# 19. Security

*Sungyoung Lee*

*College of Engineering*
*KyungHee University*

# Contents

n *The Security Problem*

n *Authentication*

n *Program Threats*

n *System Threats*

n *Securing Systems*

n *Intrusion Detection*

n *Encryption*

n *Windows NT*

# The Security Problem

- Security must consider external environment of the system, and protect it from:
  - unauthorized access
  - malicious modification or destruction
  - accidental introduction of inconsistency

- Easier to protect against accidental than malicious misuse

# Security Problem

- There is no perfectly secure system!
  - Protection can only increase the effort needed to do something bad. It cannot prevent it
  - Every system has holes, it just depends on what they look like
  - Even assuming a technically perfect system, there are always the four Bs:
    - Burglary: steal it
    - Bribery: find whoever has access to what you want and bribe them
    - Blackmail: or photograph them in a compromising position
    - Bludgeoning: or just beat them until they tell you
- Security service
  - Integrity
  - Authentication
  - Authorization
  - Access control
  - Confidentiality

# Cracker’s Basic Steps

- Gather information
  - as much information about your site as possible
- Use port scanner
  - to gather information about what services are running on hosts
  - Search for weak security services
- Get a login account
  - Doesn’t matter whose account
- Get root privilege
  - Bugs in programs or badly configured systems
- Keep root privilege
  - Leave some sort of backdoor for future access

# Authentication

- User identity most often established through *passwords*, can be considered a special case of either keys or capabilities

- Passwords must be kept secret
  - Frequent change of passwords
  - Use of “non-guessable” passwords
  - Log all invalid access attempts

- Passwords may also either be encrypted or allowed to be used only once

# Physical Security

- Hardware security
  - Restrict access to equipments
    - Smart card (ID card)
    - Bio-metric access control

- BIOS security
  - Set a boot password
  - Prevent booting from CD-ROM or floppy drives

- Session security
  - Some shells (e.g. tcsh) provide the automatic logout facility if there is no activity during the specified time period
  - vlock (for locking a virtual terminal) / xlock
  - Screen savers

# Account Security

- Authentication
  - Make sure we know who we are talking to
  - Usually done with passwords
    - First line of defense and single biggest security hole
  - Problems in passwords:
    - Users who write their password on paper for all to see
    - Type password slowly that others can see
    - Dumb passwords like “password”
    - Passwords should be long and obscure – unfortunately easily forgotten and usually written down
  - Passwords should not be stored in a directly-readable form
    - Use some sort of one-way-transformation (a “secure hash”) and store that

  - *Cf) CHAP (Challenge Handshake Authentication Protocol)*

# Account Security (Cont’d)

- Authentication alternatives
  - Some alternatives
    - Physical keys: badges, smart cards, …
    - Biometric keys: Fingerprints, iris prints, facial profiles, voice prints, hand geometry, signature analysis …
    - Passwords using images
  - Should not be forgeable or copiable
  - Can be stolen, but the owner should know if it is
    - Need to invalidate old one

# Account Security (Cont’d)

- Authorization
  - Determine if x is allowed to do y
    - Can be represented as an “access matrix”
  - Access control lists (ACLs)
    - With each object, indicate which users are allowed to perform which operations
    - Simple and used in almost all file systems
  - Capabilities
    - With each users, indicate which resources may be accessed and in what ways
    - Frequently do both naming and protection: Can only “see” an object if you have a capability for it
    - Used in systems that need to be very secure

# File System Security

- Setuid/setgid programs
  - Badly written setuid programs may contain a security hole
    - Know of all setuid and setgid programs on your system
    - Setuid programs that are not needed should be deleted
    - Never allow setuid/setgid files in user’s home directories
    - Use nosuid option in fstab file for home file system and for NFS-mounted file system
    - Maintain a check on any new setuid programs:

      ```
      find / -type f –perm 2000 –o perm 4000 –o perm 6000
      ```
    - Never write setuid/setgid shell programs

# File System Security (Cont’d)

- Search paths
  - Many users include the current directory in their search path
  - A cracker could place programs with the same name as standard commands everywhere they have write access in directory hierarchy
    - The fake program may have malicious code, or capture data from the user pretending to be the real application
  - Place current directory last in the path
    - Alternatively use full path names (e.g. /bin/su)
  - Current directory SHOULD NOT be in the search path for root user

# File System Security (Cont'd)

- Other countermeasures
  - Carefully specify default permissions: umask
  - Put a limitation on the file system usage: quota
  - Check file system integrity regularly: find, tripwire, …
    - Files without known owners may indicate unauthorized access: find / -nouser –o –nogroup
    - Files with "other" write permission (o+w) may indicate a problem: find / -type f –perm 2
  - Use encrypted file system
    - CFS (Cryptographic File System)
    - TCFS (Transparent CFS), etc.
  - Backup file system: tar, dd, …
  - Monitor system logs

# Network Security

- n Use secure protocols
  - ü Don’t let the plain password float around the network
  - ü Secure shell (ssh) suite of programs encrypts the communications of many of protocols
    - § ssh (telnet), slogin (rlogin), sftp (ftp)
  - ü Use secure http (https) for secure connection
  - ü Secure Socket Layer (SSL) provides data encryption of all data that passes between clients and server
  - ü IPsec protocol: encrypt every IP packet
    - § Required for IPv6, optional for IPv4

# Network Security (Cont'd)

- TCP wrappers
  - Monitors/filters Internet services such as telnet, ftp, finger, etc.
  - Similar to Internet super daemon, inetd
  - Before connecting the client to the service program, log the activity and check if it should be permitted
    - /etc/hosts.allow, /etc/hosts.deny
  - You should be able to detect cracking intention or activity from the log

# Network Security (Cont’d)

- Firewalls
  - Firewall
    - Creates a filter or protective layer between an organization’s internal networks and any external networks to which they are connected
  - ipchain: packet filtering firewall
    - Examine each packet header to decide the action
    - e.g. block incoming ICMP echo requests:

      ipchains –A input –i eth0 –p icmp –s 0/0 –d 0/0 –l –j REJECT
  - Proxy firewall
    - Standard: require client-side configuration. Client connects to a special port
    - Transparent: similar to packet filter firewall, but controls traffic

