UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 537
Intro to Operating Systems

A. Arpaci-Dusseau
Spring 2000

Security Solutions and Encryption

Questions answered in these notes:

- How does one increase level of security?
- What is public key encryption?
- What is private key encryption?
- How is each used for security?
- How is each used for authentication?

Reading for topic: Chapter 19 & 20

Regaining Security

May be impossible to secure system once penetrated

- Not all possible to tell that security violation occurred
  - Villain can remove all traces from log files
- Hooks could have been left around for the imposter to regain control
- Cannot restore system from backup tapes
  - Attack could have occurred earlier than suspected

Only solution

- Remove all files from disk and reinstall all software

Why Less a Problem with Humans?

Humans do not easily forget events

- Computer memory is volatile
- May leave no trace of past events

Humans usually know who they are interacting with

- Anonymity occurs easily on computers
- Cannot tell who is doing what
- Assume person logged as true self

Do not usually trust personal property to new acquaintances

- More trusting of computers
- Any program you run could modify any of your files

Security Solutions

Logging

- Record all important events and uses of privilege in an indelible file
  - Write-once disk
- Examples
  - Attempts to specify an incorrect password
  - All logins
  - All super-user actions
- Can be used to catch imposters during initial attempts and failures
- Even better to get humans involved at key steps
  - One of the solutions for Electronic Funds Transfer (EFT)
  - Periodically check logs for strange events
- Drawback: Can leak passwords into logs

More Security Solutions

Caller identification

- Telephone: traditional for callers to be anonymous, but not receivers
  - Very difficult to catch electronic thieves
- Need a change of **policy** to eliminate caller anonymity
- Solution: Callback

Principle of minimum privilege (need-to-know):

- Each piece has access to minimum information, for minimum time
- Example
  - File system cannot touch memory map, memory manager cannot touch disk blocks
- Reduces chances of accidental or intentional damage
- Impossible to provide absolute information containment
  - Covert channels

Encryption

Goals

- Secure communication: No one can eavesdrop
- Authentication : Establish identify of source; info cannot be modified

Mechanism: Convert data to form that does not make sense

- Initial readable text that needs protection: *clear text*
- Encrypt the clear text so that it does not make sense: cipher text
  - Controlled by function or number: *encryption key*
  - Encrypted text can be stored in readable file or transmitted over unprotected channels
- To make sense of cipher text, *decrypt* it back into clear text
  - Performed with secret function or number: *decryption key*
  - Based on factoring very large numbers (product of two primes)

Necessary Conditions

Encryption function cannot be easily inverted

- Cannot discover clear text unless know decryption key

Keys must be protected

- If encryption and decryption keys are identical, cannot leak either key

Encryption and decryption must be done in safe place

- Otherwise, could snoop clear text
- Trusted computing base (TCB)
  - Software and hardware that must behave correctly

Public Key Encryption

Two keys for every user: Public and private key

- Everyone knows all public keys
- Only host knows the private key (secret key)

Requirements

- Cannot derive one from knowing the other
- Public and private keys are inverses of the other
  - Encode with private key of A --> Decode with public key of A
  - $\{Message\}^{SA}$
  - Encode with public key of A --> Decode with private key of A
  - $\{Message\}^{PA}$

Authentication with Public Keys

Authentication

- Reliably identify the sender of a message
- Example: A sends to B; B must know A sent message
  - A->B: $\{Message\}^{SA}$
  - B: Can decode $\{Message\}^{SA}$ with PA
  - No one else but A could have encoded a valid message

Positive Identification

- Example: "I agree to pay Mary $100 per year for duration of my life"
- If message can be decrypted with your public key, then written by you
  - Anyone can verify author of message
- Electronic signature: Can be legally binding

Security & Authentication with Public Keys

Secure communication

- Ensure that no one can snoop on messages
- Example: A sends to B
  - A->B : $\{Message\}^{PB}$
  - B: Can decode $\{Message\}^{PB}$ with SB
  - No one else but B can decode $\{Message\}^{PB}$
- Anyone can send such a message to B

Combine above strategies for both security and authentication

- Example: A sends a message to B that only B can read;
  B knows that only A could have created message
  - A->B : $\{\{Message\}^{PB}\}^{SA}$
  - B: Can decode $\{\{Message\}^{PB}\}^{SA}$ with PA to $\{Message\}^{PB}$
  - B: Can decode $\{Message\}^{PB}$ with SB to get Message

Example with Public Encryption

How to encrypt a message given the following requirements?

