

# COMP3052.SEC Computer Security

## Session 01: Introduction to COMP3052.SEC



# ACKNOWLEDGEMENTS

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Toweys, ...

# OVERVIEW

- Convenor & Teacher Information
- Module Information
- Assessment
- Motivation for the Module
- Module Contents
- Textbook and Additional References
- Summary

# TEACHING TEAM INFO

## Convenors Information:

- Name: Dr. Alejandro Guerra Manzanares
- E-mail: [Alejandro.Guerra@nottingham.edu.cn](mailto:Alejandro.Guerra@nottingham.edu.cn)
- Room: PMB435
- Office Hours: TBA
- Name: Dr. Wooi Ping Cheah
- E-mail: [Wooi-Ping.Cheah@nottingham.edu.cn](mailto:Wooi-Ping.Cheah@nottingham.edu.cn)
- Room: PMB323
- Office Hours: **Wed 1pm-3pm**

## Technician Information:

- Name: Ms. Jane Zhao
- E-mail: [Jane.Zhao@nottingham.edu.cn](mailto:Jane.Zhao@nottingham.edu.cn)

## Teaching Assistant:

- Name: Mr. Leshan Tan
- E-mail: [Leshan.Tan@nottingham.edu.cn](mailto:Leshan.Tan@nottingham.edu.cn)

# MODULE INFORMATION

- Class Sessions
  - Classes and labs, ... and maybe some other stuff, too
  - We'll frontload a bit
  - Wednesdays, 11am-1pm, IAMET-326 (Weeks 1-3)
  - Thursdays, 1pm-3pm, IAMET-326
  - Fridays, 4pm-6pm, IAMET-406
- Labs (and Coursework)
  - ~5 main labs
    - ... more details soon!
  - All materials on Moodle module page

# TENTATIVE SCHEDULE

(Version: 2025 Spring)

Week	Week Commencing	Wednesday (11am-1pm) (IAMET-326)	Thursday (1pm-3pm) (IAMET-326)	Friday (4pm-6pm) (IAMET-406)
01 (23)	17-Feb	Introductions	Motivating Examples	Foundations
02 (24)	24-Feb	Authentications	Access Control	Lab 1: Intro to Kali
03 (25)	3-Mar	Firewalls	Reference Monitor	Lab 2: Passwords
04 (26)	10-Mar		Network Security Internet Security	Internet Security Unix/Linux Security
05 (27)	17-Mar		Windows Security	Lab 3: Firewalls
06 (28)	24-Mar		Intrusion Detection	Lab 4: Packet Sniffing
07 (29)	31-Mar		Software Vulnerabilities	Public Holiday
08 (30)	7-Apr	Data Security (1pm-3pm) (IAMET-406)	Crypto I	Lab 5: Attack & Defend
09 (31)	14-Apr		Crypto II & III	Lab Revision
10 (32)	21-Apr	Revision / Q&A (11am-1pm) (IAMET-326) Crypto IV & V (3pm-5pm) (IAMET-406)	Metamorphic Security	Revision / Q&A
11 (33)	28-Apr		Public Holiday	Public Holiday
12 (34)	5-May		No Teaching	No Teaching

# ASSESSMENT

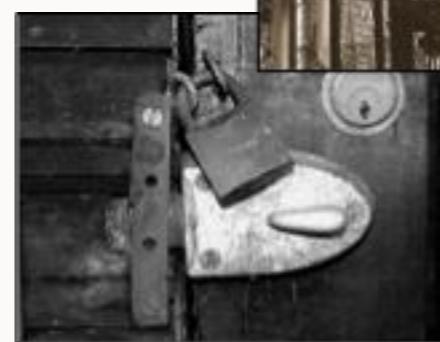
- 1 hour written examination **60 %**
- Coursework **40 %**
  - More details later ... but it will almost certainly be based on your experiences and reflections on the series of lab activities

# ACTIVITY ...

- Come up with definitions for “security,” and “computer security,” and “security engineer”
- Why are we, as humans, concerned with these issues?

# MOTIVATION

- People have protected their property and privacy for generations  
(Locks, Fences, Signatures, Seals, etc...)
- Big change
- Everything becoming electronic
- And security?
- What about the future?



# ACTIVITY ...

- List some points about what you *expect* to learn from this module, and what you *hope* to learn.
- Why?

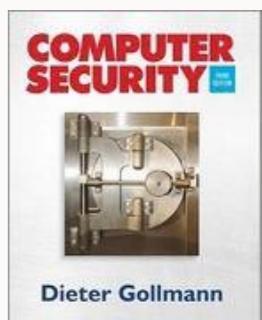
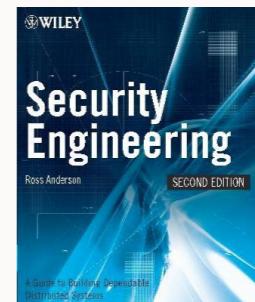
# LEARNING OUTCOMES

- What is computer/information security ?
- Why is it so important ?
- How can we evaluate and measure it ?
- How can we enforce it ?
- How can we minimise its risks ?
- The bad guy's point of view
- The victim's point of view

# RESOURCES

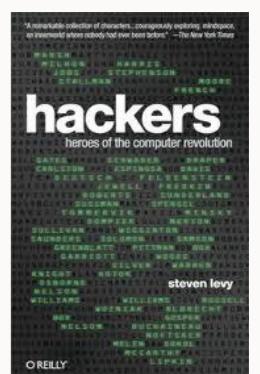
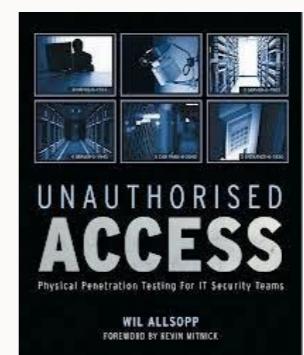
- Core text:

- Security Engineering – Ross Anderson 2<sup>nd</sup>/3<sup>rd</sup> edition (some available online, for free)
- Computer Security – Dieter Gollmann 3d edition (Amazon)



- Additional Reading:

- Secrets & Lies – Bruce Schneier
  - Unauthorised Access - Will Allsopp
  - Hackers - Steven Levy
- Module materials on Moodle



# INTRODUCTION TO SECURITY



# OUTLINE

- On Security
- Attacks and Attackers
- Security Management
  - Security Policies
  - Measuring Security
  - Standards
- Risk and Threat Analysis
  - Assets
  - Vulnerabilities
  - Threats
  - Risks
  - Countermeasures

# SECURE SYSTEMS

- A secure system is one which does not exist...

*An almost secure system is one which is locked up in a nuclear bunker within an air locked titanium safe and disconnected from anything else in the world.....and even such a system is not 100% secure!*

- It is not about achieving complete security
- It is about minimising risk to systems
- Both from a technical, and social, point of view

# WIKILEAKS SERVER BUNKER



- [http://www.youtube.com/watch?feature=player\\_embedded&v=wn8pz1HLYp8](http://www.youtube.com/watch?feature=player_embedded&v=wn8pz1HLYp8)

# ON SECURITY

- Original focus on systems with **single, or few users**
- Today focus on **ubiquitous end systems**
- Systems interconnected by **networks**
- Danger of possible attacks from '**un-trustworthy**' nodes
- Both **remotely** as well as **locally** (insiders)
- Primarily a **management** issue!

# ACTIVITY ...

- Based on your own impressions or knowledge, who are the “*attackers*”?
- What are the “*attacks*”?

# ATTACKS AND ATTACKERS

- Landscape is changing
- Hackers -> Organised crime
- Website defacement -> Personal data harvesting
- Peer appreciation -> Earning money
- Viruses -> Trojans and Denial-of-Service attacks
- Complexity of our systems is increasing
- Our understanding of the system's intricacies can't keep up

# SECURITY

- Reliability – Accidental failures
  - Usability – Operating mistakes
  - Security – Intentional failures
- 
1. ‘Security is a people problem’
  2. Legal system defines boundaries of acceptable behaviour
  3. Management responsible for security

# SECURITY MANAGEMENT

- Management **responsible** for assets
- Security measures must have clear full **support** of senior management
- Security **awareness** programs
- **User** is not (usually) the enemy!
- **Developers** need even more awareness!

# SECURITY POLICIES

- State what should be protected
- And how this should be achieved
- Security Policy Objective
- Organisational Security Policy
- Automated Security Policy

# MEASURING SECURITY

- Very difficult
- Measures only exist for some aspects of security
  
- Product Security
- System Security
- Cost of an Attack
- Cost of Assets



# RISK AND THREAT ANALYSIS

- Risk Analysis
  - All information assets
  - IT infrastructure
  - Perform during development
- Risk – **Possibility** of an incident or attack to cause **damage** to your enterprise
- Risk = **Assets \* Vulnerabilities \* Threat**



# ACTIVITY ...

- What are assets, vulnerabilities, and threats?
  - Come up with definitions, and list some examples

# ASSETS

- Software
- Hardware
- Data and Information
- Reputation
- Identification easy, valuation difficult
- Data, Information, Reputation – difficult to measure

# VULNERABILITIES

- Weaknesses of a system that could be accidentally or intentionally exploited to damage assets
- Badly configured accounts
- Programs with known flaws
- Weak access control
- Weak firewall configuration
- Can be rated according to impact

# THREATS

- Actions by adversaries who try to exploit vulnerabilities to damage assets
- Categorisation by damage done to assets
- Identification of source of attacks
- Analysis of attack execution (Attack Graphs)
- Can be rated according to likelihood
- Attack Graphs
  - formalised and structured
  - assessable, reproducible

# RISK

- Quantitative Risk Analysis
  - + probability theory based on **mathematical** theory
  - - quality of results depends on **quality of inputs**
  - - not always **feasible**
- Qualitative Risk Analysis
  - + more **applicable**
  - - scaling based on **judgements** of security experts

# COUNTERMEASURES

- Risk analysis **generates** recommended countermeasures
- **Up to date/continuous** risk analysis not always possible
- Baseline protection – security requirements for **typical cases** with recommended countermeasures

# SUMMARY

- Current security landscape
- Management is vital to security
- How security can be measured
- What is Risk and how it is analysed

Read Anderson: Chapter 1

# COMP3052.SEC Computer Security

## Session 02: Motivating Example



# Acknowledgements

---

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towey,...

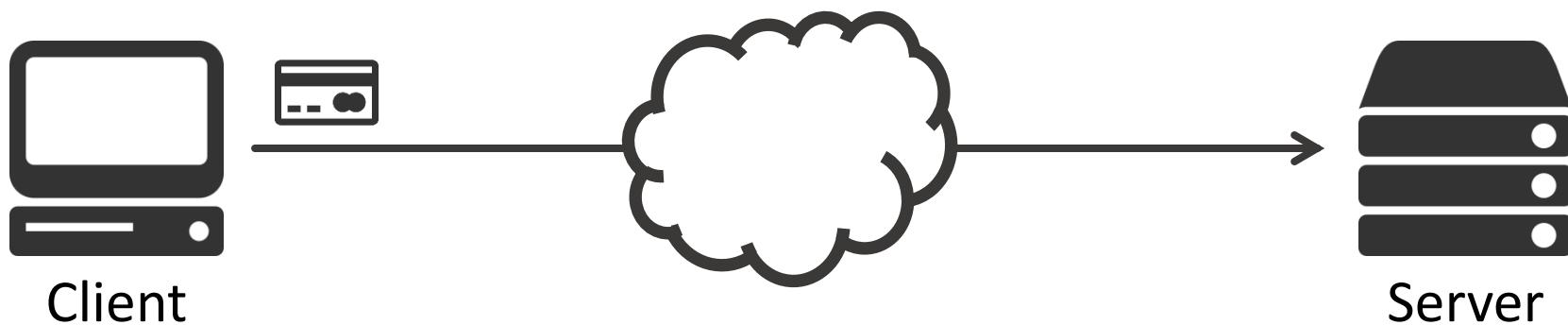
# Who needs security anyway?

---

# Who needs security anyway?

---

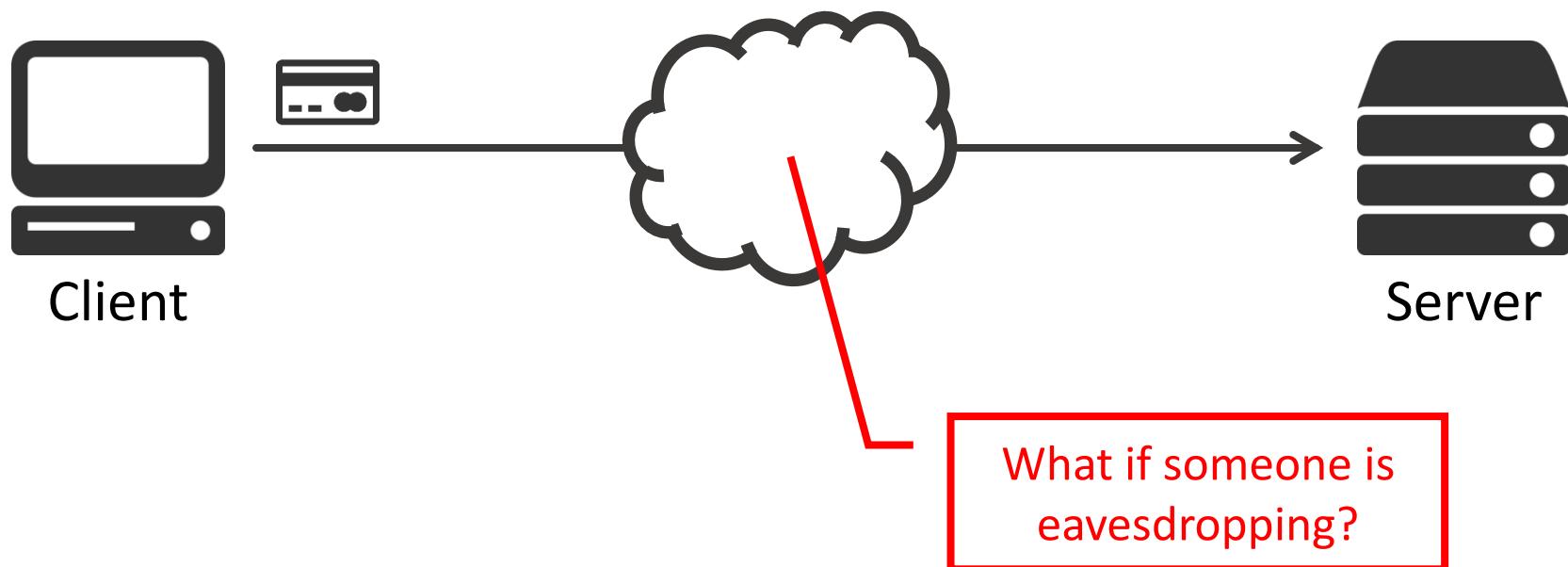
- People buy things online using card details all the time



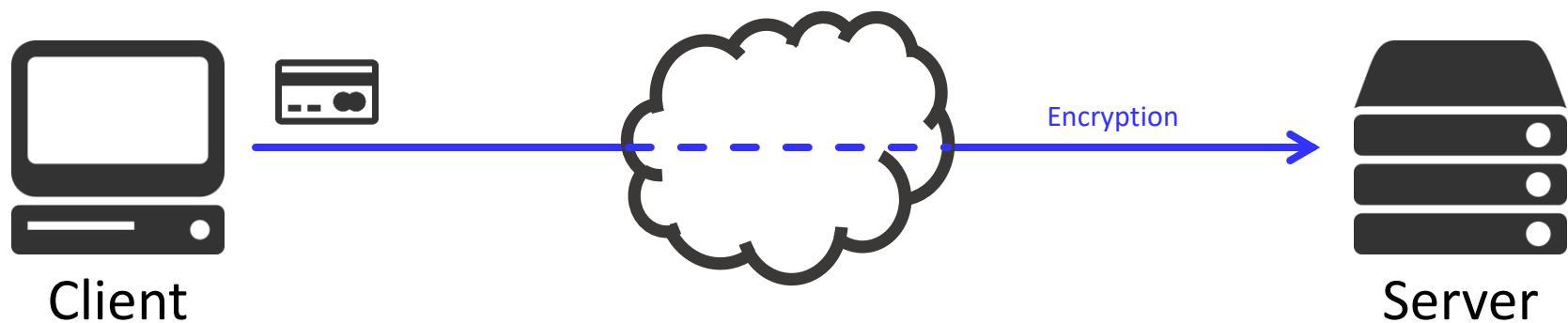
- What are the things that could go wrong here?

