UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 537
Introduction to Operating Systems

Andrea C. Arpaci-Dusseau
Remzi H. Arpaci-Dusseau

Encryption

Questions answered in this lecture:

- How does encryption provide privacy?
- How does encryption provide authentication?
- What is public key encryption?
- What is symmetric key encryption?

Motivation

Scenario: Communicate between two endpoints over unprotected channel (e.g., network)

- Others can eavesdrop on channel
- Others can inject messages on channel

Goals

- Provide secure communication
  - No one can understand message contents
- Provide authentication
  - Establish identify of sender
  - Ensure that message contents are not altered

Encryption Mechanism

Convert data to form that does not make sense to others

Terminology

- Clear text (plain text): Initial readable text that needs protection
- Cipher text: Encrypted version of clear text

Steps

- Sender: Encrypt clear text to cipher text
  - Apply function with encryption key to clear text
- Cipher text can be stored in readable file or transmitted over unprotected channels
- Receiver: Decrypt cipher text to clear text
  - Apply function with decryption key to cipher text

Based on factoring very large numbers (product of two primes)

Necessary Conditions

1) Decryption function or key remains secret
   - If encryption and decryption keys are identical, cannot leak either key
2) Encryption function cannot be easily inverted
3) Encryption and decryption must be done is safe place
   - Trusted Computing Base (TCB)
   - Clear text must never be leaked outside of TCB

Public Key Encryption

RSA algorithm (Rivest, Shamir, Adleman)

Two keys for every user

- Public key: Known by everyone
- Secret (private) key: Known only by user

Requirements

- Cannot derive one from knowing the other
- Public and secret keys are inverses of each other
  - Encode with secret key of A --> decode with public key of A
    - $\{\{Msg\}^{SA}\}^{PA}$ --> Msg
  - Encode with public key of A --> decode with secret key of A
    - $\{\{Msg\}^{PA}\}^{SA}$ --> Msg

Security with Public Keys

Secure communication: Ensure that no one can read content of messages

- Example: A sends Msg to B
- A->B: $\{Msg\}^{PB}$
- B can decode $\{Msg\}^{PB}$ with SB
- No one else but B can decode Msg
- Anyone can send messages to B

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: $\{Msg\}^{SA}$
- B can decode $\{Msg\}^{SA}$ with PA
- No one else but A could have encoded a valid message

Use for electronic signatures

- Provides positive identification
- Example: "I agree to pay Mary $100 per year for life"
- If message can be decrypted with your public key, then written by you
- Anyone can verify author of message

Security and Authentication with Public Keys

Require both security and authentication, then combine both strategies

- Example: A sends a message to B that only B can read; B knows that only A could have created message
- A->B: $\{\{Msg\}^{PB}\}^{SA}$
- B: How does it decode $\{\{Msg\}^{PB}\}^{SA}$ to get $\{Msg\}^{PB}$?
- B: How does B decode $\{Msg\}^{PB}$ to get Msg?
- Would A->B: $\{\{Msg\}^{SA}\}^{PB}$ work?

Example with Public Keys

How to encrypt message given following requirements:

- All communication channels are insecure
- Three parties involved
  - P (professor): Original sender of message
  - S (student): intermediary of message
  - E (employer): final receiver of message
- 1) Only E can read message
- 2) E must know that the message was written by P
- 3) Message must pass through S before getting to E
  - P ensures that S gives approval

How?

Potential Limitations of Encryption

How do you trust public keys?

- You hear "A's public key is K"
- Problem: Who said this? What if K is really C's public key?

Solution: Authentication Server (AS) everyone trusts

- Everyone knows public key of authentication server (PAS)
- AS->B: $\{\text{Public key of A is PA}\}^{SAS}$
- B decodes with PAS and know only AS could have sent message

Used by HTTPS and Secure Socket Layer (SSL)

- Certificate from server A
  - $\{\text{AS, A, PA, time}\}^{SAS}$

More Problems with Encryption

Eavesdroppers can replay old messages

- Replaying can confuse parties or cause unwanted behavior (even though eavesdropper doesn't know what they are replaying)
- Solution: Sequence numbers (nonces) or timestamps in messages

Relatively slow to encode and decode messages

- Single private key is faster

Symmetric Key Encryption

Also called Single key or Private key encryption

- Example: Advanced Encryption Standard (AES), Data Encryption Standard (DES)

Idea: Associate conversation key (CK) with session between two users

Same key used for both encoding and decoding

- $\{\{\text{Msg}\}^{CK}\}^{CK}$ --> Msg

Problem: How do you exchange conversation key?

- Cannot send private key over insecure channel!
- Example: A wants to talk securely with B, B must authenticate A is sender (do not worry about replay attacks)
- Simplified Solution #1: Use public key encryption to exchange session key
  - A->B: $\{\text{A, } \{\text{CK}\}^{SA}\}^{PB}$

Exchanging Conversation Keys

Solution #2

- Each user has private key, Ki
- Authentication server knows private keys of all users

Simplified algorithm:

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

Who can listen to conversation?

Secure Signatures

Problem: How to know if binary file is modified in transit?

- Could encrypt entire file, but slow if not concerned with eavesdropping

Solution: Secure checksum, message digest, digital fingerprint

- Function(Msg) --> large integer (e.g., 1024 bits)
- Difficult for adversary to find another msg that maps to same integer

Example: A sends file to B

- A calculates checksum of file; ask AS to encrypt
- A sends file and encrypted checksum to B
- B receives file and computes checksum
- B ask AS to decode encrypted checksum so it can compare

Encryption Safety

How safe is encryption? Can private keys be found?

- Crack with brute force guessing, use many machines
  - Look for understandable text in decoded version
- Safety depends on length of keys
  - DES 56 bit keys --> 2^{56} possible keys
    - Cracked in < 24 hours
  - AES has 128 bit keys

Solutions

- Upgrade encryption (key length) as machines become faster
- Remove known patterns from clear text
  - (e.g., compress text first)
- Do not send large amounts of data with same key
  - Change keys frequently