- All communication channels are insecure
- There are three parties involved
  - P: the original sender of the message
  - S: an intermediary receiver of the message
  - E: the final receiver of the message
- Only E can read the message
- E must know that the message was written by P
- The message must pass through S before getting to E

Potential Limitations of Encryption

Eavesdroppers can replay messages

- Replaying old messages may confuse parties
  - Even though eavesdropper does not know what they are replaying...
- Solution: Sequence numbers (nonces) or timestamps in messages

How do you trust public key?

- You hear: "Andrea's public key is K"
  - Problem: Who said this?
- Solution: **Authentication Server** that everyone trusts
  - Everyone knows public key of authentication server: PAS
  - AS -> A: $\{\text{Public key of B is PB}\}^{SAS}$
  - A can decode and know that only AS could have sent PB

Fail-Secure: Secure if part of system fails

Traditional Encryption

Another problem: Slow encoding and decoding

- 200 Kbits/sec in hardware
- .5 Kbits/sec/MIPS in software

Alternative: Single private key for better speed

- 1200 Kbits/sec in hardware
- 400 Kbits/sec/MIPS in software

Example: Data Encryption Standard (DES)

- Associate private key with session between two users

Problem: How do you exchange private session key?

- Cannot send private key unencrypted over channel!
- Solution #1: Use public key encryption to exchange session key

Exchanging Session Keys

Authentication server: Knows private keys of all users

Example

- A wants to talk securely with B, B must authenticate A is sender

Simplified algorithm (without worrying about replay attacks)

- A asks authentication server for a session key with B
  - No encryption needed
- Authentication server replies with new conversation key CK
  - AS->A: $\{CK, \{A,CK\}^{KB}\}^{KA}$
  - If decrypted message makes sense, only AS could have sent message
  - Only A can decrypt message and get CK
- A sends message to B telling it the key
  - A->B: $\{A,CK\}^{KB}$
  - No one could modify message to change name of sender

Secure Signatures with Private Keys

Problem: How do I know if binary file was modified in transit?

- If not worried about eavesdropping, faster to not encrypt entire file

Solution: Secure checksum or characteristic value

- Also called *Message Digests* or *Digital fingerprint*
- Function(Message) = large integer (e.g., 1024 bits)
  - Difficult to find another message that maps to same integer

Example: A sends file to B

- A calculates checksum of file; ask authentication server to encrypt
- A sends message and encrypted checksum to B
  - A->B: file, $\{CK\}^{KAS}$
- B receives file and computes checksum
- B asks AS to decode CK so it can compare two
  - B->AS: $\{CK\}^{KAS}$
  - AS->B: $\{CK\}^{KB}$

Improving Encryption Safety

How safe is encryption?

- DES 56 bit key --> 2^{56} possible keys
- Crack by guessing with many machines --> RSA Challenge

Solutions

- Upgrade encryption as computers become more powerful
- Remove known patterns from the clear text
  - Example: Your name
  - Compress clear text before encryption
- Do not send large amounts of information with the same key
  - Change keys frequently

Implication for digital signatures over lifetime?

Two Philosophies for Protection

Keep the mechanism secret

- Advantage: Harder to break in if know nothing about structure
- Disadvantage: Harder to keep secure if secret is released

Publish the mechanism

- Disadvantage: Easier to break in if can find design flaws or bugs
- Advantage: Encourage bad person — good person fight
    - Offer reward for reporting security holes
    - Find holes and fix quickly

Encryption Today

Theory more advanced than practice

- Must improve to support electronic commerce

Currently in Unix

- No encryption performed for network services
- Rlogin, NFS, ftp, etc.

Kerberos

- Based on DES
- Get tickets from server