# Eavesdropping

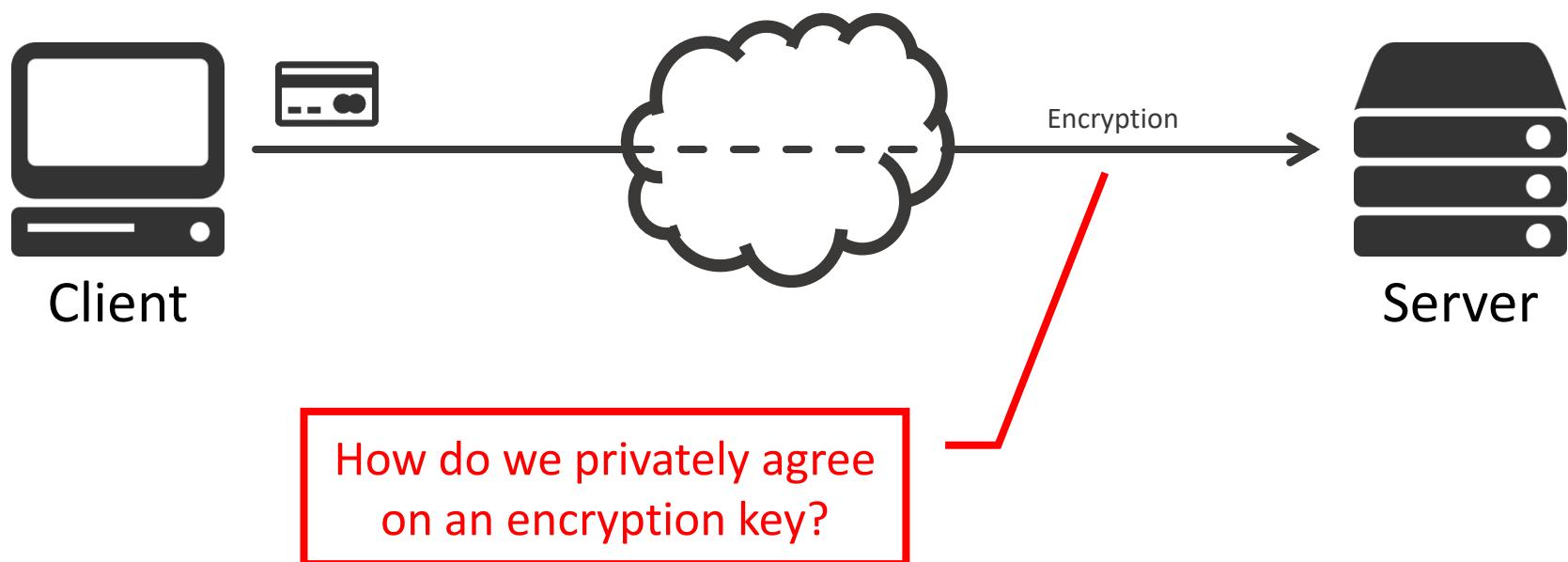
---



# HTTPS / TLS

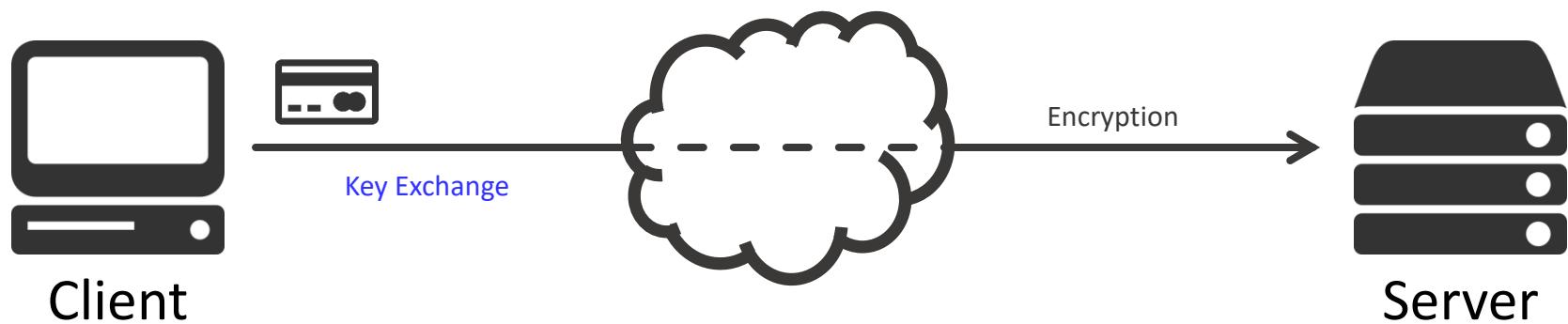


# Shared Secrets

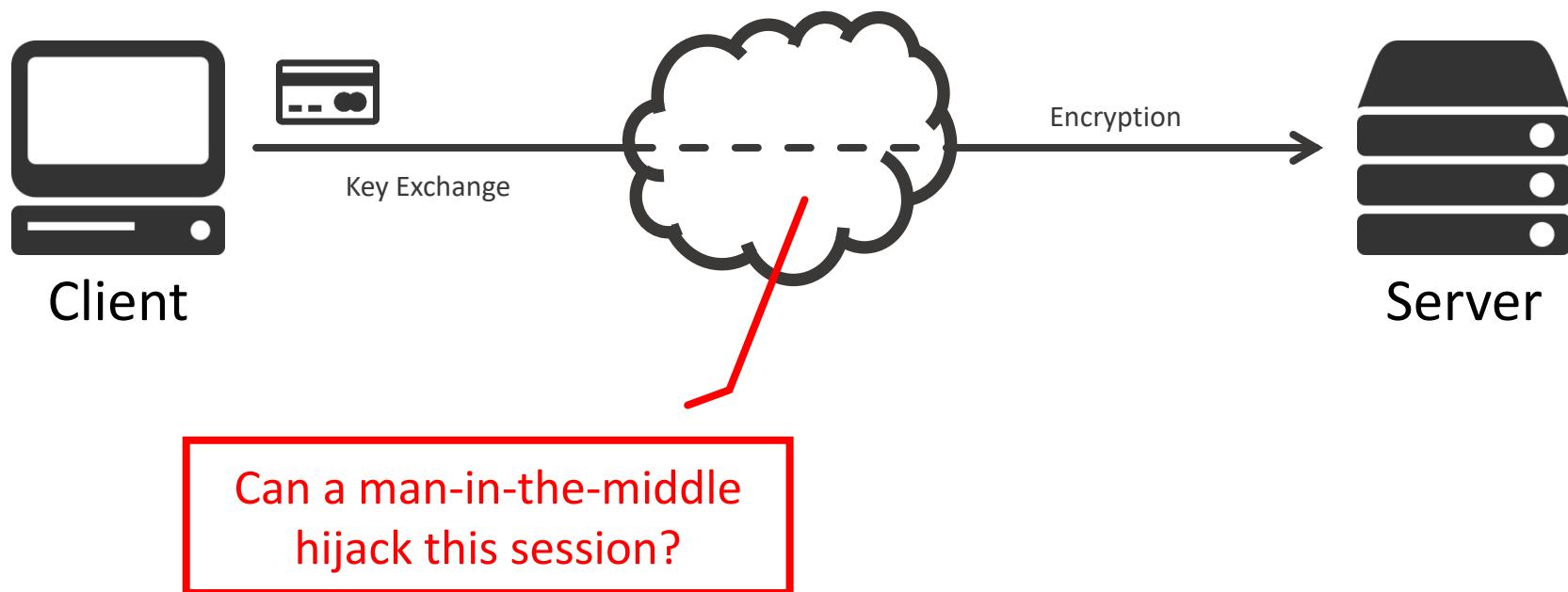


# Shared Secrets

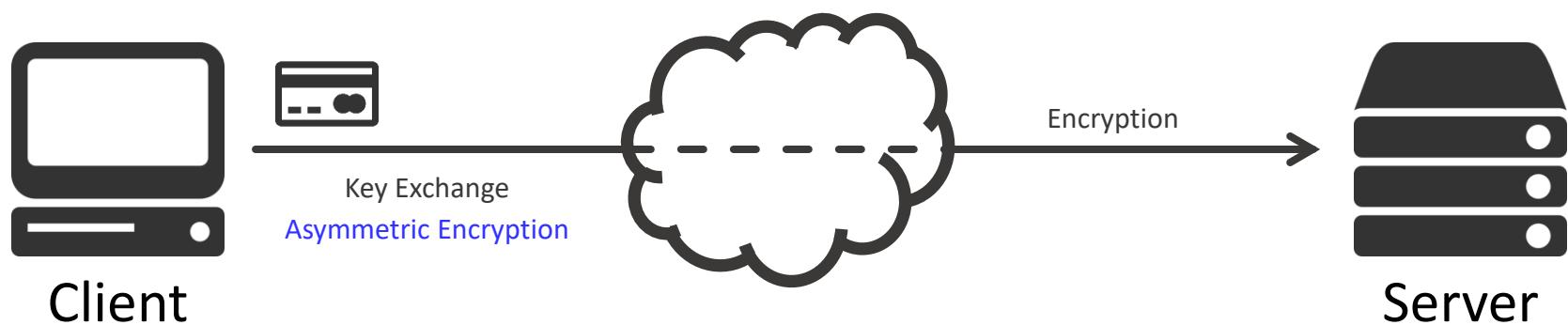
---



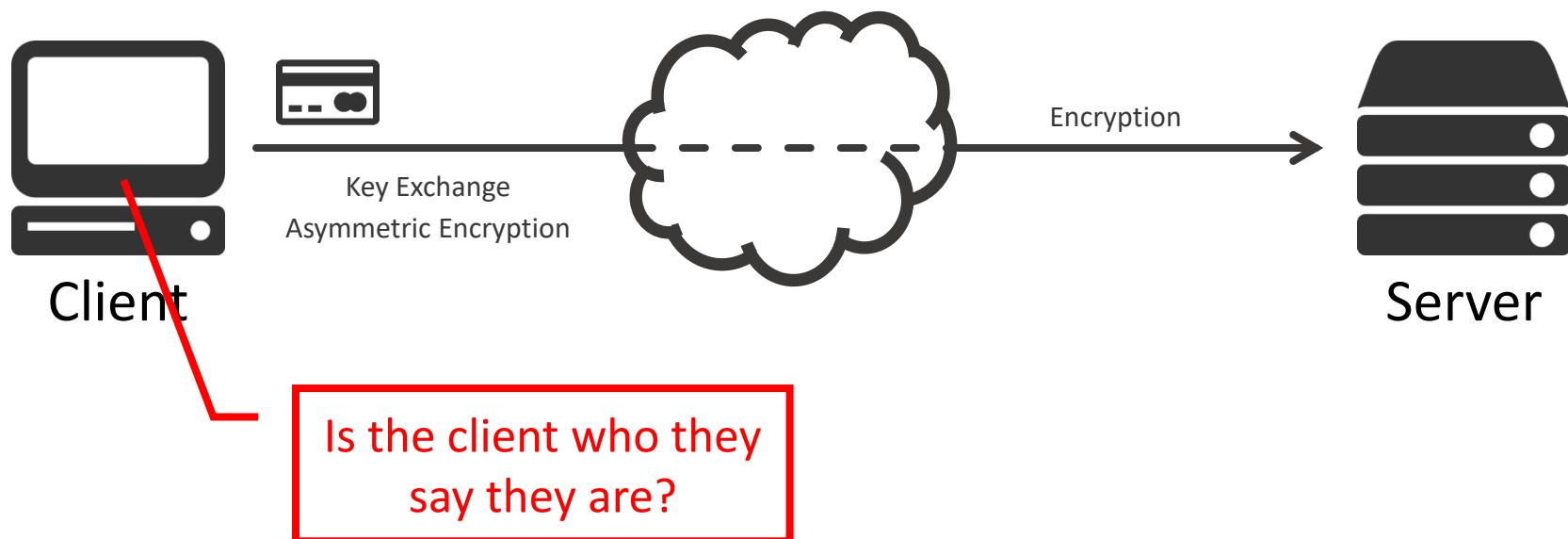
# Attack Vectors



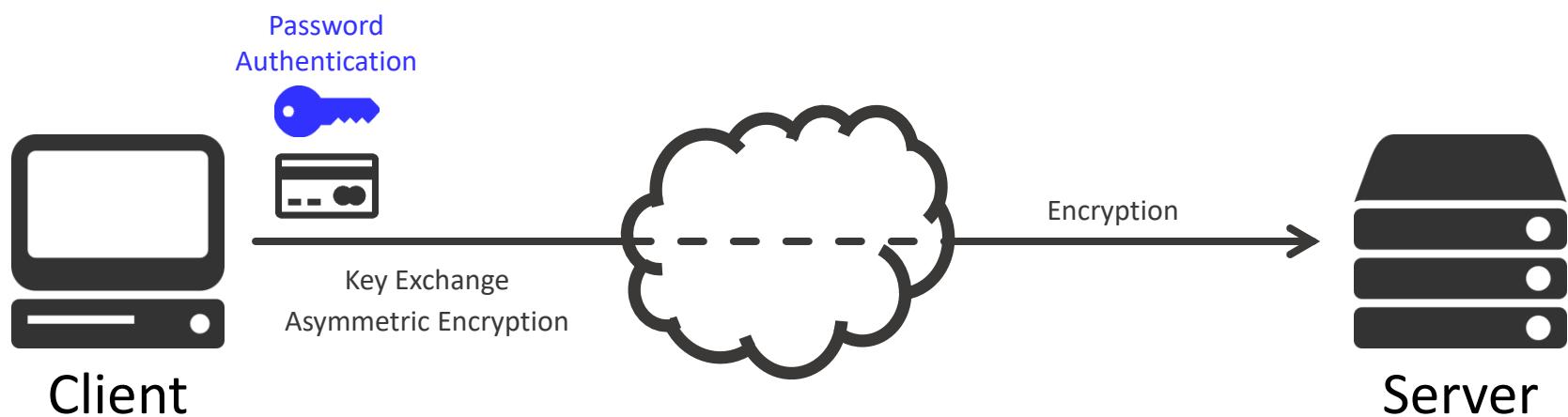
# Public-key Encryption



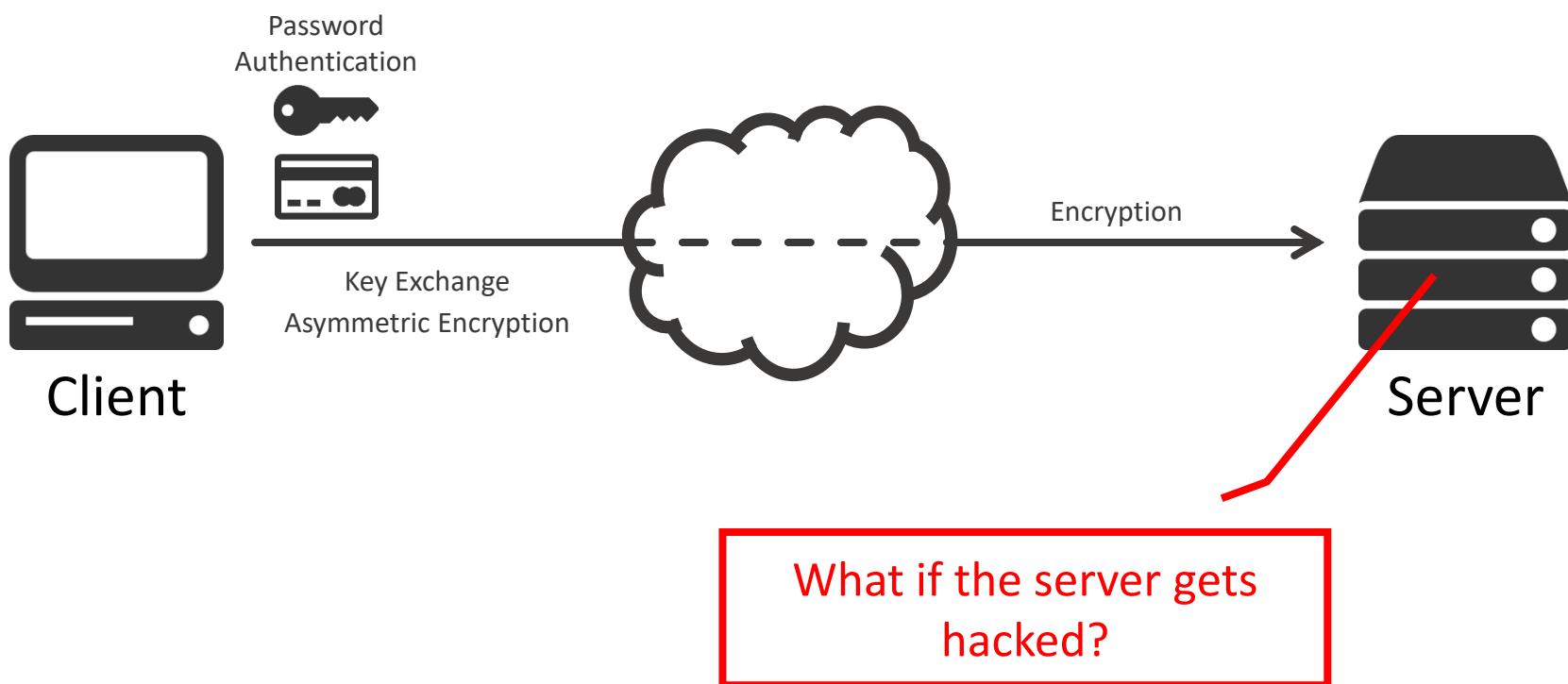
# User Authentication



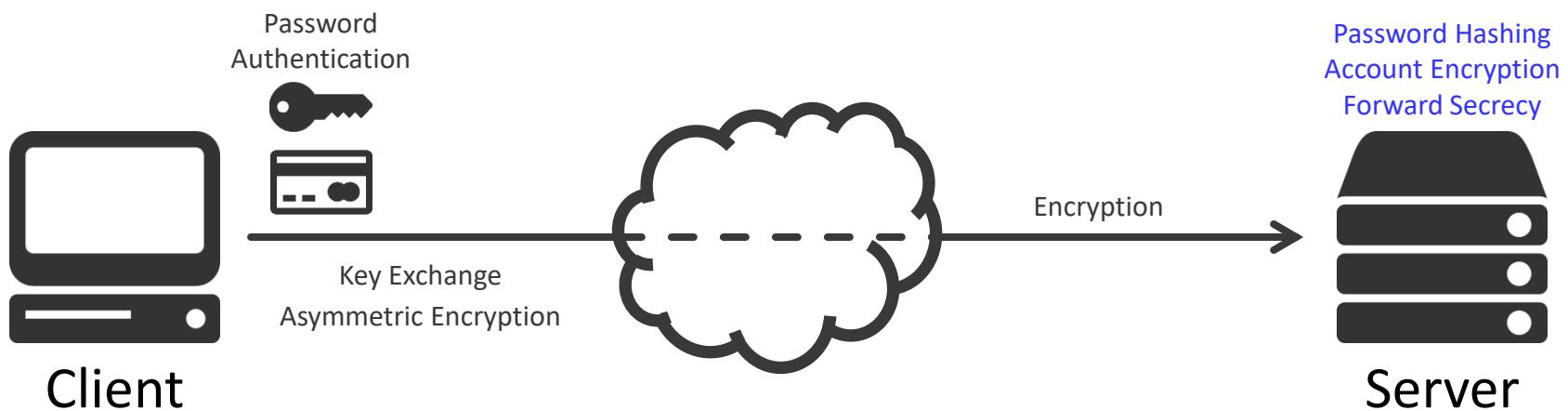
# User Authentication



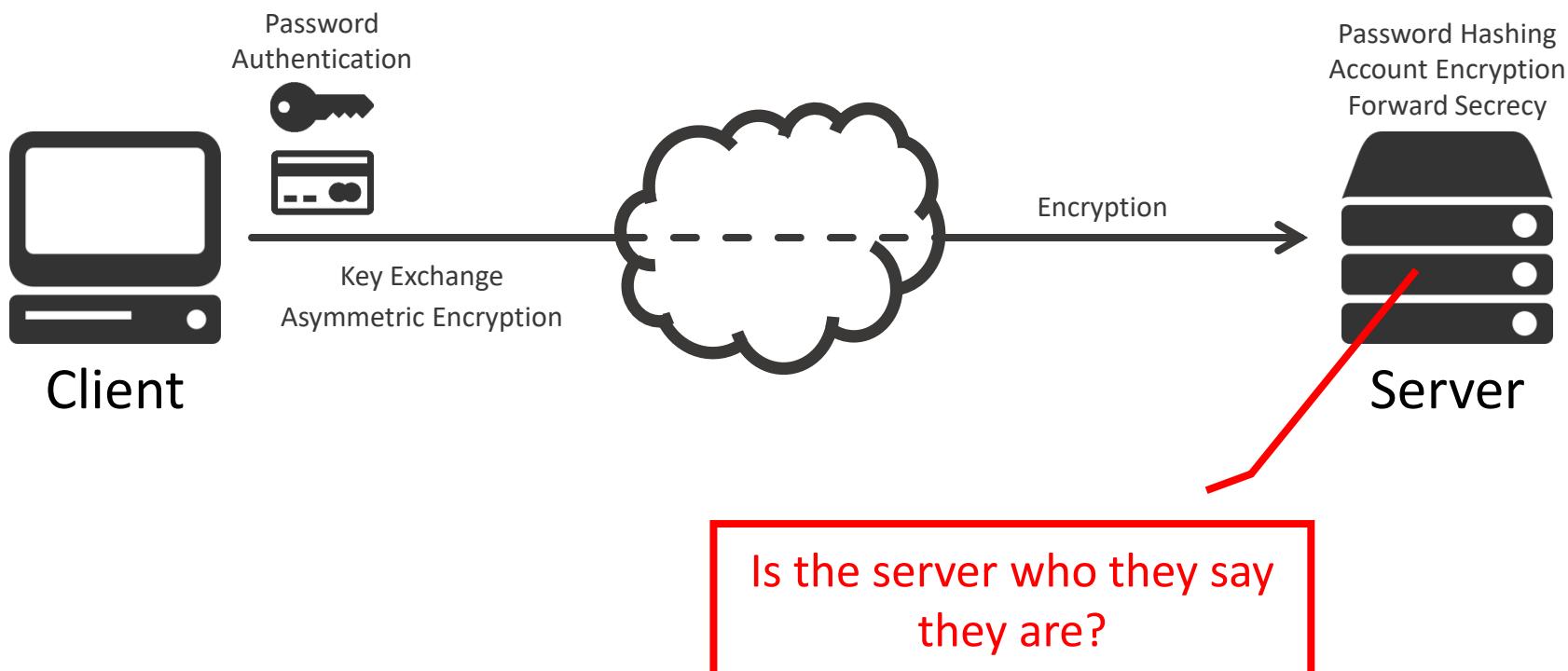
# Confidentiality



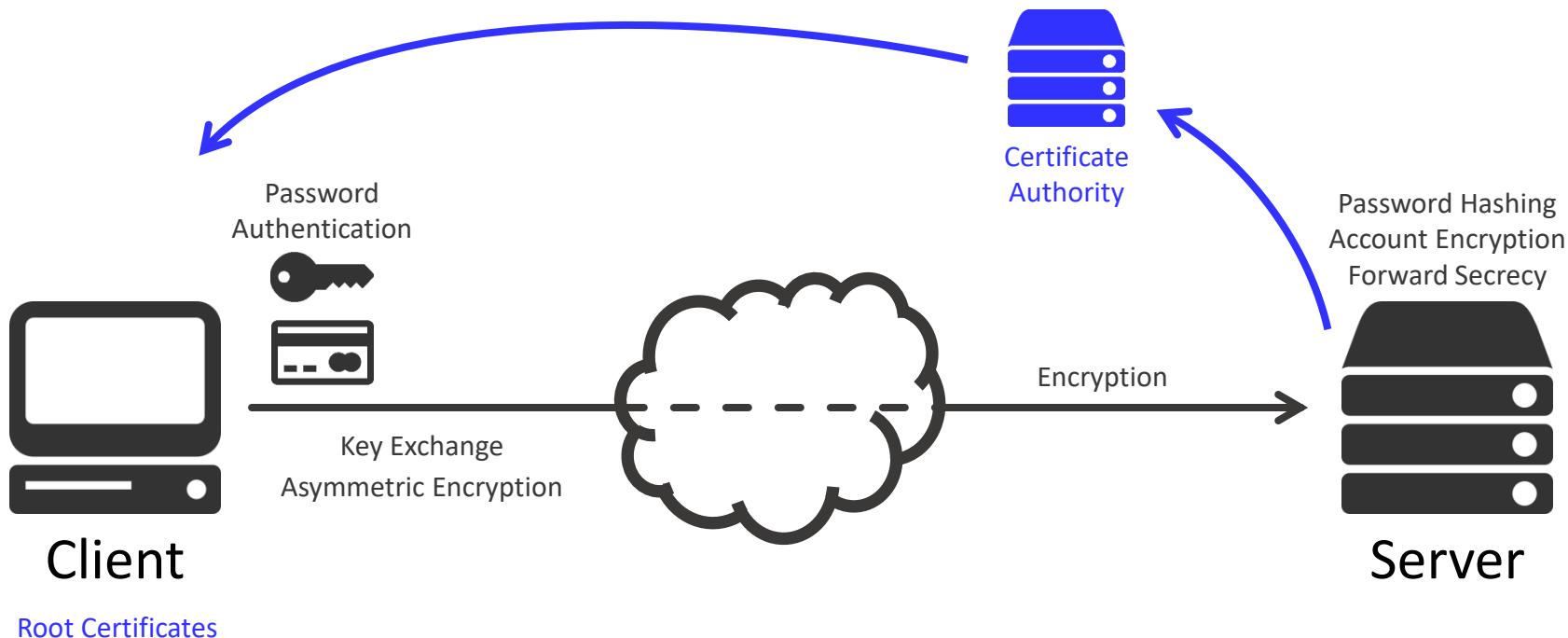
# Hash Functions



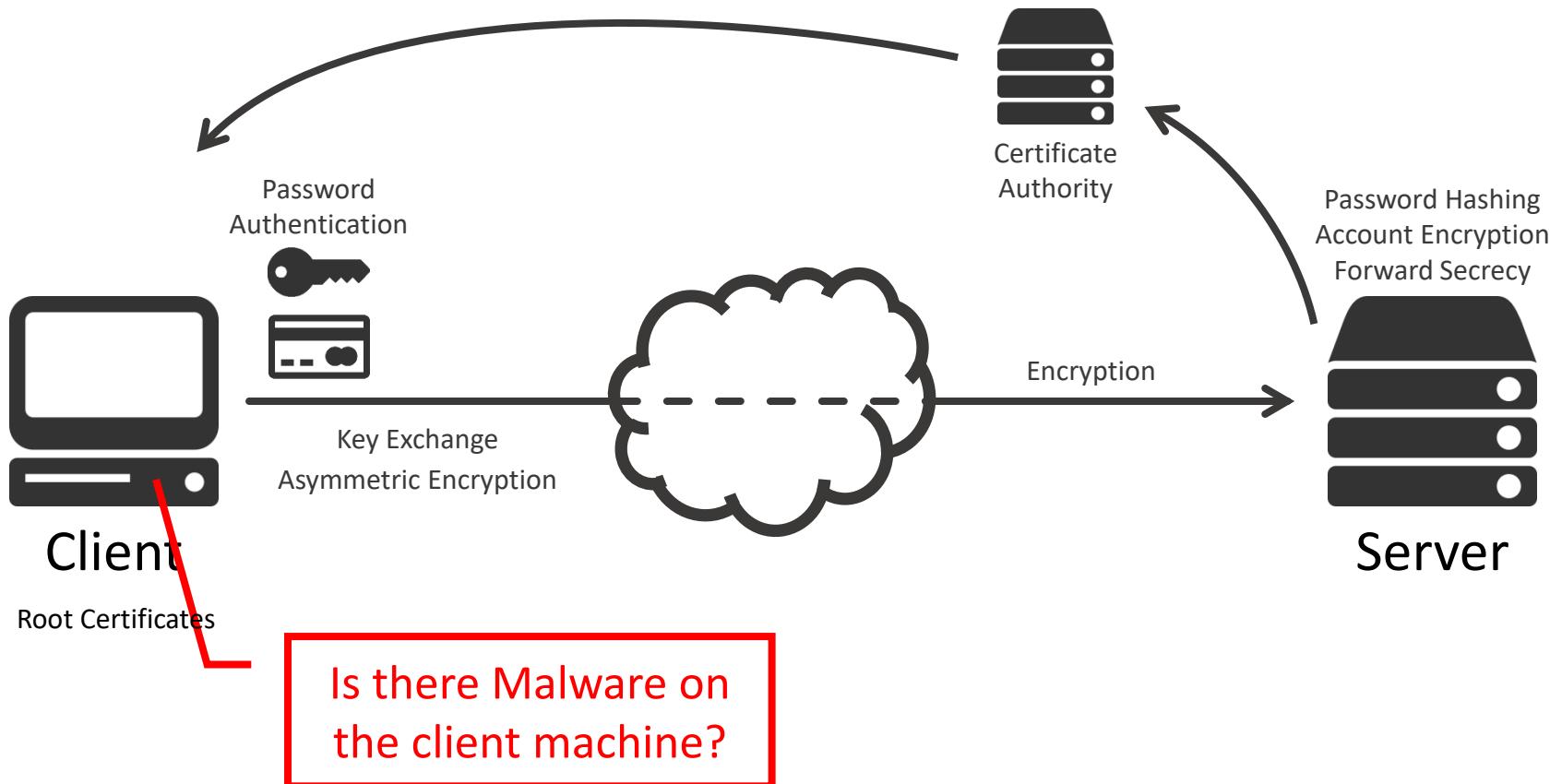
# Server Authentication



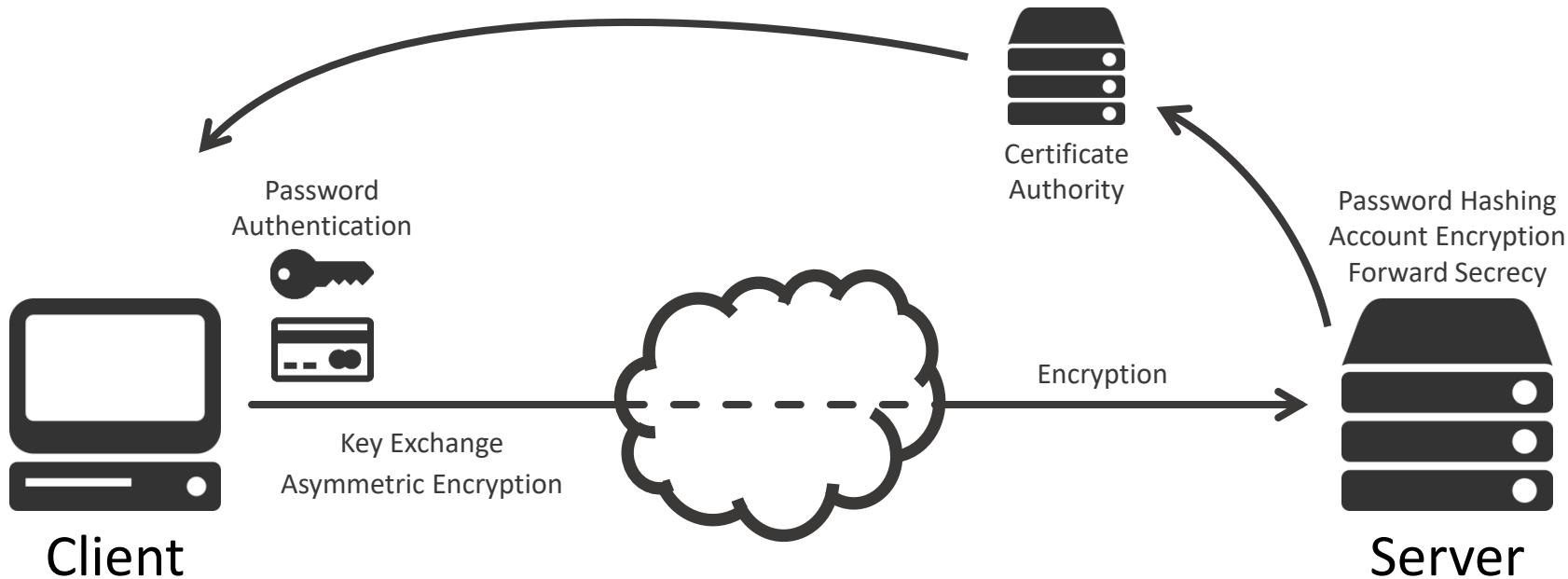
# Digital Certificates



# Malware

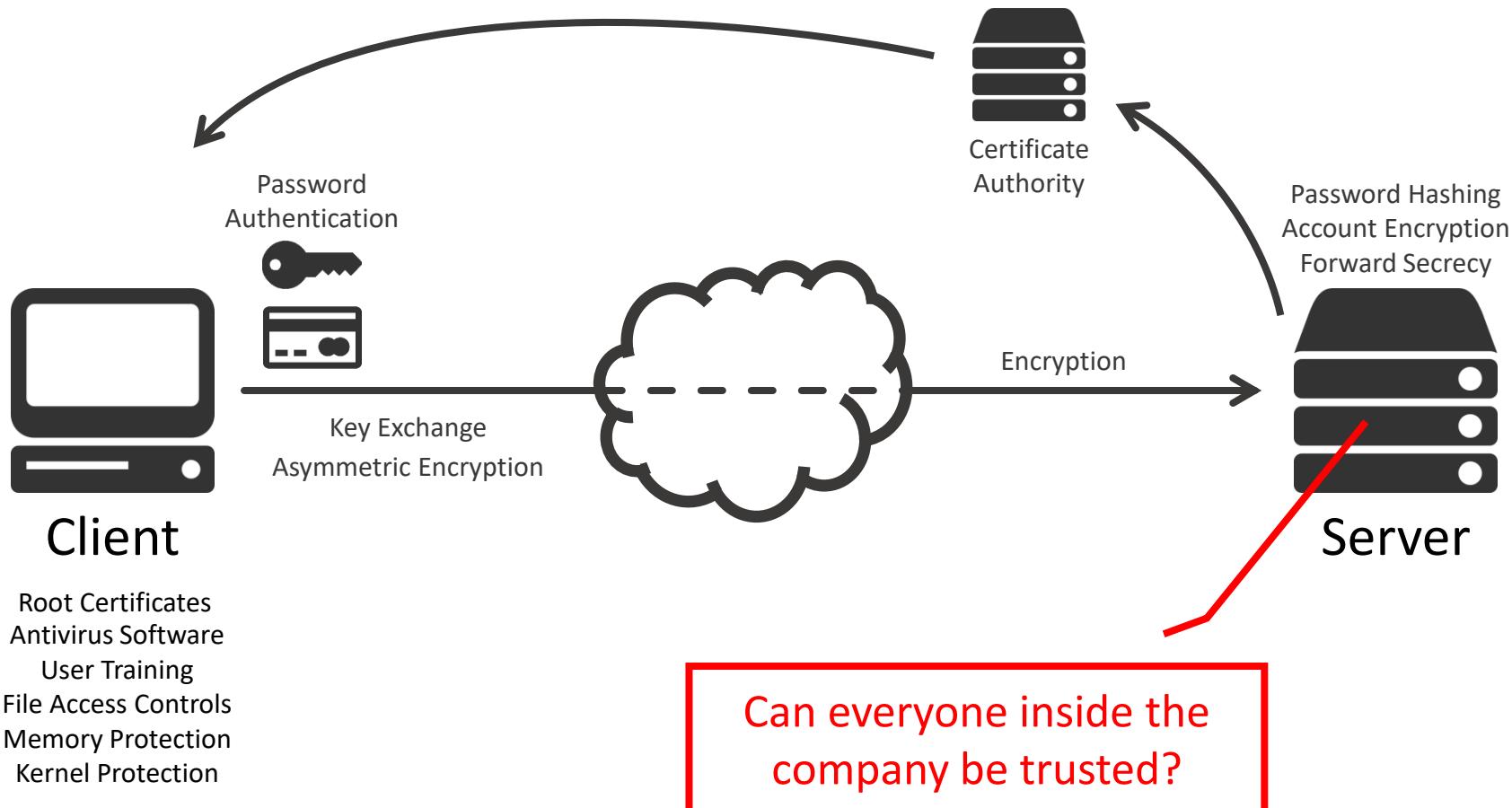


# Protection and Avoidance

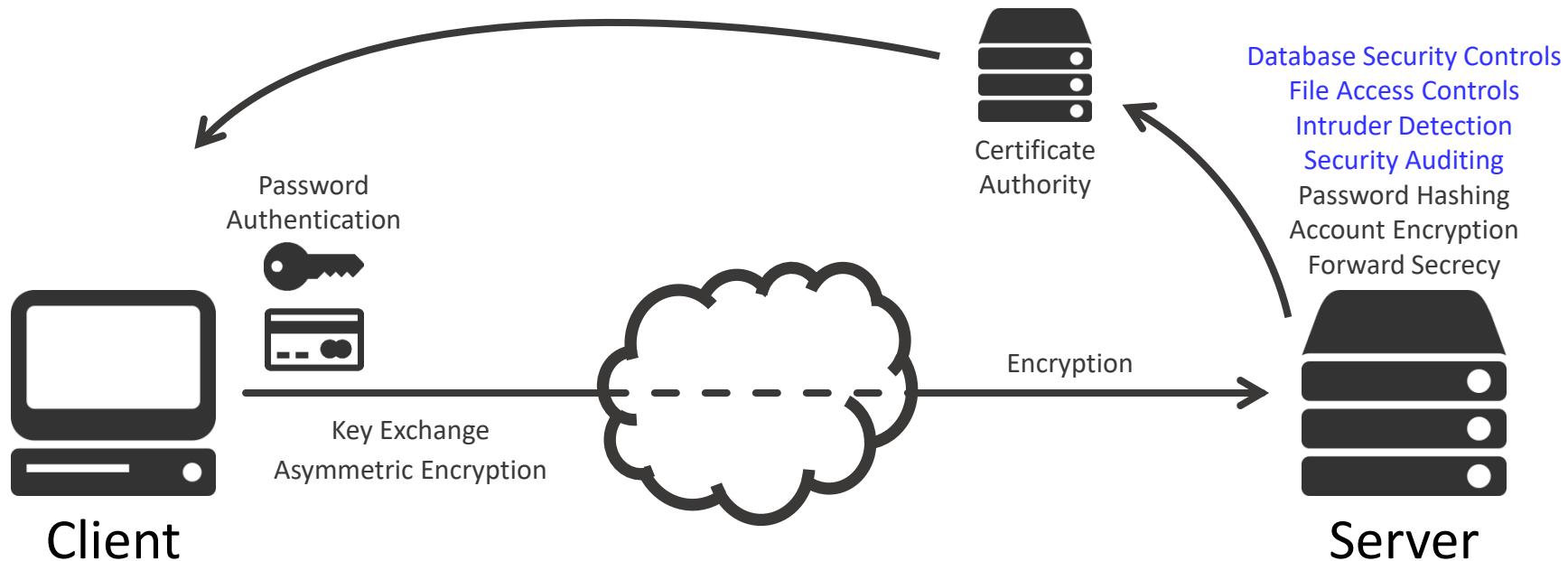


Root Certificates  
Antivirus Software  
User Training  
File Access Controls  
Memory Protection  
Kernel Protection

# Insider Attacks

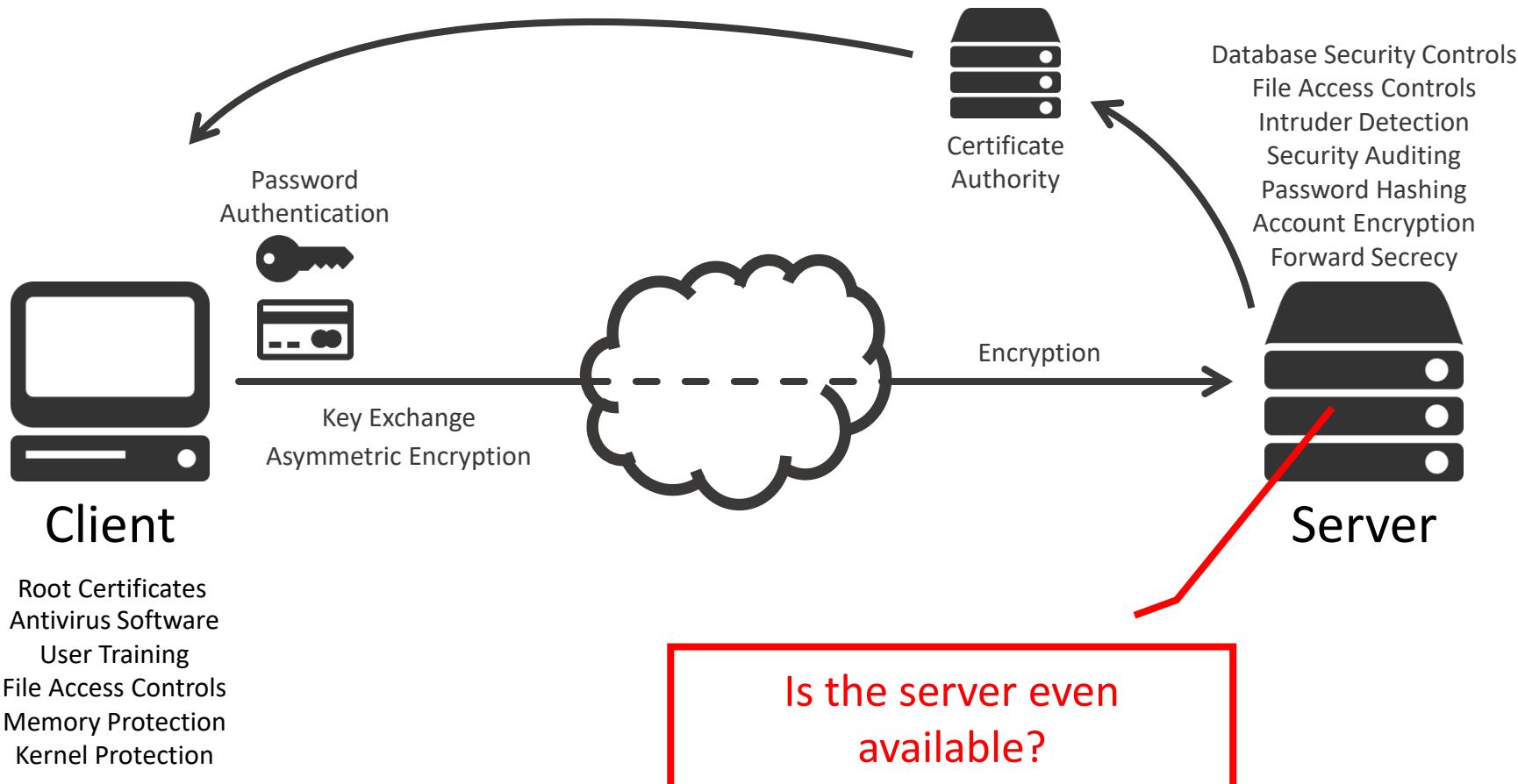


# Access Controls

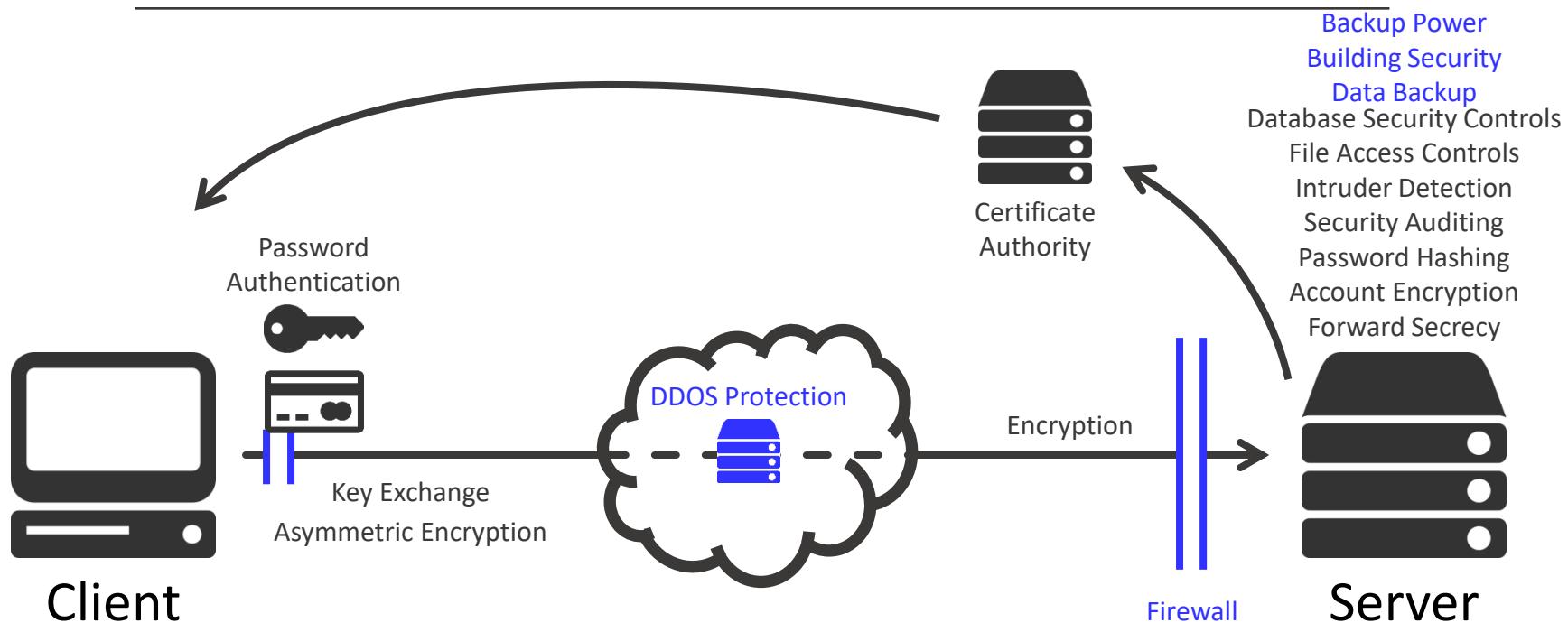


Root Certificates  
Antivirus Software  
User Training  
File Access Controls  
Memory Protection  
Kernel Protection

# Availability

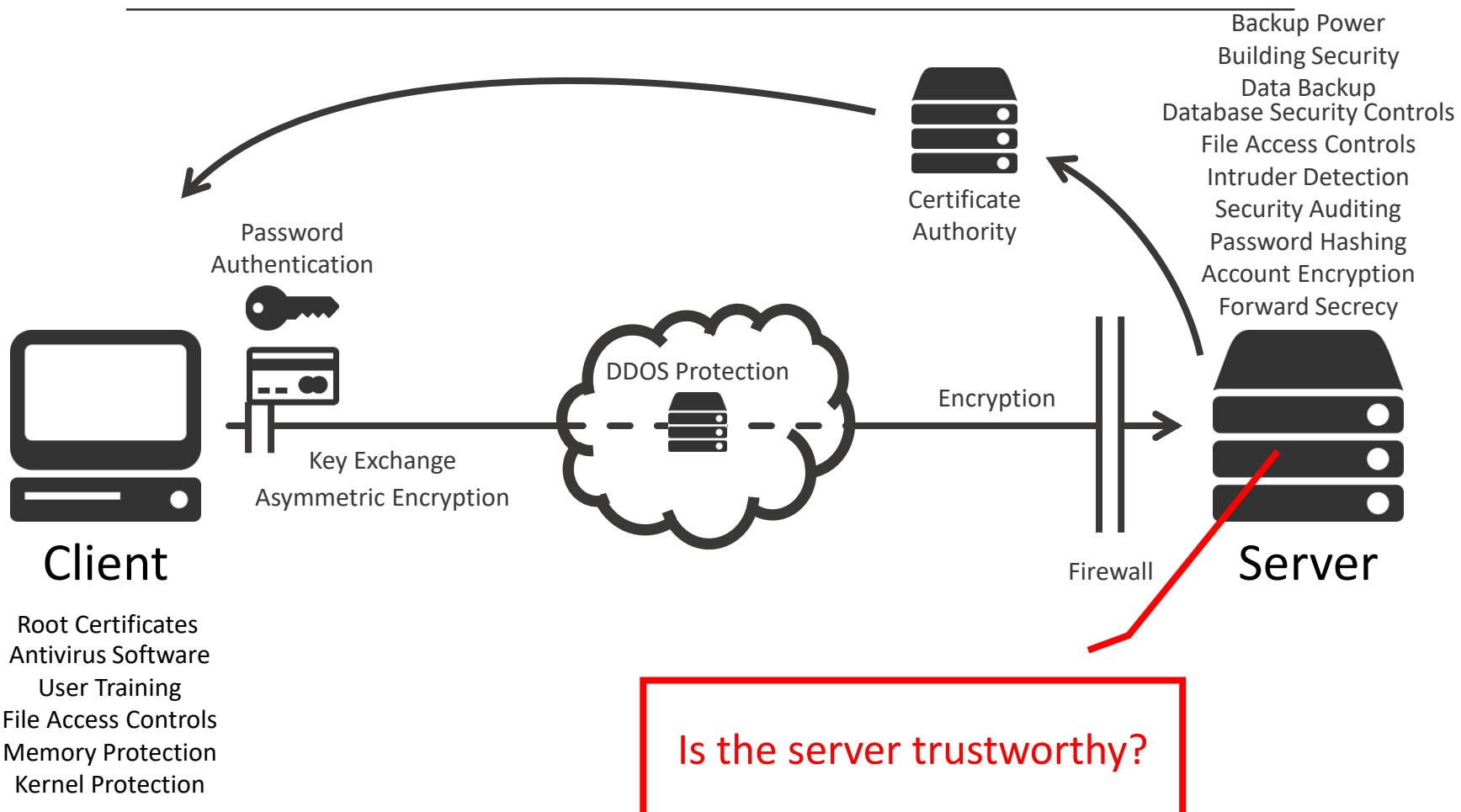


# Denial of Service

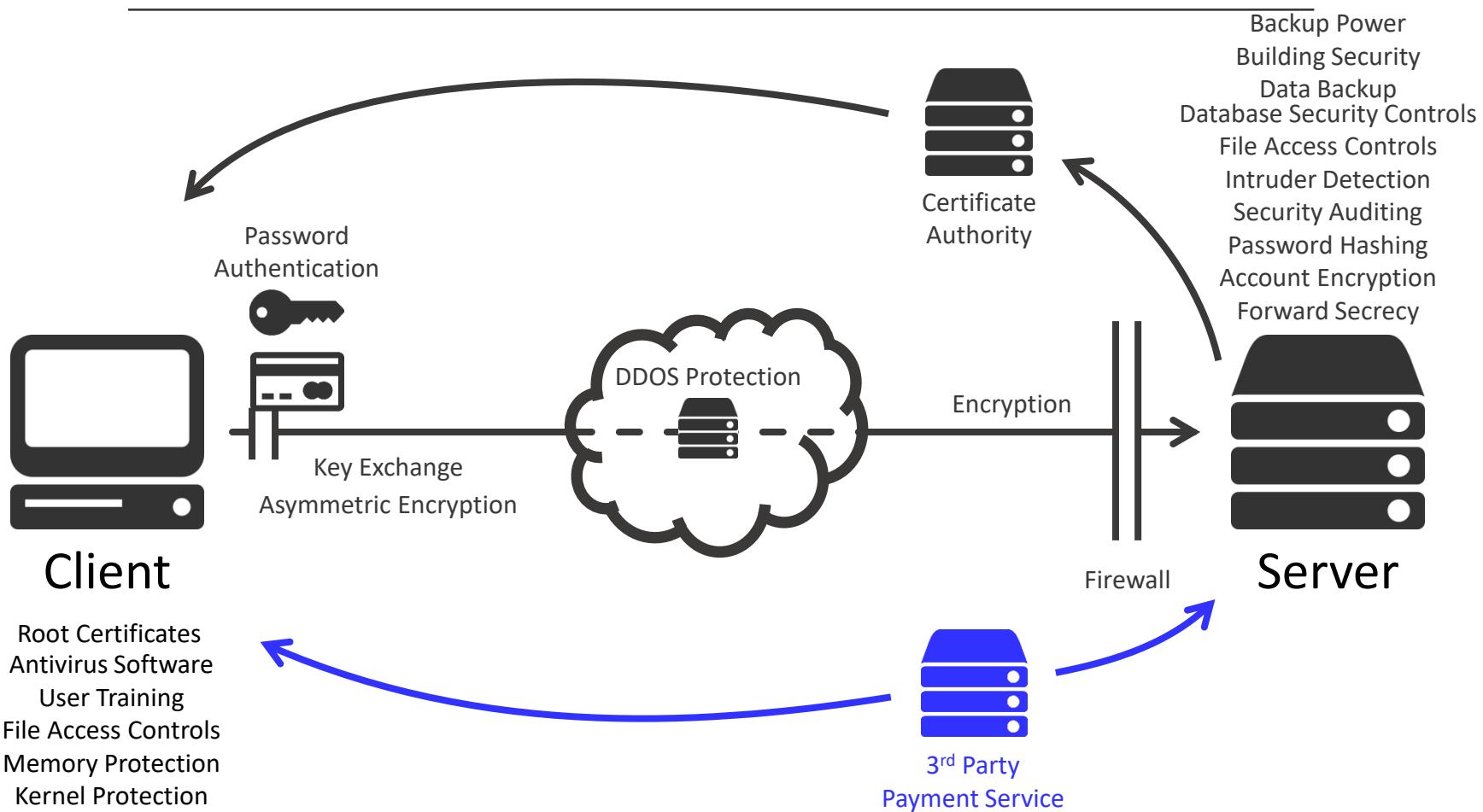


Root Certificates  
Antivirus Software  
User Training  
File Access Controls  
Memory Protection  
Kernel Protection

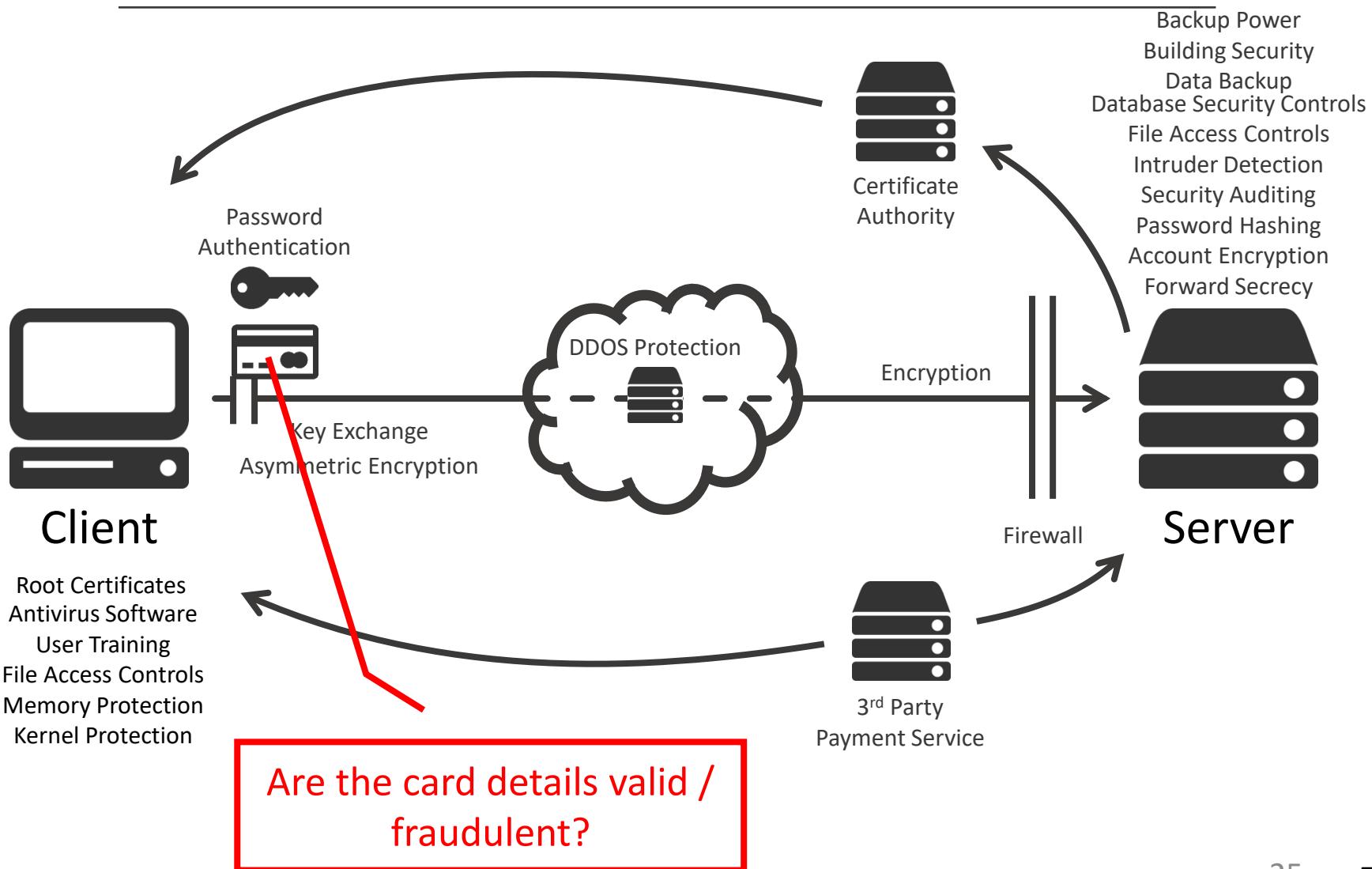
# Trust



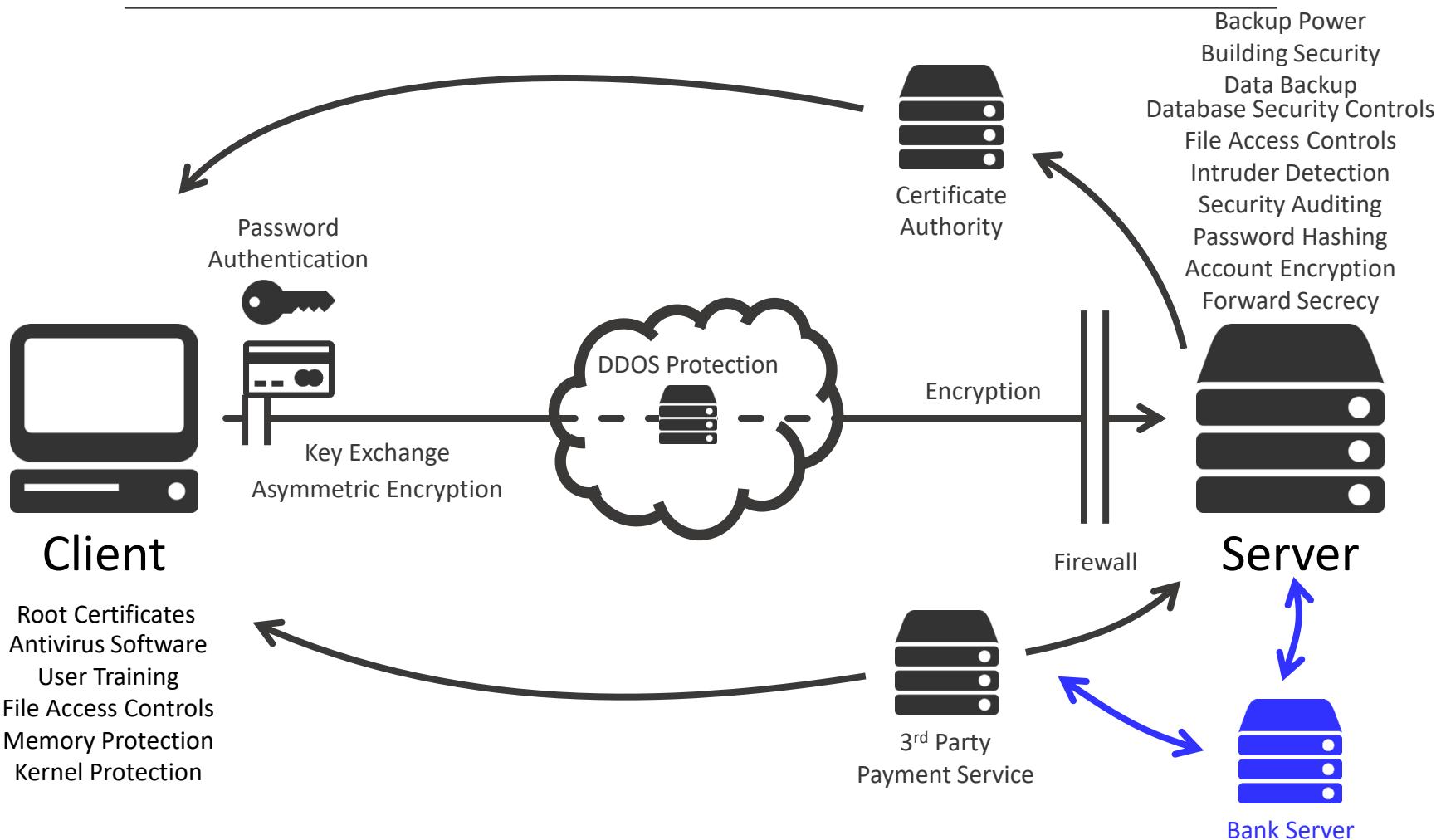
# Third Parties



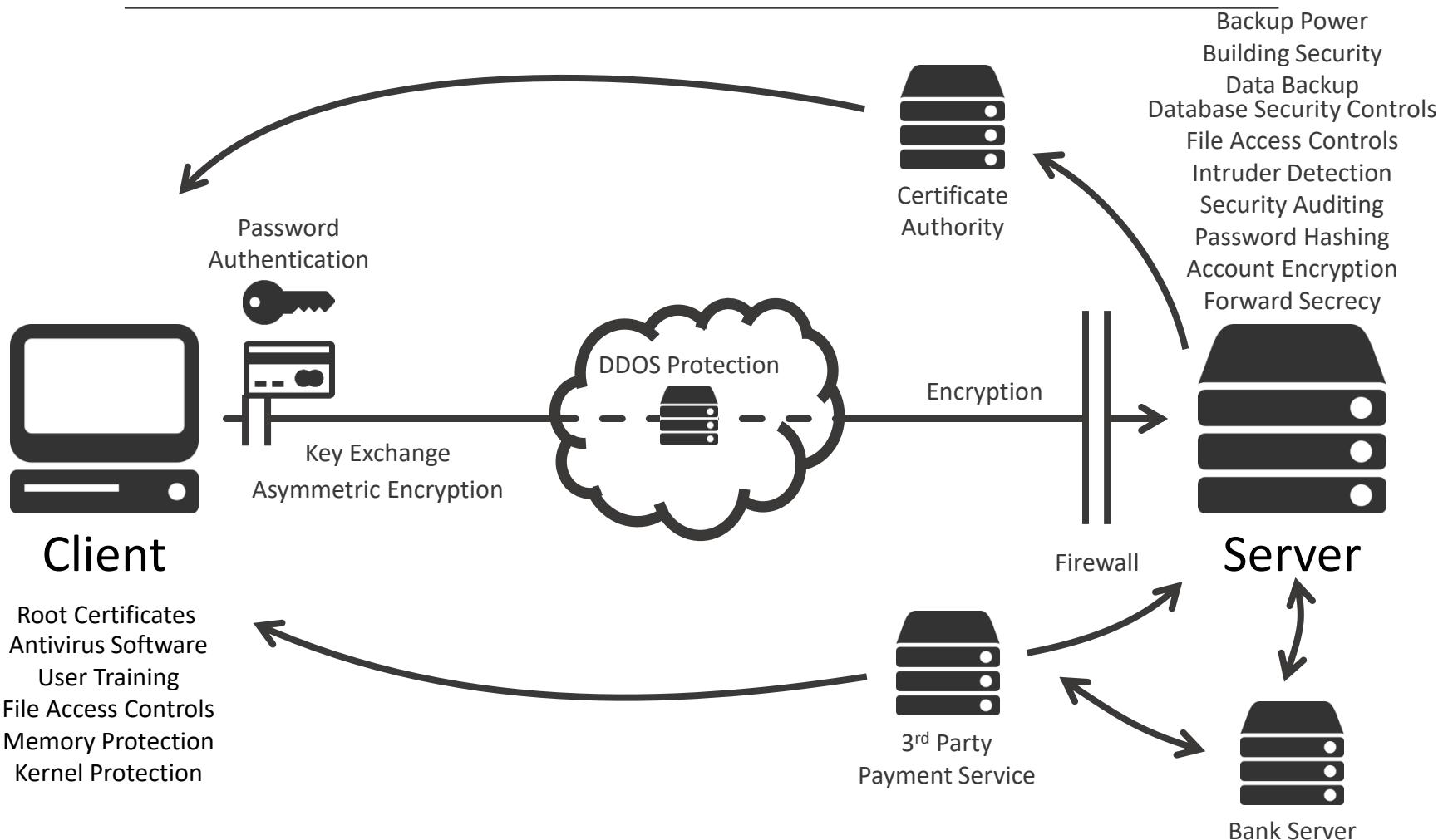
# Financial Security



# Financial Security



# Security Affects Everything



# Motivation

---

- This module will cover the core aspects of computer security
  - but of course we can't cover everything
- Security affects every aspect of modern computer science
- A brief look into common vulnerabilities and countermeasures in distributed systems

# COMP3052.SEC Computer Security

## Session 03: Foundations of Security



# ACKNOWLEDGEMENTS

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towe, ...

# OVERVIEW

- Key Concepts
- Fundamental Dilemma
- Data vs. Information
- Principles of Computer Security Design
- Summary

# KEY CONCEPTS

- Security
- Computer Security
- Confidentiality
- Integrity
- Availability
- Accountability
- Nonrepudiation

# SECURITY

- Security:
  - Security is about the protection of assets
  - Knowledge of assets and their value is vital
- Protection measures:
  - Prevention – sometimes the only feasible measure
  - Detection
  - Reaction
    - Recovery? Manual? Automatic?

# COMPUTER SECURITY

- Traditionally defined by three areas: **CIA**
  - Confidentiality
    - *prevention of unauthorised disclosure of information*
  - Integrity
    - *prevention of unauthorised modification of information*
  - Availability
    - *prevention of unauthorised withholding of information or resources*

# ACTIVITY ...

- Write down a list of as many security measures you can think of relating to:
  - Confidentiality
  - Integrity
  - Availability
- Are there any other areas?
- Which are higher or lower priorities?

# CONFIDENTIALITY

- The prevention of unauthorised users reading sensitive (private, secret) information
- Privacy – protection of personal data
- Secrecy – protection of data of an organisation
- Examples:
  - Hide document's content
  - Hide document's existence (Unlinkability and Anonymity)

# INTEGRITY

- Informally
  - Making sure everything is as it is supposed to be.
- Formally
  - Integrity deals with the prevention of unauthorised writing.

# INTEGRITY

- The prevention of unauthorised modification of data, and the assurance that data remains **unmodified**
- Examples:
  - Distributed bank transactions
  - Database records

I promise to pay Dave the  
sum of Twenty RMB

^

Thousand

# INTEGRITY

- Informally
  - Making sure everything is as it is supposed to be.
- Formally
  - Integrity deals with the prevention of unauthorised writing.
- Data Integrity

*“The state that exists when computerised data is the same as that in the source documents and has not been exposed to accidental or malicious alteration or destruction.” [Orange Book]*

DOO 5200.2B-STD  
Supercedes  
CSC-STD-001-83, eff. 15-Aug-83  
Library No. 5225.711



DEPARTMENT OF DEFENSE STANDARD

**DEPARTMENT OF  
DEFENSE  
TRUSTED COMPUTER  
SYSTEM EVALUATION  
CRITERIA**

DECEMBER 1985

# AUTHENTICITY

- Just because we have integrity, doesn't mean we have authenticity
  - Can we **verify the sender?**
  - Does it have **freshness?**
- Authenticity = Integrity + Freshness
- Freshness may seem trivial, but it's pretty important in bank transactions!



