Security and Cryptography

Security Threats

- Impersonation
  - Pretend to be someone else to gain access to information or services
- Lack of secrecy
  - Eavesdrop on data over network
- Corruption
  - Modify data over network
- Break-ins
  - Take advantage of implementation bugs
- Denial of Service
  - Flood resource to deny use from legitimate users

Three Levels of Defense

- Firewalls
  - Filtering "dangerous" traffic at a middle point in the network
- Network level security (e.g. IPsec)
  - Host-to-host encryption and authentication
  - Can provide security without application knowledge
- Application level security
  - True end-to-end security
  - Requires extra effort per application
  - Libraries help, like SSL/TLS

Private Key Cryptosystems

- Finite message domain M, key domain K
- Key $k \in K$
  - Known by all parties
  - Must be secret
- Encrypt: $E: M \times K \rightarrow M$
  - Plaintext m_p to ciphertext m_c as $m_c = E(m_p, k)$
- Decrypt: $D: M \times K \rightarrow M$
  - $m_p = D(m_c, k) = D(E(m_p, k), k)$
- Cryptographic security
  - Given m_c, hard to determine m_p or k
  - Given m_c and m_p, hard to determine k

One Time Pad

- Messages
  - n-bit strings $[b_1,\ldots,b_n]$
- Keys
  - Random n-bit strings $[k_1,\ldots,k_n]$
- Encryption/Decryption
  - $c = E(b, k) = b \oplus k = [b_1 \oplus k_1, \ldots, b_n \oplus k_n]$
    - $\oplus$ denotes exclusive or
  - $b = D(b, k) = c \oplus k = b \oplus k \oplus k = b \oplus [0, \ldots, 0] = b$
- Properties
  - Provably unbreakable if used properly
  - Keys must be truly random
  - must not be used too often
  - Key same size as message

Simple Permutation Cipher

- Messages
  - n-bit strings $[b_1,\ldots,b_n]$
- Keys
  - Permutation π of n
  - Let $\sigma = \pi^{-1}$
- Encryption/Decryption
  - $E([b_1,\ldots,b_n], \pi) = [b_{\pi(1)},\ldots,b_{\pi(n)}]$
  - $D([b_1,\ldots,b_n], \pi) = [b_{\sigma(1)},\ldots,b_{\sigma(n)}]$
- Properties
  - Cryptanalysis possible

Data Encryption Standard (DES)

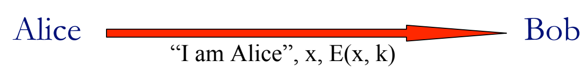
- History
  - Developed by IBM, 1975
  - Modified slightly by NSA
  - U.S. Government (NIST) standard, 1977
- Algorithm
  - Uses 64-bit key, really 56 bits plus 8 parity bits
  - 16 "rounds"
    - 56-bit key used to generate 16 48-bit keys
    - Each round does substitution and permutation using 8 S-boxes
- Strength
  - Difficult to analyze
  - Cryptanalysis believed to be exponentially difficult in number of rounds
  - No currently known attacks easier than brute force
  - But brute force is now (relatively) easy

Other Ciphers

- Triple-DES
  - DES three times
    - $m_c = E(D(E(m_p, k_1), k_2, k_3)$
  - Effectively 112 bits
  - Three times as slow as DES
- Blowfish
  - Developed by Bruce Schneier circa 1993
  - Variable key size from 32 to 448 bits
  - Very fast on large general purpose CPUs (modern PCs)
  - Not very easy to implement in small hardware
- Advanced Encryption Standard (AES)
  - Selected by NIST as replacement for DES in 2001
  - Uses the Rijndael algorithm
  - Keys of 128, 192 or 256 bits

Private Key Authentication

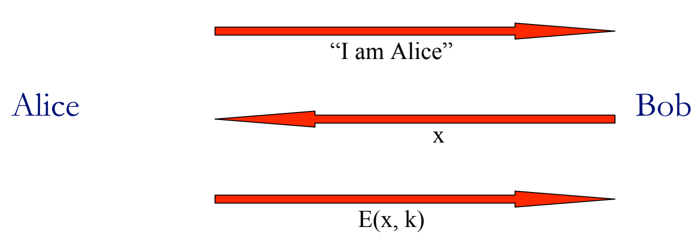
- Alice wants to talk to Bob
  - Needs to convince him of her identity
  - Both have private key k
- Naive scheme



- Vulnerability?