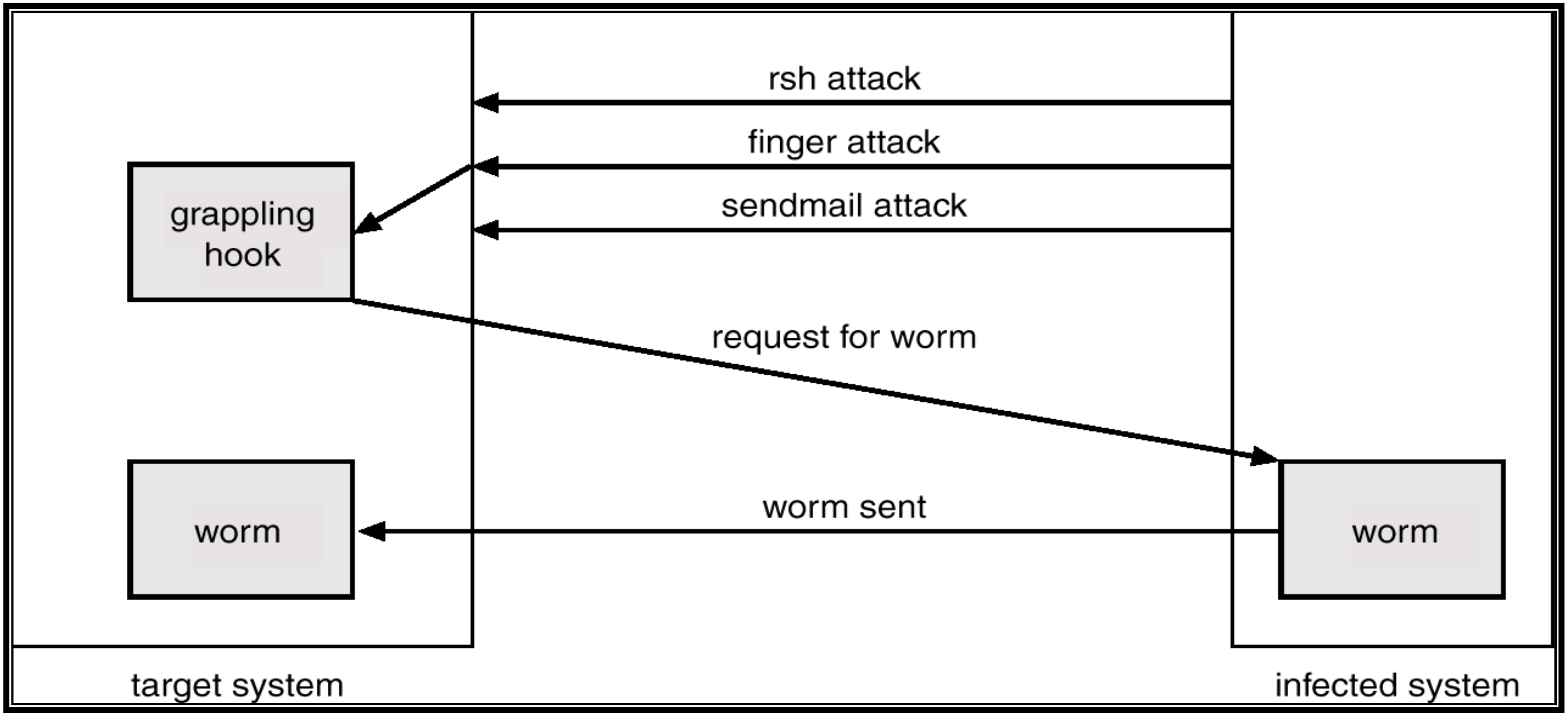
# Program Threats

- Trojan Horse
  - Code segment that misuses its environment
  - Exploits mechanisms for allowing programs written by users to be executed by other users
- Trap Door
  - Specific user identifier or password that circumvents normal security procedures
  - Could be included in a compiler
- Stack and Buffer Overflow
  - Exploits a bug in a program (overflow either the stack or memory buffers)

# System Threats

- Worms
  - Use spawn mechanism
  - Standalone program
- Internet worm
  - Exploited UNIX networking features (remote access) and bugs in *finger* and *sendmail* programs
  - Grappling hook program uploaded main worm program
- Viruses
  - Fragment of code embedded in a legitimate program
  - Mainly effect microcomputer systems
  - Downloading viral programs from public bulletin boards or exchanging floppy disks containing an infection
  - *Safe computing*
- Denial of Service
  - Overload the targeted computer preventing it from doing any useful work

# The Morris Internet Worm



# Security Threats

- Physical threats
  - Acts of nature: floods, fire, earthquake, explosion, etc.
  - Intruder takes computers, dig up network cable, or access system consoles
- Logical threats
  - Caused by problems with computer software
    - Misuse by people (e.g. easy-to-guess passwords)
    - Bugs in programs or in their interaction with each other
- Operational threats
  - No security policy, incomplete enforcement
- Denial of service
  - Prevent computer from providing services through
    - wasting resources of computer
    - flooding services on your system, thus preventing them from providing service to legitimate clients

# Attacks

- Dictionary attacks
  - crack, nutcrack, John the Ripper, etc.
  - *crack* program found 10-20% of passwords could be guessed, using a password list containing variations on login names, user’s first and last names and a list of 1800 common first names
- Login spoofing
  - Simulate login process
  - Need to have the login sequence start with a key combination that user programs cannot catch
    - CTRL-ALT-DEL in Windows 2000.

# Attacks (Cont’d)

- n Trojan horses
  - ü A seemingly innocent program contains code to perform an unexpected and undesirable function
  - ü To have the Trojan horse run, the person planting it first has to get the program carrying it executed
    - § Attract attention and encourage people to download and execute it