# AVAILABILITY



404

Page not found

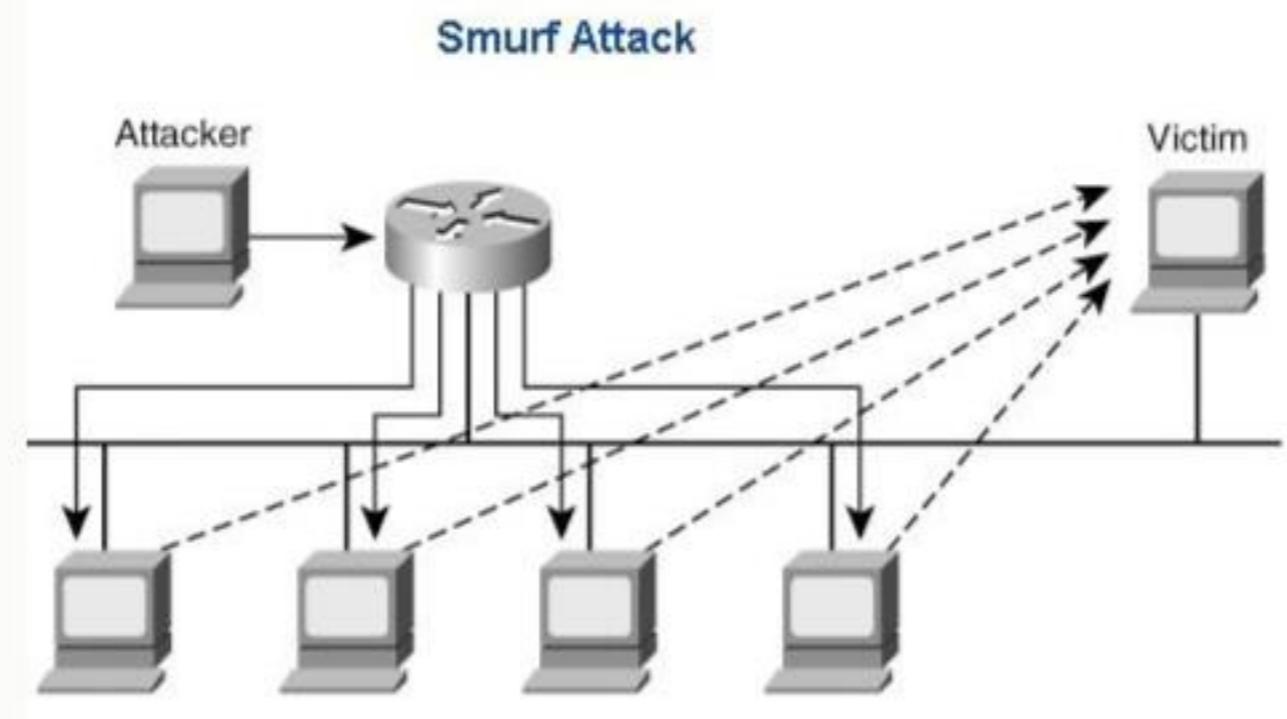
- Availability

*“The property of being accessible and useable upon demand by an authorised entity.”*

- We want to prevent denial of service (DoS):

- *“The prevention of authorised access to resources or the delaying of time-critical operations.”*

# SMURF

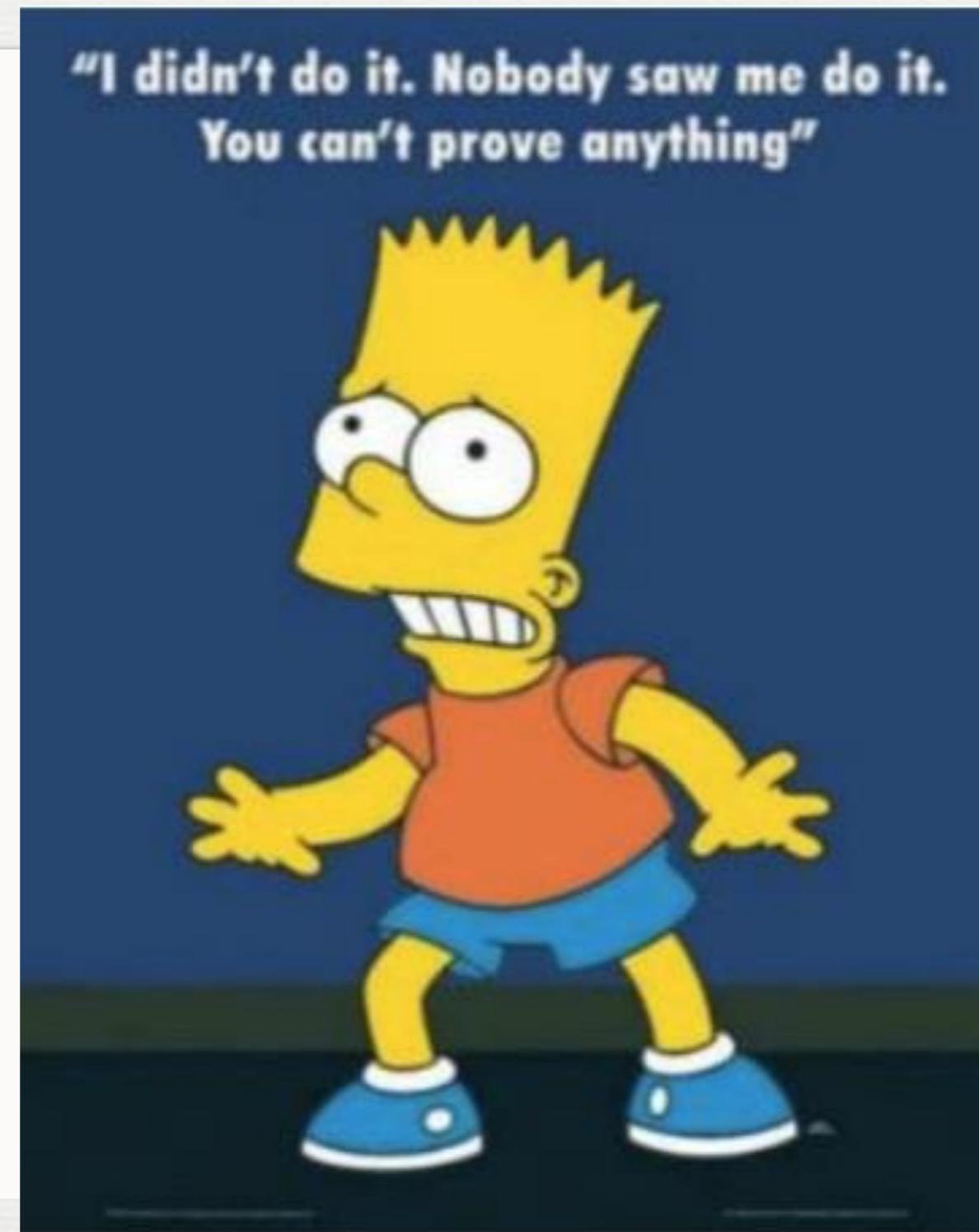


- Attacker sends ICMP echo request (ping) to broadcast address of a network, spoofing the sender address to be that of the victim

# ACCOUNTABILITY

- Users should be held responsible for their actions
- System should identify and authenticate users
- Audit trail should be kept
  - *“Audit information must be selectively kept and protected so that actions affecting security can be traced to the responsible party”*

# NON-REPUDIATION



# NON-REPUDIATION

- Non-repudiation provides un-forgeable evidence
- Evidence verifiable by a third party
  - E.g., notaries, digital certificates, ...
- Nonrepudiation of:
  - origin – sender identification
  - delivery – delivery confirmation
- Relate to physical security (keycards,...)

# RELIABILITY

- Reliability - against (accidental) failures
- Safety - impact of system failures on their environment
- Security is an aspect of reliability, and *vice versa!*
- Dependability

*“The property of a computer system such that reliance can justifiably be placed in the service it delivers”*

# OUR DEFINITION

- Computer Security – What?

*“Deals with the prevention and detection of unauthorised actions by users of a computer system”*

- Computer Security – Why?

*“Concerned with the measures we can take to deal with intentional actions by parties behaving in some unwelcome fashion”*

# REMEMBER

- No single definition of security exists
- When dealing with security material, do not confuse your notion of security with that used in the material

# FUNDAMENTAL DILEMMA

*“Security-unaware users have specific security requirements but usually no security expertise.”*

- Trade-off between security and ease of use

# FUNDAMENTAL DILEMMA

- In contrast, conflict between security and ease of use:
- Engineering trade-off:
  - Security mechanisms need **increased computational resources**
  - Security **interferes** with **working patterns** of users
  - Managing security is **work** – thus better (G)UI wins

# DATA VS INFORMATION

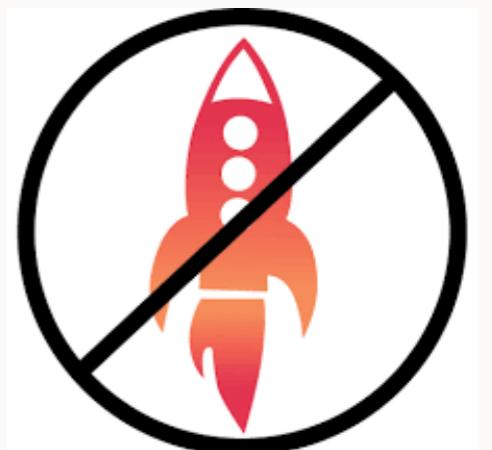
- Security is about **controlling access to information** and resources
  - This can be difficult, thus controlling **access to data** is more viable
  - **Data** – Means to represent information
  - **Information** – (subjective) interpretation of data
- Problem of inference ...

# PROBLEM OF INFERENCE

- Focusing on data can still leave information vulnerable
- Consider a medical database
  - Medical records cannot be queried
  - Aggregates like prescription totals can be
- Carefully chosen queries can narrow down who has what conditions
  - A covert channel
  - Compare:
    - “Joe’s criminal record not found in the DB”
    - “You do not have permission to access to Joe’s criminal record”

# SECURITY DESIGN: PRINCIPLES

- Computer security is NOT rocket science if:
  - approached in a systematic, disciplined & well planned manner
  - from the inception of a developed / designed system
- However:
  - if added as an **afterthought** to an **existing, complex** system ->  
**TROUBLE!**



# SECURITY DESIGN: PRINCIPLES

- Fundamental Design Principles:
  - Focus of Control
  - Complexity vs. Assurance
  - Centralised or Decentralised Controls
  - Layered Security

# FOCUS OF CONTROL

- **1st Design Decision:**

*In a given application, should the protection mechanisms in a computer system focus on:*

Data



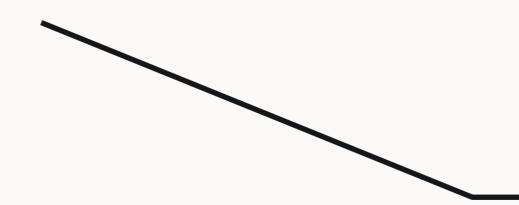
Permitted manipulation of data e.g. consistency check

Operations



Permitted invocations e.g. transfermoney()

Or users?



Permissions for specific users  
e.g. /home/name/

# COMPLEXITY VS ASSURANCE

- **2nd Design Decision:**

*Do you prefer simplicity- and higher assurance- to a feature-rich security environment?*

This decision is linked to the fundamental dilemma!

Feature-rich security systems and high assurance do not match easily

# (DE) CENTRALISED CONTROLS

- **3rd Design Decision:**

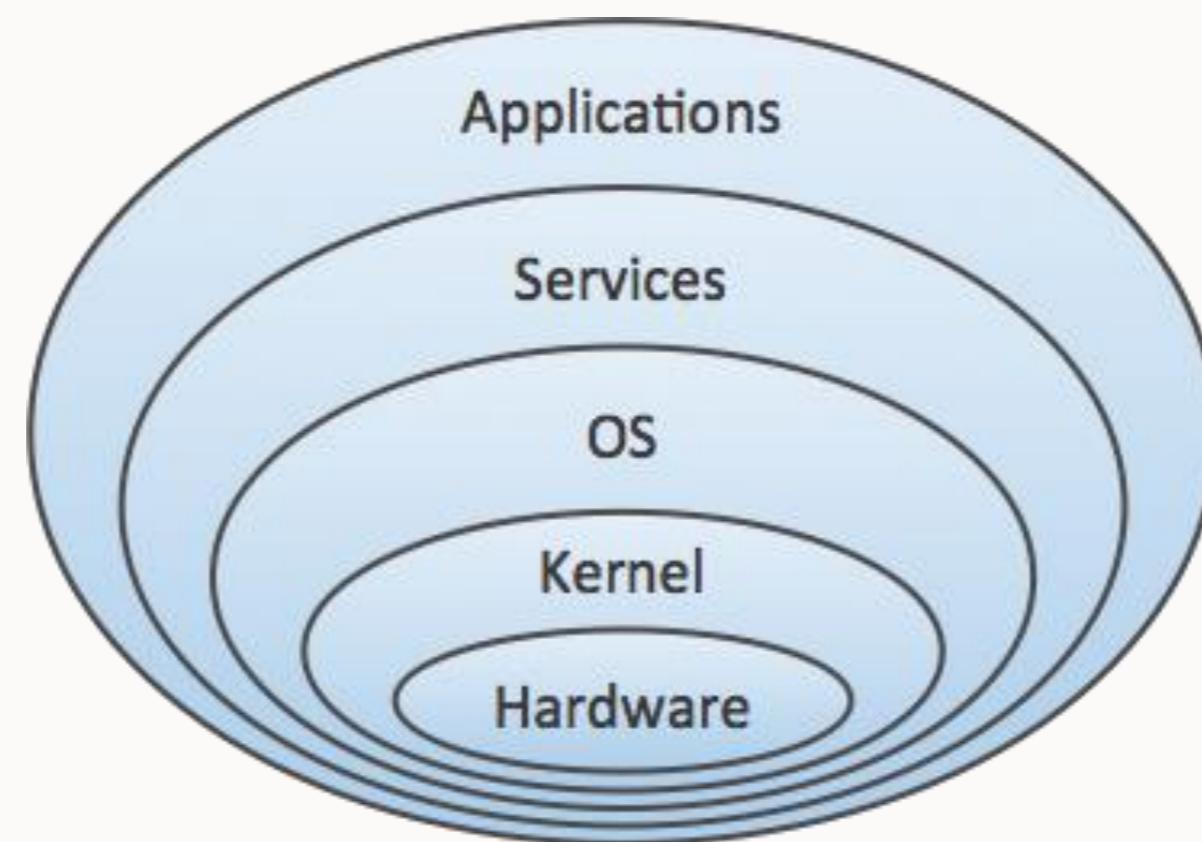
*Should the tasks of defining and enforcing security be given to a central entity or should they be left to individual components in a system?*

Central entity – could mean a bottleneck

Distributed solution – more efficient but harder to manage

# THE ONION MODEL

- We can visualise our security model in layers
- Each layer protects a boundary, and relies on the security of the layers below



# THE LAYER BELOW

- Every protection mechanism has a defined security perimeter

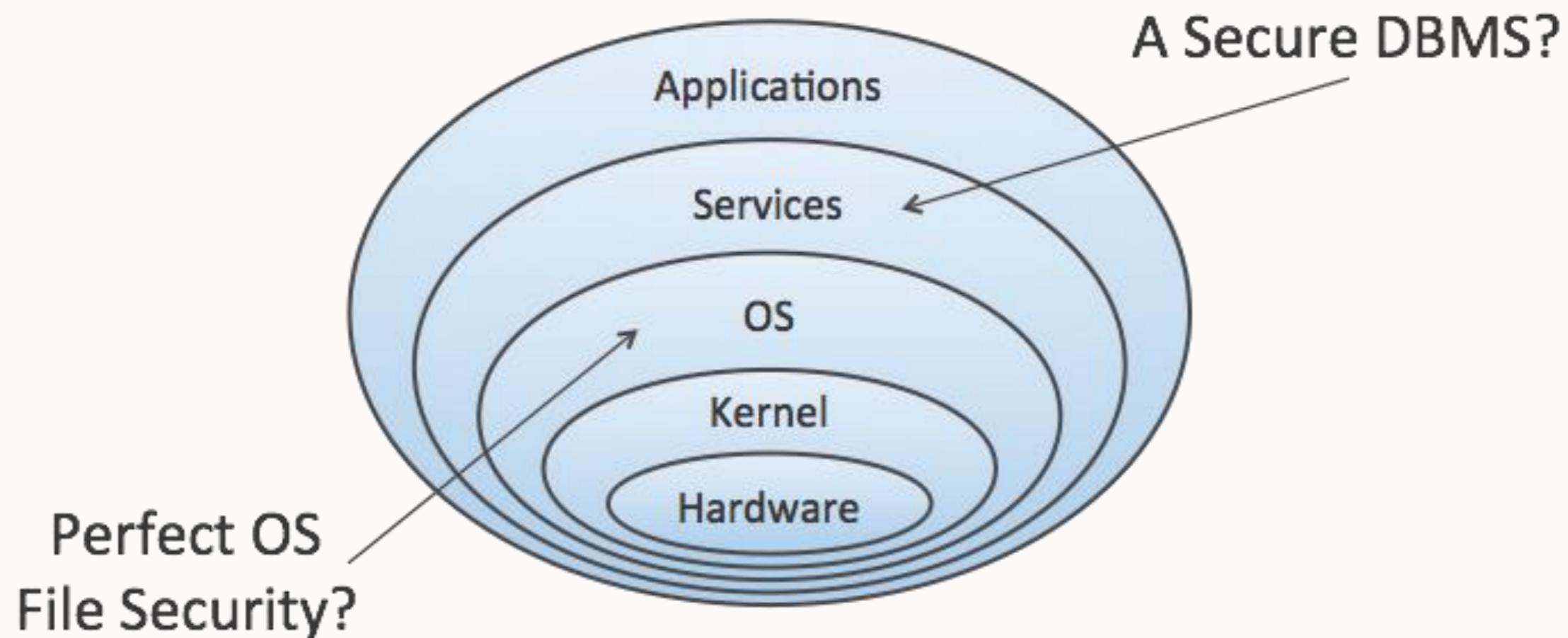
*Security perimeter – parts of a system that can be used to disable the protection mechanism lying within*

## 4th Design Decision:

*How can you prevent an attacker getting access to a layer below the protection mechanism?*

# THE LAYER BELOW

- A good security layer built upon an insecure layer is useless



# SUMMARY

- Summary:
  - Definitions
  - Fundamental Dilemma
  - Data vs. Information
  - Principles of Computer Security
  - The Layer Below

Read:

- Gollman: Chapter 3
- Anderson: Section 1.7

# COMP3052.SEC Computer Security

## Session 04: Authentication



# Acknowledgements

---

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towey, ...

# This Session

---

- Authentication
- Problems with passwords
- Storing passwords
- Cracking passwords
- Multi-factor authentication
- Biometrics

# Authentication

---

- To allow someone access to an asset we must ensure:
  - They are permitted to access that asset
  - They are who they say they are
- We can attempt to verify identity using credentials
  - Something they **are**
  - Something they **have**
  - Something they **know**

# Usernames and Passwords

---

- Identification – Who you are
- Authentication – Verify that identity
- Authentication should expire
  - “Remember my credentials” turns this into something you have
- Time of check to time of use – **TOCTTOU**
  - Repeated authentication
  - At the start and during a session

# Passwords

---

- Passwords are digital keys
  - Simple to implement using existing libraries
  - Demonstrates someone is who they say they are
- Understood by the users (mostly)
- But: In many cases passwords are a terrible way to handle authentication

# Problems With Passwords

---

- People forget them
- They can be guessed
- Spoofing and Phishing
- Compromised password files
- Keylogging
- Many of these are made many times worse by weak passwords



# Weak Passwords

- If left to their own devices, people will use terrible passwords
  - Spouse's name
  - Known dates from their life
  - Small variants on their own name
  - qwerty1234

Check Nordpass, too:

<https://nordpass.com/most-common-passwords-list/>

Rank	Password	Change from 2018
1	123456	Unchanged
2	123456789	1↗
3	qwerty	6↗
4	password	2↘
5	1234567	2↗
6	12345678	2↘
7	12345	2↘
8	iloveyou	2↗
9	111111	5↘
10	123123	

# Password Policies

---

- Many companies (and now websites) enforce password policies
  - Certain length, certain types of characters
  - No dictionary words
  - Change regularly
  - No previously used passwords



# Password Policies

---

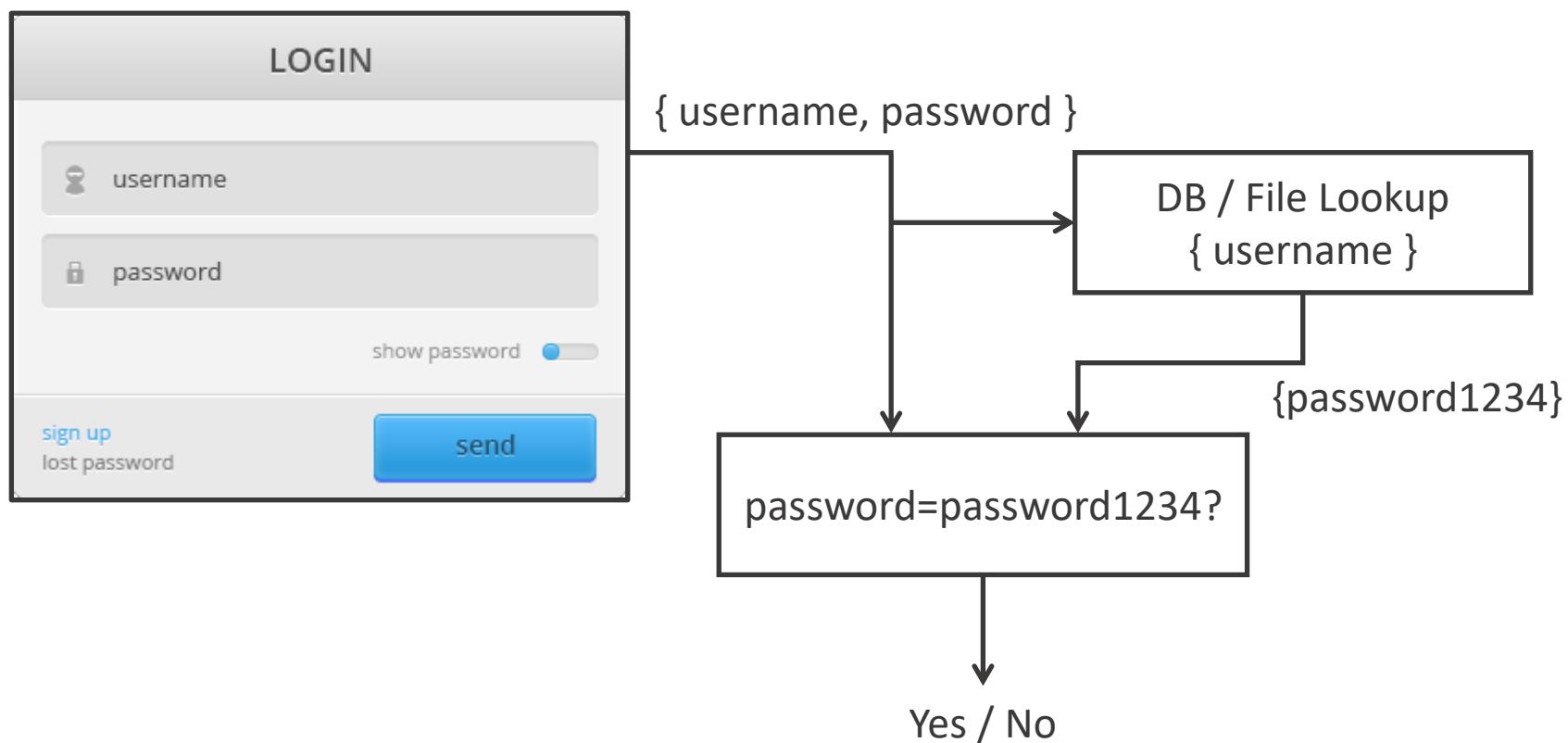
- This is not a great solution
  - People attempt to make their life easier by **re-using** passwords
  - When they're forced to change to unique passwords, they'll simply increment a counter

# Users Don't Understand Security



# Password Authentication

- The bad way:



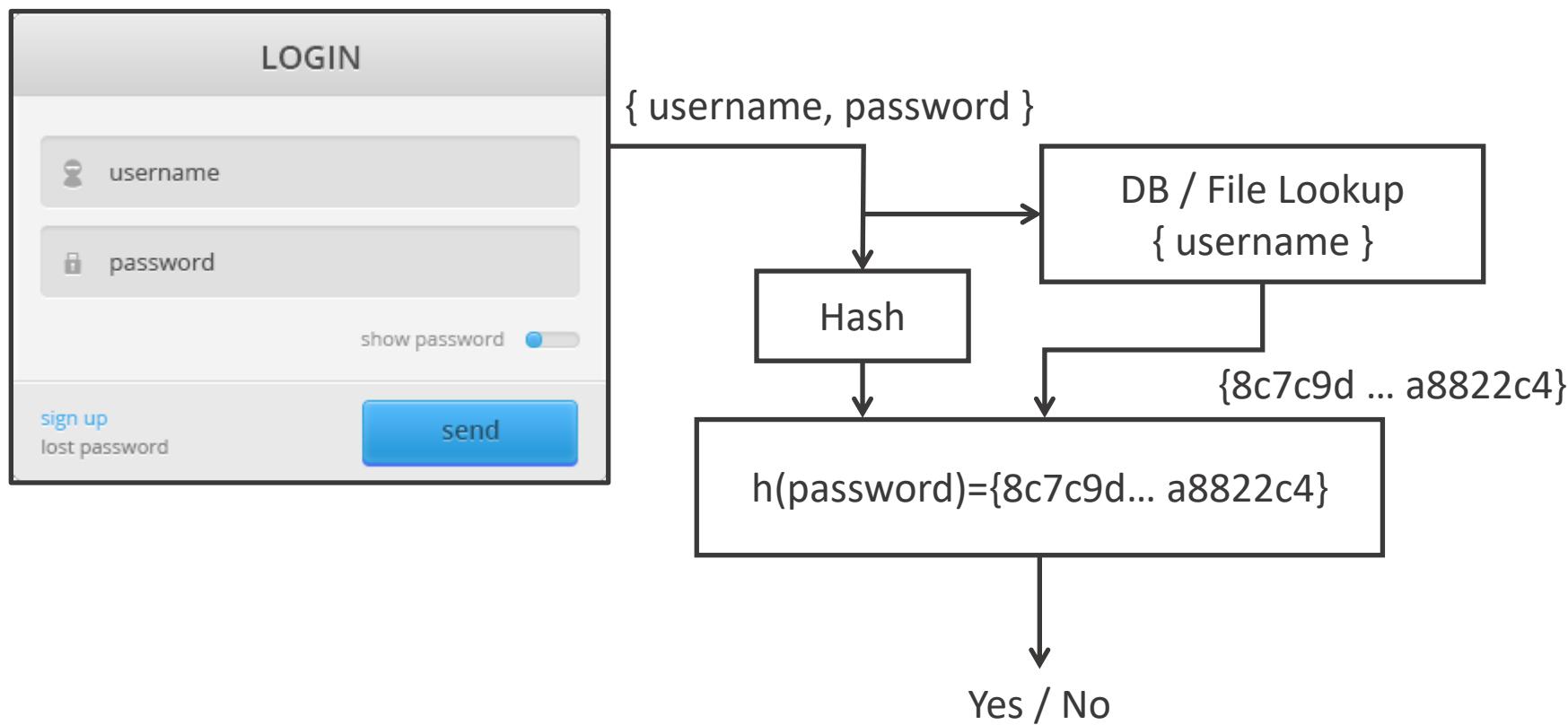
# Storing Passwords

---

- Storing passwords in plain text is a terrible idea
  - You might be hacked
  - Administrators can read them
- Storing encrypted passwords is better, but not perfect
  - Where are the keys stored?
  - Administrators can still read them

# Storing Passwords

- Using a **one-way hash function** is a much better solution:



# Password / Shadow Files

---

- Operating systems have taken steps to stop people reading hashes for offline attacks
  - Linux stores hashes in a shadow file /etc/shadow
  - Windows stores this in ..\system32\config\SAM
- These files are now **read protected**
- Administrators or people booting another OS will often find a way in

# Cracking Passwords

---

- Cracking a password isn't always illegal, though obviously it sometimes is!
- Password cracking falls into two basic types:
  - Offline: You have a copy of the password hash locally
  - Online: You do not have the hash, and are instead attempting to gain access to an actual login terminal
- Online is usually attempted with phishing

# Password Cracking

---

- Offline password cracking is simply a case of trying lots of possible passwords, and seeing if we have a hash collision with a password list
- This could be done with a Brute Force approach
  - Difficulty is calculated as  $\{char\_count\}^{length}$
  - GPUs are fast, but not fast enough for long passwords

# Dictionary Attacks

---

- Most password cracking is now achieved using **dictionary attacks** rather than brute force
  - Using a dictionary of common words and passwords
  - Apply small variations to this list, trying them all
  - Combine words from two different lists
- *qwerty1234password1* is unbreakable using brute force, but won't last against a dictionary attack

# Password Strength

---

- Which of these passwords is strongest?

michael2001

password!

N!ghtingal3

helikesfootba\_ll

# Password Strength

---

- What's the search space?

helikesfootba\_ll

3 words from 10000	$10000^3$
1 symbol from 15	15
1 position from 16	x
	240 Trillion

**For a weak hash at 8Mh/s: 1 year**

# Password Strength

---

- A small improvement?

helikesnonlinearfootba\_ll

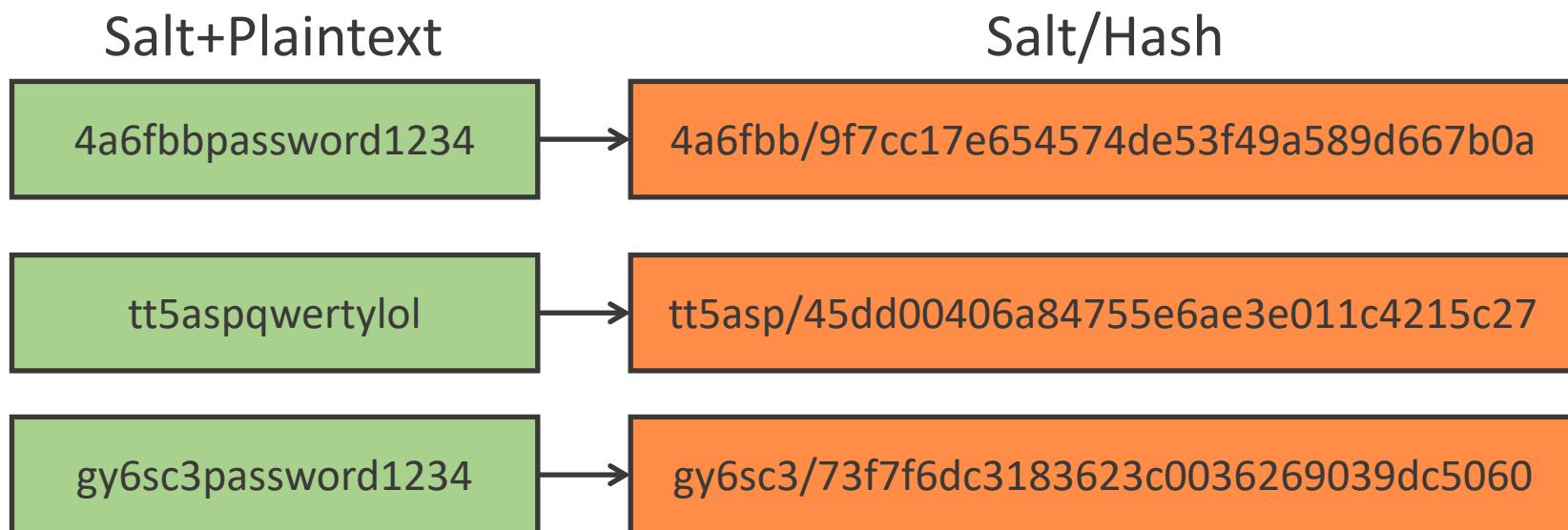
4 words from 20000	$20000^4$
1 symbol from 15	15
1 position from 25	x
	6 Quintillion

**For a weak hash at 8Mh/s: 237,000 years**

# Password Salting

---

- We can improve security by prepending a random “salt” to a password before hashing
- The salt is stored unencrypted with the hash:



# Password Salting

---

- If we use a different **random salt** for each user, we get the following security benefits:
  1. Cracking multiple passwords is slower – a hit is for a single user, not all users with that password
  2. Prevents **rainbow table** attacks – we can't pre-compute that many password combinations
- Salting has no effect on the speed of cracking a single password – so make your passwords good!

# Hashing Speed

---

- When password cracking, the most important factor is **hashing speed**
- Newer algorithms take longer
  - Partly because they're more complex
  - But some have been specifically designed to take a while
- Iterate to increase complexity - PBKDF2
- bcrypt can't be easily used on GPUs

# System Design Issues

---

- Different situations require different password strengths and policies
  - A PIN code can be short, because you can only test three before you're locked out
  - Locking people out after  $n$  tries is fine in principle, but could lead to DoS
- Poorly designed interfaces can lead to a lot of problems
  - E.g. ATM machines have to be designed to prevent “shoulder surfing”

# System Design Issues

---

- Attacks on password storage are also extremely common
  - Obtaining shadow files
  - SQL Injection of web databases
- Can be made worse by stupid mistakes
  - On systems that log failed authentication attempts, what if you type your password in the wrong place?

# If Cracking Fails: Pretexting

---

- Obtaining private details by offering some “**pretext**” as a reason for needing them
- We continue to rely on email addresses, DOB and Mother’s maiden names as our “last line of defense” for security
- How much information do we need to ring up a company as someone else?

# Alternatives

---



# Alternatives

---

- Passwords are something you **know**
  - Anyone who also knows this thing, becomes you
- What about something you **have**?
- Or something you **are**?

# Something You Have

---

- A key for a lock
  - A keycard for a door
  - A long password written down
- 
- Again, anyone who obtains this item, becomes you
  - Can be used in **combination** with something you know

# Multi-Factor Authentication

---

- Combines something you **know** with something you **have**
- Common examples:
  - Text codes to mobiles
  - One time passwords, Google Authenticator, Microsoft Authenticator etc.
  - USB devices e.g. Yubico/Yubikey
- New devices and TOCTTOU are common uses for two-factor authentication

# Biometrics

---

- Measurements of the human body, something you **are**
- Various forms, fingerprint recognition, iris / retina recognition, voice, gait, typing rhythm
- A password you always have with you, but you **can't change**



# Biometric Accuracy

---

- Usually operate by finding “features” within data, then use these to learn a template for a given individual
- The accuracy of a biometric system is extremely important
  - False positive rate
  - False negative rate
- There will usually be a trade-off between FP and FN rates

# Fingerprint Recognition

---

- The main pattern is found
  - Arch, Loop, Whorl
- Minutiae – feature points
  - Ridge ending
  - Short ridge
  - Bifurcation
  - Crossover
  - Delta
  - Island
  - Enclosure



# Fingerprint Recognition

---

- Modern fingerprint recognition is about 99% accurate
  - Is this enough?
- The iPhone fingerprint recognition was bypassed within days, but it requires custom hardware
  - Probably robust to normal people stealing phones

# Where you are

- Location of access
    - Operator console vs any console
    - Office workstation vs home PC
    - Geographical location
  - This is a useful addition to multi-factor authentication
  - Not reliable on its own



# Summary

---

- Authentication
- Problems with passwords
- Storing passwords
- Cracking passwords
- Multi-factor authentication
- Biometrics

**Anderson (2<sup>nd</sup> Ed)**  
**Chapter 2**  
**(especially 2.4)**

**Gollmann**  
**Chapter 4**

# COMP3052.SEC Computer Security

## Session 05: Access Control



# Acknowledgements

---

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towe, ...

# This Lecture

---

- Access Control fundamentals
- Principles, Subjects and Objects
- Access control structures
- Groups and Roles
- Evaluating access at runtime

# Background

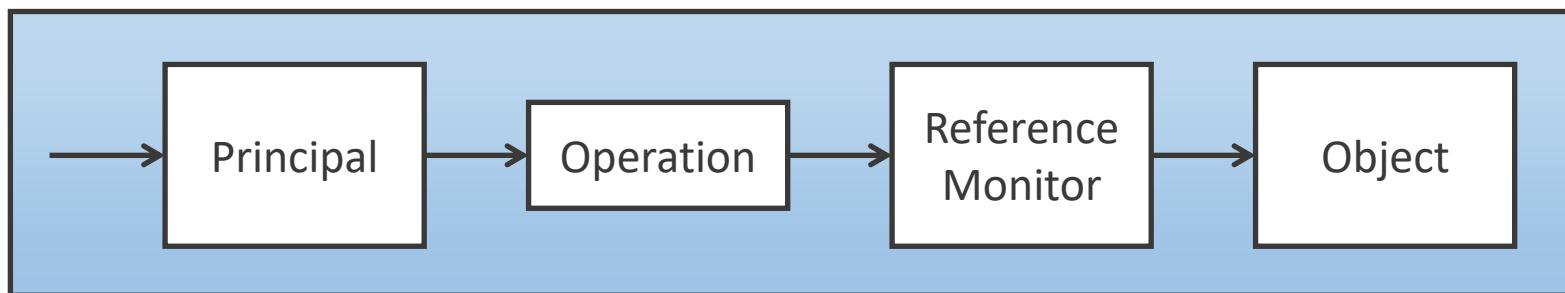
---

- Authentication lets us verify who we are to a system
- Assuming we've authenticated:
  - Some files are private, some are public
  - System files should be protected
  - We need to be able to access some applications
- We need a mechanism to enforce **access control**

# Authentication & Authorisation

---

- Subject / Principal – an active entity
- Object – resource being accessed
- Access an operation
- Reference monitor – grants or denies access



# Authentication & Authorisation

---

- Access control has two steps:
  - Authentication
    - Decide who has access to the system
  - Authorisation
    - Of those with access, who is authorised to do something to the resource (object)

# Principal vs. Subject

---

- Principal

*“An entity that can be granted access to objects or can make statements affecting access control decisions”*

- E.g user identity in an OS
- Used when discussing security policies

- Subject

*“An active entity within an IT system”*

- E.g. process running under a user identity
- Used when discussing operational systems enforcing policies

# Subject vs. Object

---

- Object – Files or resources
  - E.g. Memory, printers, directories
- Subject vs Object: Distinguish between the active and passive party in an access request
- Two options for focusing control:
  - What a subject is allowed to do
  - What may be done to an object

# Access Operations

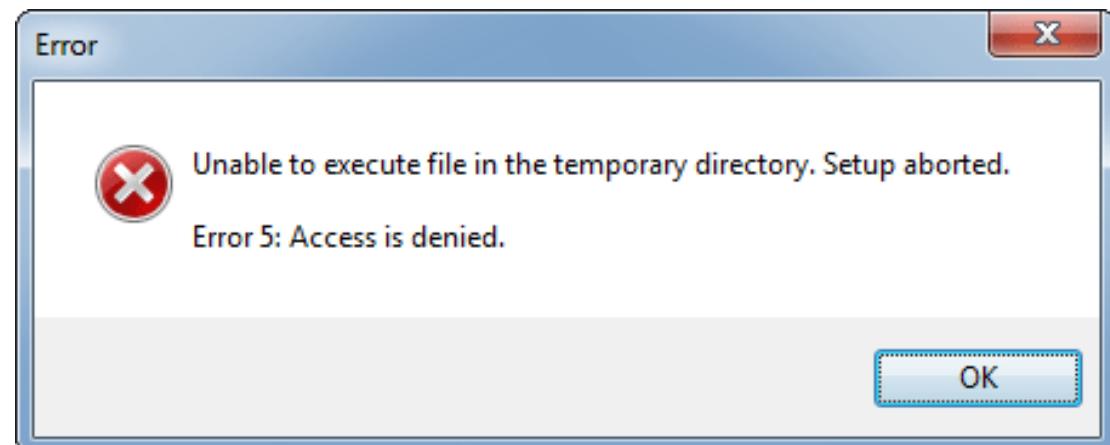
---

- From reading / writing to method calls
  - Varies from system to system
  - Sometimes similarly named operations in two systems will have different meanings
- 
- Access modes
    - Observe – Subject may look at contents of an object
    - Alter – Subject may change the contents of an object
  - Too abstract for practical use!

# General Model

---

- We'll settle on some common access on files:
  - Read – Simply viewing (Confidentiality)
  - Write – Includes changing, appending, deleting (Integrity)
  - Execute – Can run the file without knowing its contents



# Ownership

---

- Who is in charge of setting security policies?
- Discretionary: Owner can be defined for each resource
  - Owner controls who gets access
- Mandatory: Could be a system-wide policy
- Most OSs support the concept of ownership

# Access Control Structures

---

- Help express access control policy
- Often focused upon efficient lookup – resources are accessed a lot!
- Access Control Matrix
- Access Control Lists
- Capabilities

# Access Control Matrix

---

- Access rights are defined individually for each combination of subject and object
  - Quite an abstract concept, but would allow for very fine grained control
  - Not practical, think of the memory required in scaling it up!

	budget.xlsx	game.exe	msexcel.exe
Alice	r,w,e	r,e	r,e
Bob	r		r,e
Claire	r		r,e
Dave		r,w,e	r,e

# Access Control List

- Stored with an object itself, corresponding to a column of an ACM

	budget.xlsx	game.exe	msexcel.exe
Alice	r,w,e	r,e	r,e
Bob	r		r,e
Claire	r		r,e
Dave		r,w,e	r,e

The diagram illustrates how an Access Control List (ACL) is represented. A pointer originates from the 'budget.xlsx' row in the matrix and points to a summary row at the bottom. This summary row contains four entries: 'budget.xlsx' followed by the access rights for each user: Alice (r,w,e), Bob (r), Claire (r), and Dave (r,w,e). The matrix itself shows the detailed access rights for each user across three objects: budget.xlsx, game.exe, and msexcel.exe.

budget.xlsx	Alice: r,w,e	Bob: r	Claire: r
-------------	--------------	--------	-----------

# Access Control List

---

- Better:
  - Much less memory intensive
  - If stored with a file is quick to access
- However:
  - Management of individual subjects is cumbersome
  - Obtaining an overview of permissions is challenging
  - Tedious to set this up properly for all subjects and objects

# UNIX

---

- Unix simplifies the ACL structure to consider only the user, group and others
  - User is the current owner
  - Group is a named group entity
  - Everyone else
- Unix offers Read, Write and Execute access controls

# Windows

---

- Windows predominantly uses ACLs, and has done since Windows NT
- Extends the usual read, write and execute with:
  - Take ownership
  - Change permissions
  - Delete
- A higher degree of control, with the associated complexity increase!

