

# Security Basics

Platform Technologies

*Based on Tanenbaum, Modern Operating Systems*

# The Security Environment

- **Confidentiality** - if the owner of data decides to make available only to certain people and no others, the system should guarantee that release of the data to unauthorised people never occurs
- **Integrity** - unauthorised users should not be able to modify any data (changing the data, removing data and adding false data) without the owner's permission
- **Availability** - nobody can disturb the system to make it unusable, such as in the form of **denial-of-service** attacks that are increasingly common

Goal	Threat
Confidentiality	Exposure of data
Integrity	Tampering with data
Availability	Denial of service

# Operating System Security

- Often the ways to compromise the security of a computer system are not very sophisticated.
  - E.g. easy to guess passwords, writing down passwords
- Exploiting such behaviours of humans, social engineering, is a significant challenge.
  - E.g. requirement to frequent password change vs. writing down passwords
- However, operating systems should also account for targeted attacks that are more sophisticated in nature, targeting the security framework of operating systems.

# Operating System Security

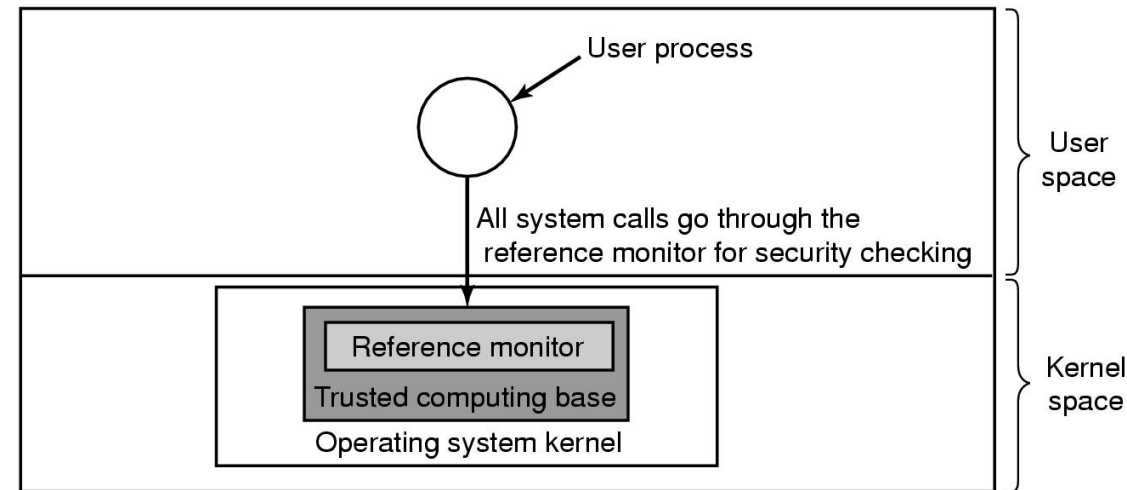
- Passive attacks
  - try to steal information passively
  - sniff the network traffic and tries to break the encryption to get to the data
- Active attacks
  - try to make a computer program misbehave
  - take control of a user's Web browser to make it execute malicious code

# Can we build secure systems?

- Is it possible to build a secure computer system?
  - In principle, software can be free of bugs and we can even verify that it is secure—as long as that software is not too large or complicated.
  - Unfortunately, computer systems today are horrendously complicated.
- If so, why is it not done?
  - People are not willing to leave what they are using, even if it's not secure
  - The only known way to build a secure system is to keep it simple. Features are the enemy of security.
  - But, today's feature-rich software have more complexity, more code, more bugs, and more security errors.

# Trusted Computing Base (TCB)

- In the security world, people often talk about **trusted systems** rather than secure systems.
- These are systems that have formally stated security requirements and meet these requirements.
- At the heart of every trusted system is a minimal **TCB (Trusted Computing Base)** consisting of the hardware and software necessary for enforcing all the security rules.
- If the trusted computing base is working to specification, the system security cannot be compromised, no matter what else is wrong.