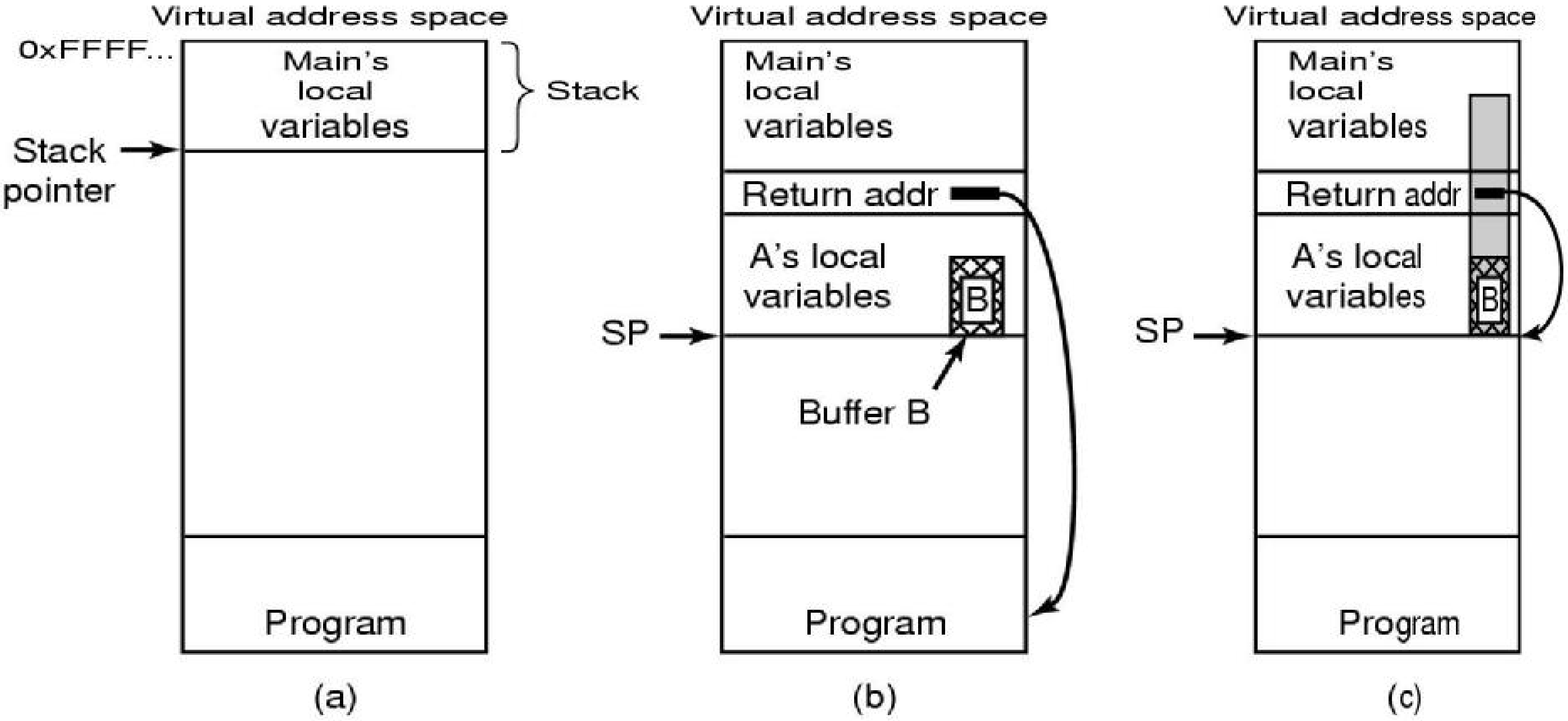
- n Logic bomb
  - ü A piece of code written by one of a company’s programmers and secretly inserted into the OS
  - ü OK as long as the programmer feeds it its daily password
  - ü If the programmer is suddenly fired, the logic bomb explodes
    - § clear the disk, erase files at random, encrypt essential files, etc.

# Attacks (Cont’d)

- Trap door
  - Created by the code inserted into the system by a system programmer to bypass some normal check
    - What happens if the programmer leaves the company?
    - Some special key sequences lead you to the “debug” mode in your mobile phone
  - Need to have code reviews as standard practice
  - Difficult to do in open-source software

# Attacks (Cont'd)

- Stack and Buffer overflow
  - Do array bounds checking!



# Attacks (Cont'd)

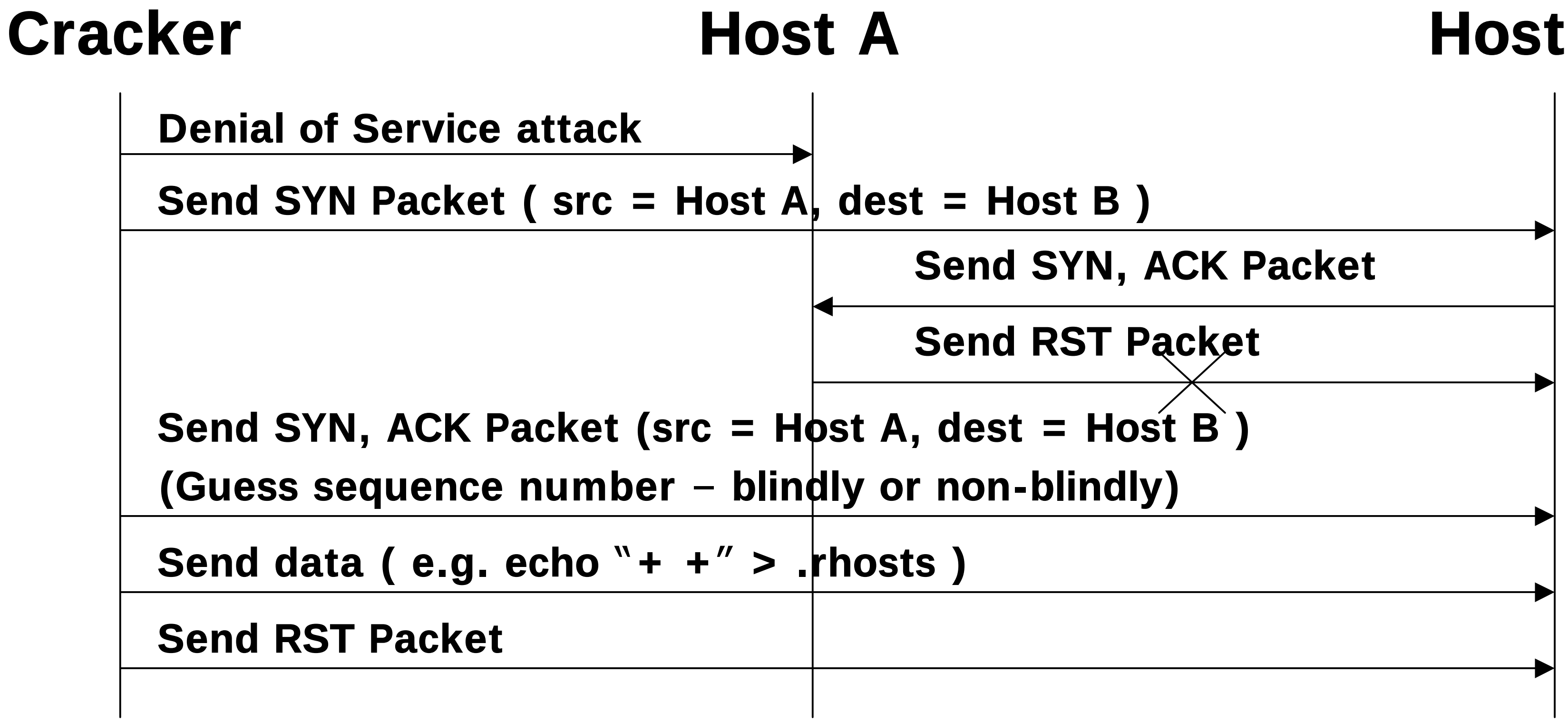
- Virus and worms
  - Virus is a program that can reproduce itself by attaching its code to another program
  - Worms are like viruses but are also capable of spreading itself from machine to machine via network
  - Types
    - memory resident viruses: e.g. intercept system call traps to infect other programs
    - boot sector viruses
    - device driver viruses: officially loaded at boot time
    - macro viruses: Microsoft Office