# Capabilities

---

- Access rights are stored with a subject, not a resource
- Every subject is given a capability:  
*“An unforgeable token specifying the subject’s access rights”*
- Corresponds to a row in an access control matrix:

Alice	budget.xlsx: {rwx}	game.exe {r,e}	msexcel.exe {r,e}
-------	--------------------	----------------	-------------------

# Capabilities

---

- Typically associated with discretionary access control
  - Subjects can pass on their capabilities
- Not widely used – e.g. exists in Linux but rare
- Difficult to get an overview of access rights on a file, and revoke them

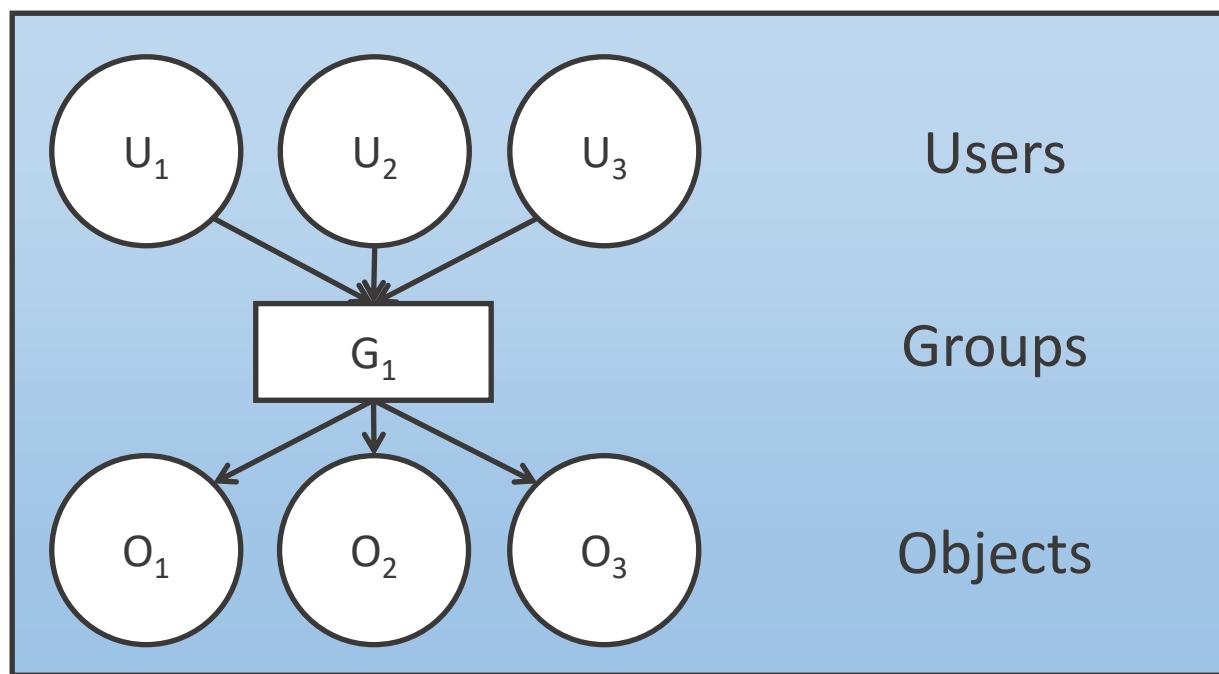
# Intermediate Controls

---

- Problems of complexity and scalability solved by indirection
  - Groups
  - Negative Permissions
  - Privileges
  - Role-based Access Control
  - Protection Rings

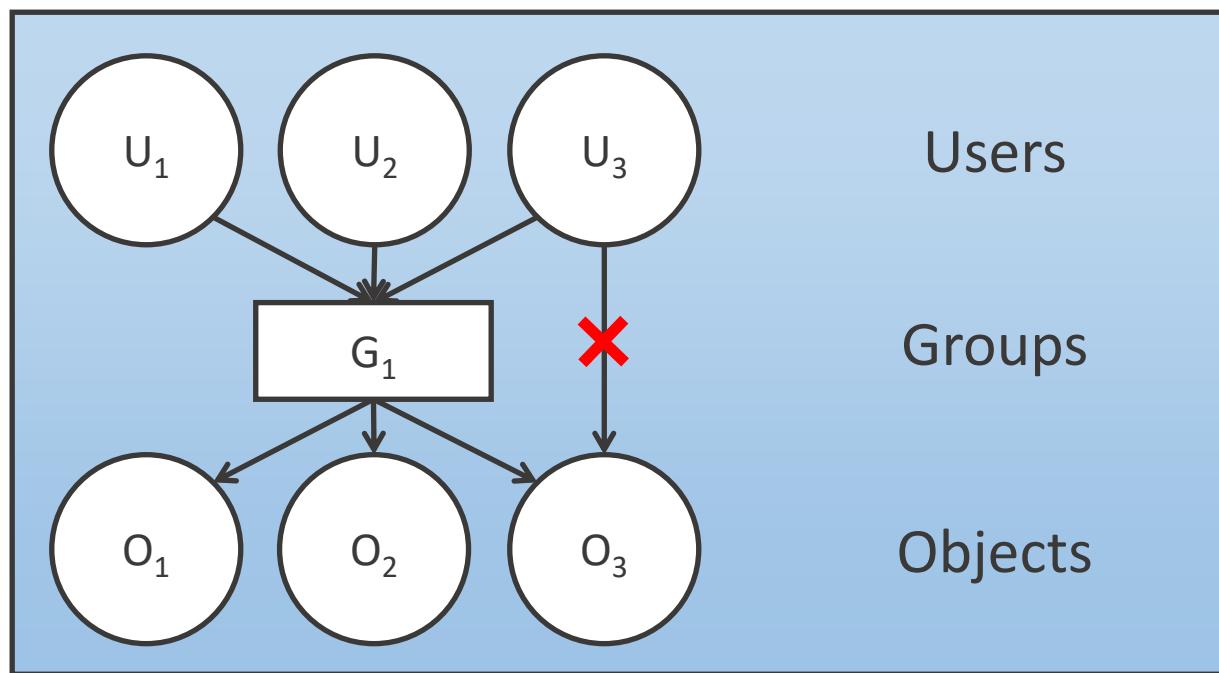
# Groups

- Users with similar access rights can be collected into groups
- Groups are given permissions to access objects



# Negative Permissions

- An operation that a user cannot perform
- Policy conflict – resolved by the reference monitor



# Alternatives

---

- Identity Based Access Control (IBAC)
  - The standard approach we've been discussing
    - e.g. ACLs
  - Scales better than a matrix, but not to enterprise level
- Role-based Access Control (RBAC)
  - Access is based on a role, e.g. accountants should access certain financial files
  - Easier to scale and use, but nothing is perfect!

# Role-based Access Control

---

- A role – Collection of application specific operations or resource access
- Subjects derive access rights from the role they perform
- RBAC focuses on users and the jobs they perform
- Much more applicable to large networks and organisations

# Role-based Access Control

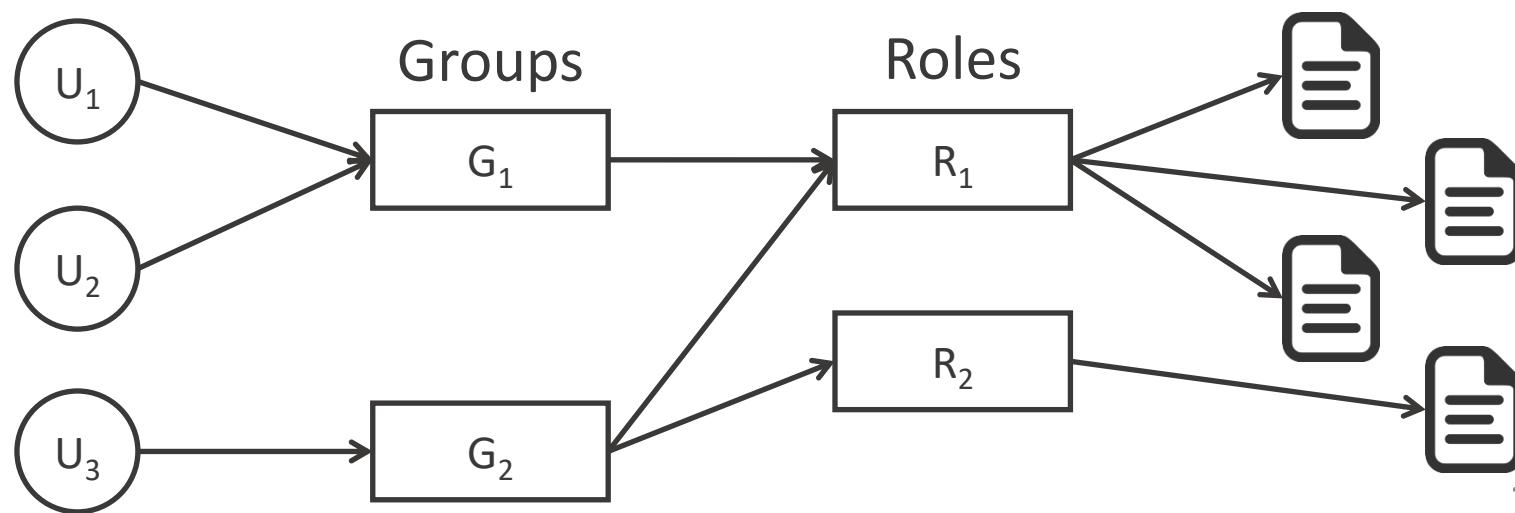
---

- Layers (between subjects and objects)
  - Roles – collection of procedures assigned to users
  - Procedures – high level access control methods
  - Data types – each object of certain data type

# Roles vs. Groups

---

- Sound the same, but are subtly different
  - Groups are collections of users
  - Roles are collections of permissions
- Most operating systems are user / group based, so role-based access can be provided using nested groups



# Evaluating Security Policies

---

- At a basic level: quality check against Access Control Entry
- More complicated:
  - Protection Rings
  - Partial Orderings
  - Lattices

# Privileges

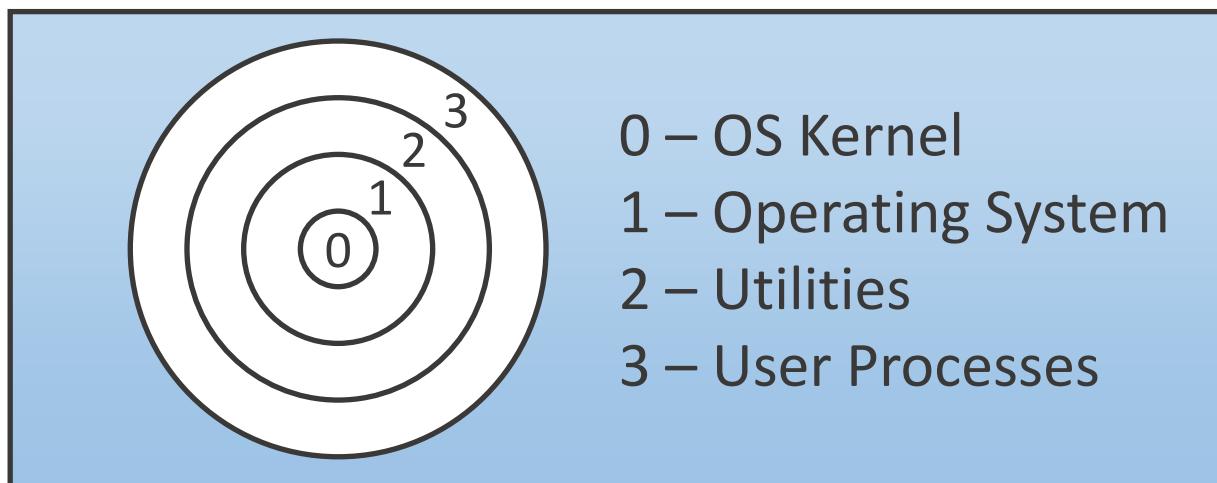
---

- A collection of rights to execute certain operations
- Come pre-defined with an OS
- An intermediate layer between subjects and operations
- Usually associated with operating system functions
  - Installing software, network access etc.

# Protection Rings

---

- Hardware-based access control
- Each subject and object assigned a number depending on importance
- Decisions are made by comparing subject's to object's numbers



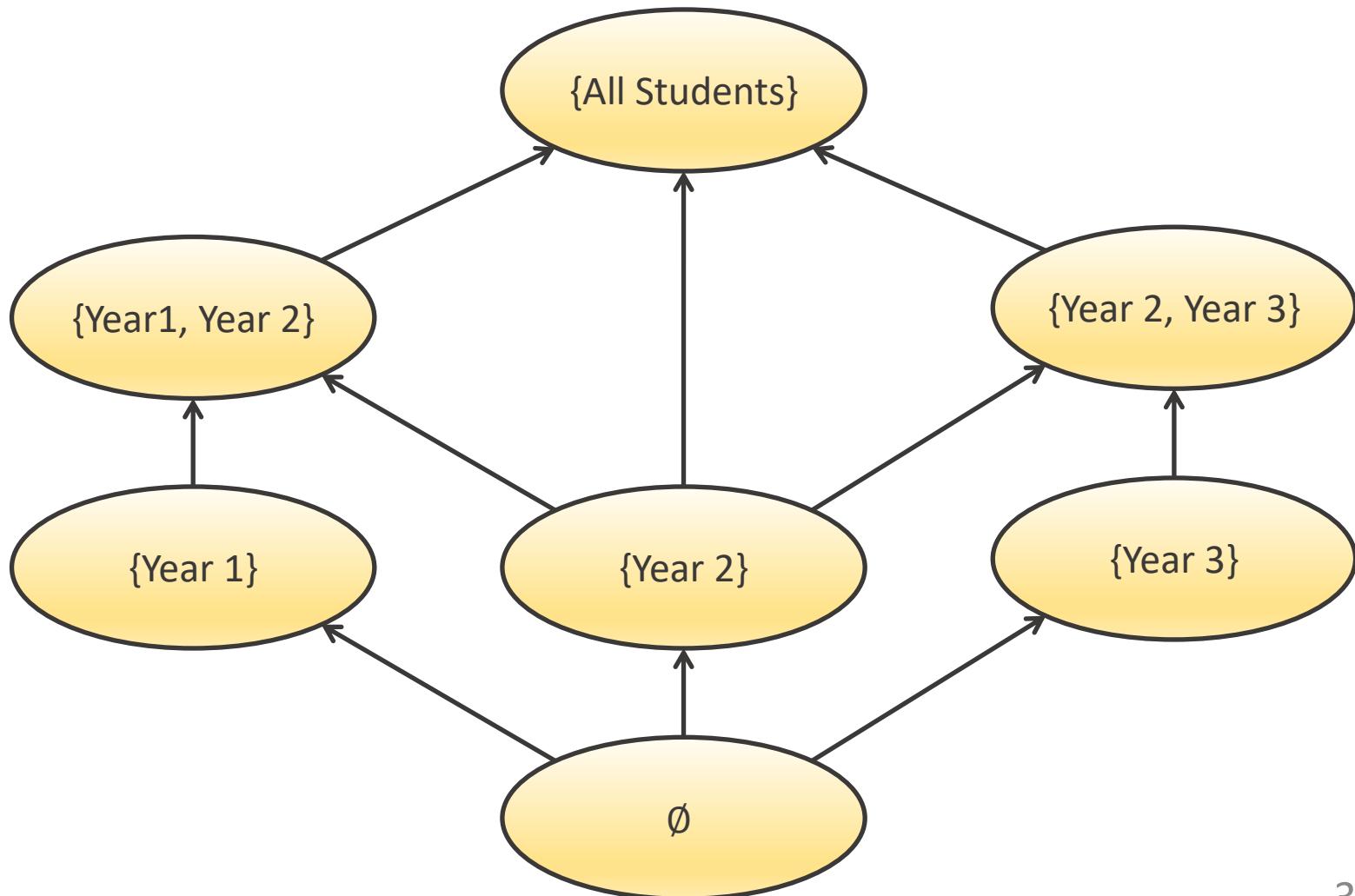
# Partial Ordering

---

- Imagine groups containing students in year 1, year 2 etc.
- Partial orderings ( $\leq$ ) provide relations between subsets of groups
- For example,  $\{\text{Year1}\} \leq \{\text{Year1}, \text{Year2}\}$
- A security policy might grant access to an object if the subject label is  $\leq$  object label

# Partial Ordering

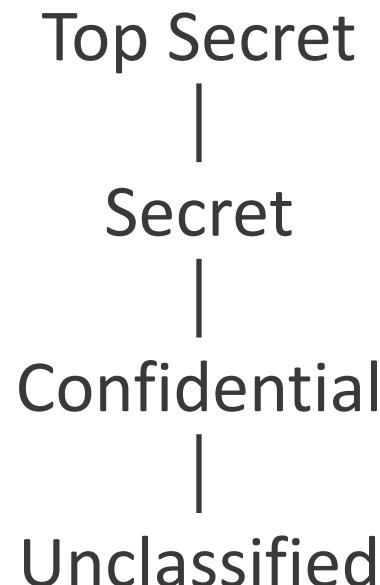
---



# Multi-Level Security

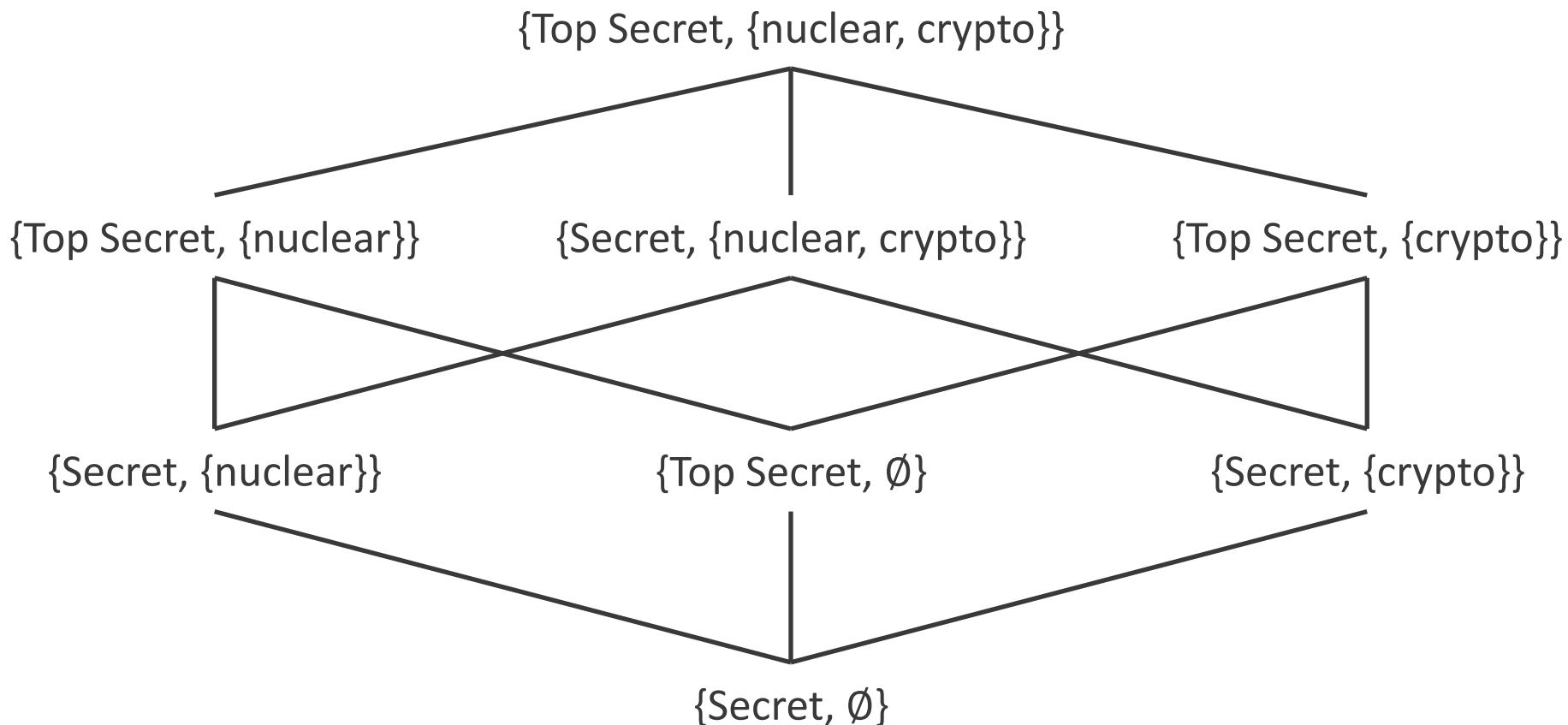
---

- Often military applications will define access based on security levels
- Easy to implement, but we can't express complex security policies



# Lattices

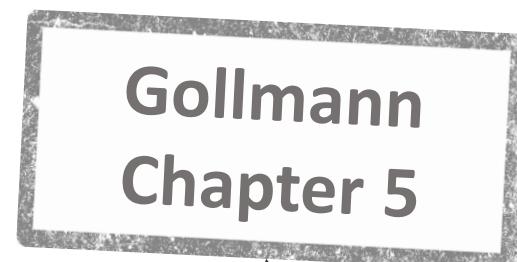
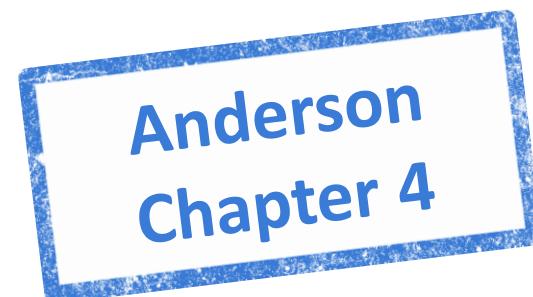
---



# Summary

---

- Access Control
- Its structures
  - Access Control Matrices, Lists
  - Capabilities
- How we manage it
  - Groups, Role-based
  - Partial ordering and lattices



Further reading  
for interested  
people!

# COMP3052.SEC Computer Security

## Session 06: Firewalls



# Acknowledgements

---

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towey, ...

# This Session

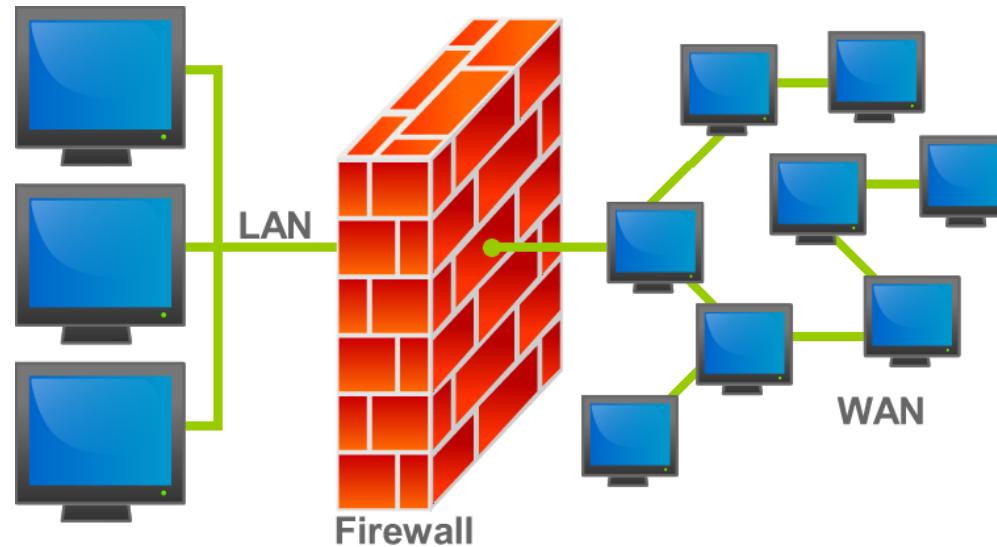
---

- Firewalls
  - Packet Filters
  - Stateful vs Stateless
- Proxies
- NAT



# Firewalls

---



- A hardware and/or software system
- Prevents unauthorised access of packets from one network to another
- All data leaving any subnet must pass through it

# Firewall Functions

---

- Implements ‘single point’ security measures
- Security event monitoring through packet analysis and logging
- Network-based access control through implementation of a rule set

# Location

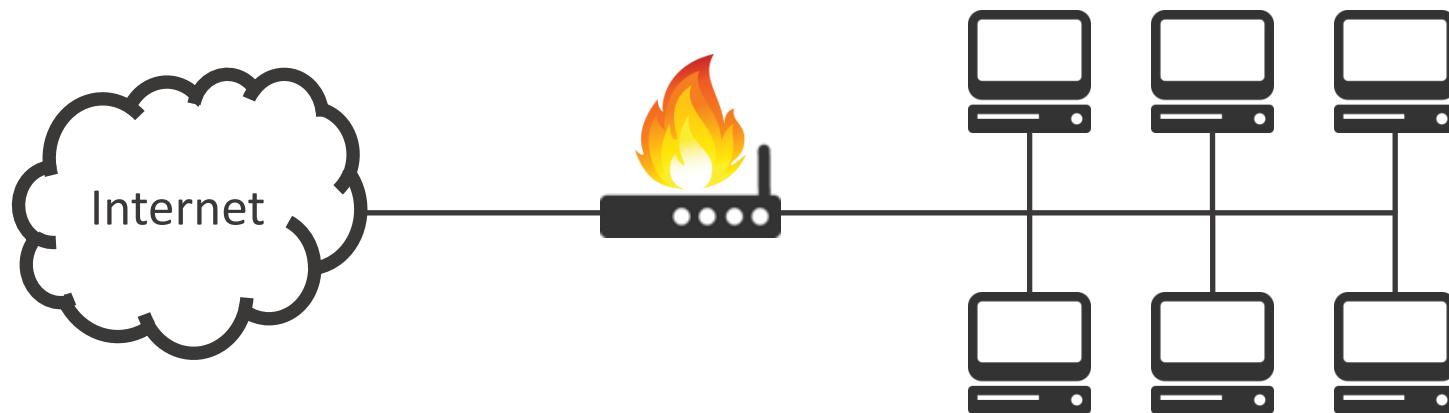
---

- Many different possible locations – thus many different network architectures
- Network Firewalls
  - Placed between a subnet and the internet
- Host-based Firewalls
  - Placed on individual machines
- All traffic must go through the firewall for it to function correctly

# Location

---

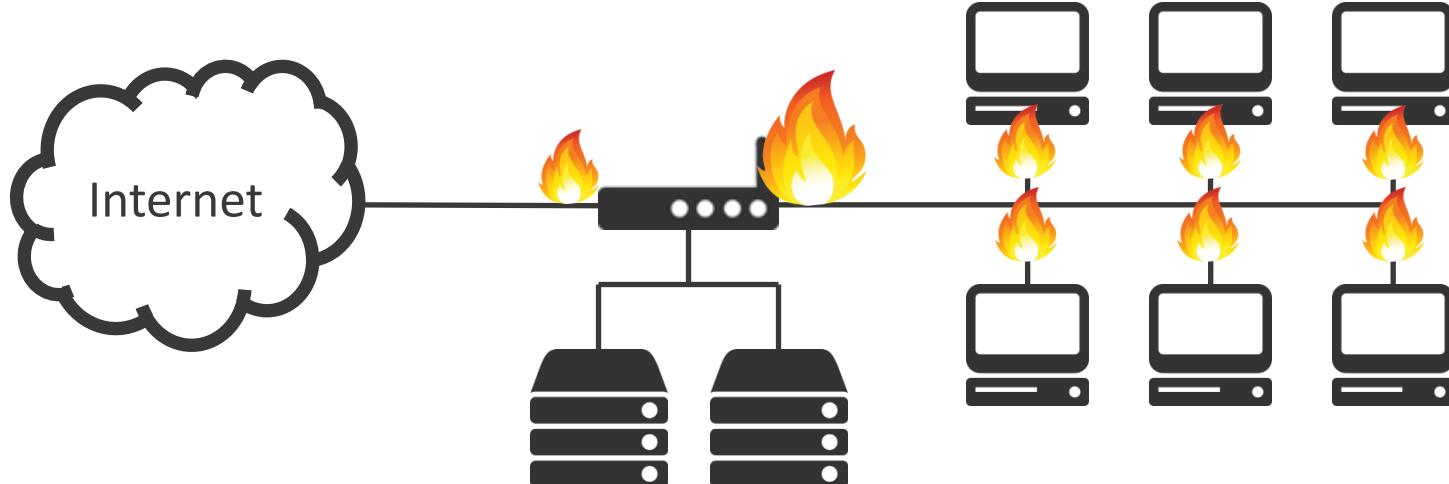
- A standard home router is a good example of a network firewall



# DMZ

---

- A demilitarized zone is a small subnet that separates externally facing services from the internal network



# Firewall Basic Function

---

- Defends a protected network against parties accessing services that should only be available internally
- Can also restrict access from inside to outside services

# Firewalls are not enough

---

- Cannot protect against attacks that bypass the firewall
  - E.g. Tunnelling
- Cannot protect against internal threats or insiders
  - Might help a bit by egress filtering
- Cannot protect against the transfer of virus-infected programs or files

# Firewall Types

---

- Packet Filters
  - Stateful Inspection
  - Proxies
  - Dynamic
  - Kernel
- 
- OSI-Based:
    - Application Layer
    - Network Layer

# Packet Filters

---

- Specify which packets are allowed or dropped
- Rules based on:
  - Source / Destination IP
  - TCP / UDP port numbers
- Possible for both inbound and outbound traffic
- Can be implemented in a router by only examining packet headers
  - Operates on OSI layer 3 (IP) or layer 4 (TCP)

# Packet Filter Rules

---

- Rule execution depends on implementation
  - IPTABLES: First rule to match is applied
  - PF: All rules are examined, the last match is applied
- Rules are organised in chains, which are logical subgroups of rules
- Depending on the packet, different chains are activated

# IPTABLES

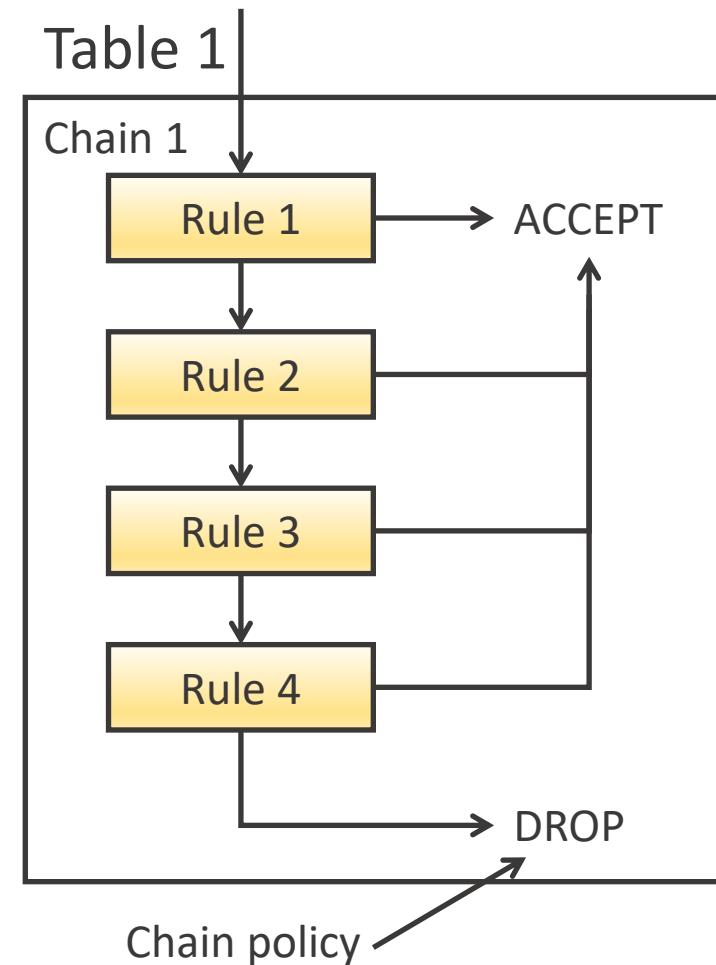
---

- An application that provides access to the Linux firewall rule tables
  - Not actually a firewall, but configures the firewall
  - The firewall is mostly implemented as *netfilter* modules



# Tables and Chains

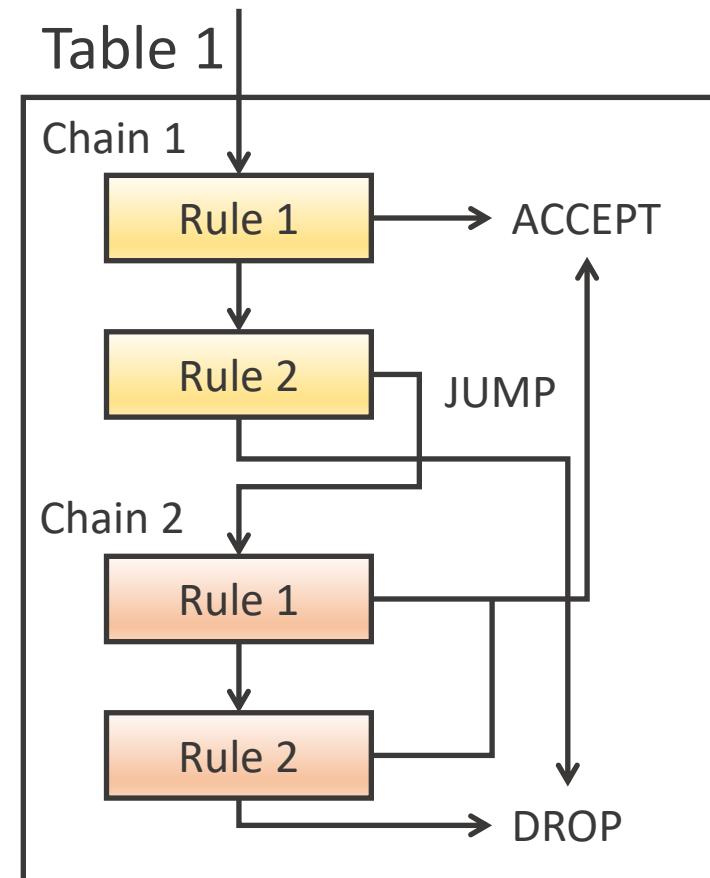
- IPTABLES uses tables to store chains
  - Default is the filtering table
- Chains are ordered lists of rules
  - Rules match, or they don't
- Matches result in a jump, else we check the next rule



# Tables and Chains

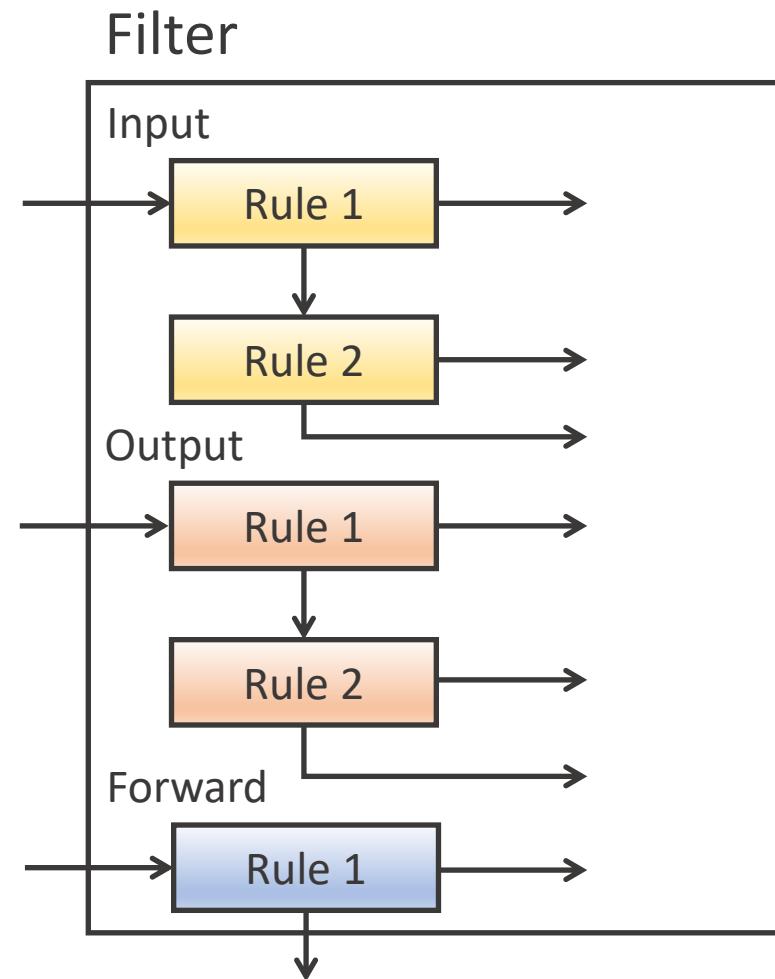
---

- There can be multiple chains per table
  - E.g. a TCP handling chain
- Jumps can go to ACCEPT, DROP, LOG or another chain
- Complex behaviour can be built up



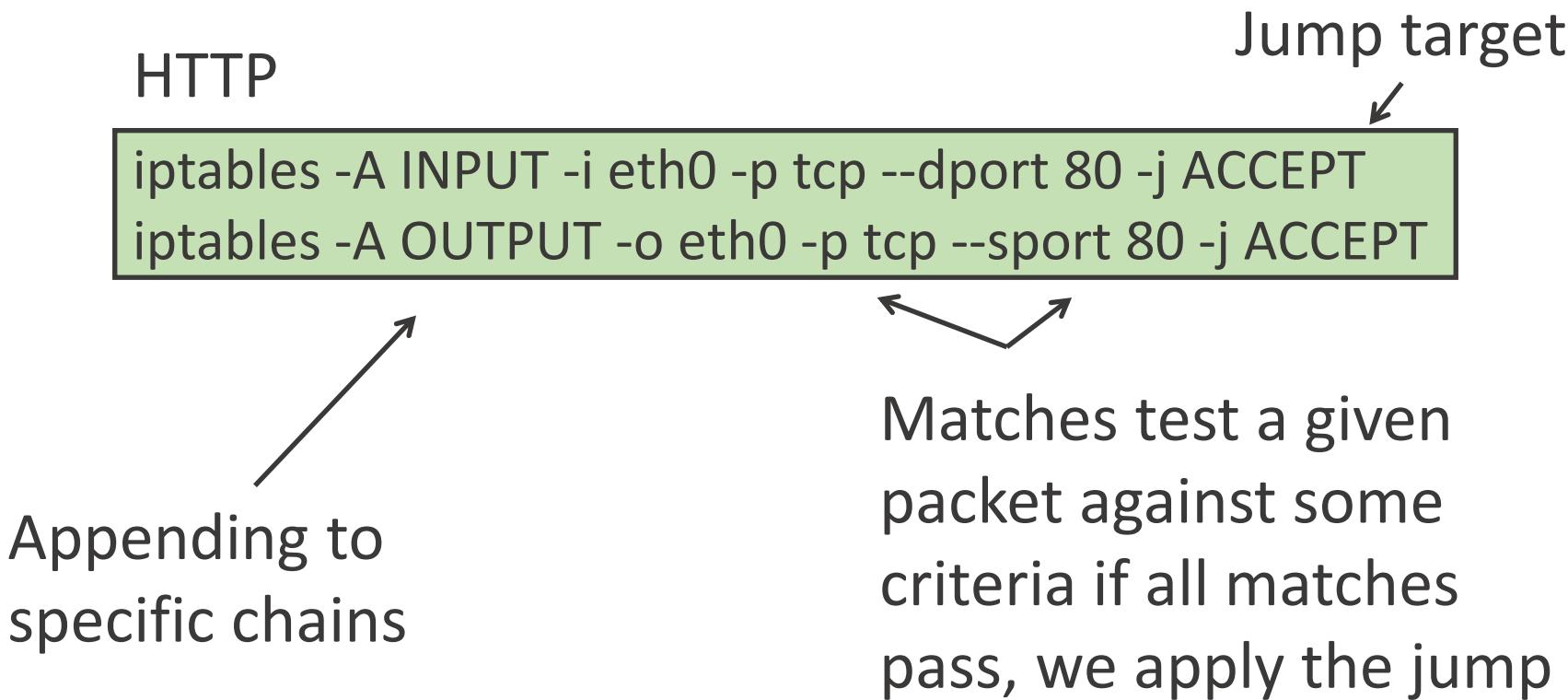
# Defaults

- There are built-in tables in IPTABLES:
  - Filter (default)
  - NAT
  - Mangle – Packet alteration
  - Raw – Skips connection tracking
  - Security – mandatory access control
- The default table is the filtering table, including Input, Output and Forward chains



# Rules Examples

- Using the command line, we add rules onto the end of chains



# Policies

---

- Permissive (Black listing) – allow everything except dangerous services
  - Easy to make a mistake or forget something
- Restrictive (White listing) – block everything except designated useful services
  - More secure by default
  - Fairly easy to DoS yourself!