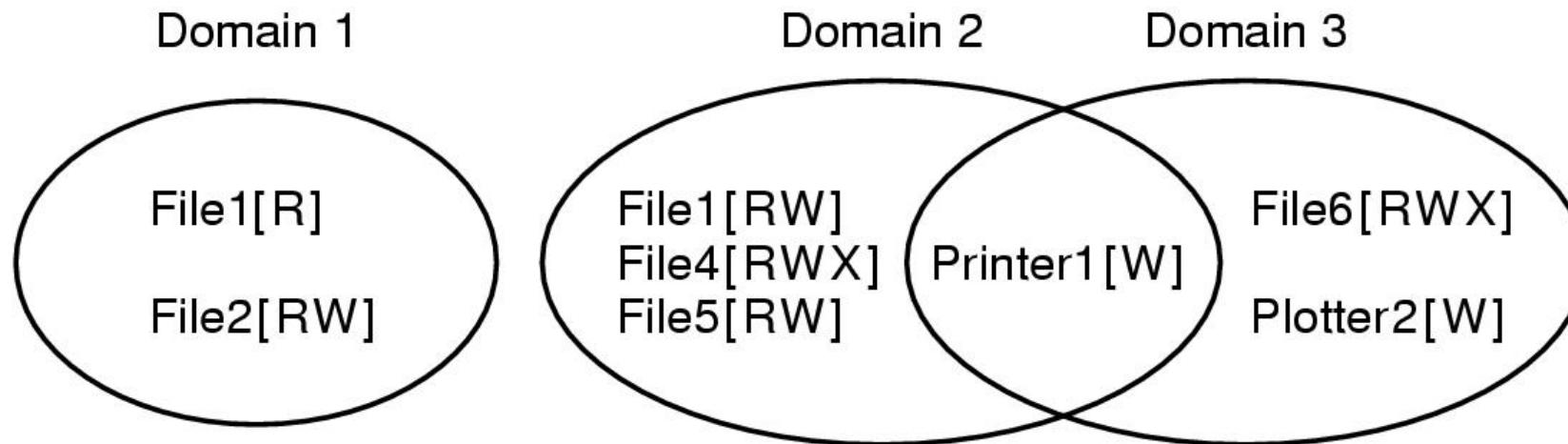


# Controlling Access to Resources

- A computer system contains many resources, or “objects,” that need to be protected.
- These objects can be hardware (e.g., CPUs, memory pages, disk drives, or printers) or software (e.g., processes, files, or databases).
- A model of what is to be protected and who is allowed to do what is necessary for the operating system.
- There are various models for doing this,
  1. Protection Domains
  2. Access Control Lists
  3. Capabilities