# Attacks (Cont’d)

- Packet sniffing
  - Listens to ethernet traffic over LAN
  - Ethernet adapter in promiscuous mode
    - Need root privilege
  - Can see all data passing between hosts on the network
  - Can gather usernames and passwords
    - telnet, ftp, httpd, pop3, imap, etc.
  - tcpdump and sniffit are software sniffers

# Attacks (Cont’d)

- IP spoofing
  - Steal an authorized IP and use it



Cracker

Host A

Host B

Denial of Service attack

Send SYN Packet ( src = Host A, dest = Host B )

Send SYN, ACK Packet

Send RST Packet

Send SYN, ACK Packet (src = Host A, dest = Host B )
(Guess sequence number – blindly or non-blindly)

Send data ( e.g. echo “+ +” > .rhosts )

Send RST Packet

# Attacks (Cont'd)

- Denial of service: internal attacks
  - Use up all resources and make system crash
  - Attacking resources: disk, memory, process, …
  - Examples
    - Shell script: while (1) { mkdir foo; cd foo; }
    - C: while (1) { fork(); ((int *) malloc(100000))[40] = 1; }
  - Done by a local user, and in most cases by accident

# Attacks (Cont'd)

- Denial of service: external attacks
  - Application level
    - Mail bombing
    - Buffer overflow
    - Java Applet attack
  - Protocol level
    - TCP SYN flooding
    - Ping flooding
  - Network level
    - UDP Storming
- Distributed DOS (DDOS)
  - Use multiple machines

# Threat Monitoring

n Check for suspicious patterns of activity

ü i.e., several incorrect password attempts may signal password guessing

n Audit log

ü records the time, user, and type of all accesses to an object

ü useful for recovery from a violation and developing better security measures

n Scan the system periodically for security holes

ü done when the computer is relatively unused

# Threat Monitoring (Cont’d)

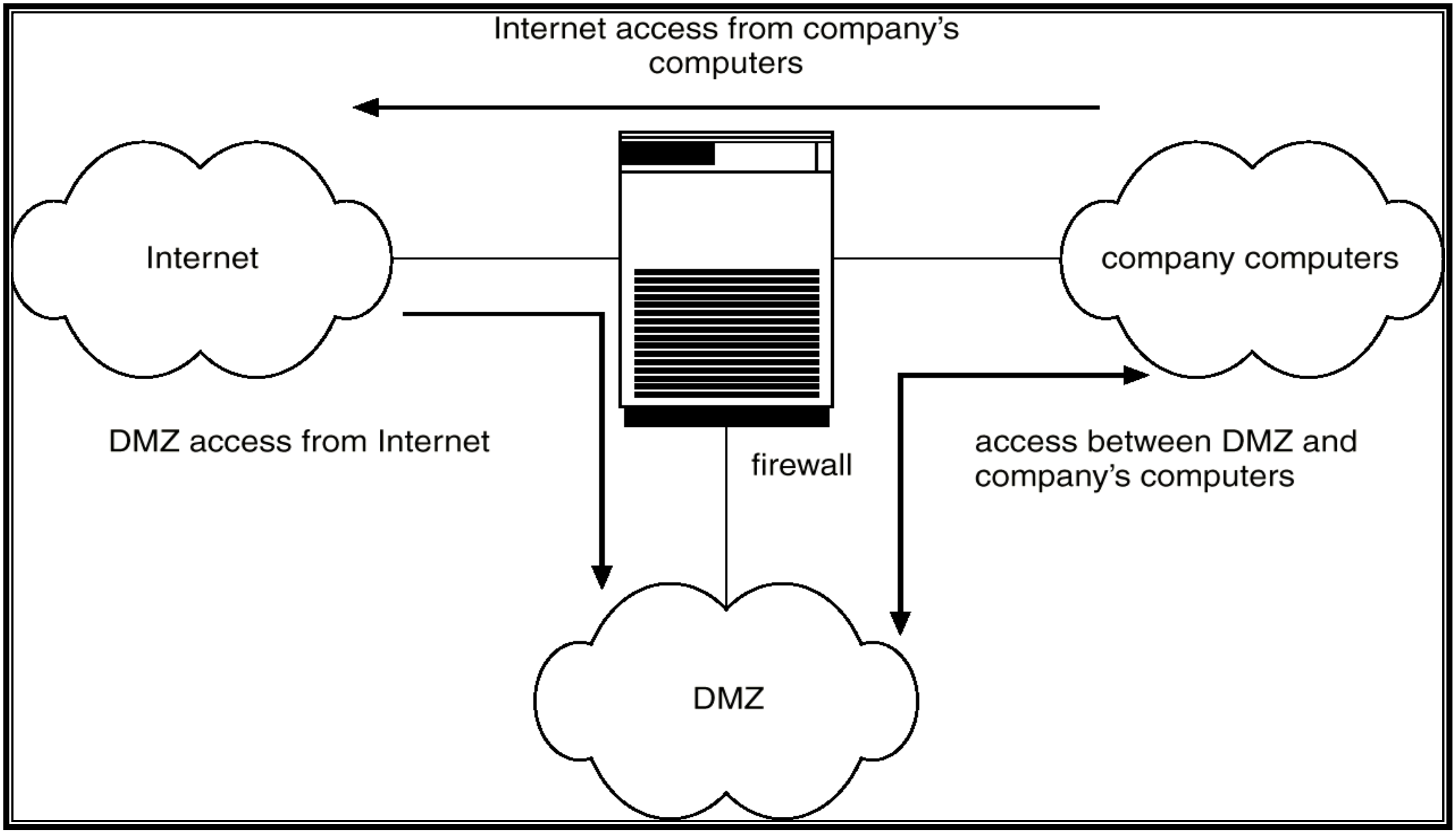
n Check for:

  - ü Short or easy-to-guess passwords
  - ü Unauthorized set-uid programs
  - ü Unauthorized programs in system directories
  - ü Unexpected long-running processes
  - ü Improper directory protections
  - ü Improper protections on system data files
  - ü Dangerous entries in the program search path (Trojan horse)
  - ü Changes to system programs: monitor checksum values

