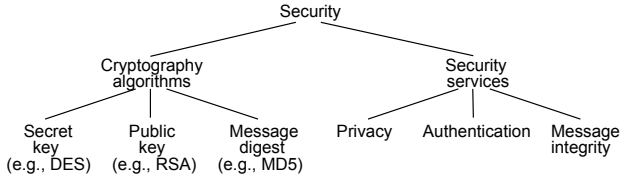
Security

Outline

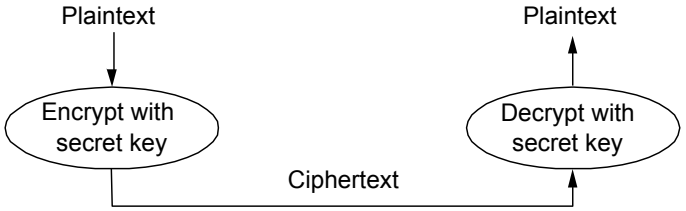
- Encryption Algorithms
- Authentication Protocols
- Message Integrity Protocols
- Key Distribution
- Firewalls

Overview

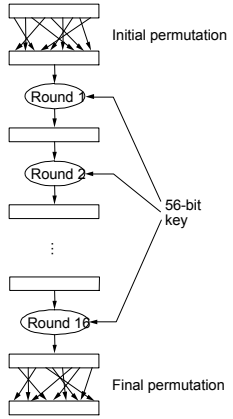
- Cryptography functions
  - Secret key (e.g., DES)
  - Public key (e.g., RSA)
  - Message digest (e.g., MD5)
- Security services
  - Privacy: preventing unauthorized release of information
  - Authentication: verifying identity of the remote participant
  - Integrity: making sure message has not been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



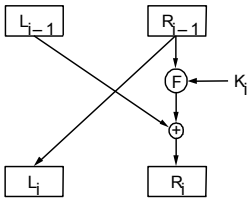
Secret Key (DES)



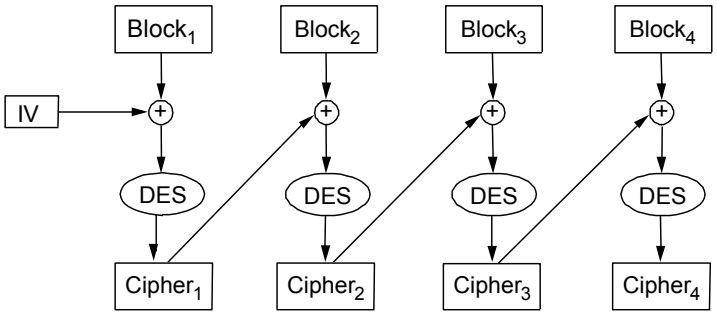
- 64-bit key (56-bits + 8-bit parity)
- 16 rounds



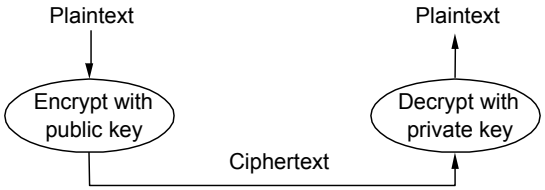
- Each Round



- Repeat for larger messages



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 $(p - 1) \times (q - 1)$ are relatively prime.
- Two numbers are relatively prime if they have no common factor greater than one
- Compute decryption key d such that

$$d = e^{-1} \bmod ((p - 1) \times (q - 1))$$

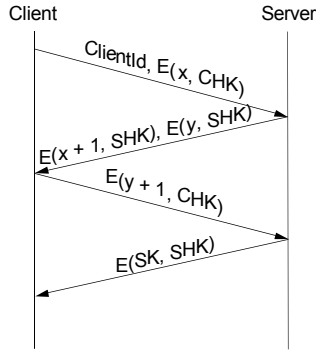
- Construct public key as (e, n)
- Construct public key as (d, n)
- Discard (do not disclose) original primes p and q

