Advanced Operating Systems: Three Easy Pieces

Security

Outline

- **Security Issues**
  - Threats
  - Method of Attack
- **Security Components**
  - Privacy
  - Integrity
  - Authentication
  - Authorization
  - Non-repudiation

Security Issues

What is a Distributed System

A distributed system is one where a machine I've never heard of can cause my program to fail.

— ***Leslie Lamport***

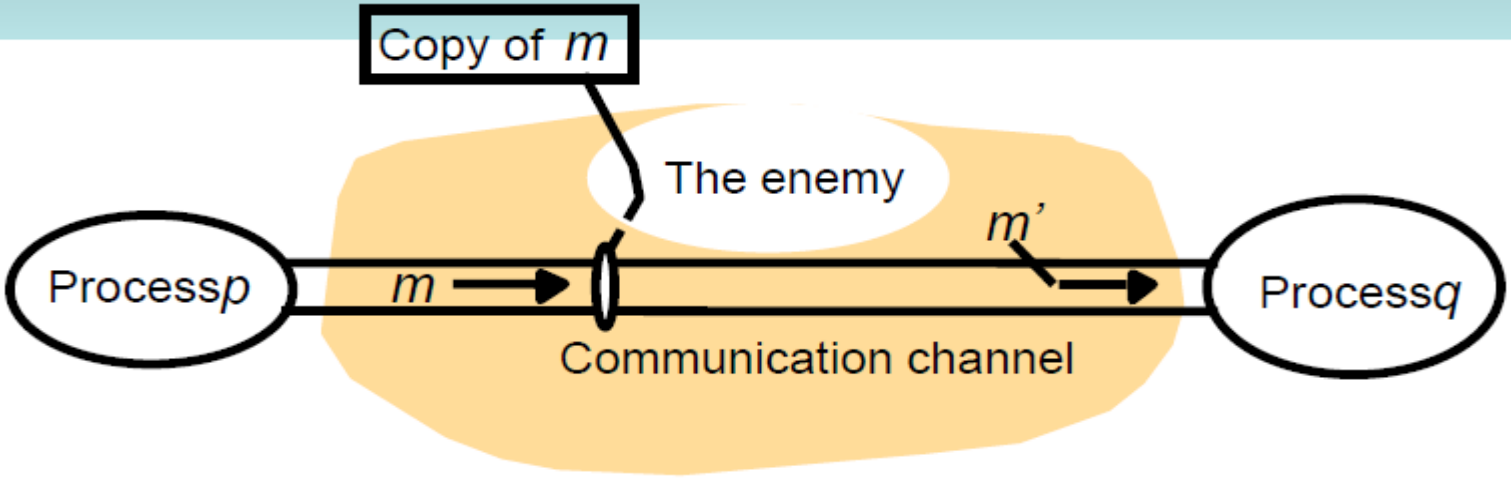
Definition:

More than 1 machine working transparently together to solve a problem

Examples:

- **Client/server:** web server and web client
- **Cluster:** page rank computation
- **Peer-2-Peer:** Twitter, etc.

Security Problems

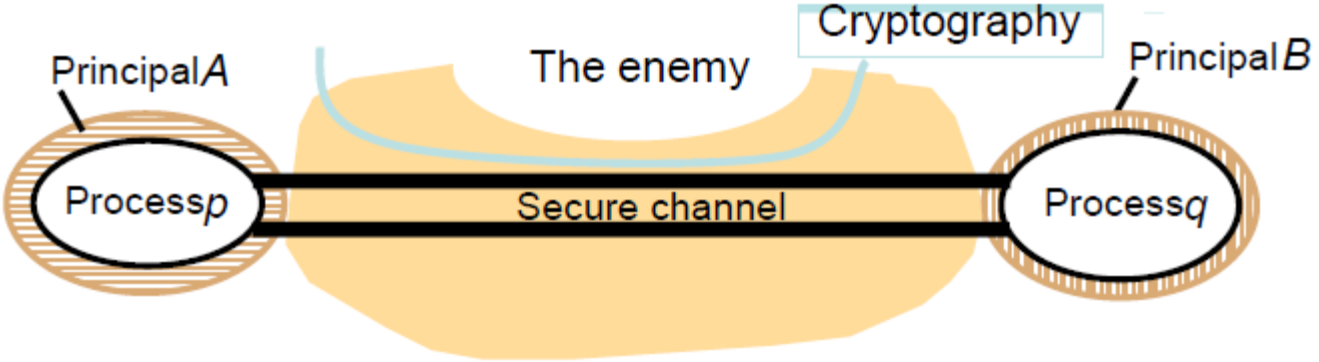


- **Attacks**
  - On applications that handle financial transactions or other information whose secrecy or integrity is crucial
- **Enemy (or adversary)**
- **Threats**
  - To processes, to communication channels, denial of service

Threats/Methods of Attacks

- **Eavesdropping:**
  - On applications that handle financial transactions or other information whose secrecy or integrity is crucial
- **Masquerading:**
  - Assume the identity of another user
- **Message tampering:**
  - Alter the content of messages in transit
    - Man-in-the-middle attack
- **Replaying:**
  - Store secure msgs and send them at a later time
- **Denial of Service:**
  - Flood a channel or other resources, denying access to others.