# Firewall

- A firewall is placed between trusted and untrusted hosts
- The firewall limits network access between these two security domains

# Network Security Through Domain Separation Via Firewall


Internet access from company's computers
Internet
company computers
DMZ access from Internet
firewall
access between DMZ and company's computers
DMZ

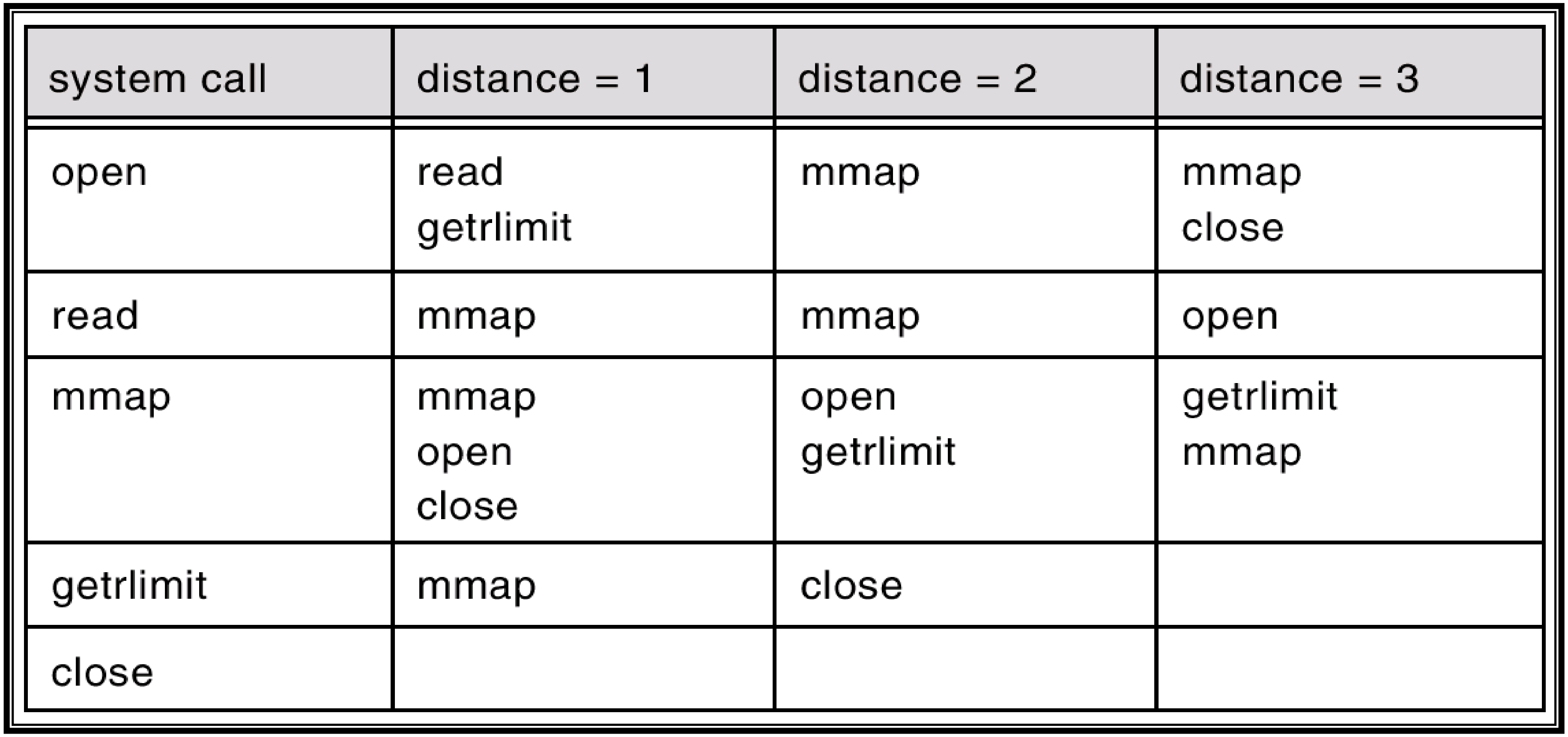
# Intrusion Detection

- Detect attempts to intrude into computer systems

- Detection methods:
  - Auditing and logging
  - Tripwire
    - UNIX software that checks if certain files and directories have been altered
    - I.e. password files

- System call monitoring

# Data Structure Derived From System-Call Sequence



| system call | distance = 1 | distance = 2 | distance = 3 |
|---|---|---|---|
| open | read<br>getrlimit | mmap | mmap<br>close |
| read | mmap | mmap | open |
| mmap | mmap<br>open<br>close | open<br>getrlimit | getrlimit<br>mmap |
| getrlimit | mmap | close | |
| close | | | |

# Encryption

- Encrypt clear text into cipher text

- Properties of good encryption technique:
  - Relatively simple for authorized users to incrypt and decrypt data
  - Encryption scheme depends not on the secrecy of the algorithm but on a parameter of the algorithm called the encryption key
  - Extremely difficult for an intruder to determine the encryption key

- *Data Encryption Standard* substitutes characters and rearranges their order on the basis of an encryption key provided to authorized users via a secure mechanism
  - Scheme only as secure as the mechanism

# Encryption (Cont’d)

- Public-key encryption based on each user having two keys:
  - public key – published key used to encrypt data
  - private key – key known only to individual user used to decrypt data

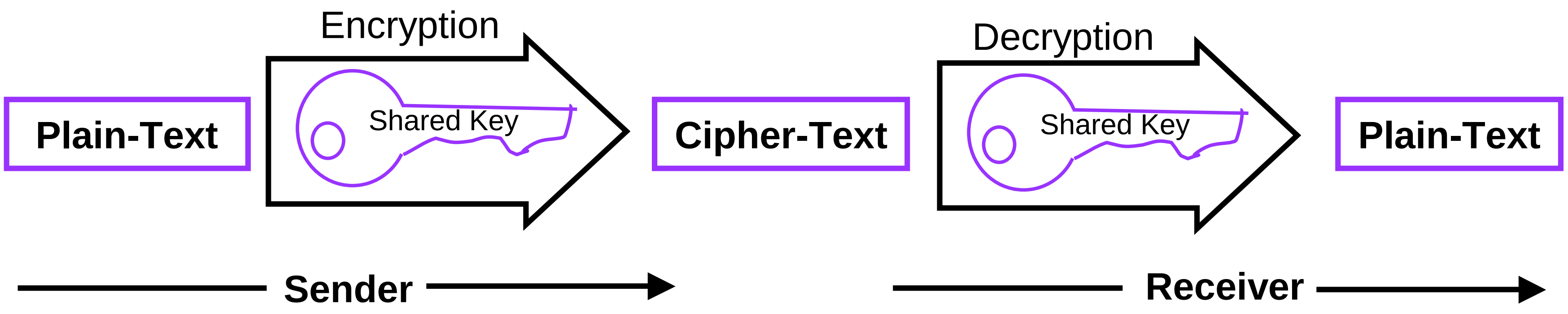
- Must be an encryption scheme that can be made public without making it easy to figure out the decryption scheme
  - Efficient algorithm for testing whether or not a number is prime
  - No efficient algorithm is know for finding the prime factors of a number