# Protection Domains

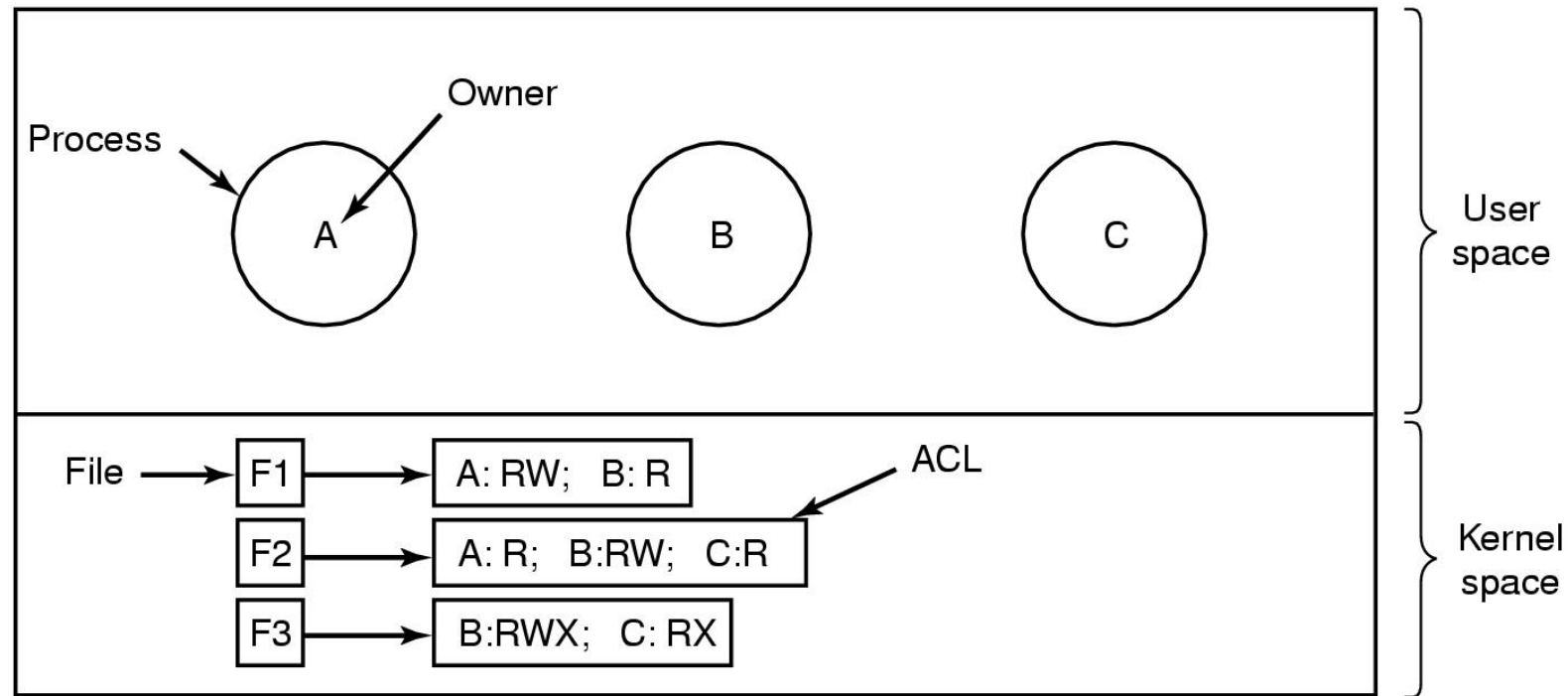
- A **domain** is a set of (object, rights) pairs.
- Each pair specifies an object and some subset of the operations that can be performed on it.
- A **right** in this context means permission to perform one of the operations.
- E.g. Unix/Linux file permissions with UID/GID