# IPTABLES Policies

---

- To use a blacklisting policy, we want to accept by default, then have rules that drop:

```
iptables -P INPUT ACCEPT  
iptables -P FORWARD ACCEPT  
iptables -P OUTPUT ACCEPT
```

```
iptables -A INPUT -s X.X.X.X -j DROP  
iptables -A OUTPUT -p tcp --dport ssh -j DROP
```

- For a whitelisting policy, we want to drop by default, then let certain packets through:

```
iptables -P INPUT DROP  
iptables -P FORWARD DROP  
iptables -P OUTPUT DROP
```

```
iptables -A INPUT -p tcp --dport ssh -j ACCEPT  
iptables -A OUTPUT -s 192.168.0.2 -j ACCEPT
```

# Packet Filter Issues

---

- Packet filters are simple, low level and have high assurance
- But, they cannot:
  - Prevent attacks that exploit application-specific vulnerabilities
  - Do not support higher-level authentication schemes

# Stateful Packet Filters

---

- A stateful firewall always keeps track of the state of network connections.
- Once a particular kind of traffic has been approved by a stateful firewall, it is added to a state table/connection table.
- Understand requests and replies  
E.g. 3-way Handshake: SYN, SYN/ACK, ACK

# Connection Table Example

---

Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.0.2	1030	210.9.88.29	80	established
192.168.0.4	22	216.32.42.123	22	established

---

- ACK packets are used to keep track of the session – the connection is ongoing
- Packets without the ACK are the connection establishment messages

# IPTABLES Rules

---

- IPTABLES has modules for stateful packet filtering
- Allow incoming / outgoing SSH connections

```
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT  
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

- Allow HTTP(S):

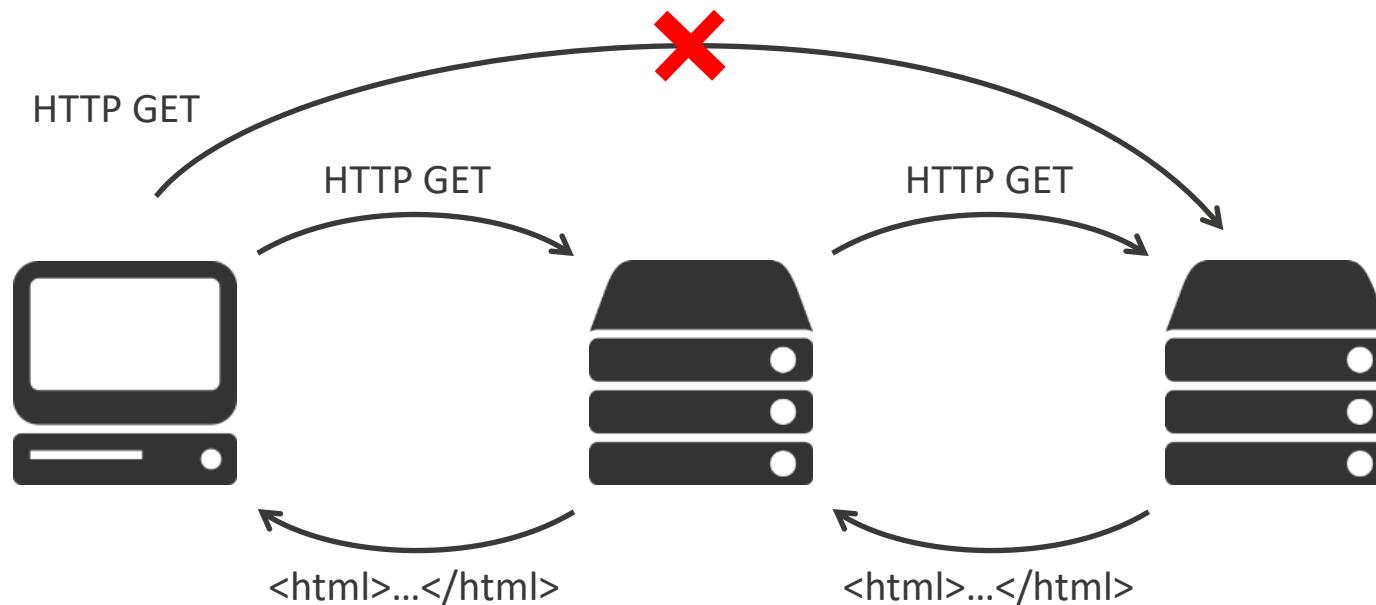
```
iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT  
iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

```
iptables -A INPUT -i eth0 -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT  
iptables -A OUTPUT -o eth0 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
```

# Proxy Servers

---

- Proxy servers initiate a connection on our behalf
- They can block certain access, and scan for malicious files or web pages



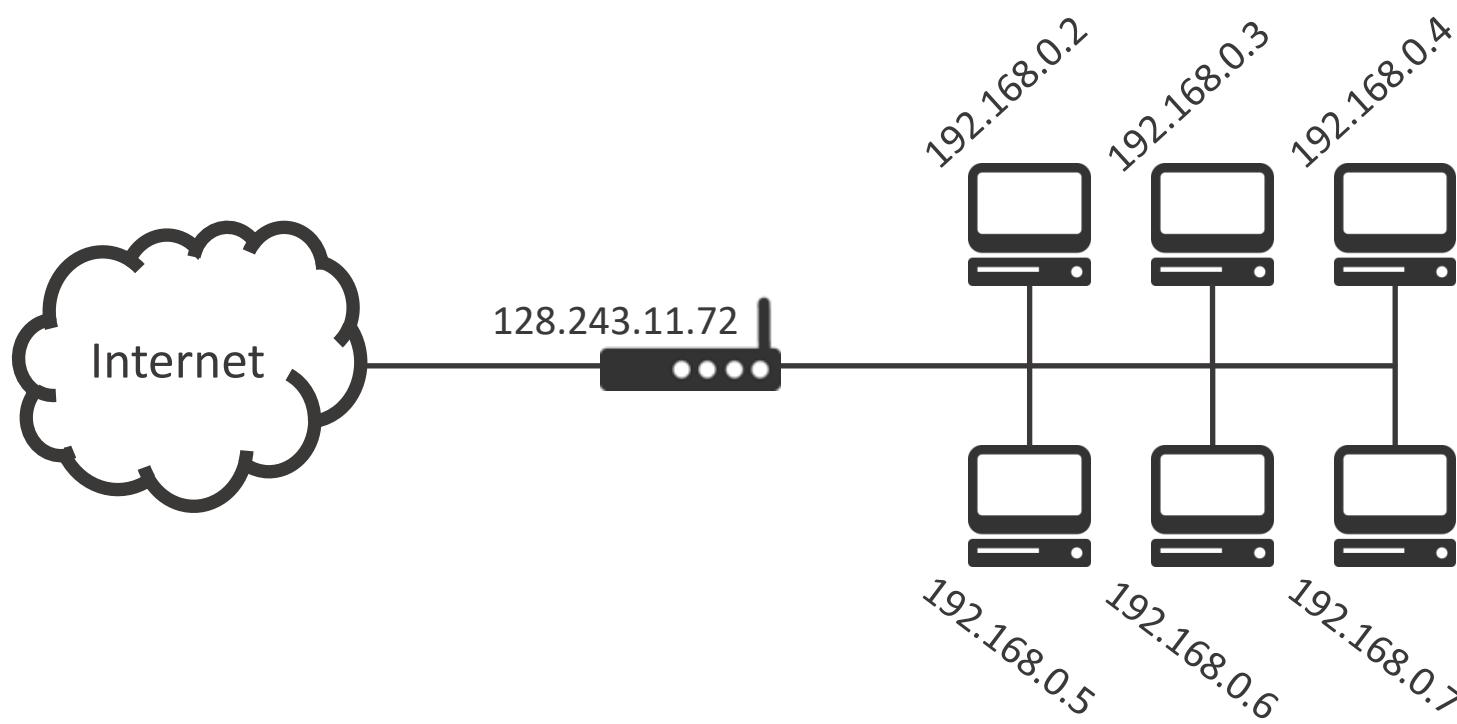
# Proxy Servers

---

- Issues
  - Large overhead per connection
  - More expensive than packet filtering
  - Configuration is complex
  - A separate server is required for each service

# Network Address Translation (NAT)

- The shortage of IP addresses mean that most routers now perform NAT automatically



# Network Address Translation

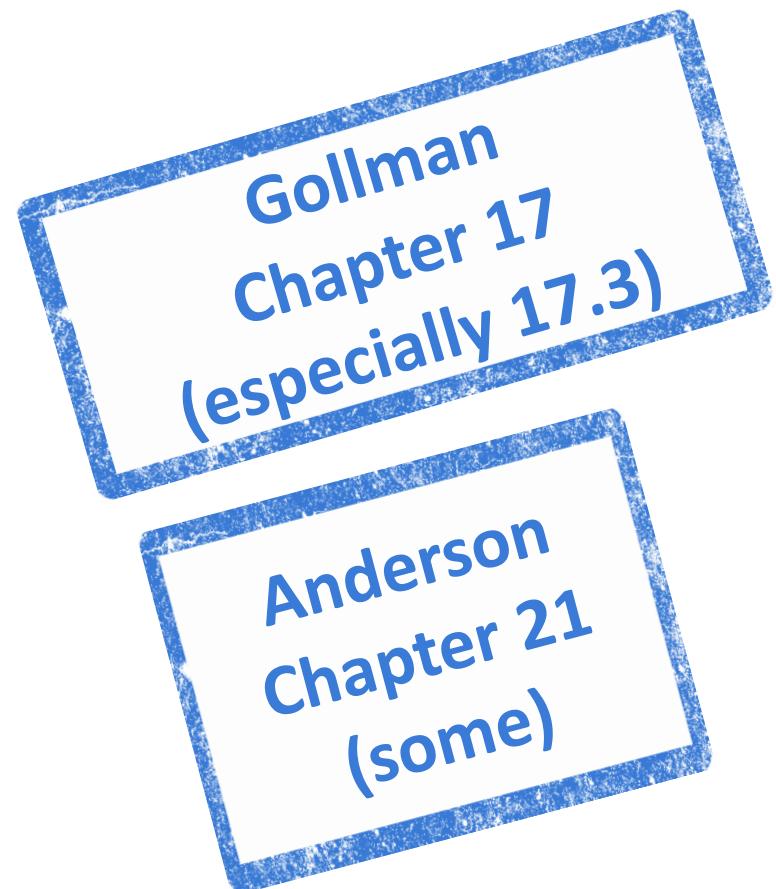
---

- The implicit advantage in NAT is that your machine is almost totally hidden from the internet
- Only established connections are forwarded to your internal machine
  - Or, specific port forwarding rules

# Summary

---

- Firewalls
  - Packet Filters
  - Stateful vs Stateless
- Proxies
- NAT



# COMP3052.SEC Computer Security

## Session 07: Reference Monitors

```
ide1: BM-DMA at 0xc008-0xc00f, BIOS settings: hdc:pio, hdd:pio
ne2k-pci.c:v1.03 9/22/2003 D. Becker/P. Gortmaker
    http://www.scyld.com/network/ne2k-pci.html
hda: QEMU HARDDISK, ATA DISK drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
hdc: QEMU CD-ROM, ATAPI CD/DVD-ROM drive
ide1 at 0x170-0x177,0x376 on irq 15
ACPI: PCI Interrupt Link [LNKC] enabled at IRQ 10
ACPI: PCI Interrupt 0000:00:03.0[A] -> Link [LNKC] -> GSI 10 (level, low) -> IRQ
    10
eth0: RealTek RTL-8029 found at 0xc100, IRQ 10, 52:54:00:12:34:56.
hda: max request size: 512KiB
hda: 180224 sectors (92 MB) w/256KiB Cache, CHS=178/255/63, (U)DMA
hda: set_multmode: status=0x41 { DriveReady Error }
hda: set_multmode: error=0x04 { DriveStatusError }
ide: failed opcode was: 0xef
hda: cache flushes supported
    hda1
hdc: ATAPI 4X CD-ROM drive, 512kB Cache, (U)DMA
Uniform CD-ROM driver Revision: 3.20
Done.
Begin: Mounting root file system... ...
/init: /init: 151: Syntax error: 0xforce=panic
Kernel panic - not syncing: Attempted to kill init!
```

# Acknowledgements

---

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towey, ...

# This Session

---

- Reference Monitors
- Operating System Integrity
- Privilege Elevation
- Memory Protection
- Page Tables

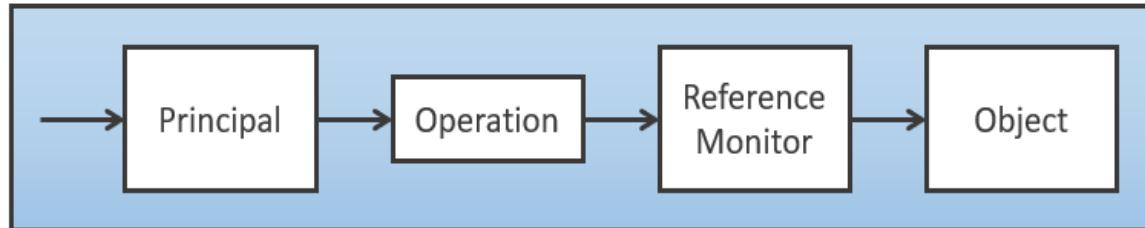
# Concepts

---

- The Reference Monitor
  - An abstract concept
- Security Kernel
  - The implementation of a reference monitor
- Trusted Computing Base (TCB)
  - Kernel + other protection measures

# Reference Monitor

---



“An access control concept that refers to an abstract machine that mediates all access to objects by subjects”

- Must be tamper proof / resistant
- **Must always be invoked** when access to an object is required
- Must be small enough to be verifiable / subject to analysis to ensure correctness

# Security Kernel

---

“The hardware, firmware and software elements of a TCB that implement the reference monitor”

- Mediates all access
- Must be protected from modification
- Must be verifiably correct
- Usually in the bottom layers of a system

# Trusted Computing Base

---

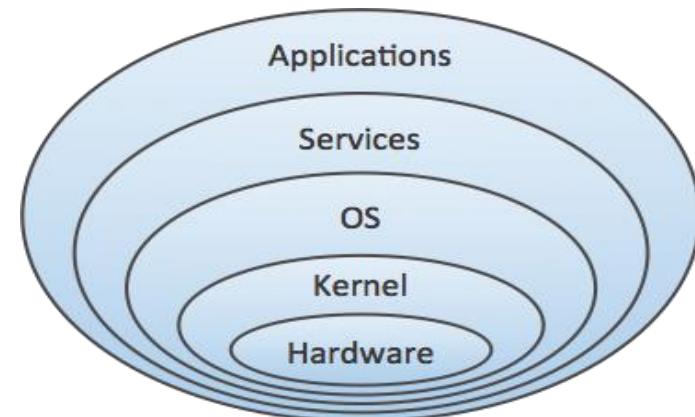
*“The totality of protection mechanisms within a computer system responsible for enforcing a security policy”*

- One or more components
- Enforce a unified security policy over a product or system
- Correct enforcement depends on components within as well as input from administrators

# Placement

---

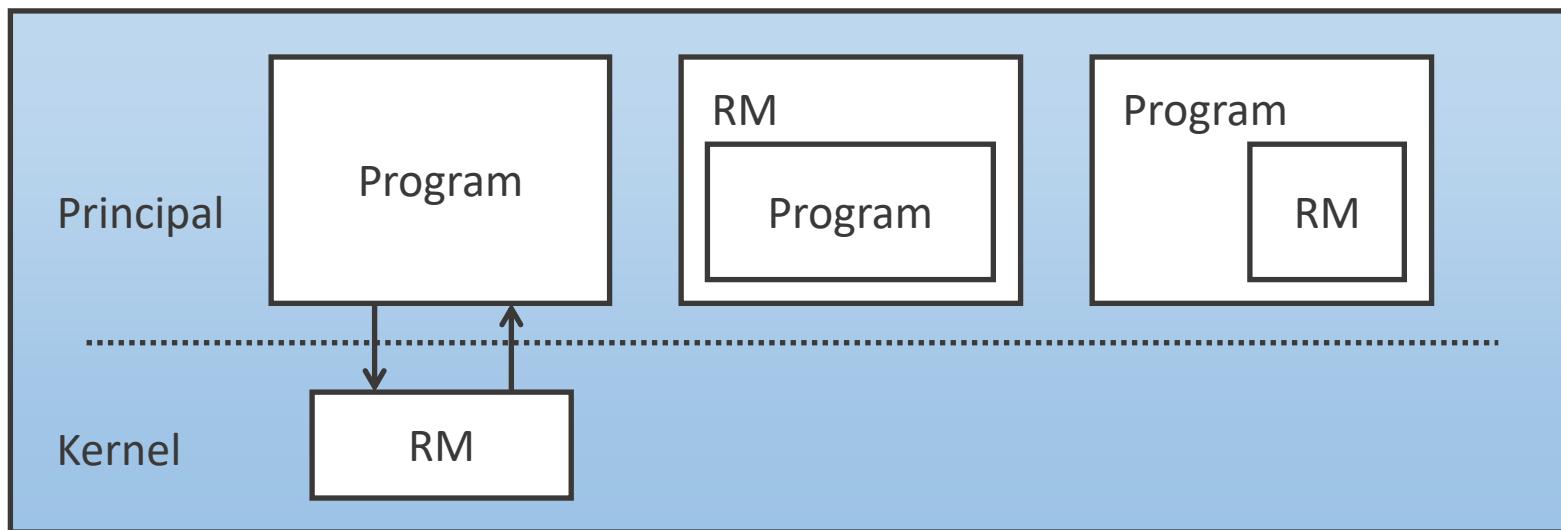
- Can be placed anywhere within a system
  - Hardware – Dedicated registers for defining privileges
  - Operating system kernel – E.g. Virtual Machine Hypervisor
  - Operating system – Windows security reference monitor
  - Services Layer – JVM, .NET
  - Application Layer – Firewalls



# Placement

---

- Reference monitors could be placed in a variety of locations relative to the program being run



# Lower Is Better

---

- Using a reference monitor or other security features at a lower level means:
  - We can assure a higher degree of security
  - Usually simple structures to implement
  - Reduced performance overheads
  - Fewer layer below attack possibilities
- However:
  - Access control decisions are far remote from applications

# OS Integrity

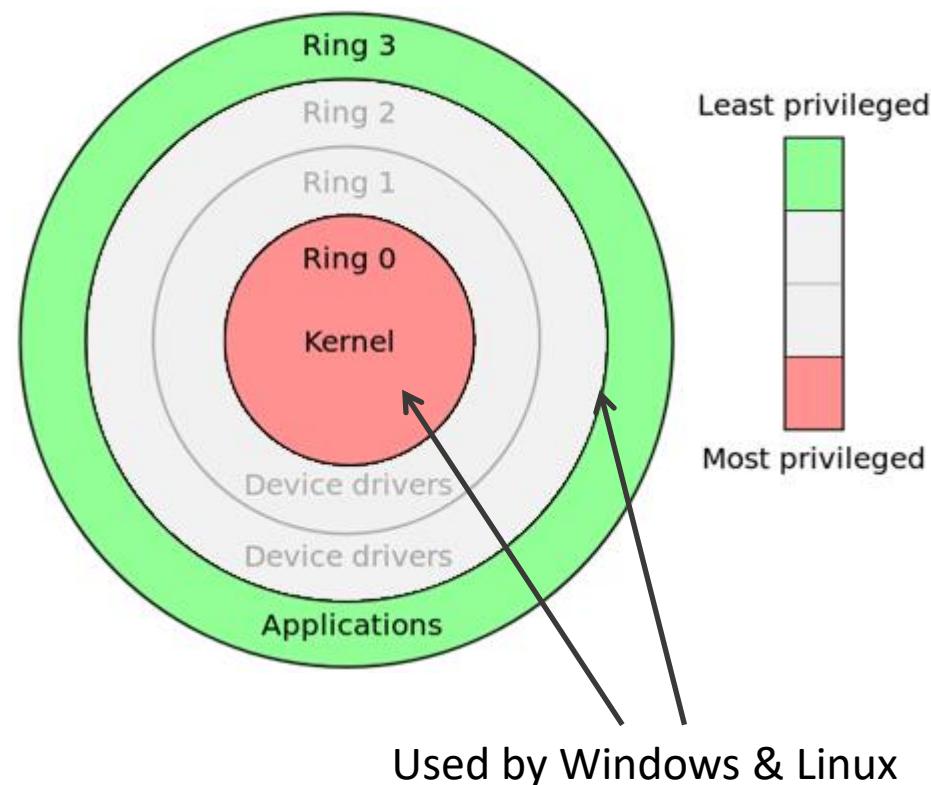
---

- The operating system:
  - Arbitrates access requests
  - Is itself a resource that must be accessed
- This is a conflict, we want to use the OS but not mess with it  
*“Users must not be able to modify the operating system”*
- Modes of operation
  - Defines which actions are permitted in which mode, e.g. system calls, machine instructions, I/O
- Controlled Invocation
  - Allows us to execute privileged instructions safely, before returning to user code

# Modes of Operation

---

- Distinguish between computations done on behalf of:
  - The OS
  - The user
- A status flag within the CPU allows the OS to operate in different modes



# Controlled Invocation

---

- Many functions are held at kernel level, but are quite reasonably called from within user level code
  - Network and File IO
  - Memory allocation
  - Halting the CPU (at shutdown only!)
- We need a mechanism to transfer between kernel mode (ring 0) and user mode (ring 3)



# The Key Point

---

**We** don't actually perform privileged operations: we ask the OS to perform them for us

The OS can refuse to do it!

# Controlled Invocation: Interrupts

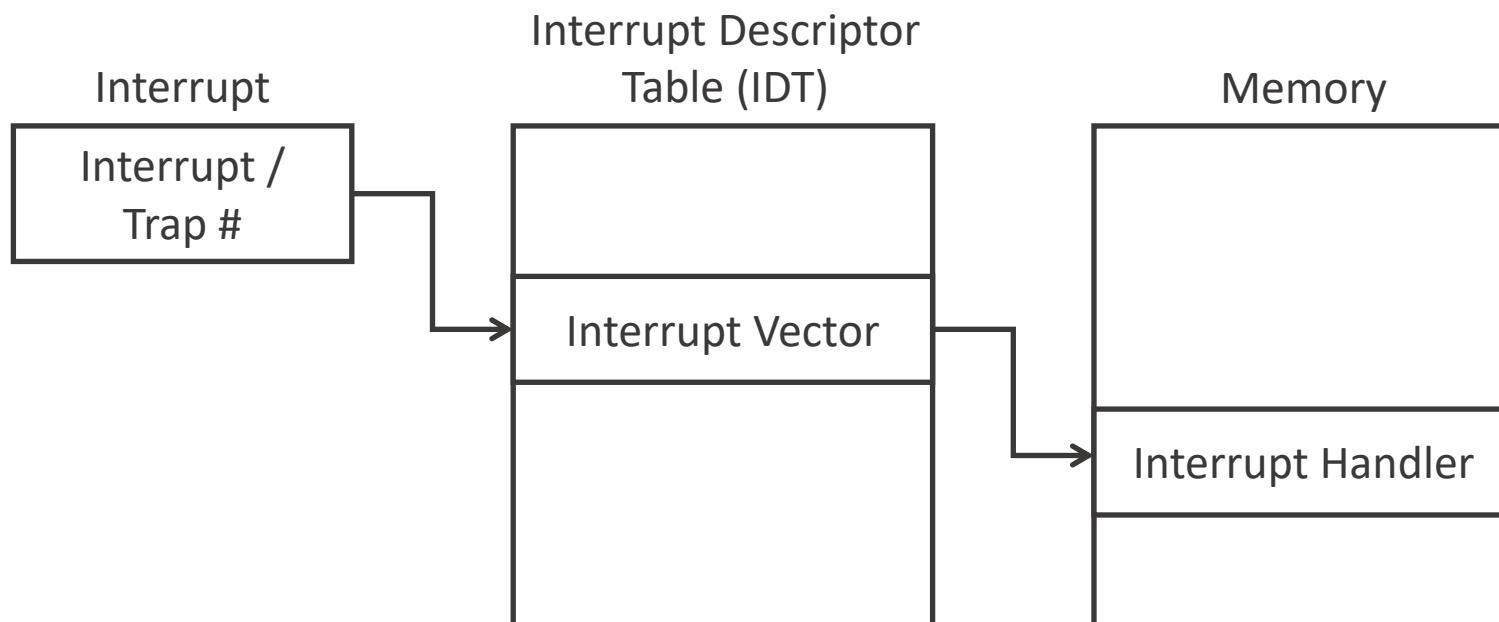
---

- Exceptions / Interrupts / Traps
  - Called various things, for now we'll just use "Interrupt"
  - In many ways is the hardware equivalent to a software exception
- Handled by an interrupt handler which resolves the issue and returns to the original code

# Processing an Interrupt

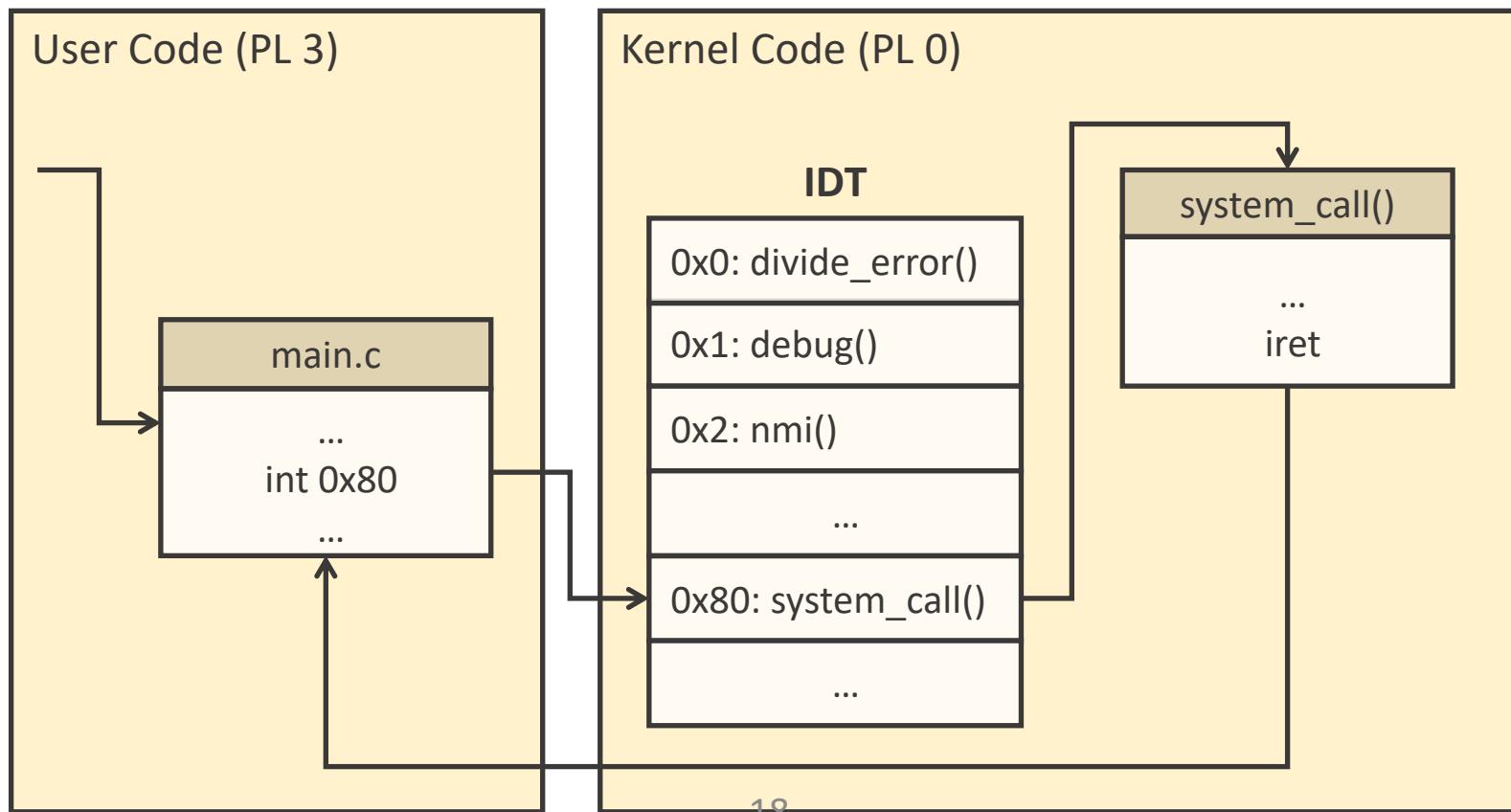
---

- Given an interrupt, the CPU will switch execution to the location given in an interrupt descriptor table (IDT)



# Privilege Elevation in x86-Linux

- Linux initialises its IDT to handle syscalls at vector 0x80



# Interrupt Example

---

```
#include <unistd.h>

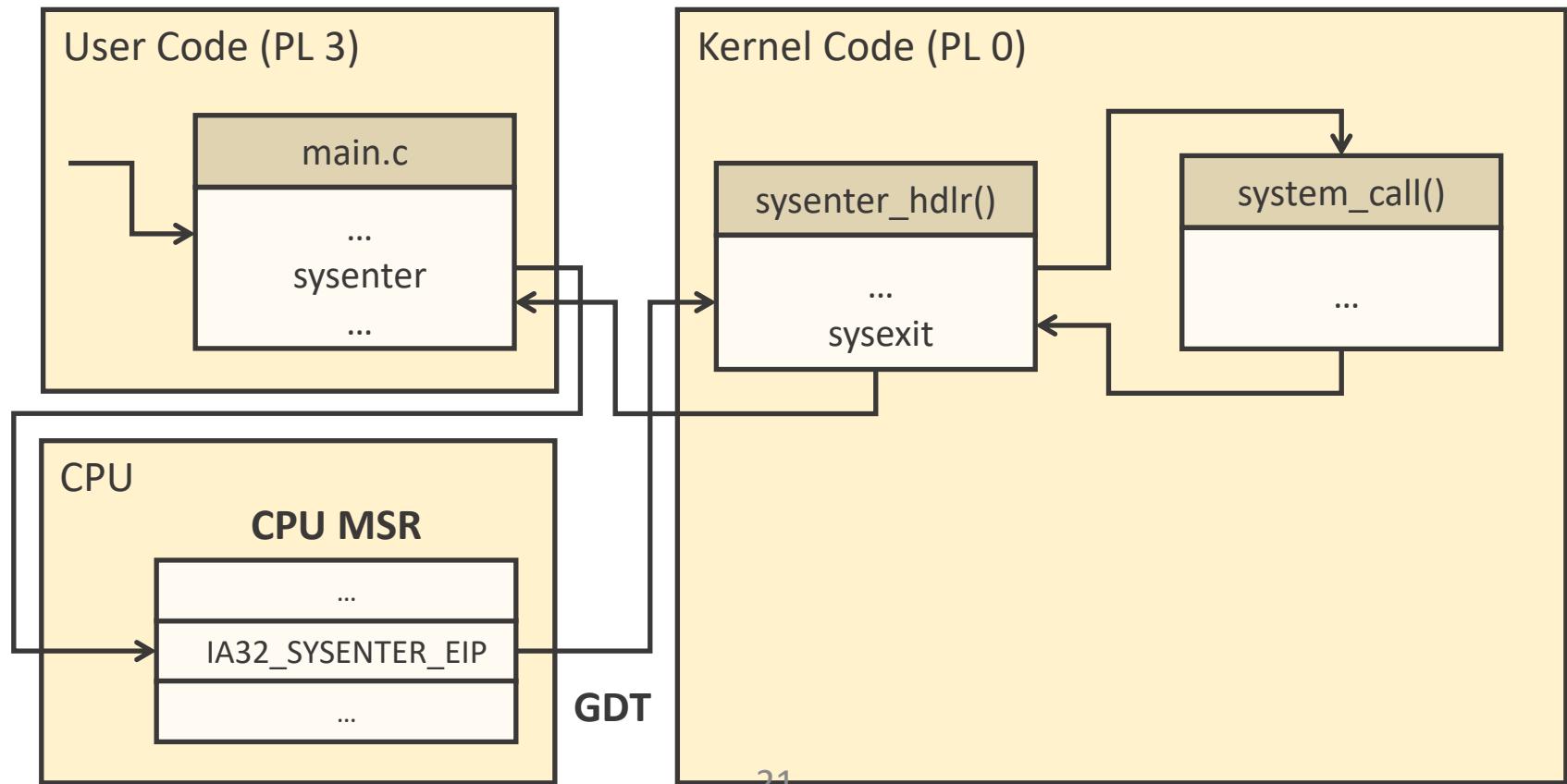
int main(int argc, char *argv[])
{
    write(1, "Hello!", 6);
    _exit(0);
}
```

```
mov    $4, %eax
mov    $1, %ebx
mov    $msg, %ecx
mov    $6, %edx
int    $0x80

mov    $1, %eax
mov    $0, %ebx
int    $0x80
```

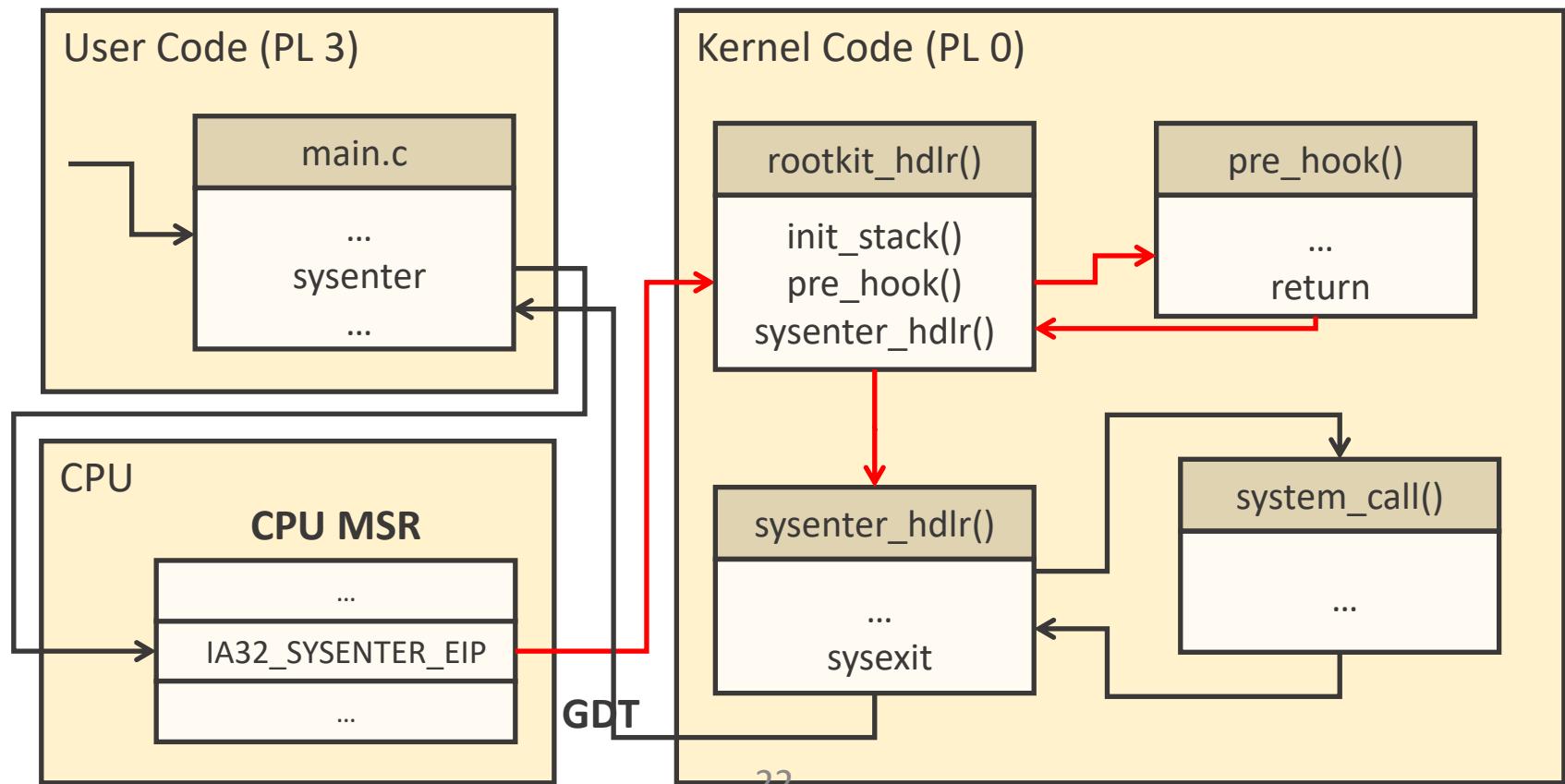
# Modern Kernels

- Intel introduced the `sysenter` and `sysexit` operations with the Pentium II – much less overhead



# Patching the Kernel

- If you can run custom PL 0 code (compromised driver?), you can insert your own handler – **Rootkit**



# Processes and Threads

---

- A process is a program being executed
- Important unit of control:
  - Exists in its own address space
  - Communicates with other processes via the OS
  - Separation for security
- A Thread is a strand of execution within a process
  - Share a common address space

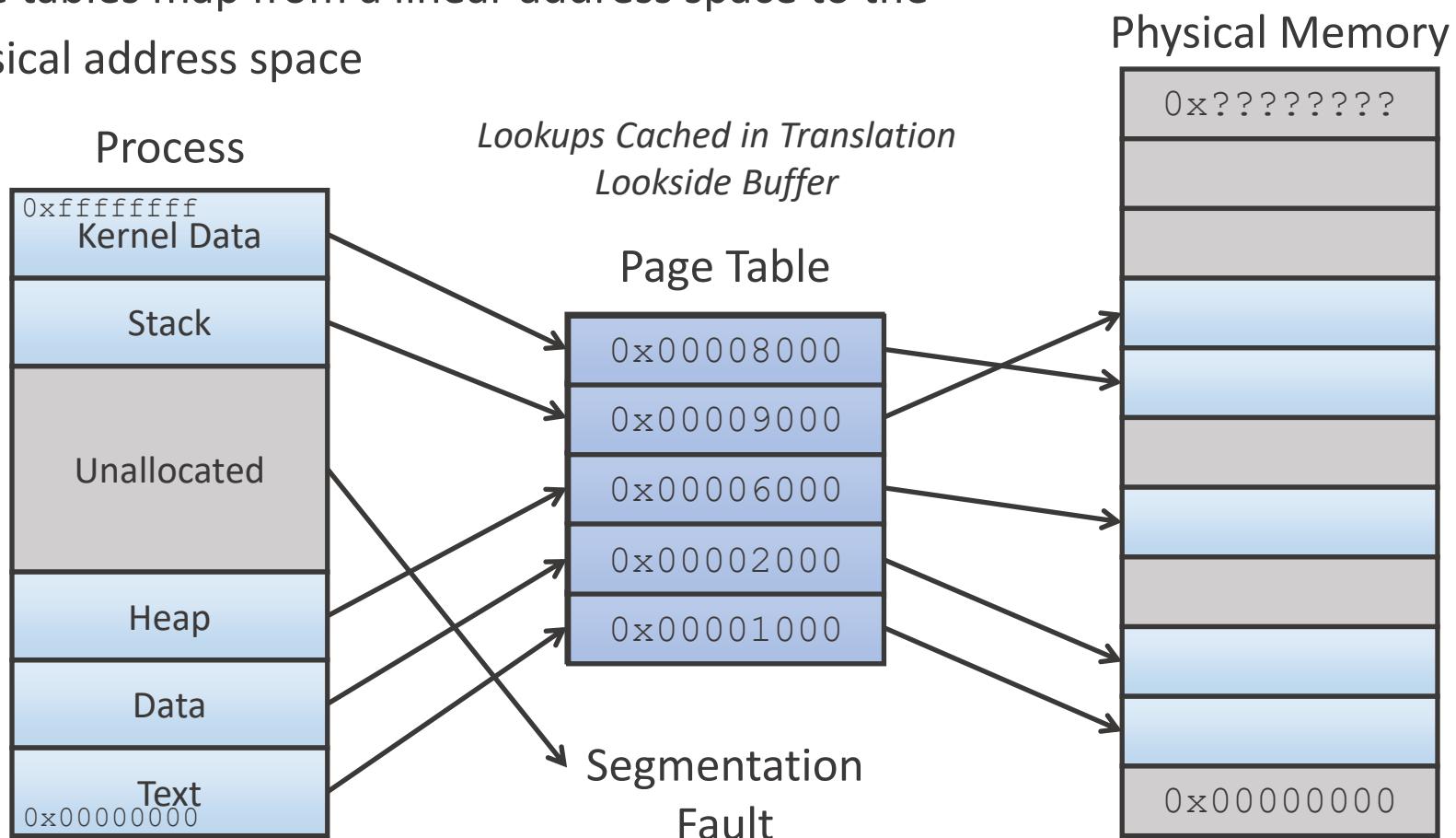
# Memory Protection

---

- Segmentation – divides data into logical units
  - Good for security
  - Challenging memory management
  - Not used much in modern OSs
- Paging – divides memory into pages of equal size
  - Efficient memory management
  - Less good for access control
  - Extremely common in modern OSs

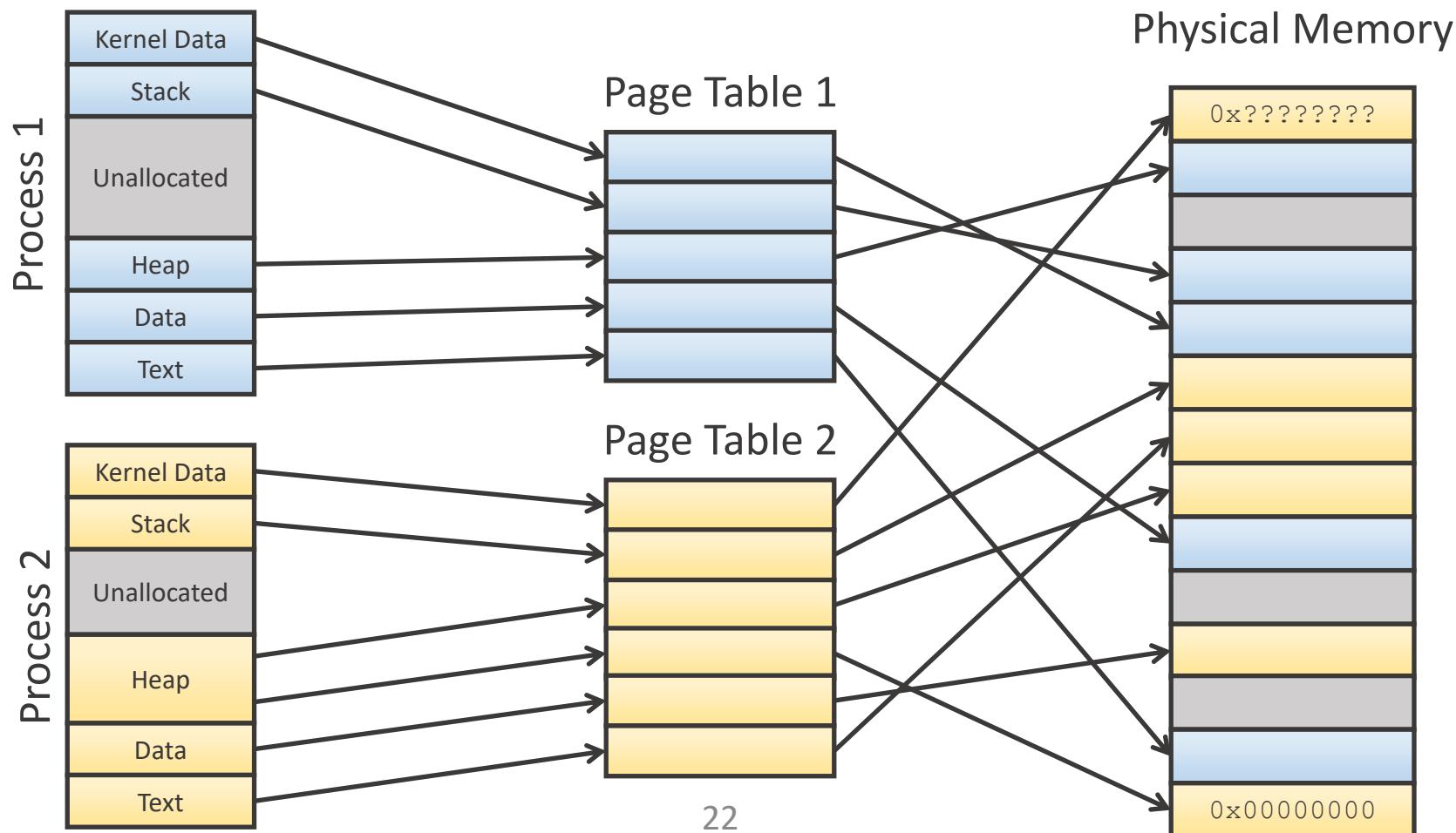
# Page Tables

- All processes see an individual linear address space
- Page tables map from a linear address space to the physical address space



# Page Tables

- Processes are separated by the page system



# Paging

---

- Page Tables have a valid / invalid bit
  - Valid pages have page numbers allocated to the currently executing process
  - Invalid pages are either non-existent (not in the page table) or are in the page table but belong to other processes
- Memory access to an invalid page results in a segmentation / page fault or bus error
  - Trap causes context switch to kernel
  - Kernel sends SIGSEGV or SIGBUS to process
    - Usual behaviour is for process to end

# Protecting Memory

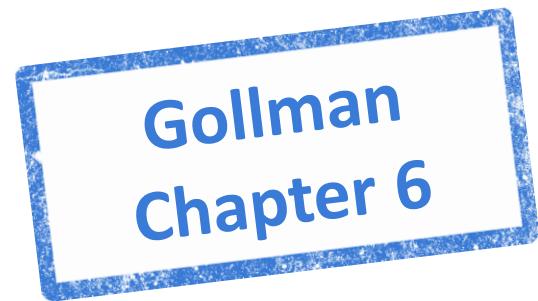
---

- OS Integrity – preserved by separation of users and kernel space
- Separation of users
  - File management – logical memory object
  - Memory management – physical memory object

# Summary

---

- Reference Monitors
- Operating System Integrity
- Privilege Elevation
- Memory Protection
- Page Tables



# COMP3052.SEC Computer Security

## Session 08: Network Security



# Acknowledgements

---

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towey, ...