# Encryption Example - SSL

- SSL – Secure Socket Layer
- Cryptographic protocol that limits two computers to only exchange messages with each other
- Used between web servers and browsers for secure communication (credit card numbers)
- The server is verified with a **certificate**
- Communication between each computers uses symmetric key cryptography

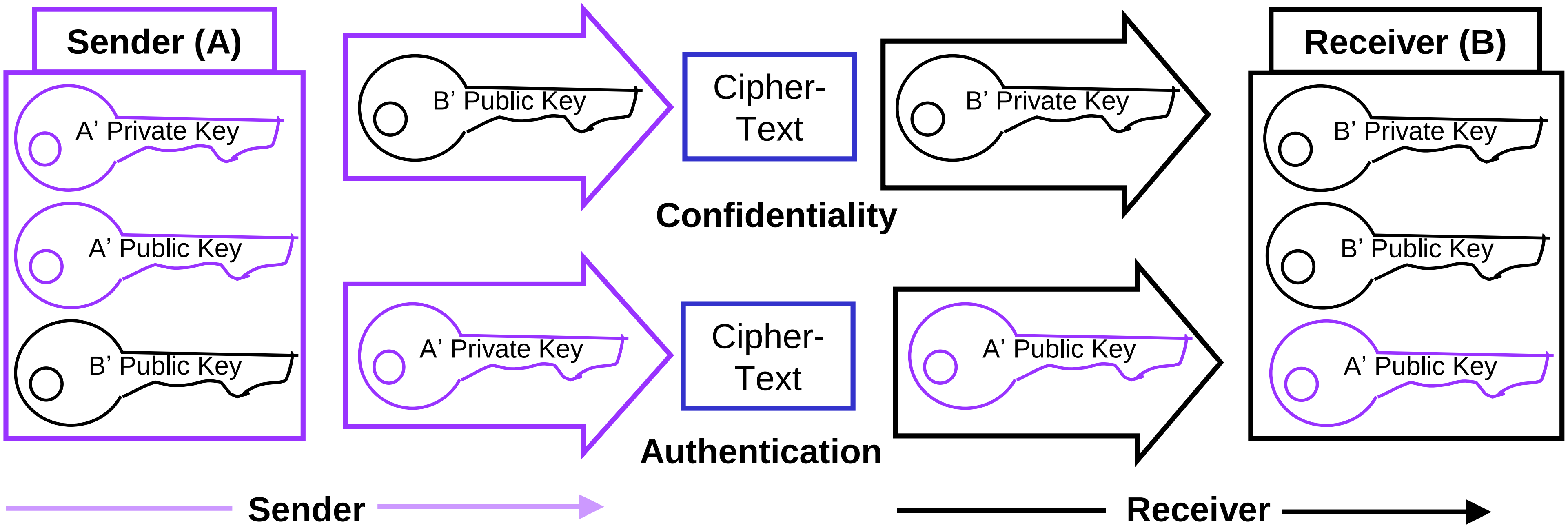
# Basic Concept of Cryptography

- n What is cryptography
  - ü The science of obfuscating data
  - ü Can provide authentication, confidentiality, data integrity and etc.
  - ü Cryptography algorithm is open, but key MUST be confidential
- n Shared Key cryptography
  - ü Both of peers share the same key
  - ü DES(Data Encryption Standard)
    - § Bit operation ( key size : 64bit, 128 bit)
    - § Can provide authentication and confidentiality
    - § ***<u>How can distribute the shared key secret and keep it secret </u>***?



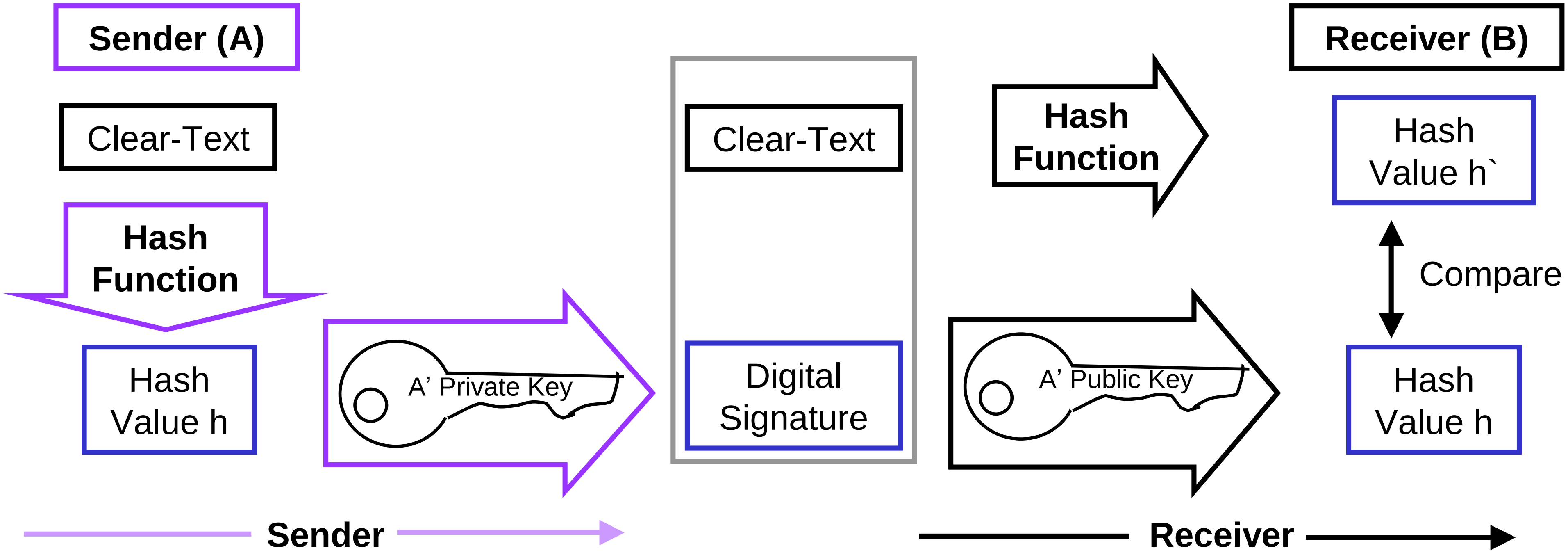
# Basic Concept of Cryptography (Cont'd)

- Public Key Cryptography
  - Both of peers have its own private key and public keys
  - Key pair
    - well known 'Public Key' and secret 'Private Key'
  - Can provide confidentiality and authentication
  - RSA : well known algorithm



# Basic Concept of Cryptography (Cont’d)

- Digital Signature
  - minimize encryption processing



- Message Authentication Code
  - Another way to provide authenticity without secrecy
  - Hash function based HMAC, MD5 and etc.