Secure Channels



- **Properties:**
  - Each process is sure of the identity of the other
  - Data is private and protected against tampering
  - Protection against repetition and reordering of data
- **Important Issues:**
  - Cryptography
  - Authentication

Security Components

Overview

1. **Privacy:** how do you ensure that the information you transmit over the internet has not been captured or passed on to a 3rd party without your knowledge?
2. **Integrity:** how do you ensure that the information you send or receive has not been compromised or altered?
3. **Authentication:** how does the sender/receiver of a message prove their identity to each other?
4. **Authorization:** how do we ensure that users can access certain necessary resources, while valuable information is protected?
5. **Non-repudiation:** how do you legally prove that a message was sent or received?

1. Privacy

1. Privacy / Encryption

- **Cryptography** transforms data by using a cipher and mathematical algorithm for encrypting the message (ciphertext).

- **There are 2 main types of ciphers:**
  - **Substitution ciphers:** every occurrence of a given letter is replaced by a different letter. The 1st known Cipher is **Caesar's cipher** where we replace every instance of a letter with the alphabetical letter of fixed distance to the right. For example, a → c, b → d, c → e, etc.

  - **Transposition ciphers:** the ordering of the letters is scrambled

1. A Simple Encryption Algorithm

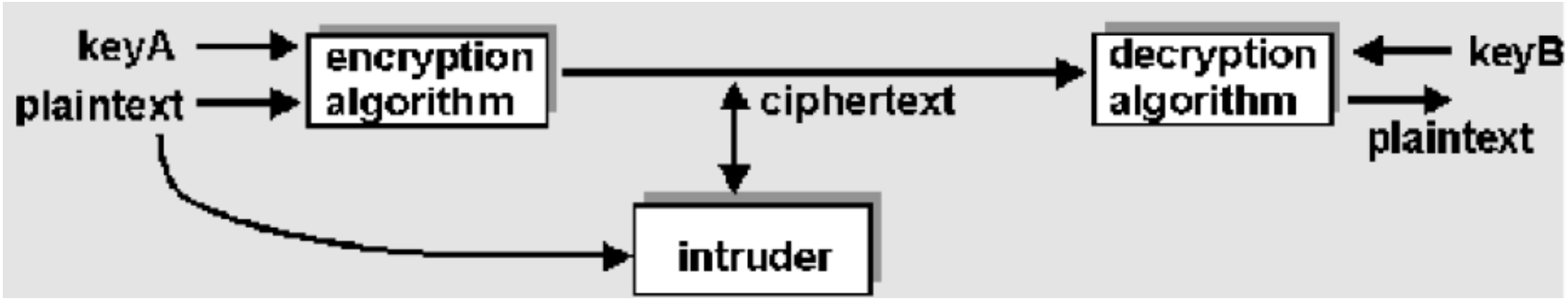
- **Substitution Cipher:**
  - Abcdefghijklmnopqrstuvwxyz
  - poiuytrewqasdfghjklmnbvczx
  - **Replace each plaintext character in the message with the matching ciphertext character.**

- **Example:**
  - **Plaintext:** Charlotte, my love
  - **Ciphertext:** iepksgmmy, dz sgby

1. A Simple Encryption Algorithm (Cont.)

- **Key is pairing (Substitution)** between plaintext characters and ciphertext characters
- **26! (approx 10^26)** different possible keys: unlikely to be broken by random trials
- **Substitution cipher** can be broken using **observed frequency of letters**:
  - ‘e’ is the most common letter in English
  - ‘the’ is the most common word in English

1. Encryption



- **Plaintext:** unencrypted message
- **Ciphertext:** encrypted form of message

- **Intruder may:**
  - Intercept ciphertext transmission
  - Intercept plaintext/ciphertext pairs
  - Identify the encryption algorithm used

1. Private/Secret/Symmetric Key Encryption

- **It uses the same key** to encrypt and decrypt messages. It is simple and efficient model.
  D(E(p)) = p

- **Advantage:** Efficient algorithm from processing perspective.
- **Disadvantage:** How do we change the key if it is compromised (**key distribution problem**)? One solution to this problem is to use 3rd trusted party (Kerberos model) that we will discuss later.