# This Session

---

- TCP/IP
- IPSec
- ARP Cache Poisoning
- DNS Spoofing
- Denial of Service Attacks

# Two Threat Models

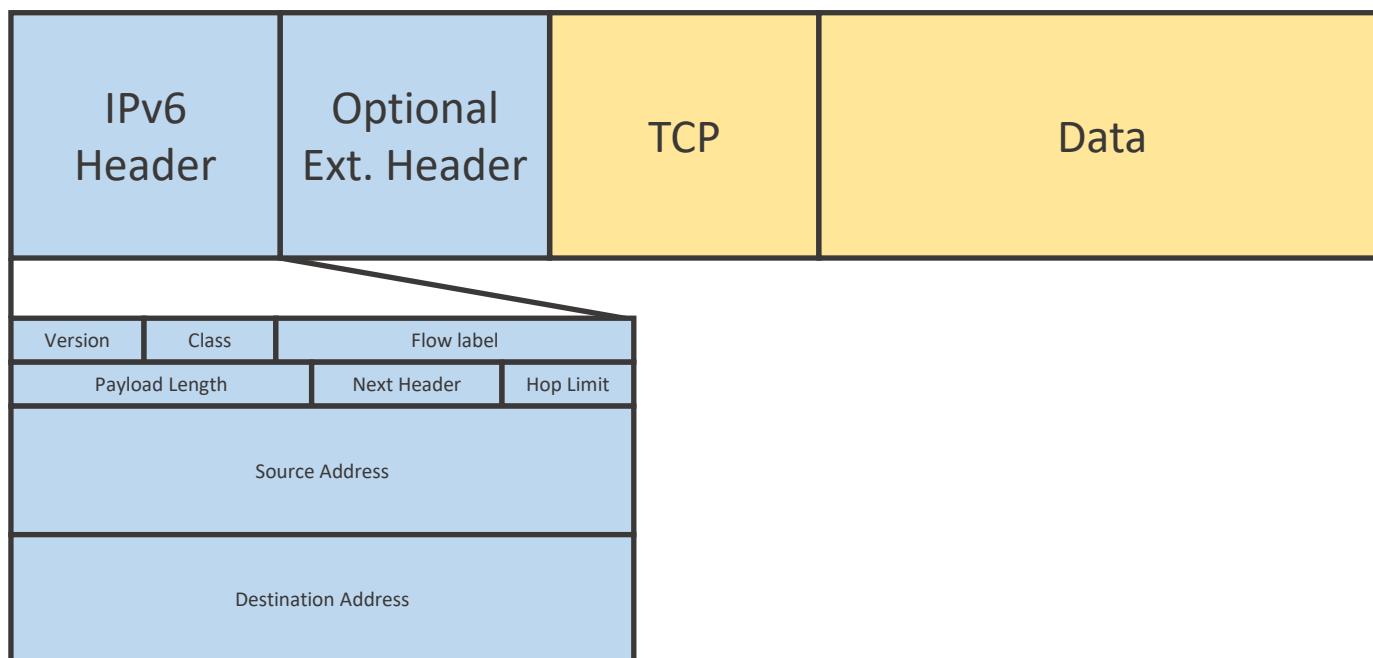
---

- Passive attackers
  - Eavesdropping / wiretapping / sniffing
  - Traffic analysis
- Active attackers
  - Spoofing attacks (phishing, email)
  - Squatting attacks

# TCP/IP - Nesting Headers

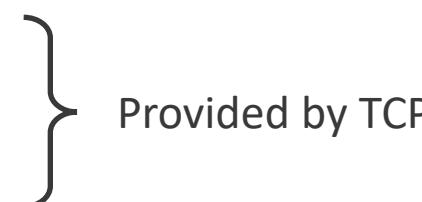
---

- Each protocol carries the protocol in the layer above by appending headers to it



# IP Security

---

- IP is connectionless and stateless
    - Best effort service
    - No delivery guarantee
    - No order guarantee
  - IPv4 No guaranteed security support
  - IPv6 security support is guaranteed - IPSec
- 
- Provided by TCP

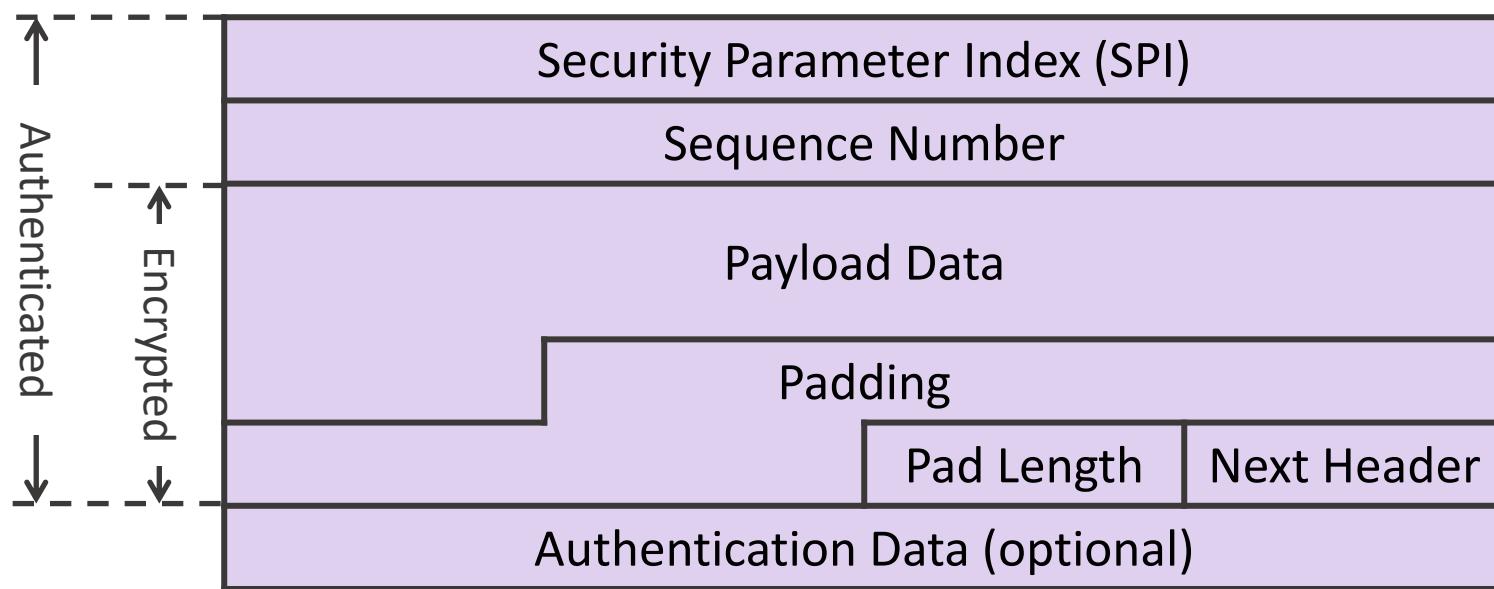
# IPSec

---

- Optional in IPv4, mandatory support in IPv6
- Two major security mechanisms
  - IP Authentication Header (AH)
  - IP Encapsulating Security Payload (ESP)
- Does not contain any mechanisms to prevent traffic analysis

# Encapsulating Security Payload (ESP)

- Includes an additional header within the IP packet that describes what encryption and authentication is in use



# Transport vs Tunnel Modes

---

- Transport mode simply encrypts packets, providing host-to-host encryption but using the original header
- Prevents contents being read, but doesn't stop traffic analysis or manipulation of the header



# Transport vs Tunnel Modes

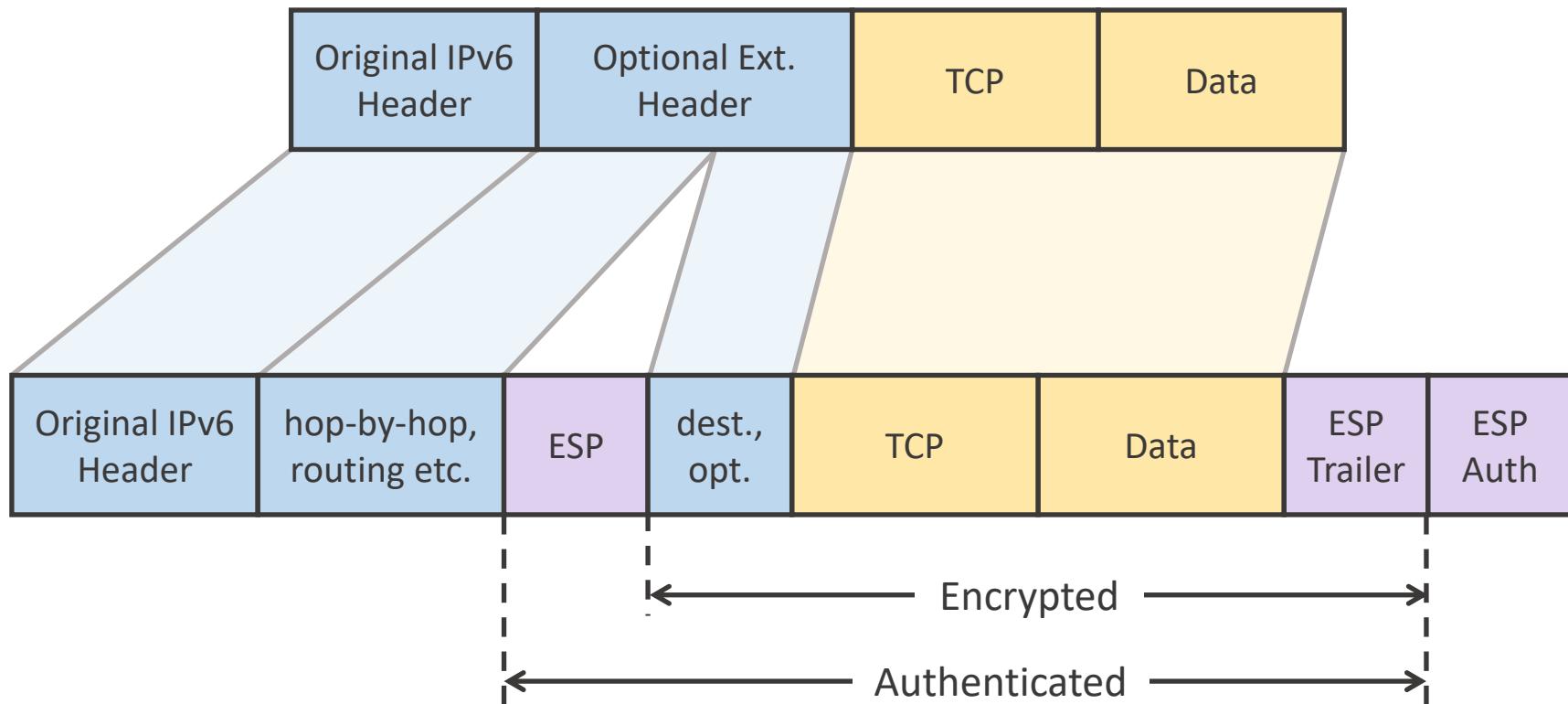
---

- Tunnel mode (usually gateway-to-gateway) protects some segment of a channel with encryption
- Provides some resistance to traffic analysis, and completely protects manipulation of the payload
- VPNs are commonly implemented this way

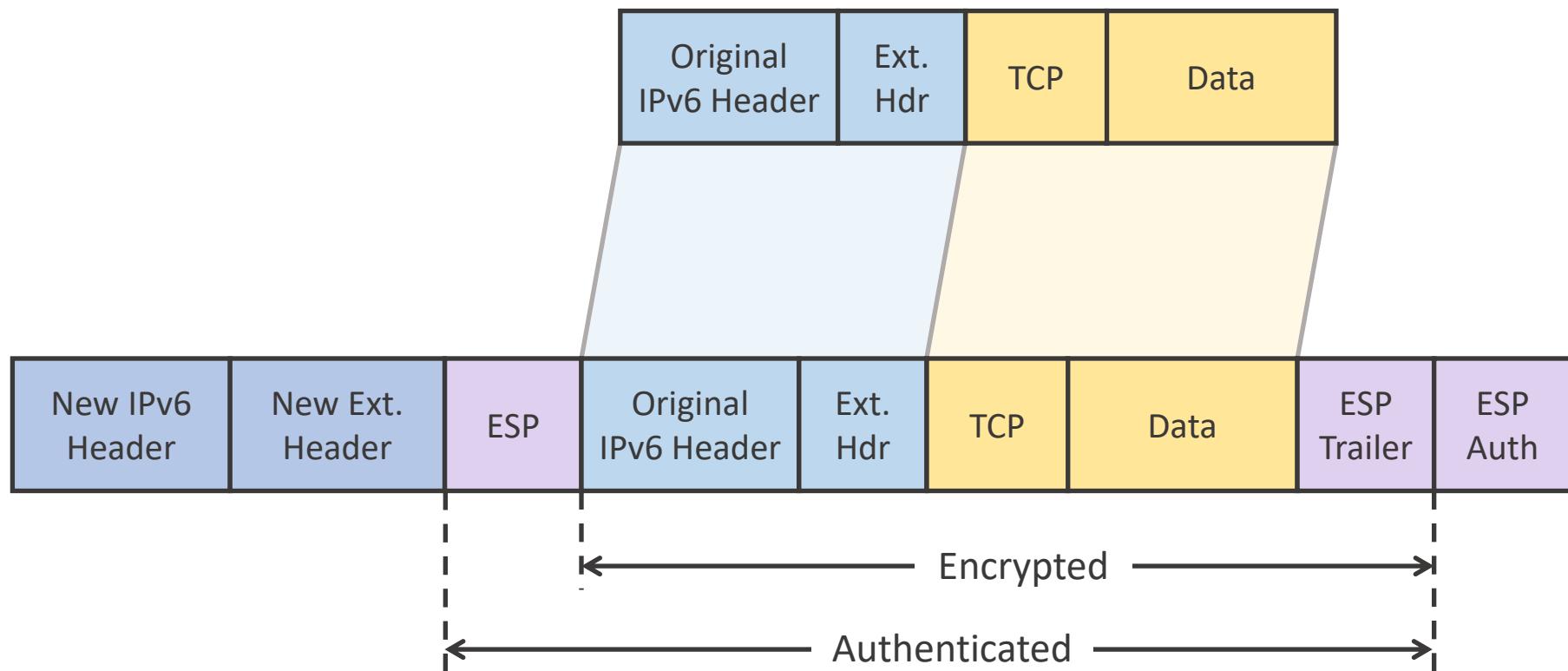


# ESP in Transport Mode

- ESP uses either Transport or Tunnel modes

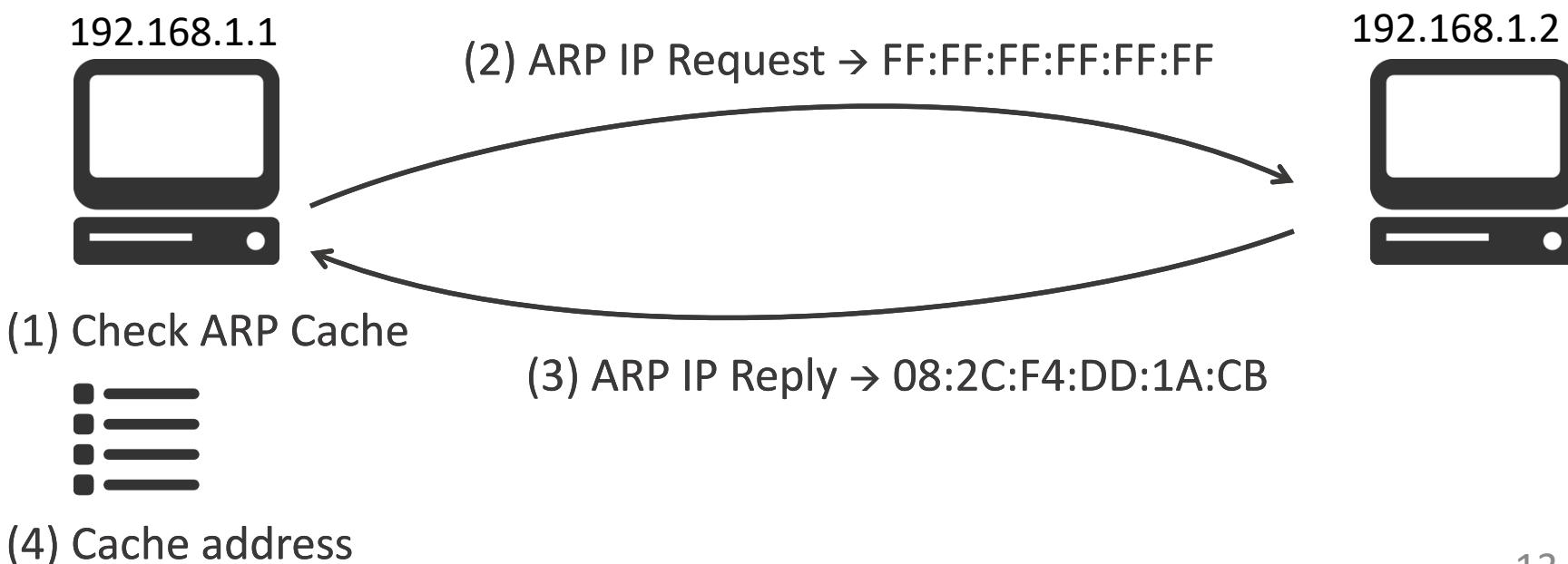


# ESP in Tunnel Mode



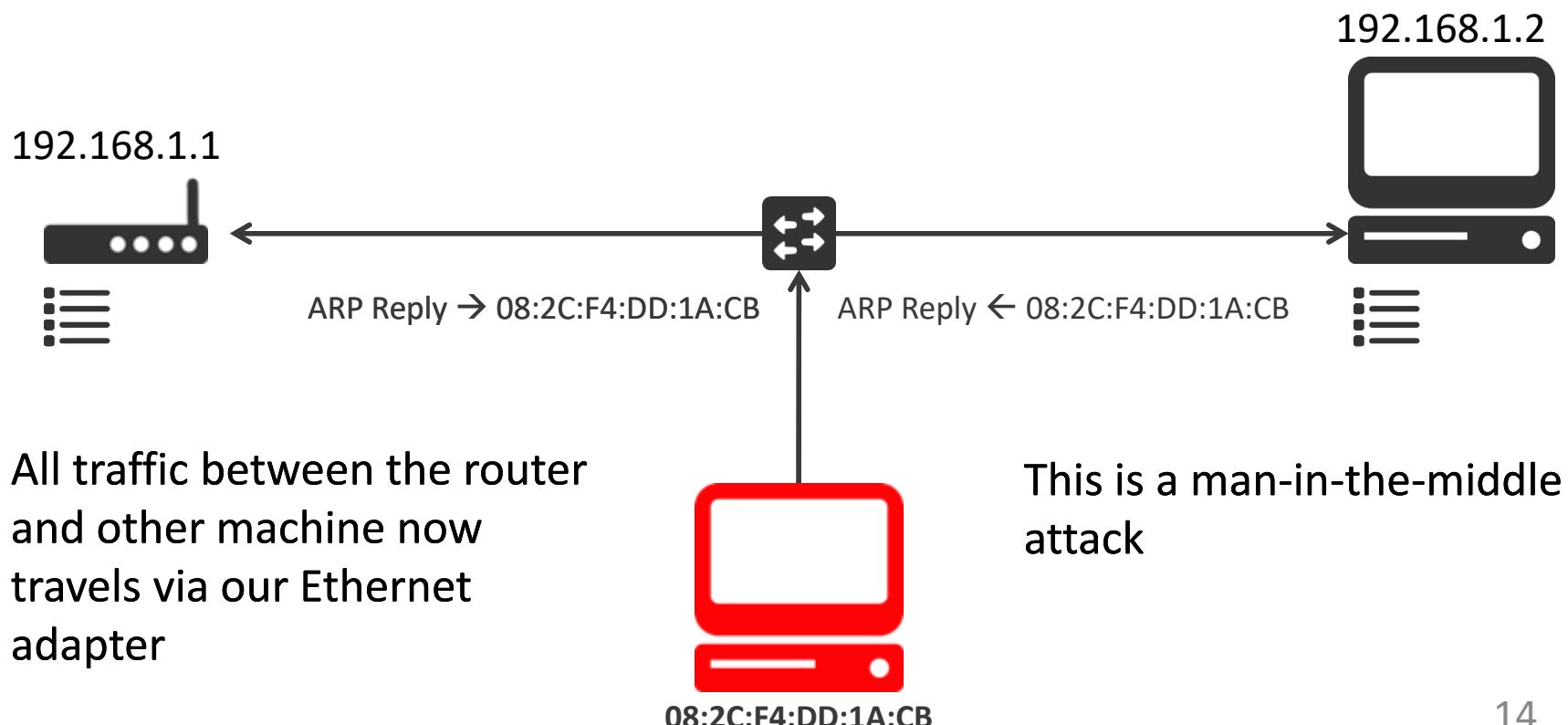
# Address Resolution Protocol (ARP)

- ARP is a protocol used (in IPv4) to obtain physical MAC addresses for given IPs
  - It is used prior to constructing IP and TCP packets for communication
  - Network layer



# ARP Cache Poisoning

- We can simply send an unrequested ARP reply, and overwrite the MAC address in a host's ARP cache with our own



# ARP Cache Poisoning

The screenshot shows the ettercap 0.8.2 interface. The menu bar includes Start, Targets, Hosts, View, Mitm, Filters, Logging, Plugins, and Info. The 'Mitm' tab is selected, revealing a dropdown menu with options: ARP poisoning..., ICMP redirect..., Port stealing..., DHCP spoofing..., NDP poisoning..., and Stop mitm attack(s). Below the menu is a table titled 'Host List' with columns for IP Address and MAC Address. Three hosts are listed: 192.168.0.1 (BC:AE:C5:EB:F3:B0), 192.168.0.2 (50:46:5D:51:4C:0A), and 192.168.0.3 (E4:11:5B:13:4C:0A). Buttons for 'Delete Host', 'Add to Target 1', and 'Add to Target 2' are present. The main window displays 'ARP poisoning victims:' followed by two groups of hosts. Group 1 contains 192.168.0.1 BC:AE:C5:EB:F3:B0. Group 2 contains 192.168.0.125 08:00:27:59:B3:9C. A message 'Unified sniffing started...' is shown, along with several captured sessions: an FTP session from 82.71.204.24:21 to a user and password, another FTP session to the same user and password, and an HTTP session to http://[REDACTED]/members/memberlogin.php with content showing 'username=[REDACTED]&password=[REDACTED]'. The interface has a light gray background with dark gray toolbars and a white main content area.

ettercap 0.8.2

Start Targets Hosts View **Mitm** Filters Logging Plugins Info

Host List x

IP Address MAC Address

192.168.0.1	BC:AE:C5:EB:F3:B0
192.168.0.2	50:46:5D:51:4C:0A
192.168.0.3	E4:11:5B:13:4C:0A

ARP poisoning... ICMP redirect... Port stealing... DHCP spoofing... NDP poisoning... Stop mitm attack(s)

Delete Host Add to Target 1 Add to Target 2

ARP poisoning victims:

GROUP 1 : 192.168.0.1 BC:AE:C5:EB:F3:B0

GROUP 2 : 192.168.0.125 08:00:27:59:B3:9C

Unified sniffing started...

FTP : 82.71.204.24:21 -> USER: [REDACTED] PASS: [REDACTED]

FTP : 82.71.204.24:21 -> USER: [REDACTED] PASS: [REDACTED]

HTTP : 82.71.204.24:80 -> USER: [REDACTED] PASS: [REDACTED] INFO: http://[REDACTED]/members/memberlogin.php

CONTENT: username=[REDACTED]&password=[REDACTED]

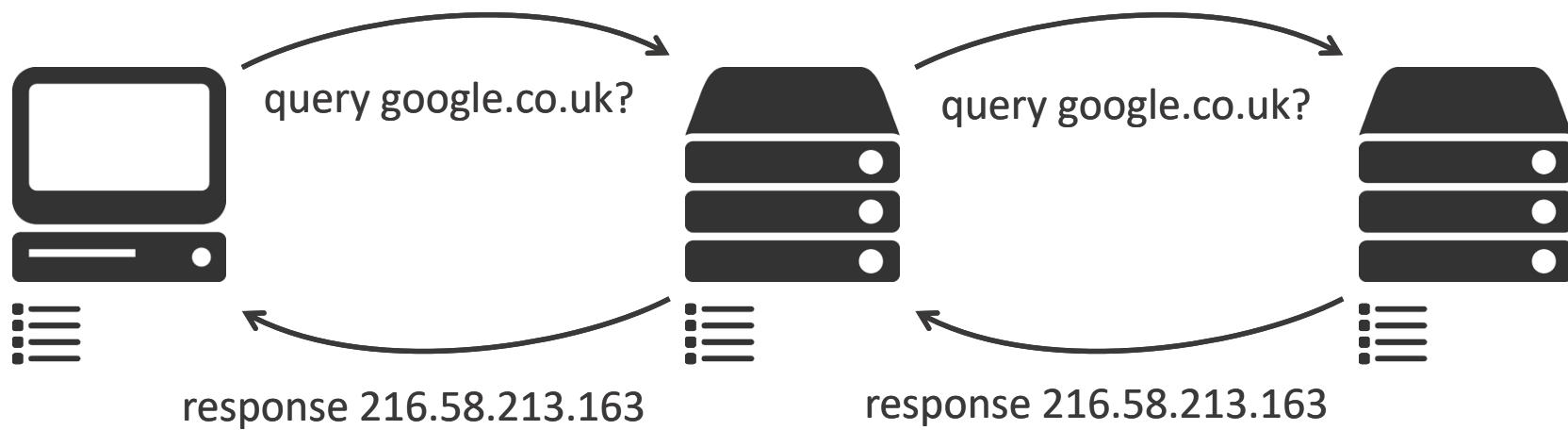
# ARP Protection

---

- Some OSs ignore unsolicited ARP requests, or can be configured to use ARP differently
- Some software, such as intrusion detection packages, will include ARP spoofing detection
  - Maintain a log of current MAC:IP assignments and ARP requests / replies
  - Allows us to spot suspicious messages such as unsolicited ARP replies

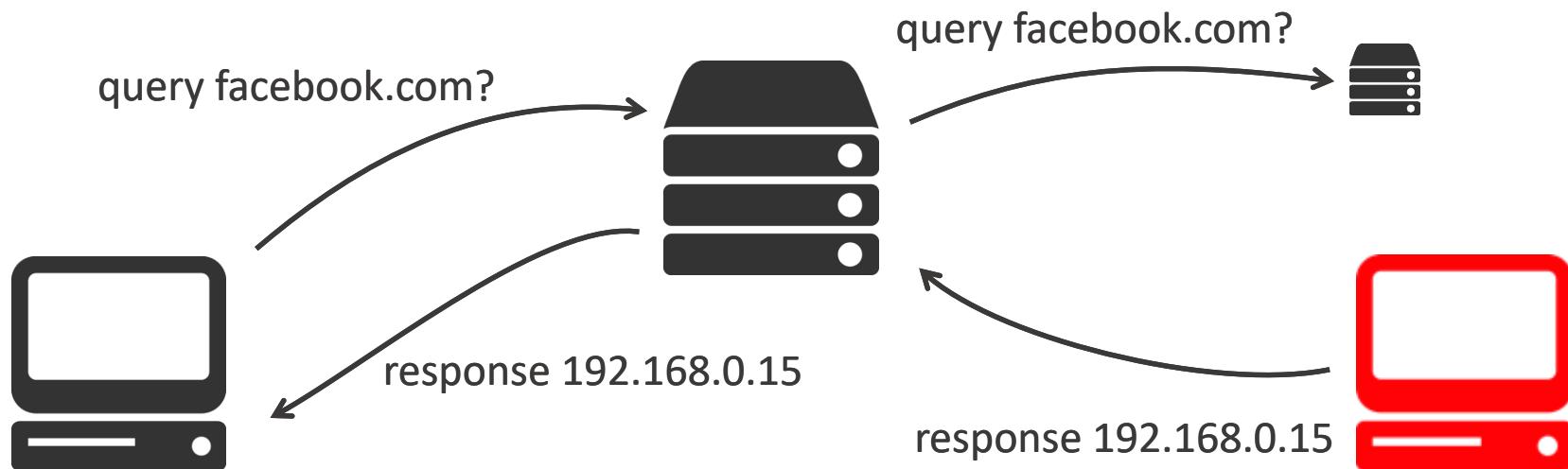
# Domain Name System (DNS)

- DNS translates domain names into IP addresses
  - E.g. nottingham.ac.uk → 128.243.80.167
- DNS packets are UDP
  - Stateless, on the transport layer
- DNS resolvers will cache the IPs for a while



# DNS Spoofing

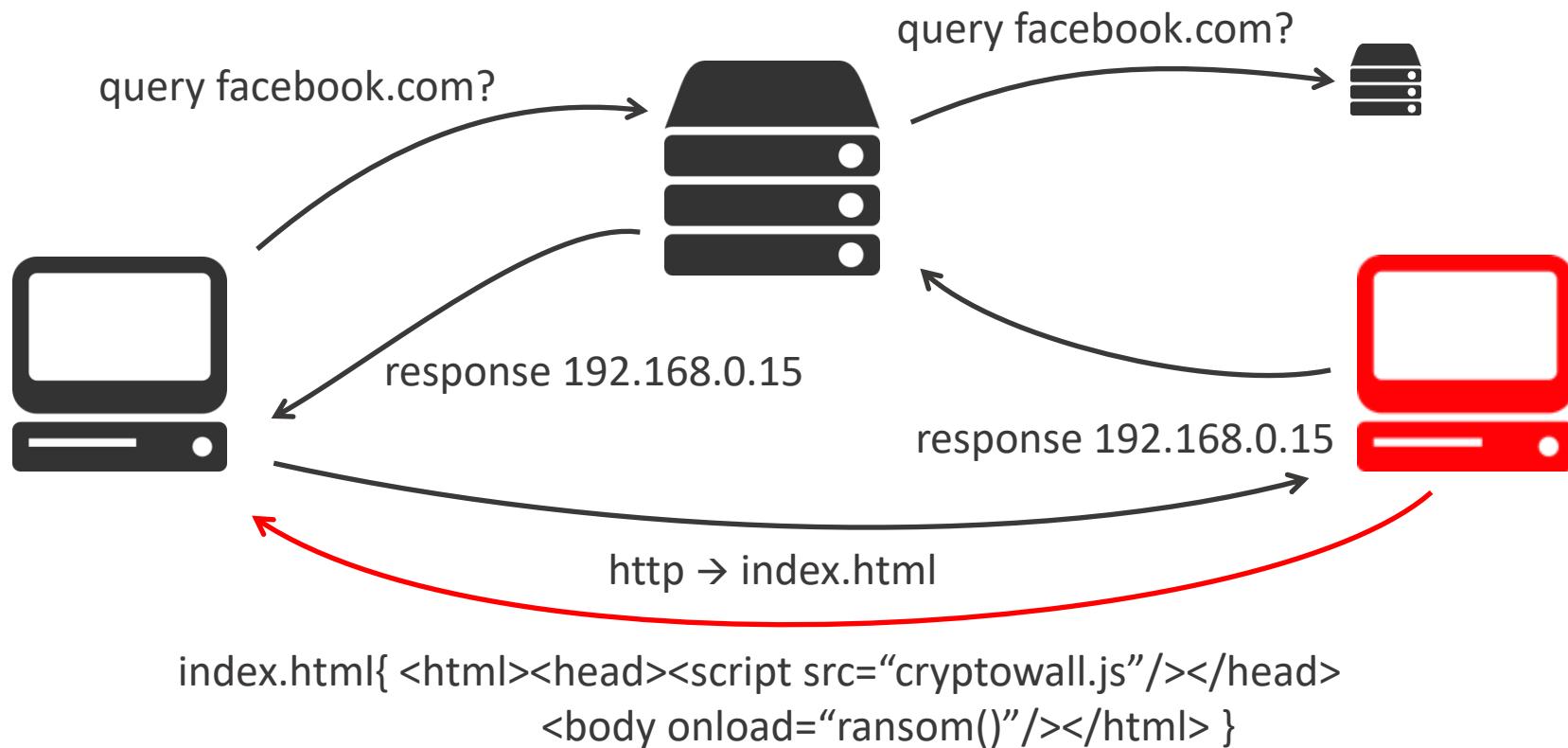
- If we can poison the cache of a nameserver people are using, we can replace a website lookup with our IP



- Can be achieved through prior ARP cache poisoning, a reply flood or a Kaminsky attack

# DNS Spoofing

- If we can poison the cache of a nameserver people are using, we can replace a website lookup with our IP



# DNS Protection

---

- Random query numbers help protect against spoof replies
- Since the Kaminsky attack, most resolvers now randomise the source port too
- DNSSEC aims to tackle DNS exploits by authenticating the name server and providing integrity for the messages



## The connection has timed out

The server at is taking too long to respond.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

[Try Again](#)

# Denial of Service

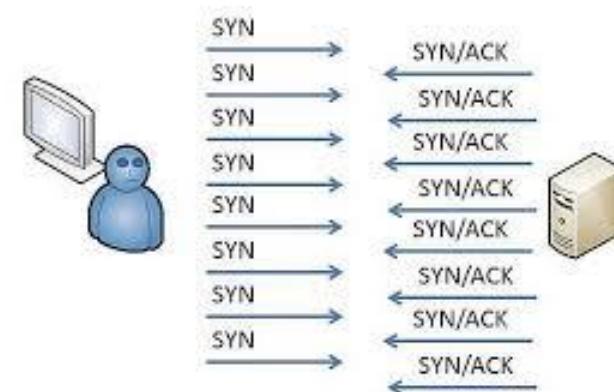
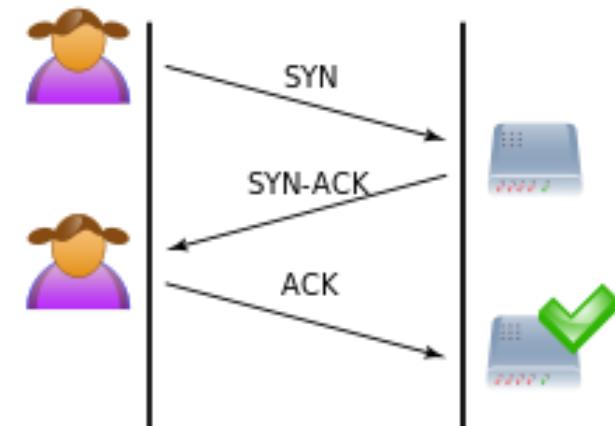
# Denial of Service Attacks

---

- A denial of service attack is an attempt to make a machine or network resource unavailable to its authorised / intended users
- This will usually involve flooding a machine with enough requests that it can't serve its legitimate purpose
  - E.g. Ping flood
- A distributed denial of service occurs where there is more than one attacking machine

# Simple Attack Example

- TCP Syn Flooding
  - Attacker initiates a genuine connection but then immediately breaks it
  - Attacker never finishes 3-way handshake
  - Victim is busy with the timeout
  - Attacker initiates large number of syn requests
  - Victim reaches its half-open connection limit
  - Denial of service



# Low and Slow

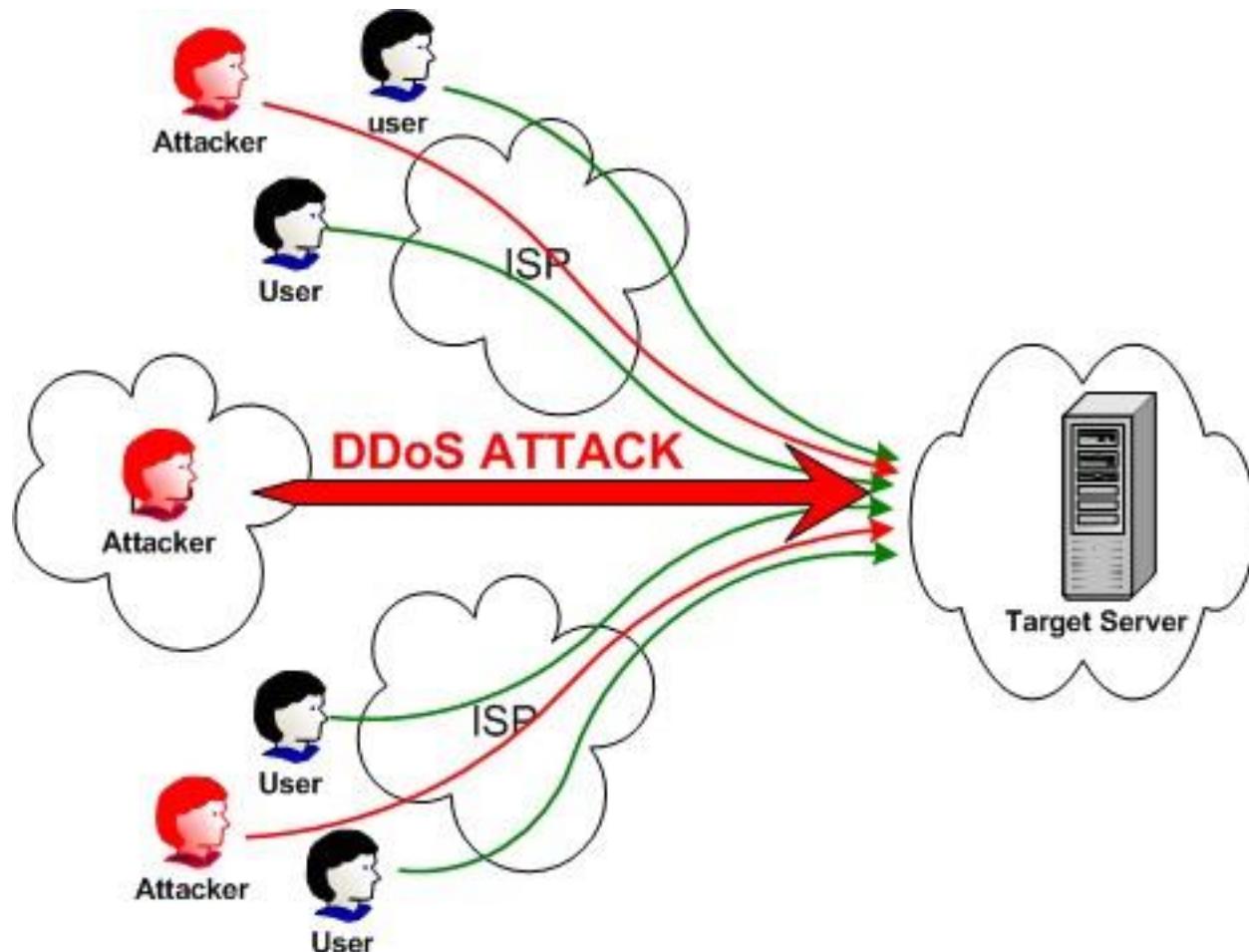
---

- **Slowloris**
  - Open numerous connections to a server
  - Begin an HTTP request, but never actually finish it
- **R-U-Dead-Yet?**
  - Similar to slowloris
  - Begin an extremely long HTTP POST, send tiny amounts at a time