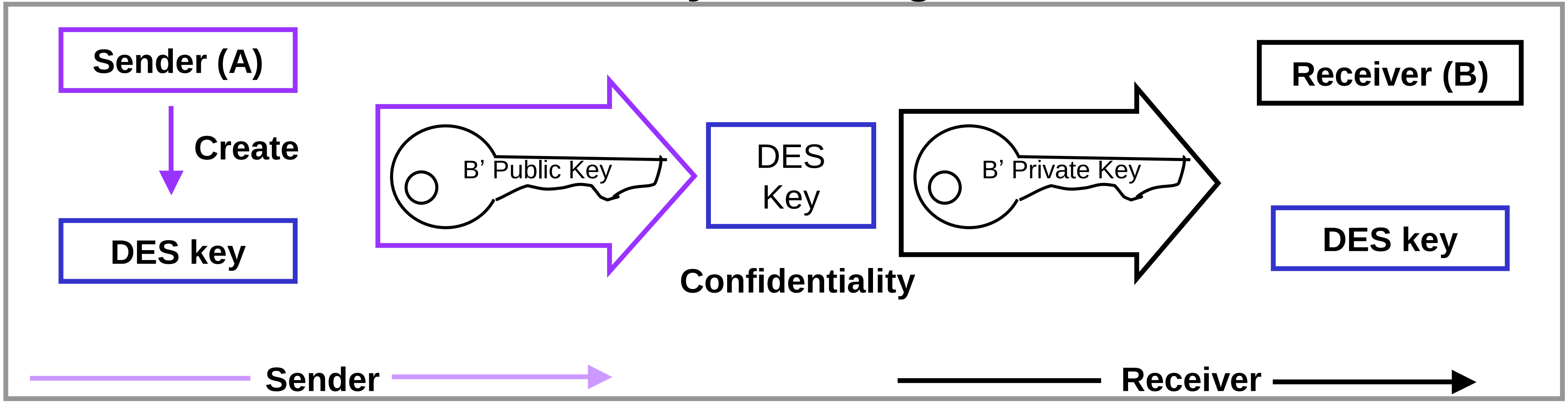
# Applied Cryptography for Network

- n Validity of key
  - ü DES < several hours
  - ü RSA < several days
  - ü ***<u>How can keep the connection secure?</u>***

- n Security
  - ü System security
    - § Create DES key every session
    - § Transfer DES key by RSA security
  - ü Connection (session ) security
    - § One time key created by DES
  - ü applied for SSL, TSL, IPSEC and etc.

# Applied Cryptography for Network (Cont'd)

## Key exchange



Sender (A)

Create

DES key

B' Public Key

DES Key

Confidentiality

B' Private Key

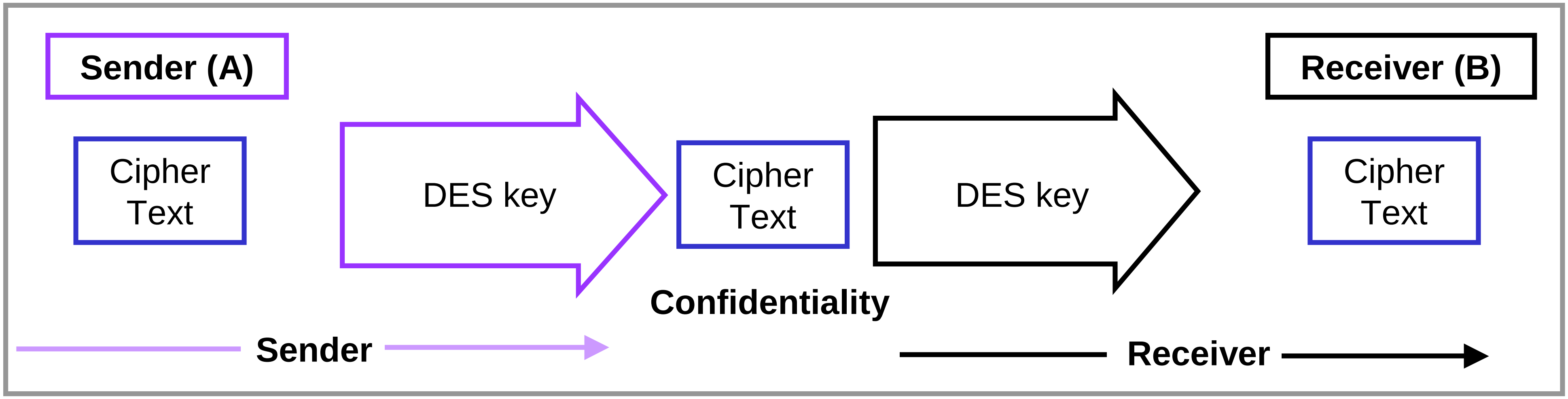
Receiver (B)

DES key

Sender

Receiver

## Data transfer



Sender (A)

Cipher Text

DES key

Cipher Text

Confidentiality

DES key

Receiver (B)

