Cryptography

COS 461: Computer Networks
Precept: 04/20/2012
Princeton University

Overview

• Network security and definitions
• Brief introduction to cryptography
  – Cryptographic hash functions
  – Symmetric-key crypto
  – Public-key crypto
  – Hybrid crypto

Internet’s Design: Insecure

• Designed for simplicity
• “On by default” design
• Readily available zombie machines
• Attacks look like normal traffic
• Internet’s federated operation obstructs cooperation for diagnosis/mitigation

Basic Components

• **Confidentiality**: Concealment of information or resources
• **Authenticity**: Identification and assurance of origin of info
• **Integrity**: Trustworthiness of data or resources in terms of preventing improper and unauthorized changes
• **Availability**: Ability to use desired info or resource
• **Non-repudiation**: Offer of evidence that a party indeed is sender or a receiver of certain information
• **Access control**: Facilities to determine and enforce who is allowed access to what resources (host, software, network, ...)

Eavesdropping - Message Interception (Attack on Confidentiality)

• Unauthorized access to information
• Packet sniffers and wiretappers (e.g. tcpdump)
• Illicit copying of files and programs

Integrity Attack - Tampering

• Stop the flow of the message
• Delay and optionally modify the message
• Release the message again
AuthenCcity Attack - Fabrication

- Unauthorized assumption of other’s identity
- Generate and distribute objects under identity

Impact of Attacks

- Theft of confidential information
- Unauthorized use of
  - Network bandwidth
  - Computing resource
- Spread of false information
- Disruption of legitimate services

What is Cryptography?

- Comes from Greek word meaning “secret”
  - Primitives also can provide integrity, authentication
- Cryptographers invent secret codes to attempt to hide messages from unauthorized observers
- Modern encryption:
  - Algorithm public, key secret and provides security
  - May be symmetric (secret) or asymmetric (public)

Attack on Availability

- Destroy hardware (cutting fiber) or software
- Modify software in a subtle way
- Corrupt packets in transit
- Blatant denial of service (DoS):
  - Crashing the server
  - Overwhelm the server (use up its resource)

Introduction to Cryptography

Cryptographic Algorithms: Goal

- Given key, relatively easy to compute
- Without key, hard to compute (invert)
- “Level” of security often based on “length” of key
Three Types of Functions

• Cryptographic hash Functions
  – Zero keys

• Secret-key functions
  – One key

• Public-key functions
  – Two keys

Cryptographic hash functions

Cryptography Hash Functions

• Take message, $m$, of arbitrary length and produces a smaller (short) number, $h(m)$

• Properties
  – Easy to compute $h(m)$
  – Pre-image resistance: Hard to find an $m$, given $h(m)$
    • “One-way function”
  – Second pre-image resistance: Hard to find two values that hash to the same $h(m)$
    • E.g. discover collision: $h(m) = h(m')$ for $m \neq m'$
  – Often assumed: output of hash fn’s “looks” random

How hard to find collisions?

Birthday Paradox

• Compute probability of different birthdays
• Random sample of $n$ people taken from $k=365$ days
• Probability of no repetition:
  \[ P = 1 - (1)(1-1/365)(1-2/365) \ldots (1-(n-1)/365) \]
  \[ P \approx 1 - e^{-n(n-1)/2k} \]
  – Let $k=n$, $P \approx 2^{n(n/2)}$

How Many Bits for Hash?

• If $m$ bits, takes $2^{m/2}$ to find weak collision
  – Still takes $2^m$ to find strong (pre-image) collision

• 64 bits, takes $2^{32}$ messages to search (easy!)