1. Public/Asymmetric Key Encryption

- **In 1976 Stanford Researcher** proposed this approach. Most commonly used in RSA developed in 1977.
- **It uses 2 inversely related keys** (mathematically related): *public key* and *private key*.
  - **Private Key** is kept secret by its owner, while the public key is freely distributed.
  - **If the public key** is used to encrypt a message, only the corresponding private key can decrypt it, and vice versa.
- **Advantage:** very secure + solves the key distribution problem
- **Disadvantage:** very CPU intensive + maintain many keys (4 keys in contrast to 1 key with secret key Encryption).

1. Public/Asymmetric Key Encryption

- **Example:**
  - Assume users A, B will communicate using public Key model.
  - **Each user will have their own private and public keys:**
    - User A will have: E_A(public), D_A. User A will keep D_A confidential.
    - User B will have: E_B, D_B. User B will keep D_B confidential.
    - Private Key is kept secret by its owner, while the public key is freely distributed.
    - If user A want to send encrypted msg (p) to user B: they will send $E_B(p)$
    - When user B receives the encrypted message, they decrypted it: $D_B(E_B(p))$ which results in the plain message (p).

1. SSL Encryption

- **HTTPS** ≡ (HTTP + SSL/TLS {Transport Layer Security – SSL 3.0} + TCP)
- **SSL** was developed by **Netscape** to provide secure connections over the web. SSL implements RSA (public key encryption) and digital certificates to authenticate the server and to protect private information as it is being passed among parties over the web.
- There will be digital certificate (https://searchsecurity.techtarget.com/definition/digital-certificate) on the server (generated by CA) to be send to the client for authentication and it includes the server public key. With help of certificate, client/server can establish secure session key
- **Typical scenario:** client sends a message to the server, server responds with a digital certificate (includes the server public key) to the client for authentication, using public key encryption the client and server negotiate a ***session key*** (private/symmetric key) to continue the transaction during that session. Once the session key is established, communication proceeds between the client and the server using the session key and the digital certificate.

2. Integrity

2. Integrity: Message Digest (MD)

- **Message Digests (MD)** verifies that data has not been tampered with. It is also a step toward creating and verifying a digital signature.

- **MD implements** the SHA (Secure Hash Algorithm) on the message text and generates a hash value that is passed with the original text.

- **The idea** is sender applies SHA on the msg body and generates a **hash tag** value. Then sends the “message + hash tag value”. Receiver re-computes the hash tag value on the received msg body and compare it with the received hash tag value. Equal hash tag values means the message is not tampered with.

2. Secure Message Digest

- **Secure Message Digest** is called Message Authentication Code (MAC). It requires a secret key to encrypt the hash value. The secret key is shared between the Sender and the Receiver.

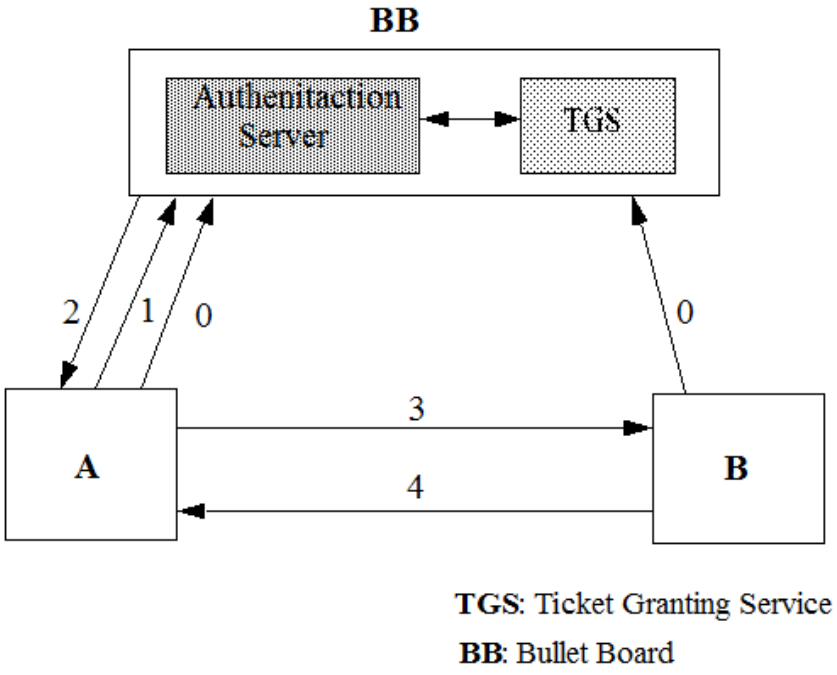
3. Authentication

3. Authentication

- **Authentication** is to Ensure that users are who they claim to be. Java recently introduced Java Authentication and Authorization Service (JAAS). JAAS is based on a plug-in framework, which allows **Kerberos** and sign-on to be implemented for authentication and authorization.