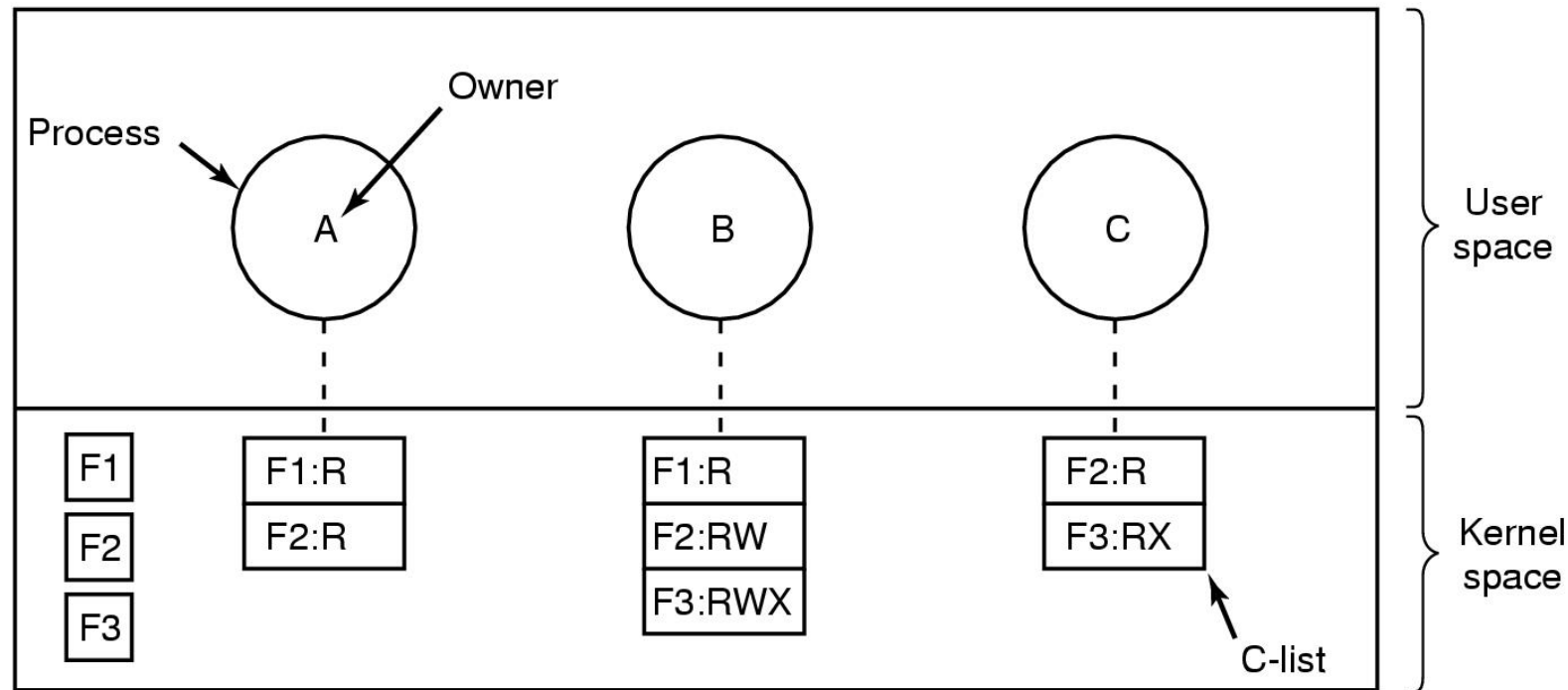
# Access Control Lists

- An **Access Control List (ACL)** consists of associating with each object an (ordered) list containing all the domains that may access the object, and how.



# Capabilities

- A **capability list** (or **C-list**) is a list of objects associated with each process that may be accessed, along with an indication of which operations are permitted on each, in other words, its domain.



# Authentication

- Every *secured* computer system must require all users to be authenticated at login time.
- General principles of authenticating users:
  1. Something the user knows – Known things password, PIN
  2. Something the user has – Physical objects like smartcard, phone
  3. Something the user is – Biomatrices like fingerprint, iris scan

# Authentication

- A key problem with password login is the use of weak passwords
- **Challenge-Response Authentication** is a variation on the password idea is to have each new user provide a long list of questions and answers that are then stored on the server securely, and asked for at the time of authentication
- Authentication Using a Physical Object or Authentication Using Biometrics can add additional layer of security to the authentication process

# Common Intruders

- Casual prying by nontechnical users.
- Snooping by insiders.
- Determined attempts to make money.
- Commercial or military espionage.

# Attack Surfaces

- An attack surface consists of the reachable and exploitable vulnerabilities in a system
- Examples:
  - Open ports on outward facing Web and other servers, and code listening on those ports
  - Services available in a firewall
  - Code that processes incoming data, email, XML, office documents, etc.
  - Interfaces and Web forms
  - An employee with access to sensitive information vulnerable to a social engineering attack

# Attack Surfaces

- Network attack surface
  - Refers to vulnerabilities over an enterprise network, wide-area network, or the Internet
    - E.g. DoS, intruders exploiting network protocol vulnerabilities
- Software attack surface
  - Refers to vulnerabilities in application, utility, or operating system code
- Human attack surface
  - Refers to vulnerabilities created by personnel or outsiders
  - E.g. social engineering, insider traitors

# Common network attacks and countermeasures

- Finding a way into the network
  - Firewalls
- Exploiting software bugs, buffer overflows
  - Intrusion Detection Systems
- Denial of Service
  - Ingress filtering, IDS
- TCP hijacking
  - IPSec
- Packet sniffing
  - Encryption (SSH, SSL, HTTPS)
- Social problems
  - Education



# Denial of Service

- An interruption in an authorised user's access to a computer network, typically one caused with malicious intent.
- To make a network service unusable, usually by overloading the server or network
- Many different kinds of DoS attacks
  - SYN flooding
  - SMURF
  - Distributed attacks
  - Mini Case Study: Code-Red

# Distributed Denial of Service

- Same techniques as regular DoS, but on a much larger scale
- Example: Sub7Server Trojan and IRC bots
  - Infect a large number of machines with a “zombie” program
  - Zombie program logs into an IRC channel and awaits commands
  - Example:
    - Bot command: `!p4 207.71.92.193`
    - Result: runs `ping.exe 207.71.92.193 -l 65500 -n 10000`
    - Sends 10,000 64k packets to the host (655MB!)
  - Read more at: <http://grc.com/dos/grcdos.htm>

# Malware & Spyware


- Malicious is software that is commonly spread over the internet, which can be used for a form of blackmail
  - Example: Encrypts files on victim disk, then displays a message asking for money transfer to decrypt the file system
- Spyware is software that is stealthily loaded onto a PC without the owner's knowledge and runs in the background doing things behind the owner's back.