Message Digest

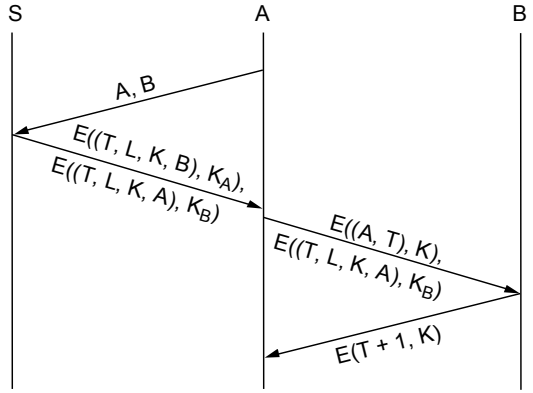
- Cryptographic checksum
  - just as a regular checksum protects the receiver from accidental changes to the message, a cryptographic checksum protects the receiver from malicious changes to the message.
- One-way function
  - given a cryptographic checksum for a message, it is virtually impossible to figure out what message produced that checksum; it is not computationally feasible to find two messages that hash to the same cryptographic checksum.
- Relevance
  - if you are given a checksum for a message and you are able to compute exactly the same checksum for that message, then it is highly likely this message produced the checksum you were given.

Authentication Protocols

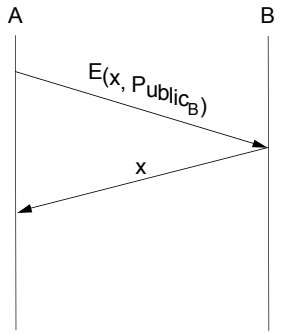
- Three-way handshake



- Trusted third party (Kerberos)



- Public key authentication



Message Integrity Protocols

- Digital signature using RSA
  - special case of a message integrity where the code can only have been generated by one participant
  - compute signature with private key and verify with public key
- Keyed MD5
  - sender: m + MD5(m + k) + E(k, *private*)
  - receiver
    - recovers random key using the sender's public key
    - applies MD5 to the concatenation of this random key message
- MD5 with RSA signature
  - sender: m + E(MD5(m), *private*)
  - receiver
    - decrypts signature with sender's public key
    - compares result with MD5 checksum sent with message

Message Integrity Protocols

- Digital signature using RSA
  - special case of a message integrity where the code can only have been generated by one participant
  - compute signature with private key and verify with public key
- Keyed MD5
  - sender: m + MD5($m + k$) + E(E(k, *rcv-pub), private*)
  - receiver
    - recovers random key using the sender's public key
    - applies MD5 to the concatenation of this random key message
- MD5 with RSA signature
  - sender: m + E(MD5(m), *private*)
  - receiver
    - decrypts signature with sender's public key
    - compares result with MD5 checksum sent with message

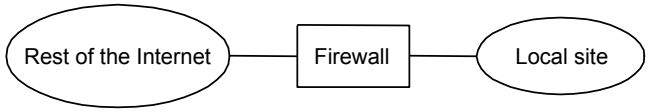
Key Distribution

- Certificate
  - special type of digitally signed document:
    "I certify that the public key in this document belongs to the entity named in this document, signed X."
  - the name of the entity being certified
  - the public key of the entity
  - the name of the certified authority
  - a digital signature
- Certified Authority (CA)
  - administrative entity that issues certificates
  - useful only to someone that already holds the CA's public key.

Key Distribution (cont)

- Chain of Trust
  - if X certifies that a certain public key belongs to Y, and Y certifies that another public key belongs to Z, then there exists a chain of certificates from X to Z
  - someone that wants to verify Z's public key has to know X's public key and follow the chain
- Certificate Revocation List

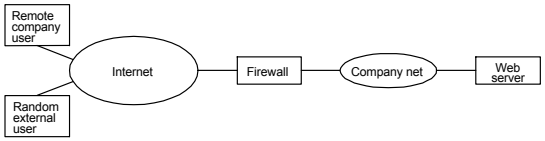
Firewalls



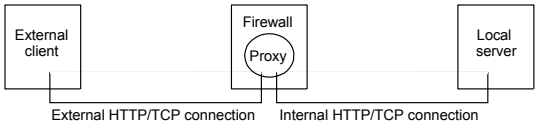
- Filter-Based Solution
  - example
    (192.12.13.14, 1234, 128.7.6.5, 80)
    (*,*, 128.7.6.5, 80)
  - default: forward or not forward?
  - how dynamic?

Proxy-Based Firewalls

- Problem: complex policy
- Example: web server



- Solution: proxy



- Design: transparent vs. classical
- Limitations: attacks from within

Denial of Service

- Attacks on end hosts
  - SYN attack
- Attacks on routers
  - Christmas tree packets
  - pollute route cache
- Authentication attacks
- Distributed DoS attacks