# Distributed Denial of Service (DDoS) Attack Example

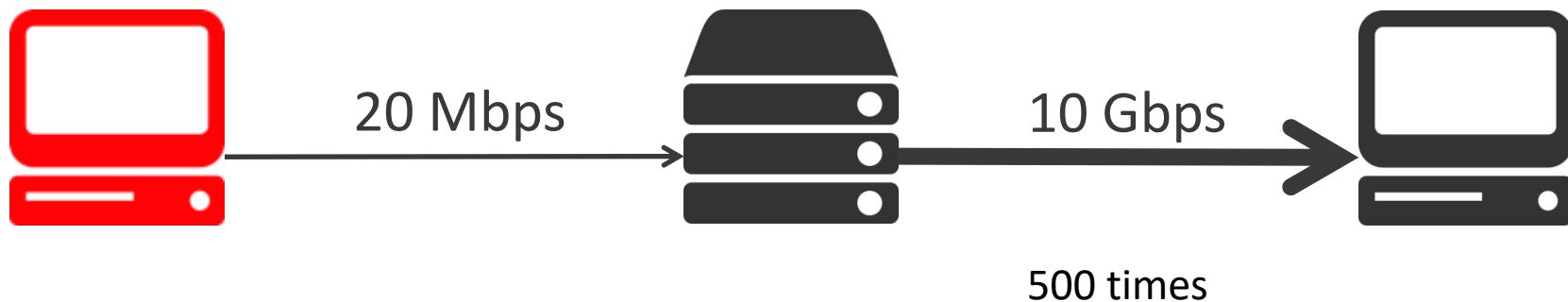
---



# Amplification Attacks

---

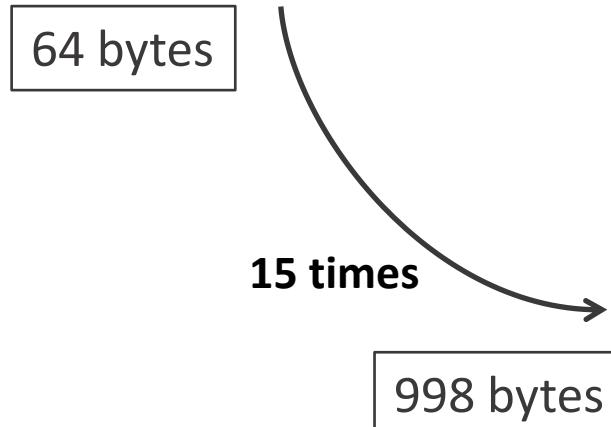
- Regular attacks are attacker's bandwidth vs attacker's target's
- Amplification attacks utilise some aspect of a network protocol to increase the bandwidth of an attack



# DNS Amplification

- Recursive resolvers respond to DNS queries then return a response
- This response can be many times larger than the query

```
dig +bufsize=4096 +dnssec ANY gov.uk
```



```
; >>>HEADER<<- opcode: QUERY, status: NOERROR, id: 58774
;; flags: qr rd ra; QUERY: 1, ANSWER: 10, AUTHORITY: 0, ADDITIONAL: 16

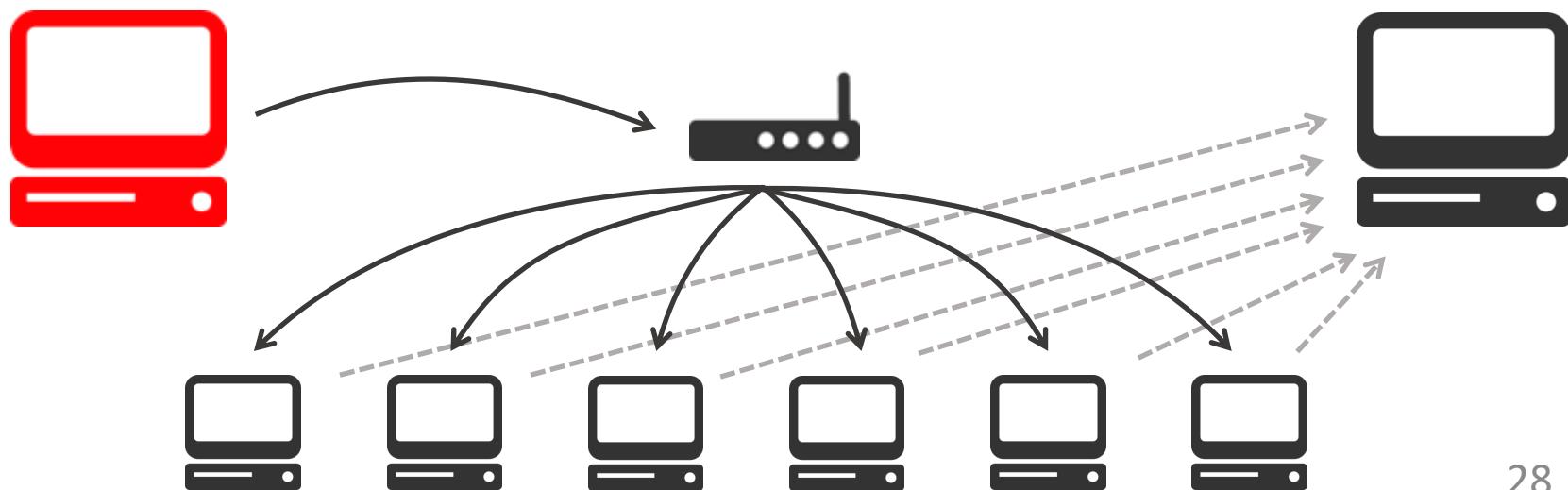
;; OPT PSEUDOSECTION:
;EDNS: version: 0, flags: do; udp: 4000
;; QUESTION SECTION:
;gov.uk.           IN ANY

;; ANSWER SECTION:
gov.uk.        42518 IN A 23.235.37.144
gov.uk.        42518 IN A 23.235.33.144
gov.uk.        42518 IN NS ns2.ja.net.
gov.uk.        42518 IN NS ns0.ja.net.
gov.uk.        42518 IN NS ns4.ja.net.
gov.uk.        42518 IN NS auth50.ns.de.uu.net.
gov.uk.        42518 IN NS auth00.ns.de.uu.net.
gov.uk.        42518 IN NS ns1.surfnet.nl.
gov.uk.        42518 IN NS ns3.ja.net.
gov.uk.        42518 IN RRSIG A 8 2 86400 (
20160306140746 20160205140746 64425 gov.uk.
IhXkrom/IFKOnSInHgnv/me9/CVTP3eZS5102Dyjq/C
4J1YoSg3JPDvLgz8Ucs0q02y+ohcmDCvyQB7SX72L31V
fZBbRwQAYkwNJ5/LrG13oLqwwQaBVbCWuLq8eE8h45BS
KEdznOh9X7VWj9T+uIy8o7oY+i2s2Ykl8ZNd2k=)
```

# Smurf and Fraggle Attacks

---

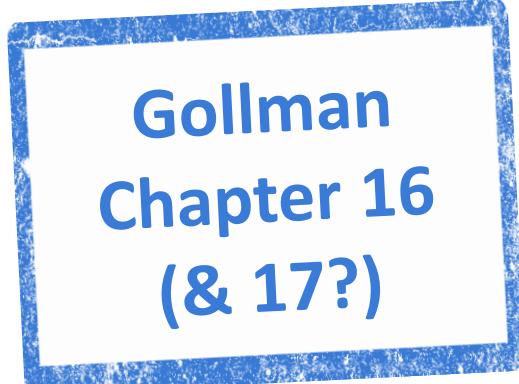
- Smurf attacks broadcast an ICMP Ping request to a router, but with a spoofed IP belonging to the victim
- A Fraggle attack is identical in principle, using UDP echo packets



# Summary

---

- TCP/IP
- IPSec
- ARP Cache Poisoning
- DNS Spoofing
- Denial of Service Attacks



Gollman  
Chapter 16  
(& 17?)

# COMP3052. SEC Computer Security

## Session 09: Internet Security



# Acknowledgements

---

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towey, ...

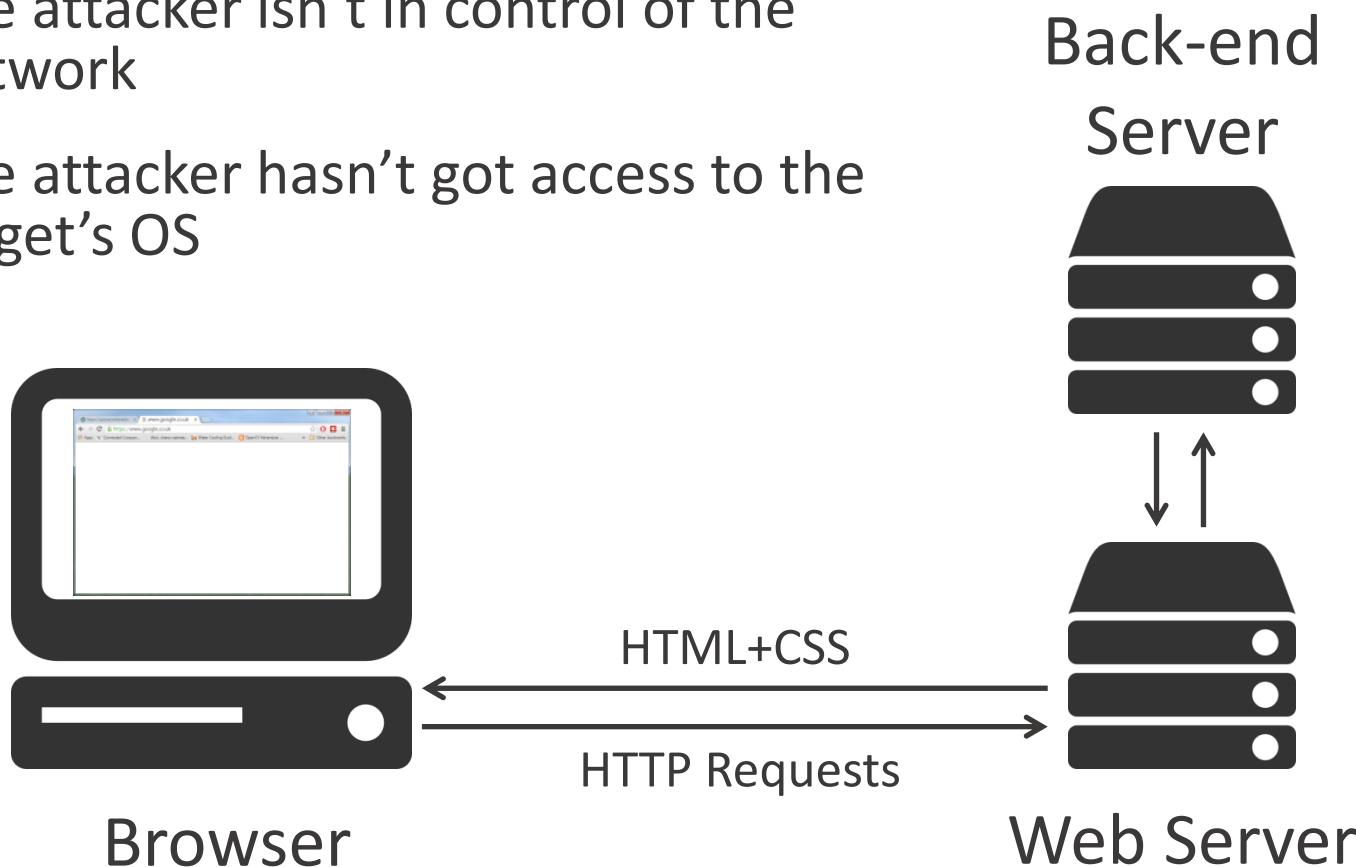
# This Session

---

- Internet Security
- Cookies – Stealing & Tracking
- Cross-site Scripting
- Cross-site Request Forgery
- SSL / TLS
  - Vulnerabilities

# Browser Server Model

- Different from other threat models:
  - The attacker isn't in control of the network
  - The attacker hasn't got access to the target's OS



# Threat Model

---

- Different from other threat models:
  - The attacker isn't in control of the network
  - The attacker hasn't got access to the target's OS
- Instead, the attacker
  - Sees messages addressed to themselves or others
  - Sees data obtained from security compromises
  - Can make educated guesses

# Cookies

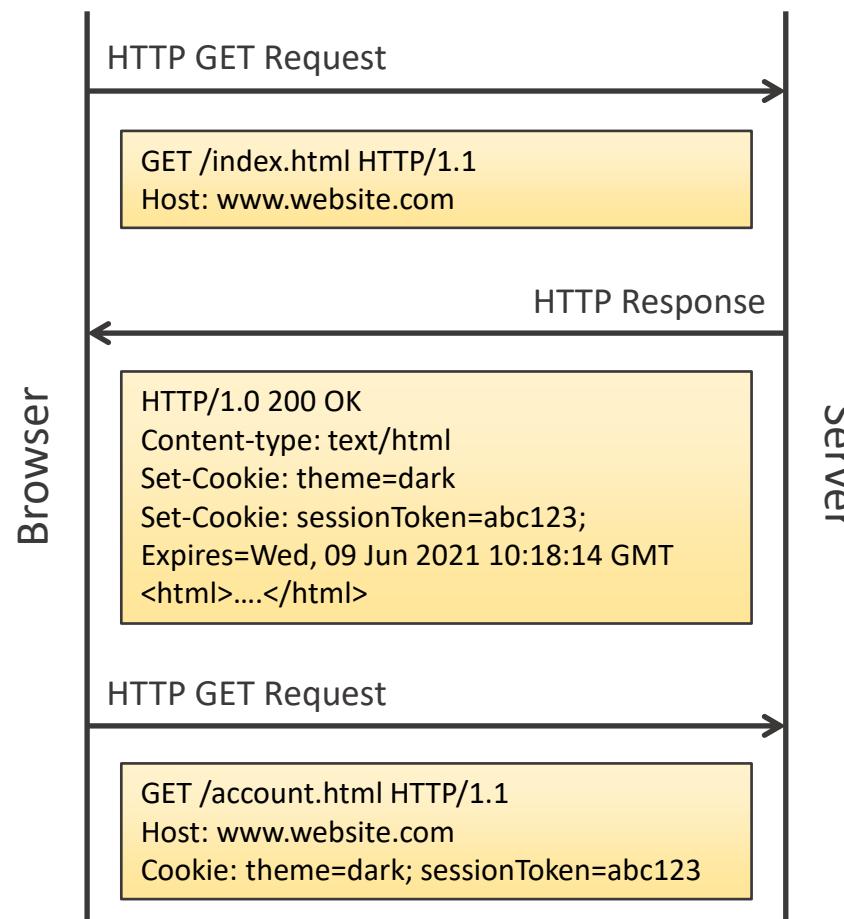
---

- HTTP is a stateless protocol
- Most of what we do online is stateful
- Cookies are small text files used to provide persistence



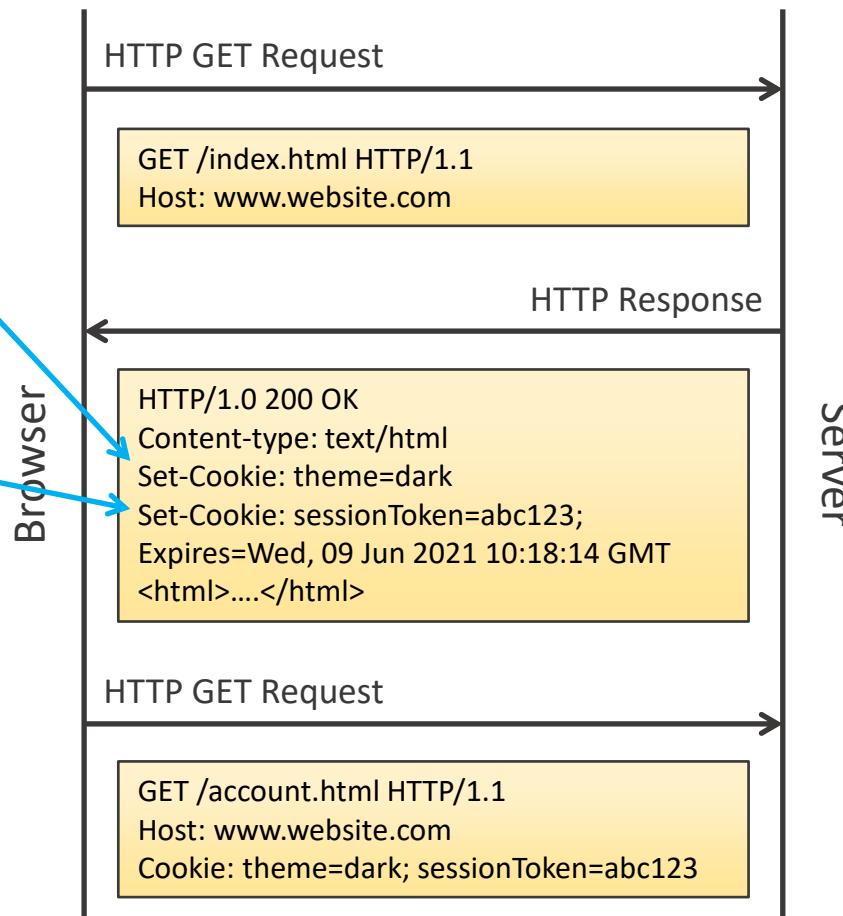
# Cookie

- Servers can provide cookies during HTTP responses, using Set-Cookie
- Browsers will return any cookies for a given domain in GET and POST requests



# Types of Cookie

- Session – Deleted when the browser exits, contain no expiration date
- Persistent – Expire at a given time
- Secure – Can only be used over HTTPS
- HTTPOnly – Inaccessible from JS



# Third Party Cookies

- Cookies are associated with the domains that produced them
  - Amazon.com cookies don't go to google.co.uk
- Some websites include requests to other domains, such as 3<sup>rd</sup> party advertisers
  - These serve cookies – *a lot*

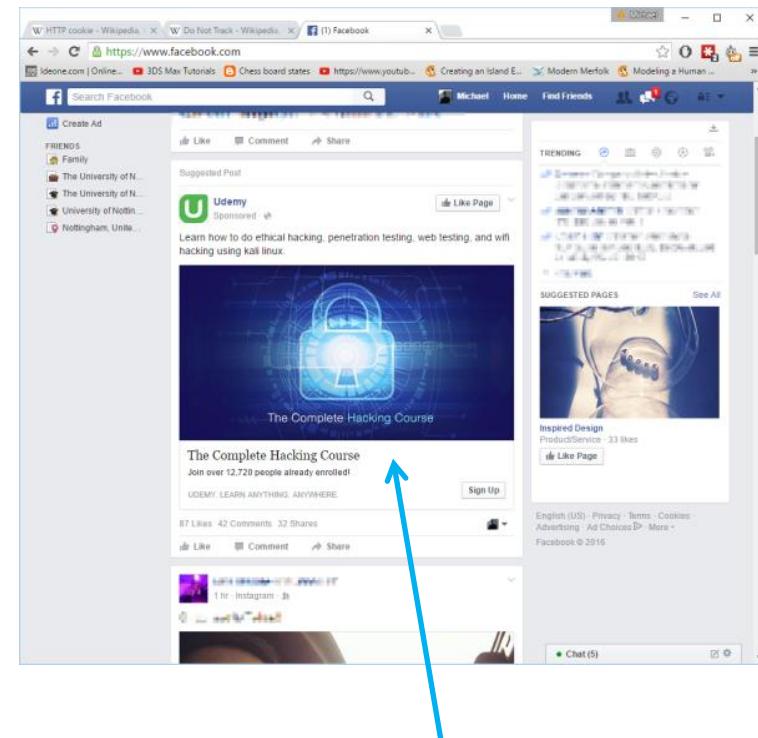


Cookie: [www.thetimes.co.uk](http://www.thetimes.co.uk)

Cookie: [happybanners.net](http://happybanners.net)

# Tracking Users

- Third party cookies are returned to the advertiser every time any website includes one of their ads
- Over time, very detailed information on users can be constructed



The Complete Hacking Course: Learn ethical hacking with Kali Linux!

# Cookie Vulnerabilities

---

- How a website uses a cookie is up to the server
- Many create an SID to authenticate users, for example to “keep me logged on”
- Obtaining this cookie – **Cookie Stealing** – lets you **hijack** their session
  - HTTP Cookies can be stolen simply by monitoring
  - HTTPS will require Cross-site scripting attacks or DNS poisoning

# Cross-site Scripting (XSS)

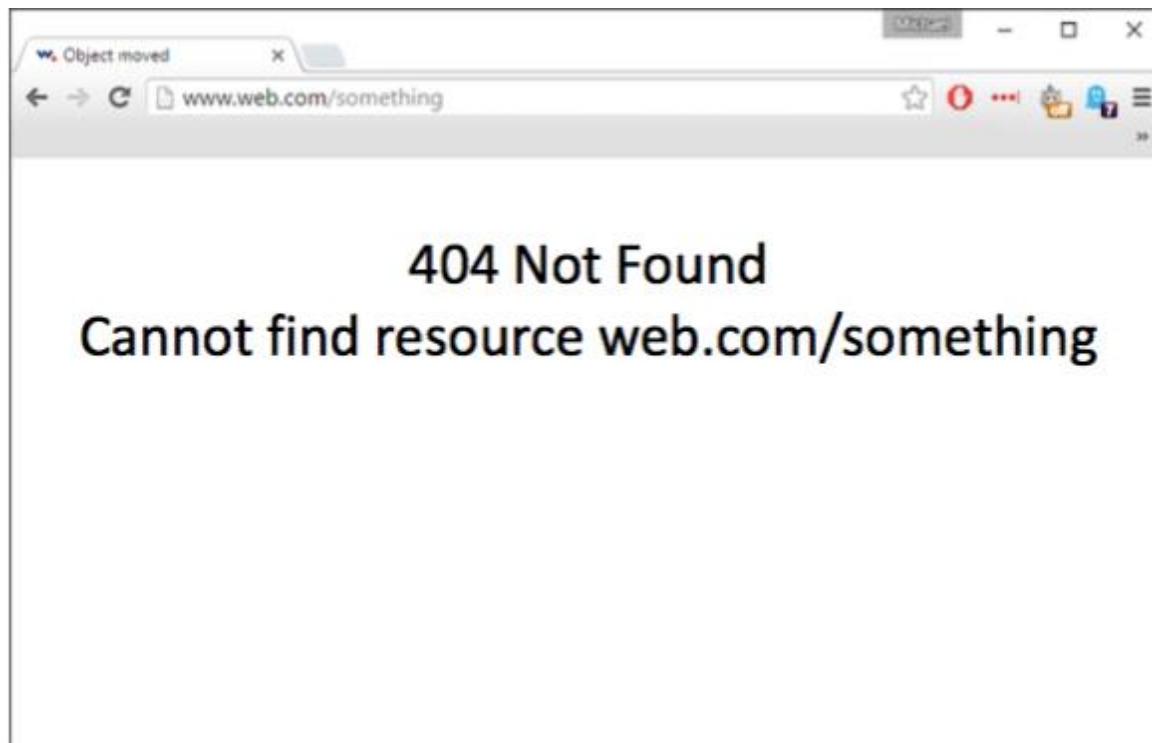
---

- A type of injection attack, similar in many ways to an SQL Injection
- HTML is read by a browser, and is a combination of content (text) and structure (html tags)
- If we can inject html structures into the content of a website, the browser will simply execute these – e.g. <script> tags

# Cross-site Scripting (XSS)

---

- A malicious URL that inserts an exploit directly into the page returned by a server
- Consider a 404 page at [www.web.com/something](http://www.web.com/something):



# Cross-site Scripting (XSS)

---

- Now consider this URL:



# Preventing XSS

---

- Websites must aggressively escape HTML characters from *any* user input / output
- When you consider all of the things people input on interactive websites, this can be a real problem

# Cross-site Request Forgery (XSRF)

---

- When a user puts in an HTTP request, they will also send any relevant session cookies
  - E.g. an SID from having logged in
- If the user has already authenticated, a malicious URL can then perform some action on their account

`http://shop.com/account.php?act=editemail&e=attacker@mail.com`

# XSRF in POST

---

- Most websites use POST, this is little defence
- The phishing email just points to a convincing website with a malicious form on it

```
<form action="http://bank.com/transfer.do" method="POST">
<input type="hidden" name="acct" value="attacker"/>
<input type="hidden" name="amount" value="10000"/>
<input type="submit" value="View my pics!"/>
</form>
```

```
<body onload="document.forms[0].submit()">
<form...
```

You don't even  
need to have  
them click..

# Preventing CSRF

---

- XSS vulnerabilities make CSRF a lot easier!  
Fix these!
- Use synchronizer tokens
  - Each website form has a one-time token that the server validates when the form is submitted

```
<form action="http://bank.com/transfer.do" method="POST">
<input type="hidden" name="sToken" value="OWY4NgmQdnw">
<input type="hidden" name="acct" value="attacker"/>
<input type="submit" value="Transfer Money"/>
</form>
```

# SSL/TLS

---

- There are dangers associated with sending plain text cookies, passwords etc.
- SSL, and the newer TLS provide authenticated and encrypted sessions
- Secure Socket Layer (SSL) came first, then after v3.0 it became Transport Layer Security (TLS), currently v1.3

*We will treat SSL and TLS here interchangeably*

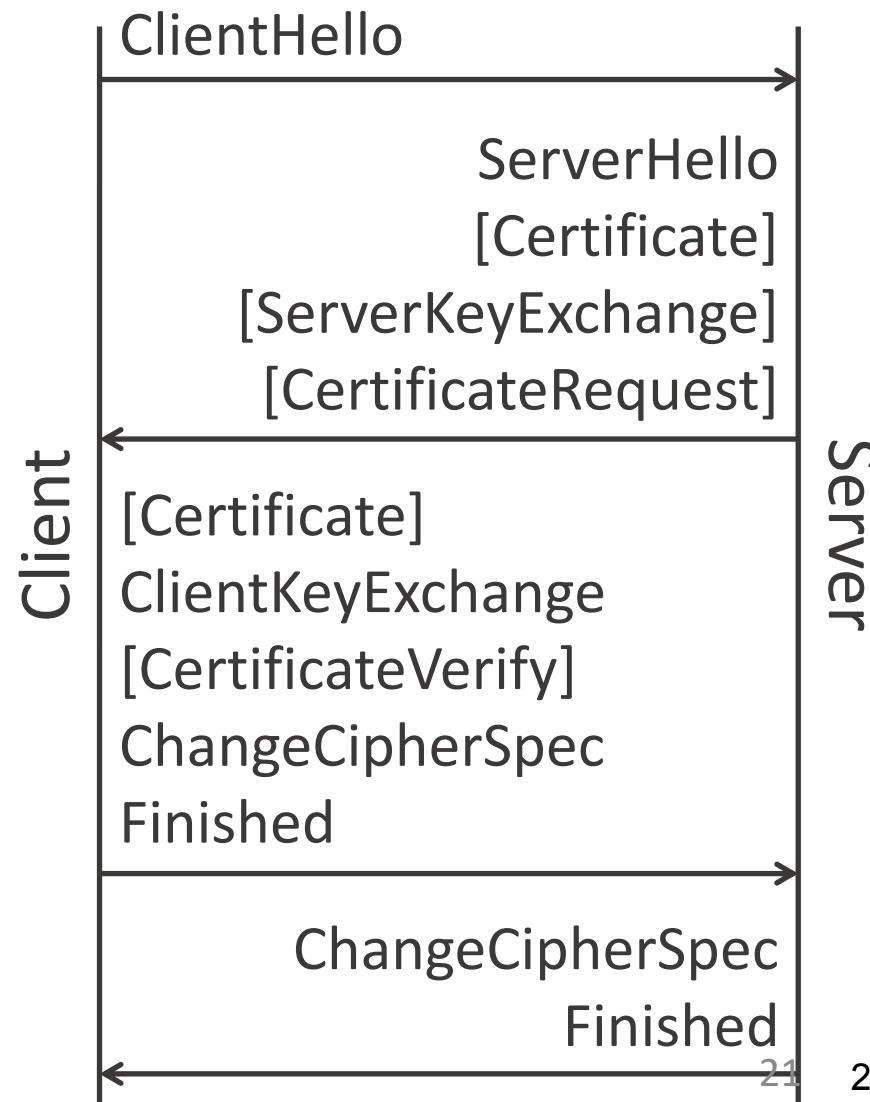
# TLS

---

- Transport Layer Security has two layers:
- The Record Layer
  - Using established symmetric keys and other session info, will encrypt application packets, very like IPSec
- The Handshake Layer
  - Used to establish session keys, as well as authenticate either party – usually the server using a Public-Key Certificate

# TLS Handshake

- The TLS handshake allows us to:
  - Establish the master secret
  - Resume sessions
  - Authenticate the identity of the server or client
- This example is for ECDHE\_RSA



# TLS Handshake

## ClientHello

Supported TLS version: 1.2

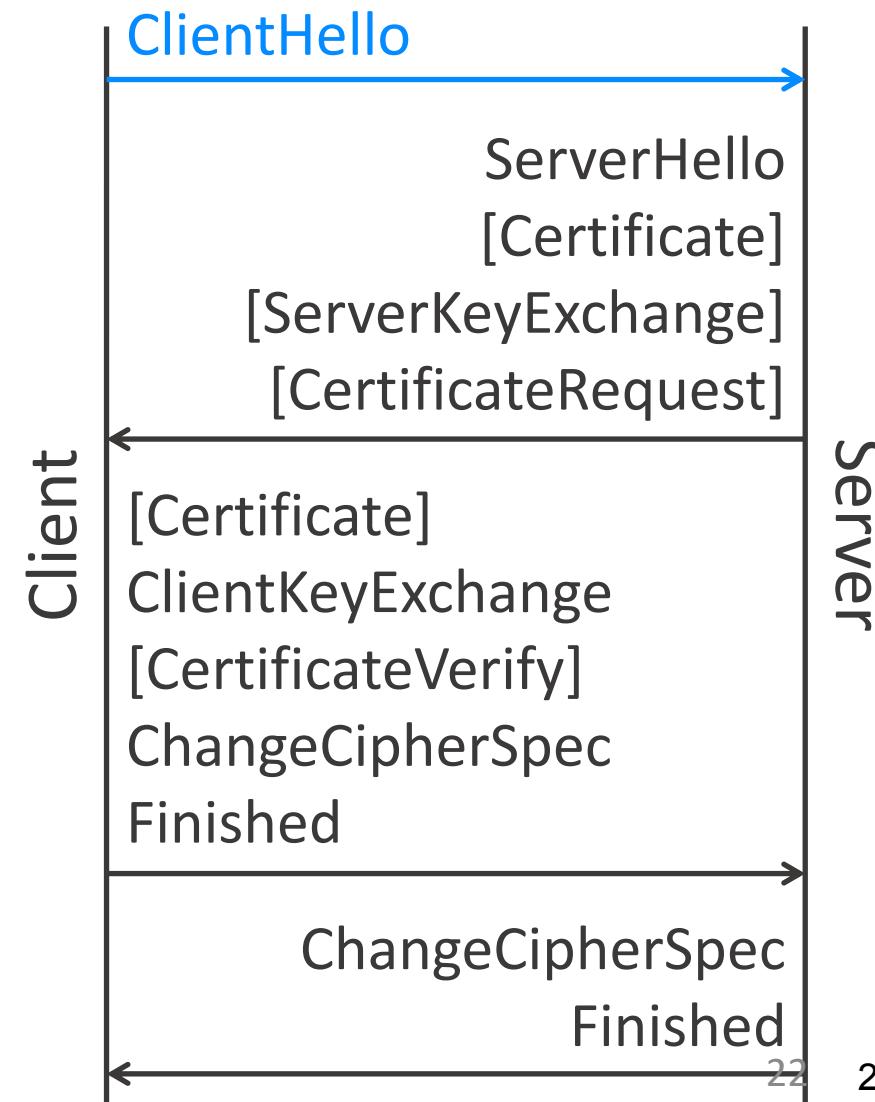
Random number: f3bc12ad..

## Supported Ciphers

```
{  
    TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256  
    TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256  
    TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA  
}
```

[Extensions]

[Session ID]



# TLS Handshake

ServerHello

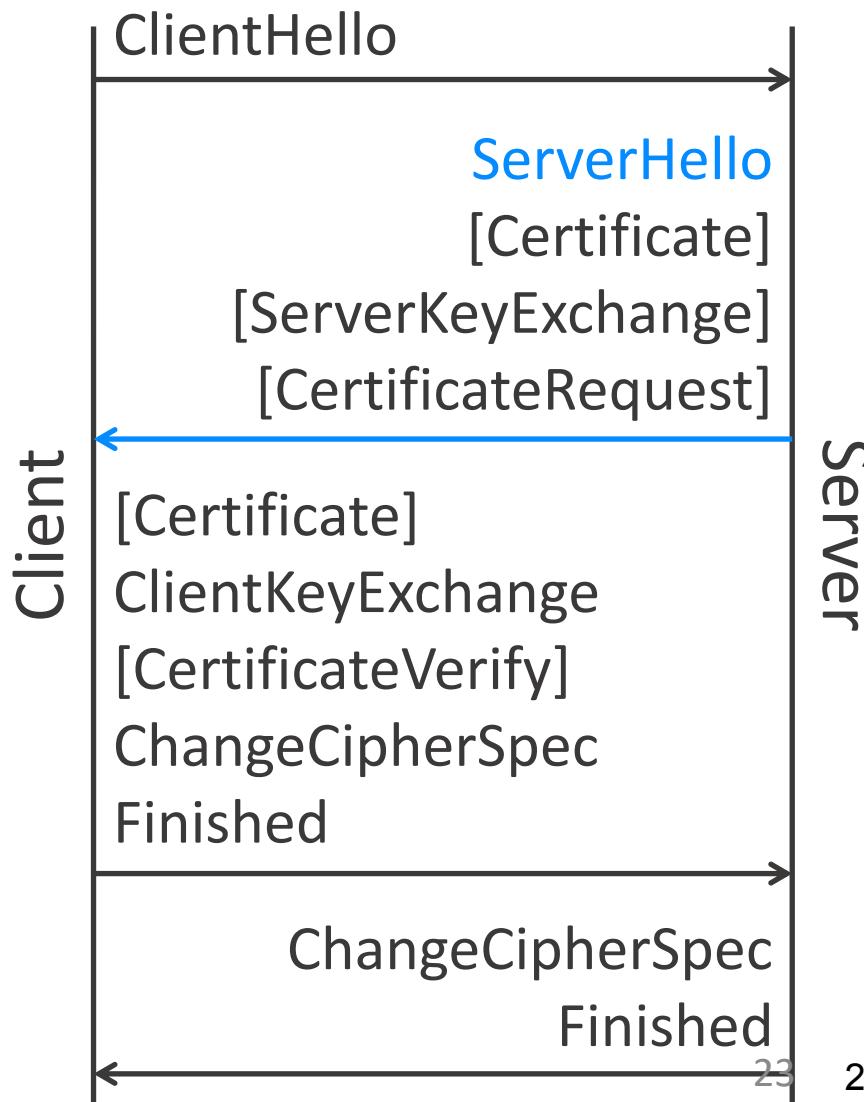
Version: 1.2

Random number: 16cf90ed..

Suite :

TLS\_ECDHE\_RSA\_WITH\_  
AES\_128\_GCM\_SHA256

[Session ID]



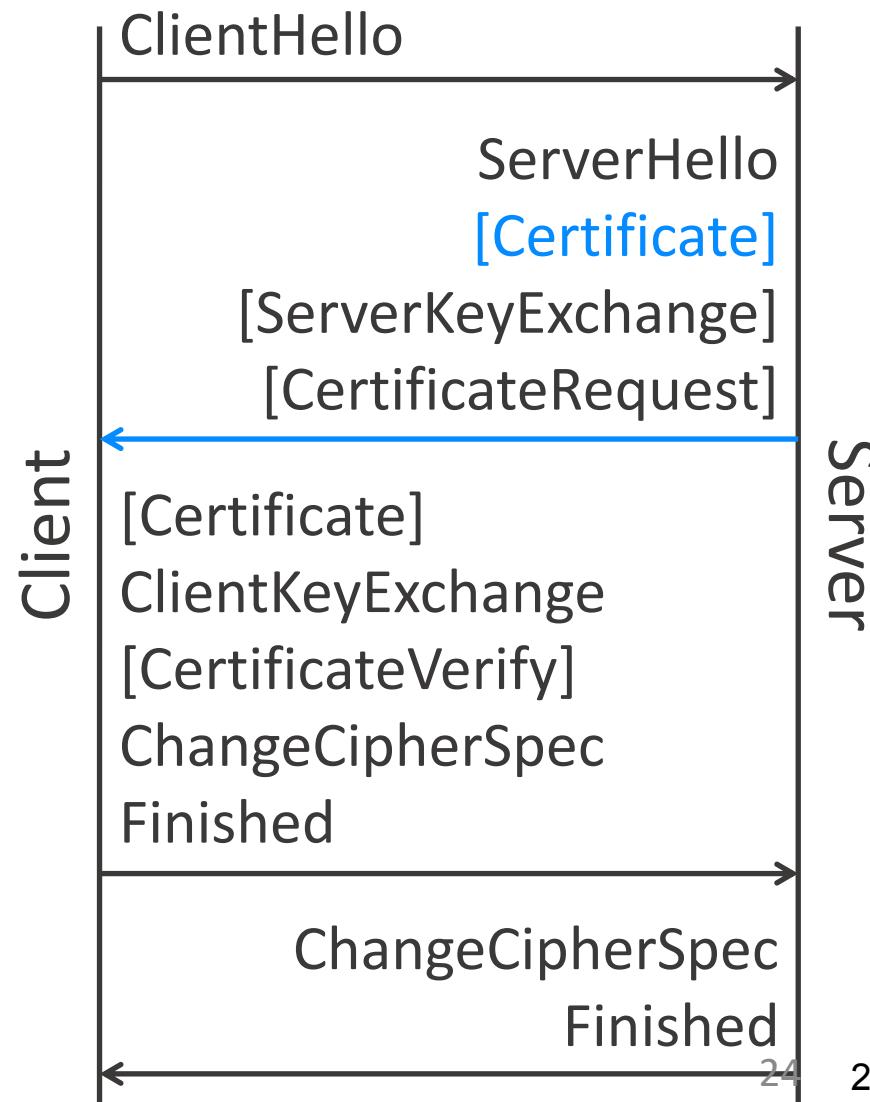
# TLS Handshake

## Certificate

The server sends its public-key certificate to the client

The client checks it using its browser root certificates, or via a CA cert

Digital signature using server private key, client uses this to confirm server identity



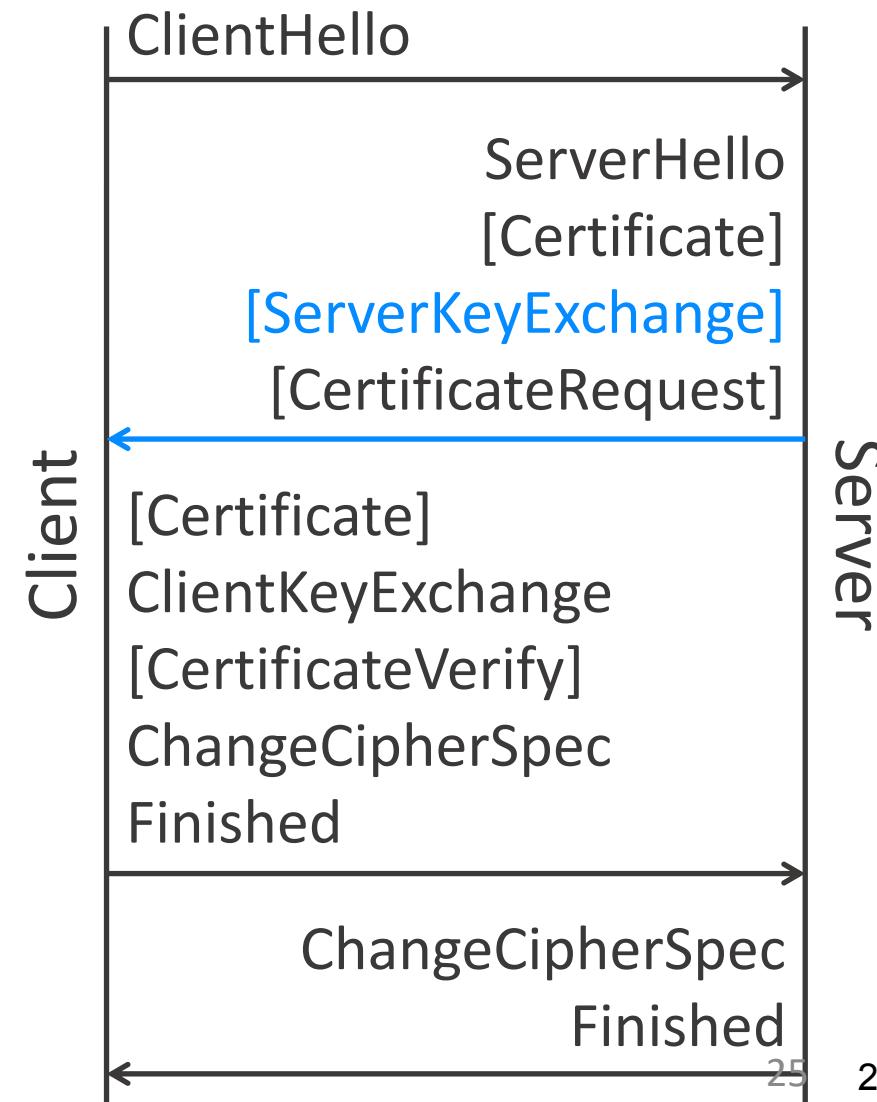
# TLS Handshake

## ServerKeyExchange

Elliptic Curve Diffie-Hellman  
Parameters:

Named Curve: secp256r1  
(0x0017)