# Insider Attacks

- These are executed by programmers or employees of the company running the computer to be protected or making critical software.
- Logic Bombs
  - a piece of code written by one of a company's (currently employed) programmers and secretly inserted into the production system
  - in the event of their firing and absence of a daily input of password, the system can do any pre-programmed malicious actions
- Back Doors
  - a programmer could add code to the login program to allow anyone to log in using the login name "zzzzz" no matter what was in the password file
- Login Spoofing

# Social Engineering

- Social engineering is a collection of techniques intended to trick people into divulging private information. Includes calls emails, web sites, text messages, interviews, etc.



**Hello, I'm calling from Technology for America – we're a non-profit organization, working to help ensure that the U.S. stays at the forefront of computer technology.**

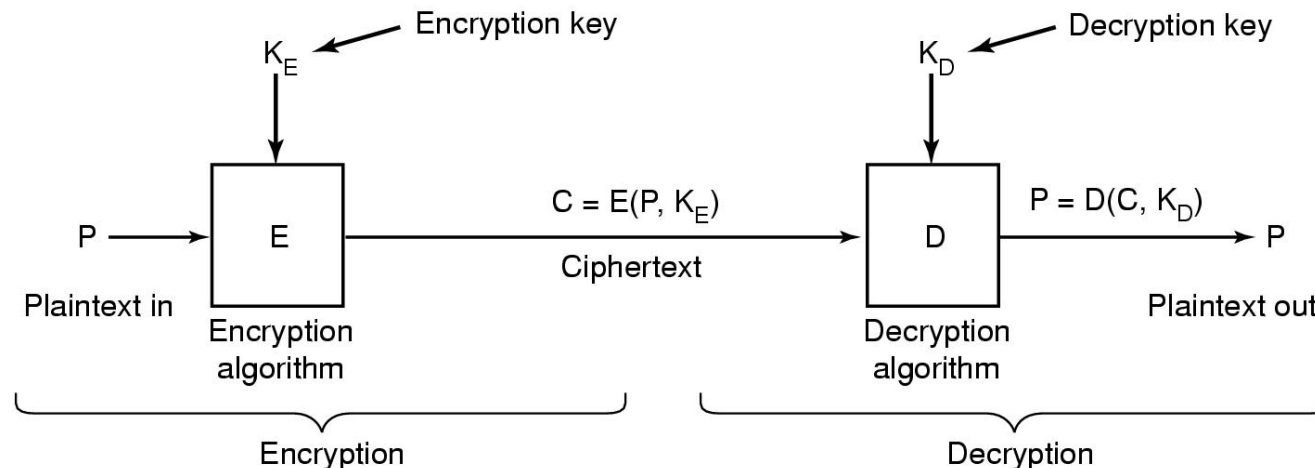
**Today we're conducting a telephone survey about the usage of computer systems. Can I ask you a few questions about your computer system?**

# Defences

- **Defence in depth:**
  - there should be multiple layers of security so that if one of them is breached, there are still others to overcome
- There are various layers of security that can be applied to an OS:
  - Anti Virus
  - Code Signing
  - Model-based Intrusion Detection
  - Sandboxing
  - Firewalls
  - Virtual Private Networks

# Basic Cryptography

- Cryptography plays an important role in security and operating systems use cryptography in many places.
  - Some file systems can encrypt all the data on disk
  - Protocols like IPSec may encrypt and/or sign all network packets
  - Most operating systems scramble authentication passwords
- Take a message or file, called the **plaintext**, and encrypt it into **ciphertext** in such a way that only authorized people know how to convert it back to plaintext.