Preventing Replay Attacks

- Bob can issue a challenge phrase to Alice



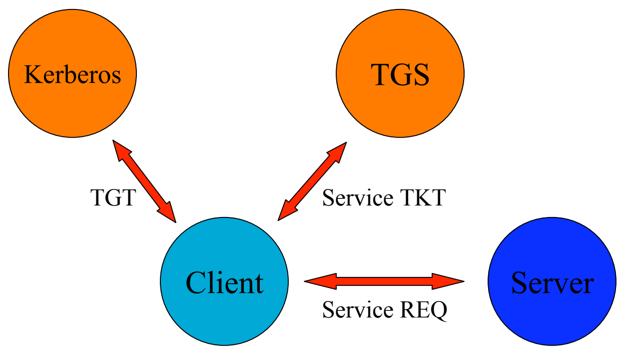
Key Distribution

- Have network with n entities
- Add one more
  - Must generate n new keys
  - Each other entity must securely get its new key
  - Big headache managing n^2 keys!
- One solution: use a central keyserver
  - Needs n secret keys between entities and keyserver
  - Generates session keys as needed
  - Downsides
    - Only scales to single organization level
    - Single point of failure

Kerberos

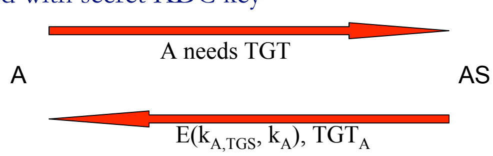
- Trivia
  - Developed in 80's by MIT's Project Athena
  - Mythic three-headed dog guarding the entrance to Hades
- Uses DES, 3DES
- Key Distribution Center (KDC)
  - Central keyserver for a Kerberos domain
  - Authentication Service (AS)
    - Database of all master keys for the domain
    - Users' master keys are derived from their passwords
    - Generates ticket-granting tickets (TGTs)
  - Ticket Granting Service (TGS)
    - Generates tickets for communication between principals
  - "slaves" (read only mirrors) add reliability
  - "cross-realm" keys obtain tickets in others Kerberos domains

Kerberos Authentication Steps



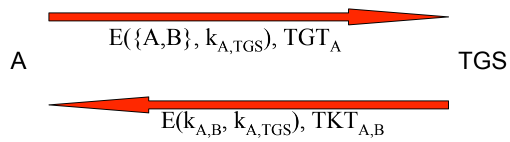
Kerberos Tickets

- What is a ticket?
  - Owner (Instance and Address)
  - A key for a pair of principles
  - A lifetime (usually ~1 day) of the key
    - Clocks in a Kerberos domain must be roughly synchronized
  - Contains all state
  - Encrypted for server
- Ticket-granting-ticket (TGT)
  - Obtained at beginning of session
  - Encrypted with secret KDC key

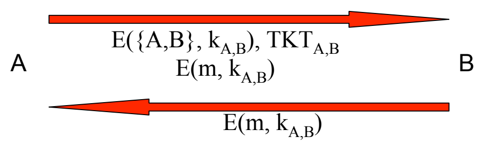


Kerberos – A wants to talk to B

- First, get ticket from TGS



- Then, use the ticket



Using Kerberos

- kinit
  - Get your TGT
  - Creates file, usually stored in /tmp
- klist
  - View your current Kerberos tickets
- kdestory
  - End session, destroy all tickets
- kpasswd
  - Changes your master key stored by the AS
- "Kerberized" applications
  - kftp, ktelnet, ssh, zephyr, etc
  - afslog uses Kerberos tickets to get AFS token

Diffie-Hellman Key Agreement

- History
  - Developed by Whitfield Diffie, Martin Hellman
  - Published in 1976 paper "New Directions in Cryptography"
- Allows negotiation of secret key over insecure network
- Algorithm
  - Public parameters
    - Prime p
    - Generator $g < p$ with property: $\forall n: 1 \leq n \leq p-1, \exists k: n = g^k \bmod p$
  - Alice chooses random secret a, sends Bob g^a
  - Bob chooses random secret b, sends Alice g^b
  - Alice computes $(g^b)^a$, Bob computes $(g^a)^b$ – this is the key
  - Difficult for eavesdropper Eve to compute g^{ab}

