHxb
LY7288kjwEPwpVsYjY67VYy4XTjTNP18F1dDox0YbN4zISy1Kv884bEpQBgRjXyE
pwpy1obEAxnIByl6ypUM2Zafq9AKUJsCRtMIPWakXUGfnHy9iUsiGSa6q6Jew1Xp
Mgs7AAICCACLxNC3Vth553Y90JCVyM9mPWzvrkjfEGfBiCFDZ0HONW81ywUyV6jT
O/1sUsgR7jGB26XBsnIY96a9WTpUoI+20YstFLRj8sXOVXuaP/YTmgSLv82O6SWd
Bze1S0YJcU31/zdCftsz67UWT8vg39yeGyQ5KQP83p9DKpi4Z5K4M29p8eCt9BY+
kid94h9+16ZT8JLF0iEwGapZvpaTucCNoC8t6CKPto0dGpkYp7uBYoSzLgNvUh2n
BjGVEmLuioabqbOaomDErITY2iNcW3CCgjjYvgg/Hnu7HB2xKzuVUN1NTGogcuNI
Yx88mi+d/HxTY6YNr9xNW0f0pWkZDVB0iQBMBBgRAgAMBQI6rSz8BRsMAAAAAAoJ
EJVO3RVVn3orYhIAoIQPxGvHmX8c6kaAZqko1zYCeixcAJ9tp5h/KQZrIN/BpyTW
9Xgv4qxKEA==
=Pv5O
-----END PGP PUBLIC KEY BLOCK-----

Public Key Infrastructure (PKI)

- Process of issuing, delivering, managing and revoking public keys
- E.g. Secure Sockey Layer (SSL)
  - Client C connects to Server S
    1. C requests server certificate from S
    2. S sends server certificate with Spublic to C
    3. C verifies validity of Spublic
    4. C generate symmetric key for session
    5. C encrypts Csymmetric using Spublic
    6. C transmits Csymmetric(data) and Spublic(Csymmetric) to S