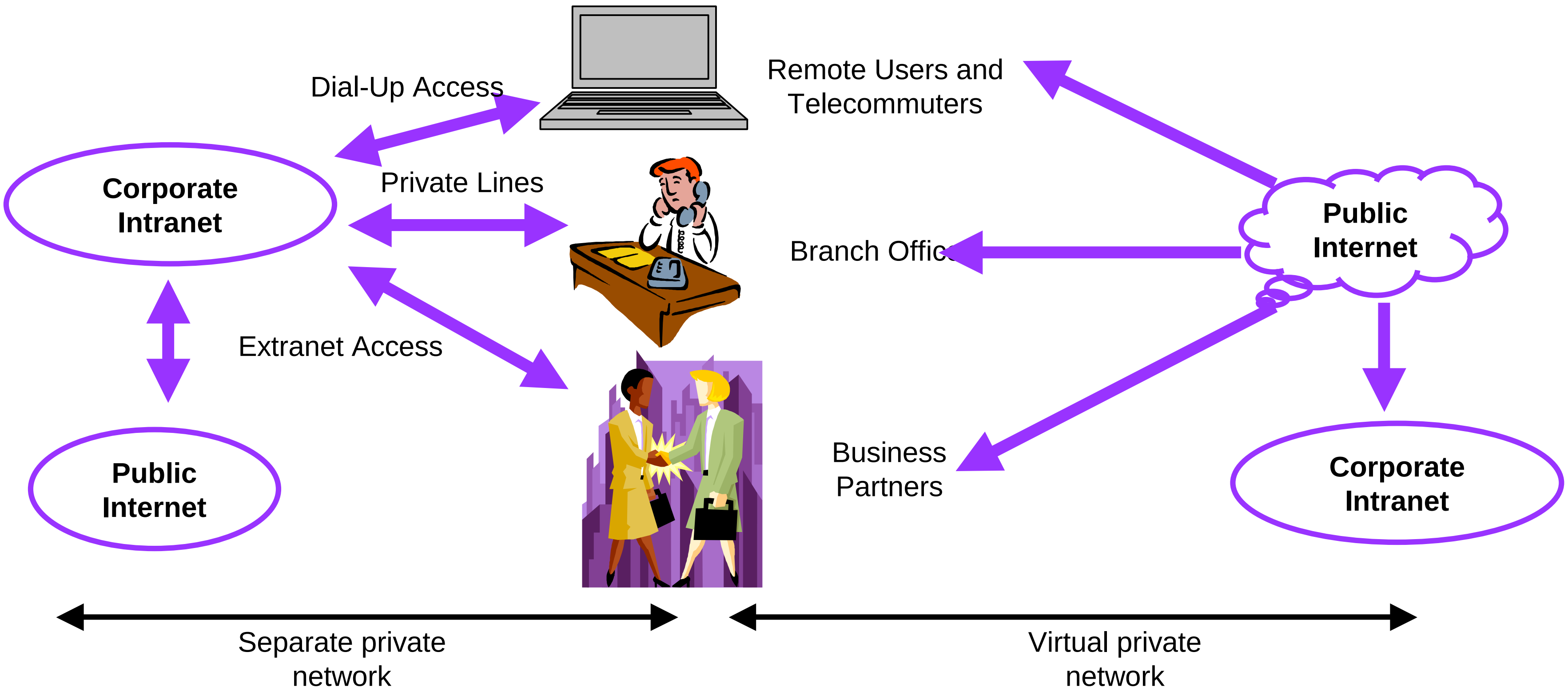
Cipher Text

Sender

Receiver

# Virtual Private Network (VPN)

- Virtual
  - ü ***No*** physical infrastructure ***dedicated*** to the private network
- Private
  - ü Keep data confidential so that it can be received by an intended receiver

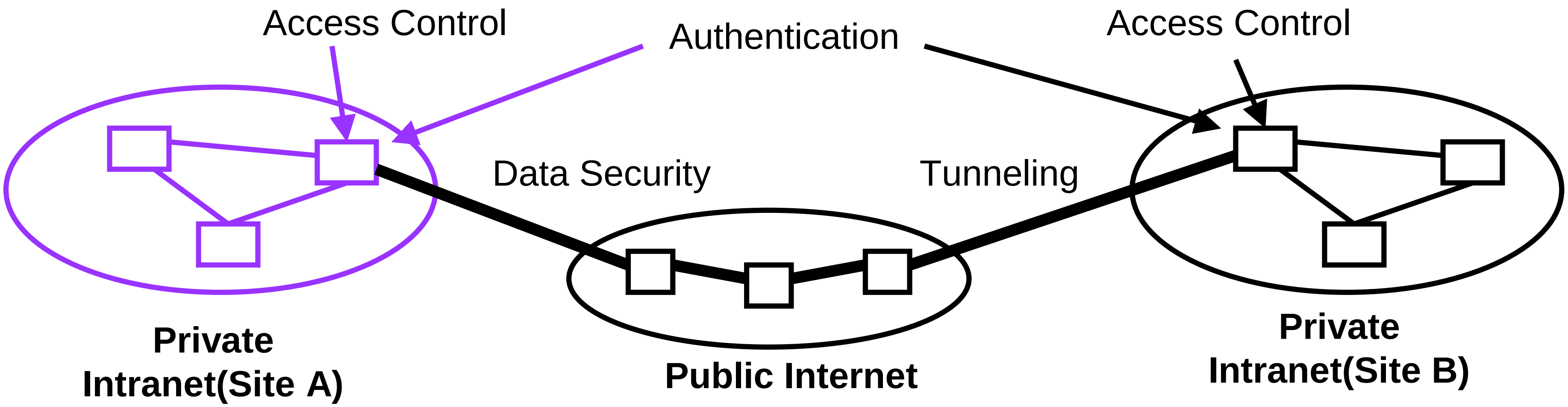


# VPN Motivation

- n Ubiquitous coverage
  - ü Easy to access the private network
  - ü Sharing the public infrastructure
  - ü ***How to keep 'Privacy'*** ?
- n Cost reduction
  - ü Impractical to have a physical separate infrastructure
  - ü Sharing an internet-based VPN
- n Security
  - ü Using cryptography
    - § Authentication, access control, confidentiality and integrity.
  - ü ***Can keep 'Privacy'***
- n E-Commerce

In conclusion,
VPNs can provide both **interconnectivity and security**

# VPN Technologies



- Tunneling
  - PPTP, L2TP, L2F, MPLS , IPsec and etc.
- Authentication
  - Radius, CHAP, PKI and etc.
- Access Control
  - PKI and etc.
- Data Security
  - IPsec, PKI, SSL, TSL and etc.

# Computer Security Classifications

- U.S. Department of Defense outlines four divisions of computer security
  - **A**, **B**, **C**, and **D**
- **D**
  - Minimal security
- **C**
  - Provides discretionary protection through auditing
  - Divided into **C1** and **C2**
  - **C1** identifies cooperating users with the same level of protection
  - **C2** allows user-level access control.
- **B**
  - All the properties of **C**, however each object may have unique sensitivity labels
  - Divided into **B1**, **B2**, and **B3**
- **A**
  - Uses formal design and verification techniques to ensure security

# Windows NT Example

- Configurable security allows policies ranging from D to C2

- Security is based on user accounts where each user has a security ID

- Uses a subject model to ensure access security
  - A subject tracks and manages permissions for each program that a user runs

- Each object in Windows NT has a security attribute defined by a security descriptor
  - For example, a file has a security descriptor that indicates the access permissions for all users