• Now, MD5 (128 bits) considered too little

• SHA-1 (160 bits) getting old

Example use #1: Passwords

• Password hashing
  – Can’t store passwords in a file that could be read
    • Concerned with insider attacks!
  – Must compare typed passwords to stored passwords
    • Does hash (typed) == hash (password) ?
  – Actually, a “salt” is often used: hash (input || salt)
    • Avoids precomputation of all possible hashes in “rainbow tables.” (available for download from file-sharing systems)
Example use #2: Self-certifying naming

- File-sharing software (LimeWire, BitTorrent)
  - File named by  $F_{name} = \text{hash (data)}$
  - Participants verify that  $\text{hash (downloaded)} == F_{name}$
    - If check fails, reject data

- Recursively applied...
  - BitTorrent file has many chunks
  - Control file downloaded from tracker includes:
    - $\forall$ all chunks, $F_{chunk\_name} = \text{hash (chunk)}$
  - BitTorrent client verifies each individual chunk

Symmetric (Secret) Key Cryptography

Symmetric Encryption

- Also: “conventional / private-key / single-key”
  - Sender and recipient share a common key
  - All classical encryption algorithms are private-key
  - Dual use: confidentiality or authentication/integrity
    - Encryption vs. msg authentication code (MAC)

- Was only type of encryption prior to invention of public-key in 1970’s
  - Most widely used
  - More computationally efficient than “public key”

Symmetric Cipher Model

Use and Requirements

- Two requirements
  - Strong encryption algorithm
  - Secret key known only to sender / receiver

- Goal: Given key, generate 1-to-1 mapping to ciphertext that looks random if key unknown
  - Assume algorithm is known (no security by obscurity)
  - Implies secure channel to distribute key

Distribution of Symmetric Keys

- Options: (between A and B).
  - A selects a key and physically delivers it to B.
  - If A and B already have a viable key, it can be used to distribute a new key.
  - A trusted third party key distribution center (KDC) selects a key and physically delivers it to A and B (through a secure channel).
**Distribution of Symmetric Keys**

- Manual delivery is challenging...
- The number of keys grows quadratically with the number of endpoints \(n*(n-1)/2\)
  - Further complexity for application/user level encryption
- Key distribution center (KDC) a good alternative
  - Only \(n\) master keys required
  - KDC generate session key for Alice and Bob

**Public-Key Cryptography**

**Why Public-Key Cryptography?**

- Developed to address two key issues:
  - **Key distribution**: Secure communication w/o having to trust a key distribution center with your key
  - **Digital signatures**: Verify msg comes intact from claimed sender (w/o prior establishment)
- Public invention due to Whitfield Diffie & Martin Hellman in 1976
  - Known earlier in classified community

**Security of Public Key Schemes**

- Public-key encryption is a “trap-door” function:
  - Easy to compute \(c \leftarrow F(m)\)
  - Hard to compute \(m \leftarrow F^{-1}(c)\) without knowing \(k\)
  - Easy to compute \(m \leftarrow F^{-1}(c, k)\) by knowing \(k\)
- Like private key schemes, brute force search possible
  - But keys used are too large (e.g., \(\geq 1024\)bits)
  - Hence is slow compared to private key schemes
(Simple) RSA Algorithm

- **Security** due to cost of factoring large numbers
  - Factorization takes $O(e \log n \log \log n)$ operations (hard)
  - Exponentiation takes $O((\log n)^3)$ operations (easy)

- To encrypt a message $M$ the sender:
  - Obtain public key $(e, n)$; compute $C = M^e \mod n$

- To decrypt the ciphertext $C$ the owner:
  - Use private key $(d, n)$; computes $M = C^d \mod n$

Symmetric vs. Asymmetric

- **Symmetric Pros and Cons**
  - Simple and really very fast (order of 1000 to 10000 faster than asymmetric mechanisms)
  - Must agree/distribute the key beforehand

- **Public Key Pros and Cons**
  - Easier predistribution for public keys
    - Public Key Infrastructure (in textbook)
  - Extremely slow

Hybrid Encryption

- **User’s public key** (in certificate)
- Randomly-Generated symmetric “session” key
- **Recipient’s private key**
- Digital envelope contains “session” key encrypted using recipient’s public key
- Session key must be decrypted using the recipient’s private key

Summary

- Network security and definitions
- Introduction to cryptography
  - Cryptographic hash functions
    - Zero keys, hard to invert, hard to find collisions
  - Symmetric-key crypto
    - One key, hard to invert, requires key distribution
  - Public-key crypto
    - Two keys, hard to invert, more expensive
  - Hybrid scheme
- Mon: IPSec, HTTPS, DNS-Sec, other security problems