Diffie-Hellman Weakness

- Man-in-the-Middle attack
  - Assume Eve can intercept and modify packets
  - Eve intercepts g^a and g^b, then sends Alice and Bob g^c
  - Now Alice uses g^{ac}, Bob uses g^{bc}, and Eve knows both
- Defense requires mutual authentication
  - Back to key distribution problem

Public Key Cryptosystems

- Keys P, S
  - P: public, freely distributed
  - S: secret, known only to one entity
- Properties
  - x = D(E(x,S), P)
  - x = D(E(x,P), S)
  - Given x, hard to determine E(x, S)
  - Given E(x, P), hard to determine x

Using Public Key Systems

- Encryption – Bob sends to Alice
  - Bob generates and sends m_c = E (m_p, P_A)
  - Only Alice is able to decrypt m_p = D(m_c, S_A)
- Authentication – Alice proves her identity
  - Bob generates and sends challenge x
  - Alice response s = E(x, S_A)
  - Bob checks: D(s, P_A) = x
- Weakness – key distribution (again)
  - If Bob gets unauthentic P_A, he can be easily attacked

Cryptographic Hash Functions

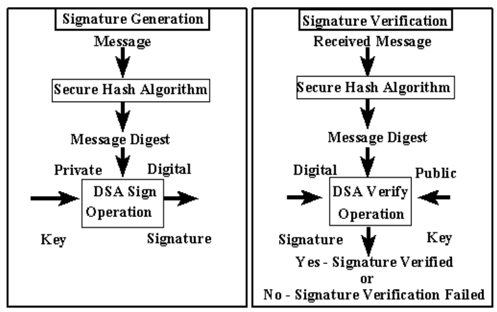
- Given arbitrary length m, compute constant length digest d = h(m)
- Desirable properties
  - h(m) easy to compute given m
  - One-way: given h(m), hard to find m
  - Weakly collision free: given h(m) and m, hard to find m' s.t. h(m) = h(m')
  - Strongly collision free: hard to find any x, y s.t. h(x) = h(y)
- Example use: password database, file distribution
- Common algorithms: MD5, SHA

Comparative Performances

- According to Peterson and Davie
- MD5: 600 Mbps
- DES: 100 Mbps
- RSA: 0.1 Mbps

Digital Signatures

- Alice wants to convince others that she wrote message m
  - Computes digest d = h(m) with secure hash
  - Signature $s = S_A(d)$
- Digital Signature Standard (DSS)



Authentication Chains

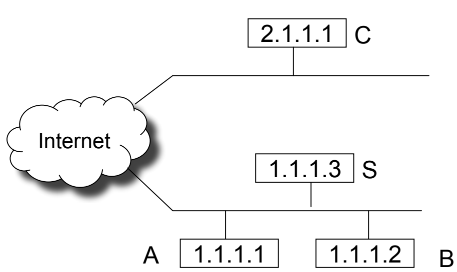
- How do you trust an unknown entity?
- Trust hierarchies
  - Certificates issued by Certificate Authorities (CAs)
    - Certificates are signed by only one CA
    - Trees are usually shallow and broad
    - Clients only need a small number of root CAs
      - Roots don't change frequently
      - Can be distributed with OS, browser
    - Problem
      - Root CAs have a lot of power
      - Initial distribution of root CA certificates
  - X.509
    - Certificate format standard
    - Global namespace: Distinguished Names (DNs)
      - Not very tightly specified – usually includes an email address or domain name

Security Vulnerabilities

- Security Problems in the TCP/IP Protocol Suite – Steve Bellovin - 89
- Attacks on Different Layers
  - IP Attacks
  - ICMP Attacks
  - Routing Attacks
  - TCP Attacks
  - Application Layer Attacks

Security Flaws in IP

- The IP addresses are filled in by the originating host
  - Address spoofing
- Using source address for authentication
  - r-utilities (rlogin, rsh, rhosts etc..)



•Can A claim it is B to the server S?

•ARP Spoofing

•Can C claim it is B to the server S?