# Secret-Key Cryptography

- Monoalphabetic substitution:
  - Plaintext:                ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - Ciphertext:              QWERTYUIOPASDFGHJKLZXCVBNM
  - Plaintext ATTACK would be transformed into the ciphertext QZZQEA
  - Decryption key:        KXVMCNOHPQRSZYIJADLEGWBUFT
- Given the encryption key it is easy to find the decryption key.
  - With a small amount of ciphertext, the cipher can be broken.
  - Symmetric-key cryptography
- The basic attack takes advantage of the statistical properties of natural languages.
  - In English, for example, *e* is the most common letter, followed by *t*, *o*, *a*, *n*, *i*, etc. The most common two-letter combinations, called **digrams**, are *th*, *in*, *er*, *re*, and so on.
- Also, sender and receiver must both be in possession of the shared secret key.

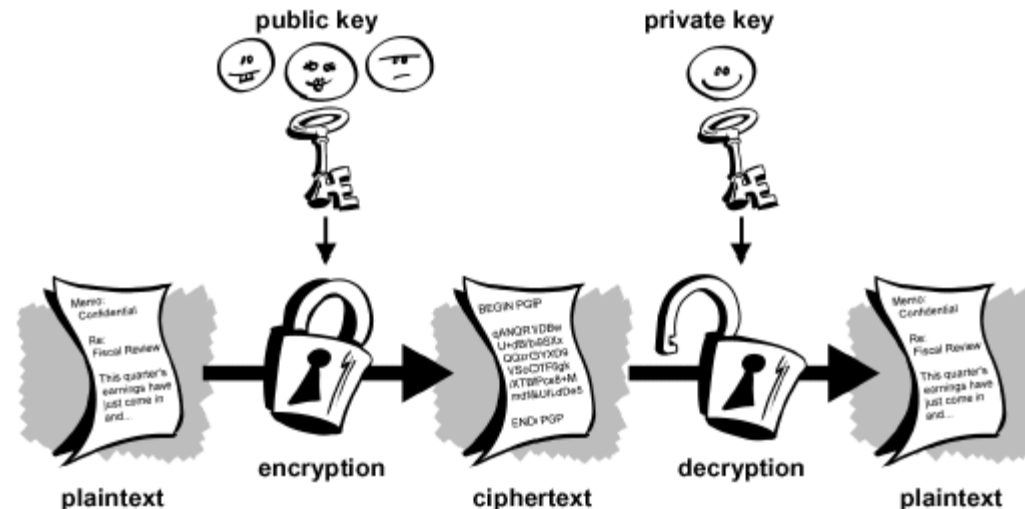


# Public-Key Cryptography

- In this system, distinct keys are used for encryption and decryption.
- Given a well-chosen encryption key, it is virtually impossible to discover the corresponding decryption key.
- Encryption makes use of an "easy" operation, such as how much is  $314159265358979 \times 314159265358979$ ?
- Decryption without the key requires you to perform a hard operation, such as what is the square root of  $3912571506419387090594828508241$ ?
- The main problem with public-key cryptography is that it is a thousand times slower than secret-key cryptography.
- Public-Key Cryptography - [https://www.youtube.com/watch?v=GSIDS\\_lvRv4](https://www.youtube.com/watch?v=GSIDS_lvRv4)
- Instant Messaging and the Signal Protocol - <https://www.youtube.com/watch?v=DXv1boalsDI>