- **Kerberos:** Kerberos is a freely available, open source, protocol developed by MIT. It uses secret key technology to authenticate users in a network. It is used to generate session keys between a client and a server that is valid only during that session, i.e., session key has expiration time-out value.

3. Kerberos



- **Step 0**: K_A, K_B
- **Step 1**: A, B
- **Step 2**: $E_A\{A, K_{A,B}\}$, $E_B\{A,B,K_{A,B}\}$
- **Step 3**: $E_B\{A,B,K_{A,B}\}$, $E_{A,B}\{A,R\}$
- **Step 4**: $E_{A,B}\{R+1\}$

3. Digital Signature

- **This is the electronic equivalent** of written signature. It is based on public key encryption to solve the problem of authentication and integrity.

- **Authentication aspects:**
  - The sender “A” will encrypt the message (P) with its own private key (K_A) - (E_{KprA} (P)) then encrypt the output with the receiver’s “B” public key (K_B) - (E_{KpuB} ((E_{KprA} (P)).
  - On the other side, the receiver decrypt with its own private key first getting (E_{KprA} (P)) then decrypt using the sender’s public key getting (P).
  - Given that B can produce both (P) and E_{KprA}(P); it is clear that only “A” can send E_{KprA}(P) from P!

3. Digital Signature

- **Adding the Integrity aspects:**
  - Use Message Digest (SHA-1 or MD5) algorithm on the Plain text message. The sender uses the above algorithm (E_{KpuB} ((E_{KprA} (MD)) to generate the digital signature. In addition, the sender encrypts the MD with the receiver's public key.

  - **The receiver** uses the above algorithm on the digital signature to get MD value. The receiver decrypt the body using its own private key, apply the same Message Digest algorithm and compares the output with the above MD value. If they match, then the message is authenticated and the integrity of the body of the message is guaranteed.

4. Digital Certificate

- **Server** needs to go to Certificate Authority (CA) to generate a digital certificate. The digital certificate includes: identity of the server, CA issuer, version, validity period, Signature to the certificate, algorithm used to generate the signature, server public key.

- **When server wants to exchange secure messages with client**: It sends to the client the certificate which includes the server public key. The client verify the digital signature and if valid, client now can use the server public key to establish secure communication between the client and the server.

5. Firewalls

- **Firewall:** network components (host/router + software) sitting between inside ("us") and outside ("them)
- **Packet filtering firewalls:** drop packets on basis of source or destination address (i.e., IP address, port)
- **Application gateways:** application specific code intercepts, processes and/or relays application specific packets:
  - e.g., email of telnet gateways
  - application gateway code can be security hardened
  - can log all activity

4. Authorization

4. Authorization

- **Authorization:** What operations on a given resource a user (already authenticated user) can perform?

- **Access Control List (ACL)** An ACL is a data structure with multiple ACL entries. Each ACL entry contains a set of permissions associated with a particular Principal (Principal is an individual, user, group, etc.). Also, each ACL entry is specified as being positive or negative. Positive means permissions are to be granted to that principal, and negative means permissions are to be denied.

- **ACL** is independent of authentication or encryption. The ACL is always consulted after the authentication phase.

4. Authorization

- **Protection Domains:** It is a grouping of “code source” and permissions, i.e., represents all the permissions granted to a code source (such as downloaded code).

- **Construct a protection domain** based on the given code source and set of permissions. When associated with a class, a PD means that the given class was loaded from the site specified in the code source, was signed by the public key specified in the code source and should have permissions to perform the set of operations represented in the permission collection object.

5. Non-repudiation

5. Non-repudiation

- **There is fundamental difference** between digital signature and handwritten signature. Handwritten signature is the same on any document while digital signature is created using the content of the document, and as a result it is different for every document.

- **Timestamping Digital Signature**: Digital signatures do not provide proof that the message has been sent. As an example, a contractor sends a company a digitally signed contract, which later-on he would like to revoke. He can do so by claiming losing his private key.

5. Non-repudiation

- **One technique** to address the above problem is **time stamping**. It binds a time and date to a digital document.
- **Now,** the contractor digitally sign the contract, then send it to a 3rd party to digitally time-stamped (time stamping agency) which affix a timestamp and encrypt the whole package with their own private key, i.e., timestamp cannot be altered.
- **Unless the contractor reports** the private key to have been compromised before the contract was time stamped, the contractor cannot legally prove that the contract was signed by an unauthorized 3rd party.
- **The sender** can also requires the receiver to sign and timestamp the message digitally as proof of receipt.

END