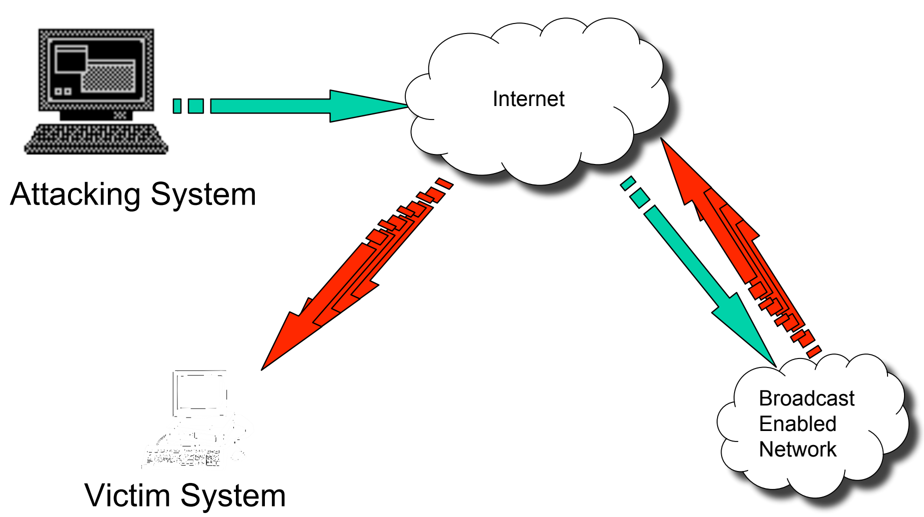
•Source Routing

Security Flaws in IP

- IP fragmentation attack
  - End hosts need to keep the fragments till all the fragments arrive

- Traffic amplification attack
  - IP allows broadcast destination

Ping Flood



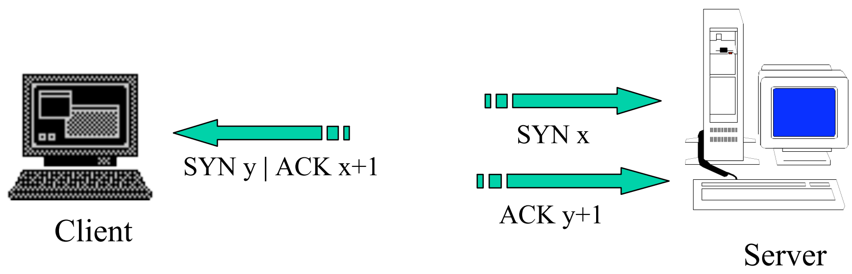
ICMP Attacks

- No authentication
- ICMP redirect message
  - Can cause the host to switch gateways
  - Benefit of doing this?
    - Man in the middle attack, sniffing
- ICMP destination unreachable
  - Can cause the host to drop connection
- ICMP echo request/reply
- Many more…
  - http://www.sans.org/rr/whitepapers/threats/477.php

Routing Attacks

- Distance Vector Routing
  - Announce 0 distance to all other nodes
    - Blackhole traffic
    - Eavesdrop
- Link State Routing
  - Can claim direct link to any other routers
  - A bit harder to attack than DV
- BGP
  - ASes can announce arbitrary prefix
  - ASes can alter path

TCP Attacks



TCP Layer Attacks

- TCP SYN Flooding
  - Exploit state allocated at server after initial SYN packet
  - Send a SYN and don't reply with ACK
  - Server will wait for 511 seconds for ACK
  - Finite queue size for incomplete connections (1024)
  - Once the queue is full it doesn't accept requests

TCP Layer Attacks

- TCP Session Hijack
  - When is a TCP packet valid?
    - Address/Port/Sequence Number in window
  - How to get sequence number?
    - Sniff traffic
    - Guess it
      - Many earlier systems had predictable initial sequence number
  - Inject arbitrary data to the connection

TCP Layer Attacks

- TCP Session Poisoning
  - Send RST packet
    - Will tear down connection
  - Do you have to guess the exact sequence number?
    - Anywhere in window is fine
    - For 64k window it takes 64k packets to reset
    - About 15 seconds for a T1
  - Can reset BGP connections

Application Layer Attacks

- Applications don't authenticate properly
- Authentication information in clear
  - FTP, Telnet, POP
- DNS insecurity
  - DNS poisoning
  - DNS zone transfer

Denial of Service

- Objective → make a service unusable by overloading
- Consume host resources
  - TCP SYN floods
  - ICMP ECHO (ping) floods
- Consume bandwidth
  - UDP floods
  - ICMP floods
- Crashing the victim
  - Ping-of-Death
  - TCP options (unused, or used incorrectly)
- Forcing more computation on routers
  - Taking long path in processing of packets

Summary

- Tools for network security:
  - Secret keys, public/private keys, digital signature
- Network security needs to be addressed at different levels
  - Better protocols, better routers, better application level features, etc.