# Public-Key Cryptography

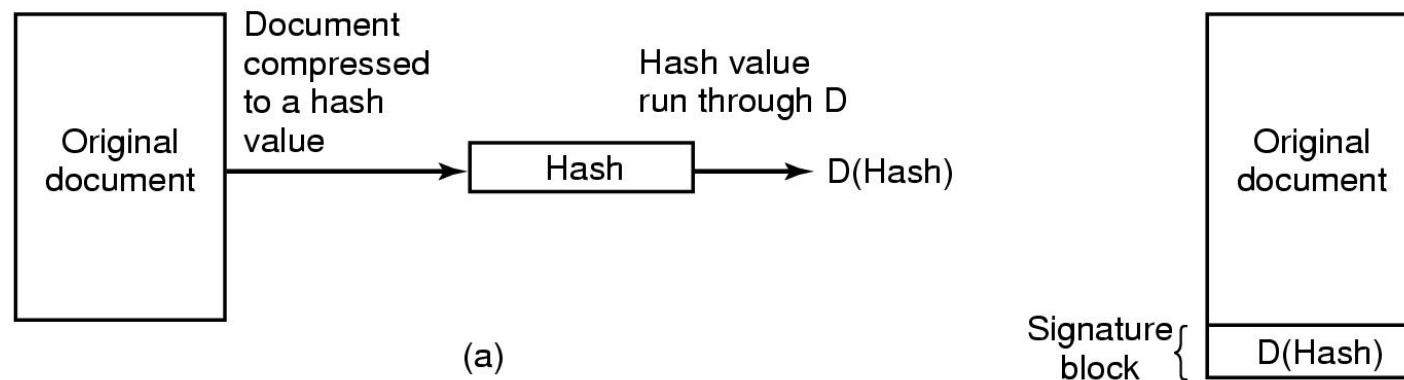
- The way public-key cryptography works is that everyone picks a (public key, private key) pair and publishes the public key.
- The public key is the encryption key; the private key is the decryption key.
- To send a secret message to a user, a correspondent encrypts the message with the receiver's public key.
- Since only the receiver has the private key, only the receiver can decrypt the message.



# Digital Signatures

<https://www.youtube.com/watch?v=704dudhA7UI>

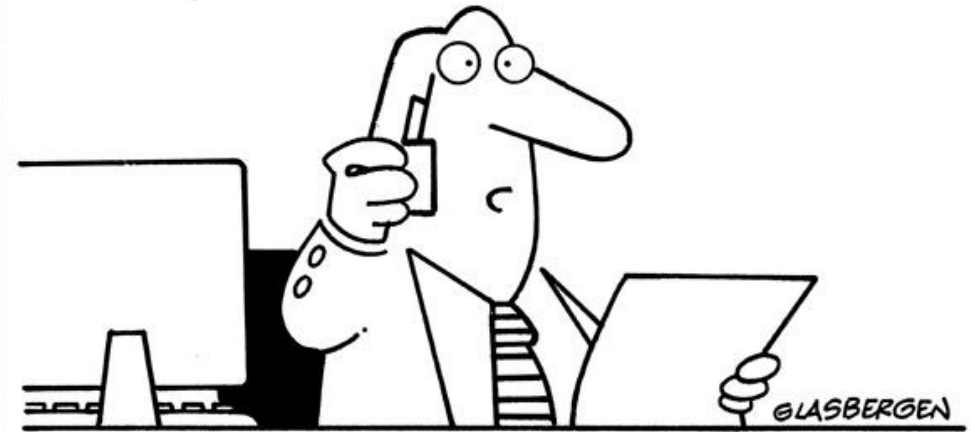
- Digital signatures make it possible to sign emails and other digital documents in such a way that they cannot be repudiated by the sender later.
- One common way is to first run the document through a one-way cryptographic hashing algorithm that is very hard to invert.
- The hashing function typically produces a fixed-length result independent of the original document size.
- The most popular hashing functions used is **SHA-1 (Secure Hash Algorithm)**, which produces a 20-byte result (NIST, 1995).



# Fundamental Dilemma of Security

- *“Security unaware users have specific security requirements but no security expertise.”*
  - D. Gollmann
- Solution: level of security is given in predefined classes specified in some common criteria
  - Orange book (Trusted Computer System Evaluation Criteria) is such a criteria

© Randy Glasbergen  
www.glasbergen.com



**“I sent my bank details and Social Security number in an e-mail, but I put ‘PRIVATE FINANCIAL INFO’ in the subject line so it should be safe.”**

# Fundamental Tradeoff

- Between security and ease-of-use
  - Security may require clumsy and inconvenient restrictions on users and processes
- *“If security is an add-on that people have to do something special to get, then most of the time they will not get it”*
  - Martin Hellman, co-inventor of Public Key Cryptography

Copyright 2004 by Randy Glasbergen.  
www.glasbergen.com



**“The boss is worried about information security,  
so he sends his messages one alphabet letter  
at a time in random sequence.”**



" IT'S A FINE LINE BETWEEN  
SECURITY AND PARANOIA. "