DH Public Key: bG



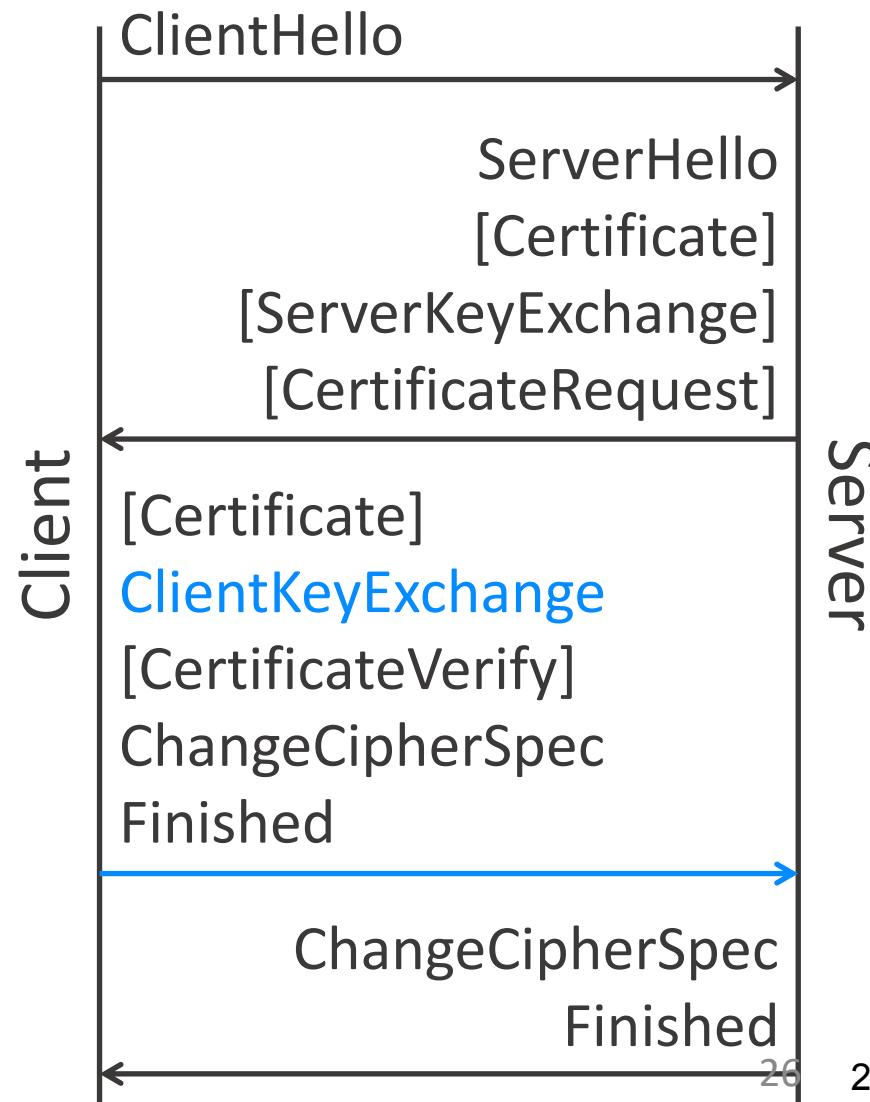
# TLS Handshake

## ClientKeyExchange

DH Public Key:  $aG$

The server and client  
combine the DH parameters  
into the pre-master secret  
 $abG$

The server and client  
combine the random  
parameters and pre-master  
into the session key



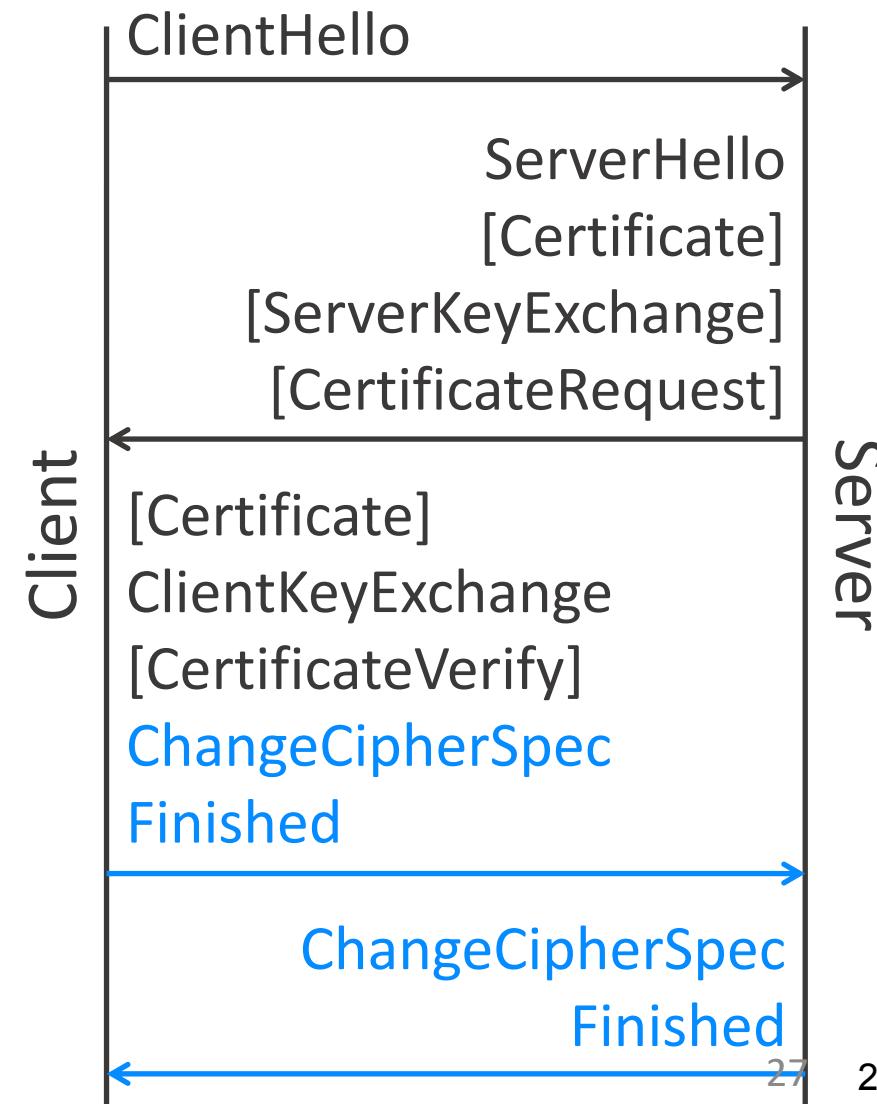
# TLS Handshake

## ChangeCipherSpec

Client then server send the ChangeCipherSpec message, which states that they are about to begin encryption

## Finished

Client then server send Finish message, including a MAC of entire handshake for verification



# TLS Vulnerabilities

---

- The man-in-the-middle vulnerabilities are usually countered using public-key authentication
- The majority of TLS problems are implementation
  - Heartbleed
- Protocol downgrade attacks are still a concern
  - many servers still allow weak cipher suites
  - FREAK and Logjam force the use of 512-bit keys

# Summary

---

- Internet Security
- Cookies – Stealing & Tracking
- SSL / TLS
  - Vulnerabilities
- Cross-site Scripting
- Cross-site Request Forgery

Gollmann  
Chapter 16.5

Gollmann  
Chapter 18.2,  
18.4, 18.5

# COMP3052.SEC Computer Security

## Session 12: Intrusion Detection



# Acknowledgements

---

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, ...

# This Session

---

- Network Attack Models
  - Insider Attacks
- Intrusion Detection Systems
  - Network and Host-based
- Protocol Analysis
- Signature Detection
- Anomaly Detection

# Intrusion & Detection

---

- **Security Intrusion:**

'A security event, or a combination of multiple events that constitutes a security incident in which an intruder gains, or attempts to gain, access to a system or asset without authorisation.'

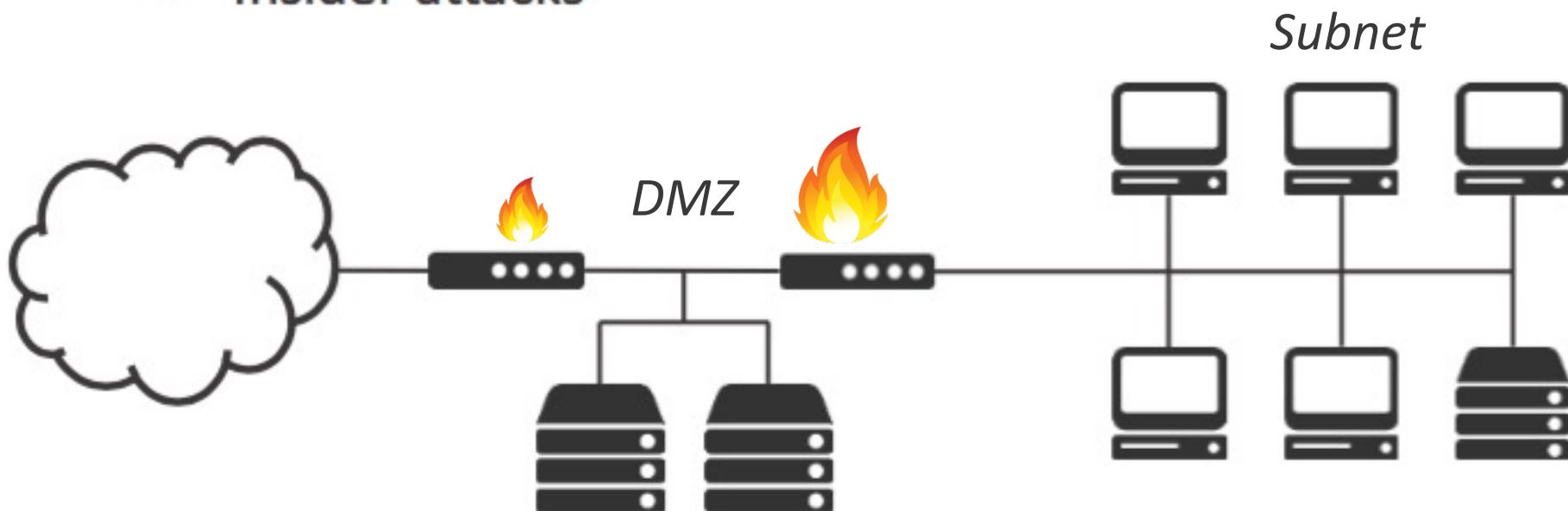
- **Intrusion Detection**

'A security service which monitors and analyses system events for the purpose of finding and providing (near to) real-time warning of attempts to access assets in an unauthorised manner'

- **Internet Security Glossary RFC 2828**

# Network Attack Models

- Firewalls don't protect against:
  - Attacks using valid protocols
  - Insider attacks



# Intruders

---

- **Masquerader**
  - An outsider who is an unauthorised individual who gains access via a legitimate user account
- **Misfeasor**
  - An insider who is a legitimate user, who misuses access permissions and privileges
- **Clandestine**
  - Subject who seizes supervisory control to evade auditing

# Insider Attacks

---

- The most difficult to detect and prevent
  - often simply an HR issue
- Employees will have intimate knowledge of both system layouts and potentially vulnerable services
- Motivated by revenge or entitlement
  - corporate espionage
  - More recently, whistleblowing
- Intrusion detection and system monitoring is the only defence against insiders

# Anti-virus Teaser

---

- **Signature-based Detection:**
  - Store some small **code signature** for each virus
  - Scan files either in bulk or at runtime, compare with the signatures on file
  - Generic signatures
- **Heuristics:**
  - Determine what actions and rules a virus program will normally adopt
  - Start the program in a VM and see what it does
- **Machine Learning**

# Misuse vs. Anomaly

---

- Misuse detection
  - Based on signatures
  - Can miss novel or variant attacks
  - Unsuitable for zero-day attack detection
- Anomaly detection
  - Detects deviations from normal behaviour
  - Can generate too many false alerts
  - What is defined as ‘normal’ can change over time

# Current IDS Issues

---

- Misuse detection is pretty straightforward
  - need to increase the speed of updating the signature database
- Anomaly detection is by no means solved
  - still massive research effort worldwide
  - look for novel solutions outside of statistical machine learning
  - cope with changing user and network behaviour

# Intrusion Detection/Prevention

---

- **Intrusion Detection Systems (IDS)**
  - Detects possible intrusion attempts
  - Generates alerts and logs for administrators
- **Intrusion Prevention Systems (IPS)**
  - Identical to IDS except also stops the attack

# IDS Deployment

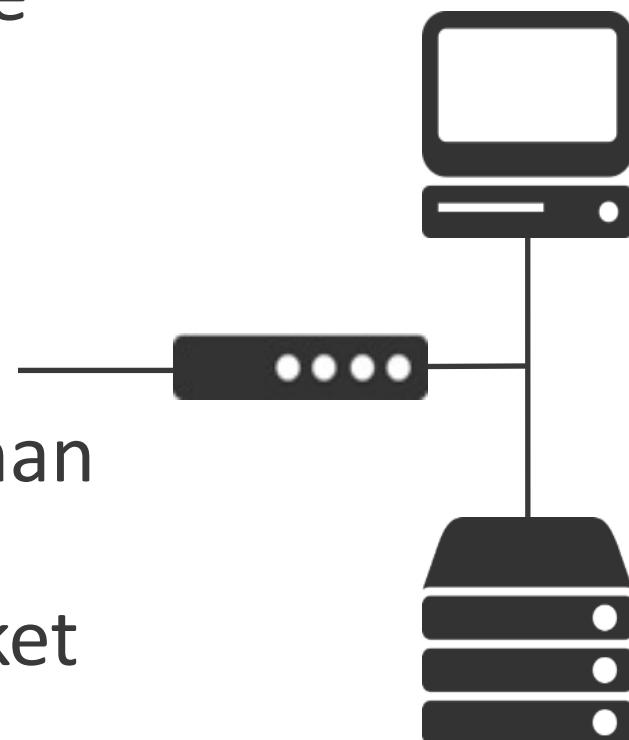
---

- Network-based:
  - Monitors **network traffic** and analyses a variety of packets from different protocols to identify suspicious activity
- Host-based:
  - Monitors the characteristics of a **single host** to find suspicious activity including resource / app usage
  - In many ways modern Anti-Virus does this

# Network-based IDS

---

- Placed at a **viewpoint on a network** to examine and analyse traffic
  - Installed on a firewall or in a DMZ
  - Installed behind a screened subnet
- May perform deeper analysis than many firewalls, e.g. stateful protocol analysis and deep packet inspection



# Network-based IDS

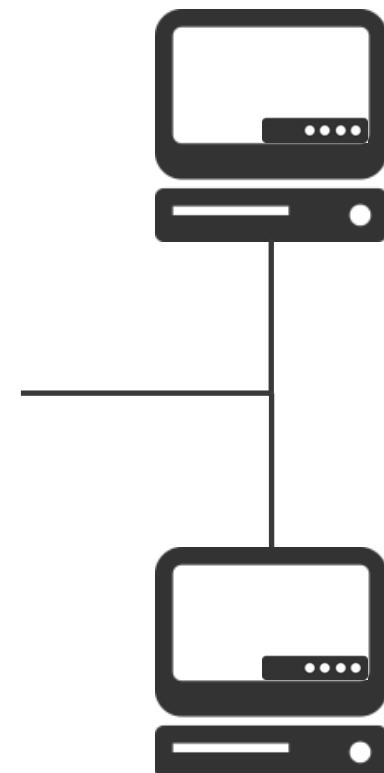
---

- Can monitor traffic from multiple hosts
  - Enables use of correlation techniques
    - which can be very powerful
- Can be difficult to detect fragmented packet based attacks
- Harder to detect phishing or trojan attacks
- Better at detecting DDoS attacks
- Deep packet inspection rarely used

# Host-based IDS

---

- Additional layer of security software running on a host within a protected LAN or VPN
- Creates a **profile of usage** for specific users
- Can monitor both the internals of a host including CPU, memory use, application use and the network stack



# Host-based IDS

---

- Can easily correlate network with host behaviour
  - can inspect more useful data
- Can perform deep packet inspection
- Can deal with packet fragmentation
- Only gets insight from a single machine
- Lends itself more to anomaly-based techniques

# Components of IDS

---

- Sensors / Agents: collect and collate data from multiple viewpoints on a network
- Analysers: ascertain if an intrusion has taken place
- Reporting: notify the administrators via alerts, usually a console or graphical interface is required

Multiple sensors allow us to distribute analysis, but centralise computing overhead

# Detection Modes

---

- Stateful Protocol Analysis
  - More complex version of a stateful packet filter
- Signature-based Detection
  - Fingerprinting sequences of operations or packets
- Anomaly-based Detection
  - Build a model of “normal” and find deviations

# Protocol Analysis

---

- Hold detailed session information on protocols being used, examine for attacks:
  - Why is this user logging in as root?
  - Why is this command being sent a 1000 byte buffer as a parameter?
- Computationally costly, and requires the IDS to have all possible versions of these protocols described in its database

# Signature-based Systems

---

- Like antivirus, signatures are created and stored in a database
  - operations rather than binaries
- If operations match a defined signature, then an alarm is triggered
- Include some form of attack language
  - Mechanisms to describe sequences of events
  - Maintain and monitor intermediate states and event transitions

# Signature-based Systems

---

- The pros and cons of these systems are identical to their anti-virus counterparts
  - Computationally efficient
  - Will always spot a known attack or vulnerability
  - Will always miss an unknown attack or vulnerability
  - Detailed signature databases must be kept up-to-date

# Example Signature

---

- What are the signs that a host on the network is performing port scans?
  - Large amounts of ICMP traffic
  - Many TCP connection (SYN) packets
  - These connections going to a variety of other hosts

“If a host establishes more than three TCP connections to different hosts in five seconds, it is port scanning”

# SNORT

---

- Snort is a powerful and well established IDS
  - Also free!
- Uses rules to analyse network packets, and then can provide alerts or logging



# Snort Rule Example

---

- Rule outline:

action proto src-ip src-port direction dst-ip dst-port (options)

- Buffer Overflow?

activate tcp any any -> 192.168.1.21 22 (activates:1;  
msg:"Possible SSH exploit"; content:"|90|"; \ offset:40;  
depth:75; dsize: >6000;)

dynamic tcp any any -> 192.168.1.21 22 (activated\_by:1;  
count:100;)



# Detecting Nmap

---

- Snort has built in rules for detecting Nmap, a logged scan may look like this:

```
08/xx-13:27:32.464097 TCP src: 10.0.4.100 dst: 10.0.4.1 sport: 3537 \
dport: 5232 tgts: 1 ports: 11 flags: *****S* event_id: 0
```

```
08/xx-13:27:32.464177 TCP src: 10.0.4.100 dst: 10.0.4.1 sport: 3538 \
dport: 5002 tgts: 1 ports: 12 flags: *****S* event_id: 7
```

```
08/xx-13:27:32.464256 TCP src: 10.0.4.100 dst: 10.0.4.1 sport: 3539 \
dport: 780 tgts: 1 ports: 13 flags: *****S* event_id: 7
```

```
08/xx-13:27:32.465642 TCP src: 10.0.4.100 dst: 10.0.4.1 sport: 3540 \
dport: 1484 tgts: 1 ports: 14 flags: *****S* event_id: 7
```

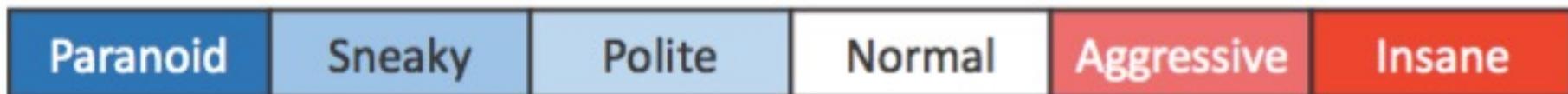
```
08/xx-13:27:32.465722 TCP src: 10.0.4.100 dst: 10.0.4.1 sport: 3541 \
dport: 2002 tgts: 1 ports: 15 flags: *****S* event_id: 7
```

etc. ...

# Nmap Timings

---

- You can avoid detection when using Nmap by reducing the speed of the scan
- This makes port scanning very hard to distinguish from general network noise
- Nmap contains 6 timing options

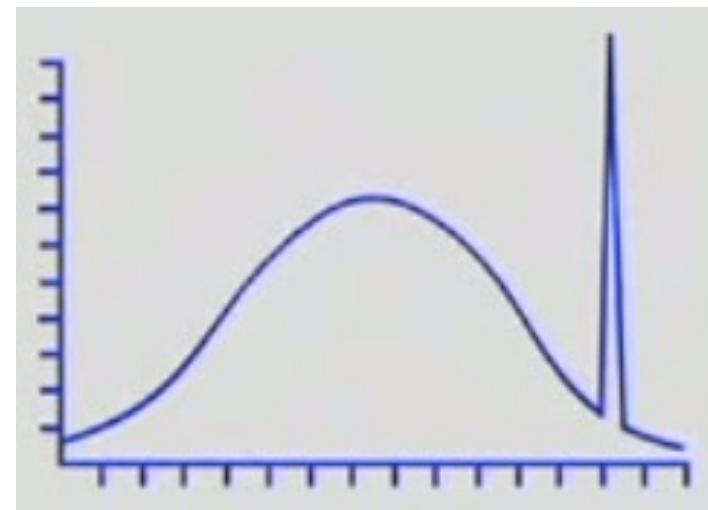


5 minutes between packets – leave overnight!

# Anomaly Detection

---

- Anomaly detection has wide-ranging applications from IDS to banking fraud
- Build up a **picture of normal usage**, and detect when usage moves beyond what is normal
  - This may involve usage of network, applications, storage, system calls etc.
- Always a trade off between **false positives** and **false negatives**



# What is Normal?

---

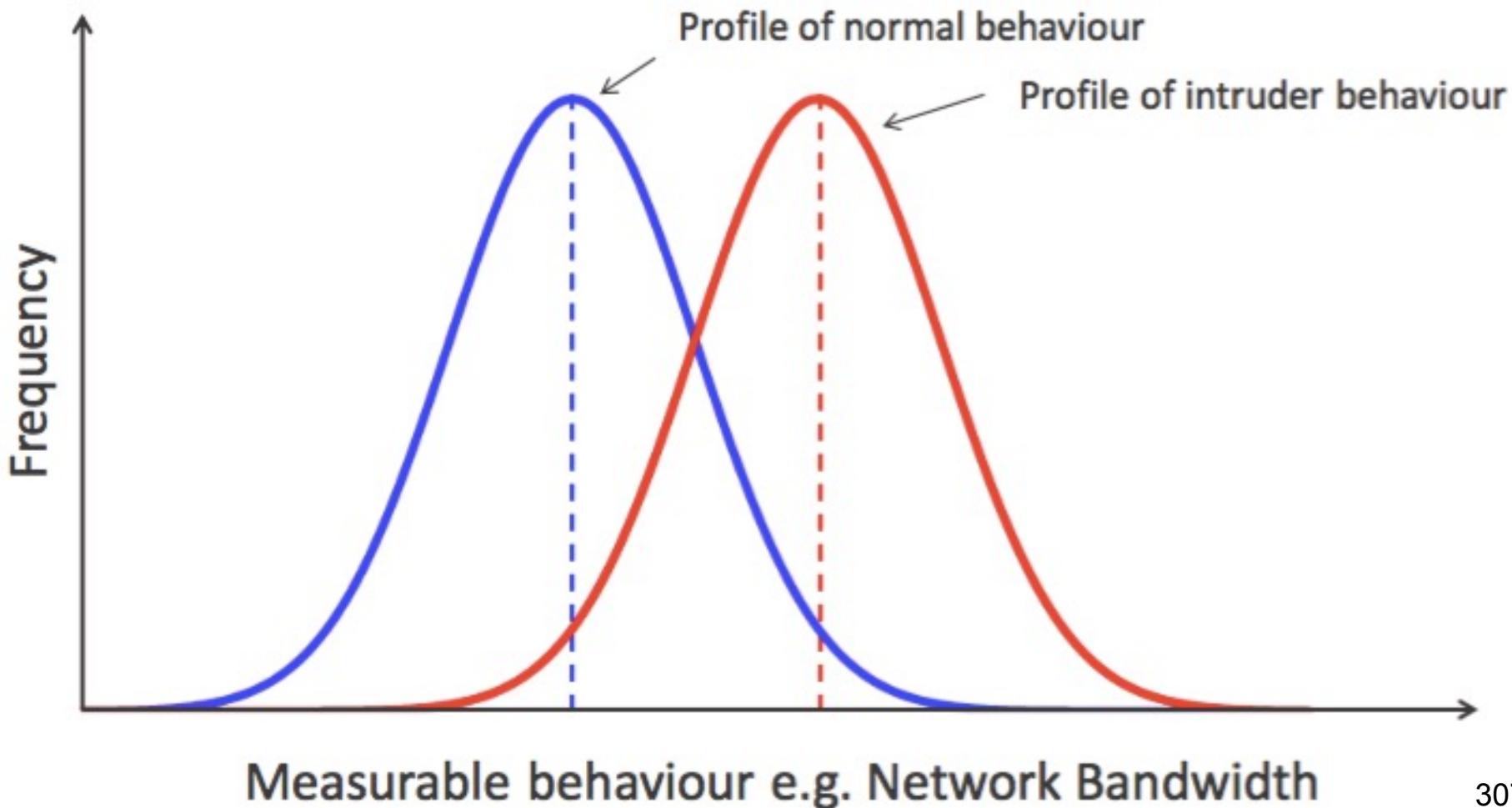
- Run a host within a quarantined environment and collect training data
- Constructed by monitoring audit logs
- Sometimes rely on analysis of sequences of system calls through normal behaviour



What is  
**NORMAL?**

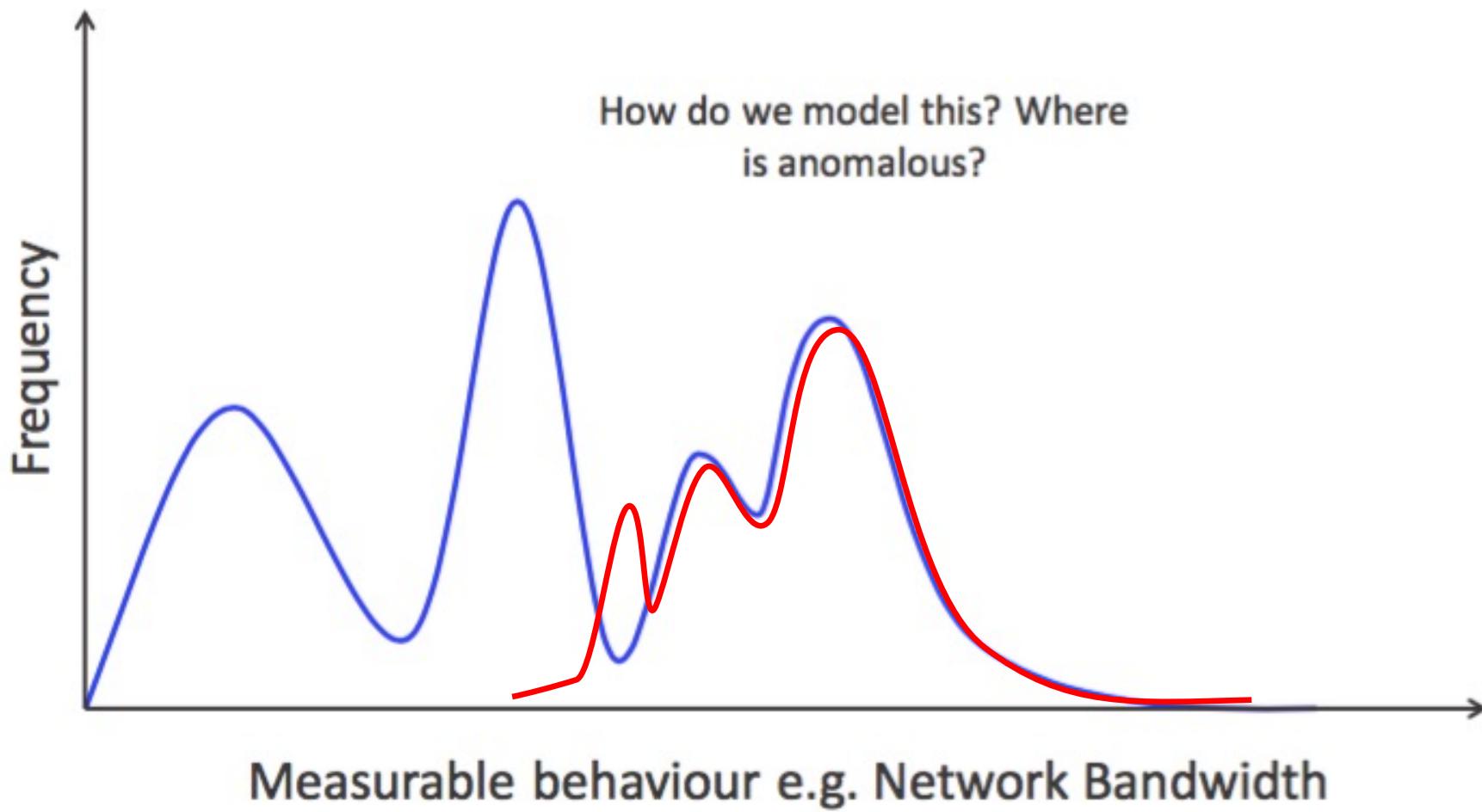
# Statistical Models of Normal

## Probability density function



# Complex Behaviours

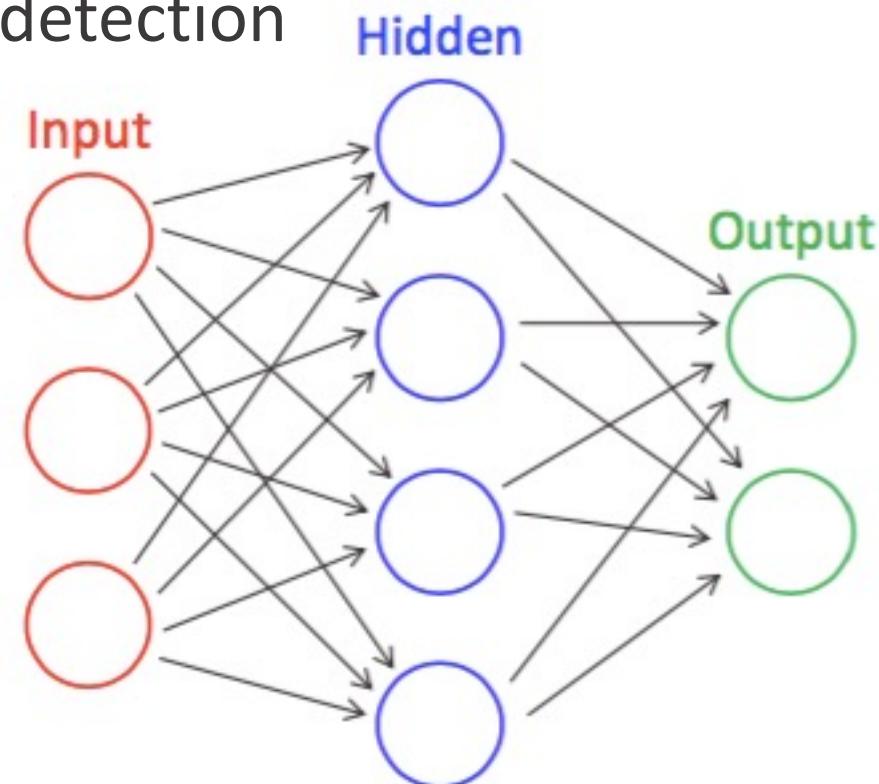
- Profiles can be complex, and can change over time



# Machine Learning

- Machine learning approaches train a model to make predictions on data
- Support Vector Machines, Neural Networks etc. all see use in intrusion detection

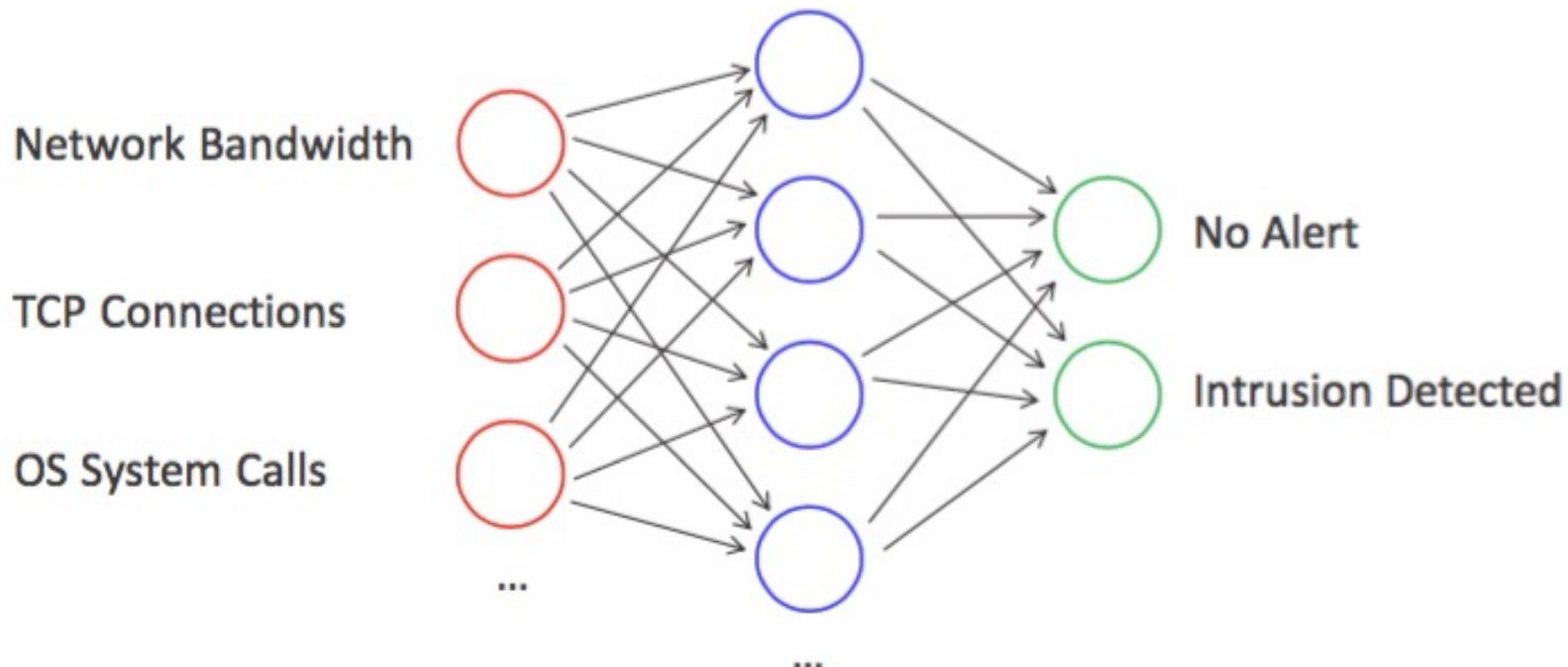
Artificial Neural Networks  
are capable of modelling  
complex non-linear  
functions



# Neural Networks for ID

---

- A network can be pre-trained
- Sensor measurements are then passed through the network
- Activations in the specific output neuron signal an alert



# Drawbacks

---

- Scaling:
  - Search space can increase exponentially
  - Real-time data
- False negatives
  - Limits in the representation
  - What is normal can change
    - do we re-train and risk learning intruder behaviour?

# Intrusion Prevention

---

- A common extension of IDS, often network based
- Actively monitors the system through stateful analysis
  - Setting alarms
  - Dropping packets, stalling connections, closing ports
  - Can be subverted to cause DoS

# Summary

---

- Network Attack Models
  - Insider Attacks
- Intrusion Detection Systems
  - Network and Host-based
- Protocol Analysis
- Signature Detection
- Anomaly Detection



# COMP3052.SEC Computer Security

## Session 13: Software Vulnerabilities



# Acknowledgements

---

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Toweys...

# This Session

---

## Malware

- Viruses
- Worms
- Trojans
- Ransomware
- Rootkits and Backdoors

## Exploits

- Coding Flaws
- Common Vulnerabilities
- Buffer overflows
- ...



## Danger: Malware Ahead!

Google Chrome has blocked access to this page

Content from valid.canardpc.com, a known malware distributor, has been inserted into this web page. Visiting this page now is very likely to infect your computer with malware.

Malware is malicious software that causes things like identity theft, financial loss, and permanent file deletion. [Learn more](#)

[Go back](#)

[Advanced](#)

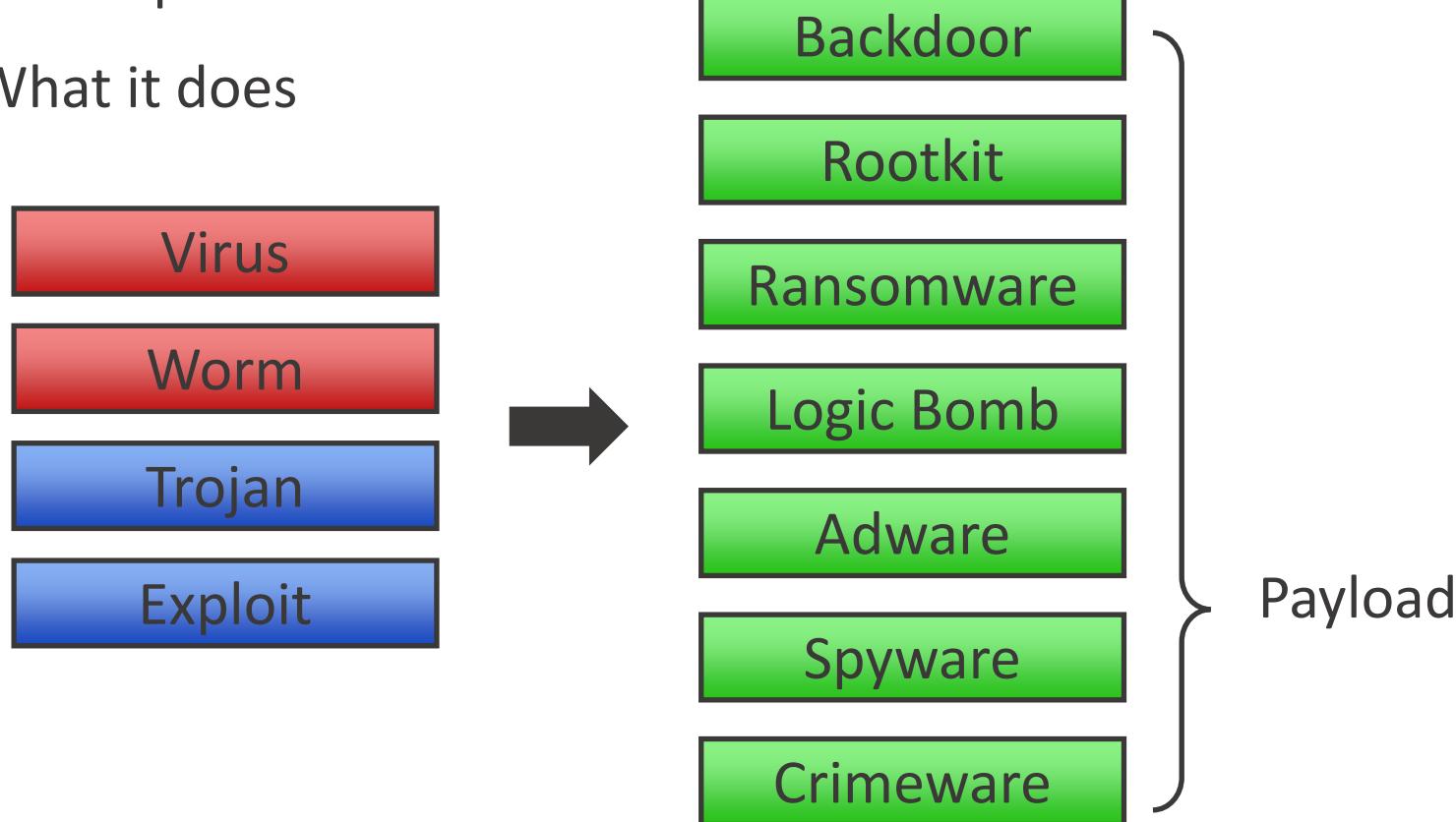


Improve malware detection by sending additional data to Google when I encounter warnings like this. [Privacy policy](#)

# Malware Malicious Software

# Malware

- A *very general* term, malware is usually categorised based on:
  - How it proliferates
  - What it does



# Vectors

---

- Vectors are the mechanism through which malware infects a machine
- Usually the vector will be either a software vulnerability, or a human error
- In the case of human error, this often means someone has clicked “Yes” to something they shouldn’t have!

# Payloads

---

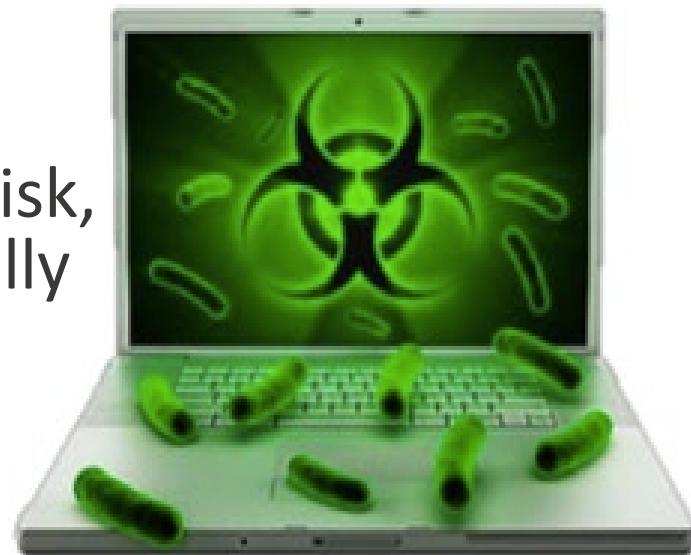
- Payloads are the actual malware deposited on the machine, or the harmful results
- They range in severity:
  - Essentially do nothing
  - Messages and adverts
  - Recruited into a botnet or mail spam
  - Stealing private information
  - System destruction



# Viruses

---

- A virus is a piece of self-replicating code
- Propagates by attaching itself to a disk, file or document, which would usually be executable
- When the file is run, the virus runs, and attempts to proliferate
- Installs without the user's knowledge or consent
- Requires human intervention to spread



# Virus Attachment Mechanisms

---

- Historically, viruses installed themselves in the boot sectors of disk partitions
- Executable files may be sent as email attachments and require social engineering to entice users
- Script files for system admins including Windows batch files and UNIX shell scripts
- Documents containing macros
  - Older office formats, newer macro enabled workbooks

# Notable Viruses

---

- 1981: Elk Cloner, the first known computer virus outbreak affects Apple II computers
- 1986: Brain, the first MS-DOS computer virus
- 1989: Ghostball, the first multipartite virus – affects both EXEs and the boot sector
- 1995: First macro virus, Concept, affects MS Word documents
- 1996: First linux virus, Staog, uses bugs in the Linux kernel

# Worms

---

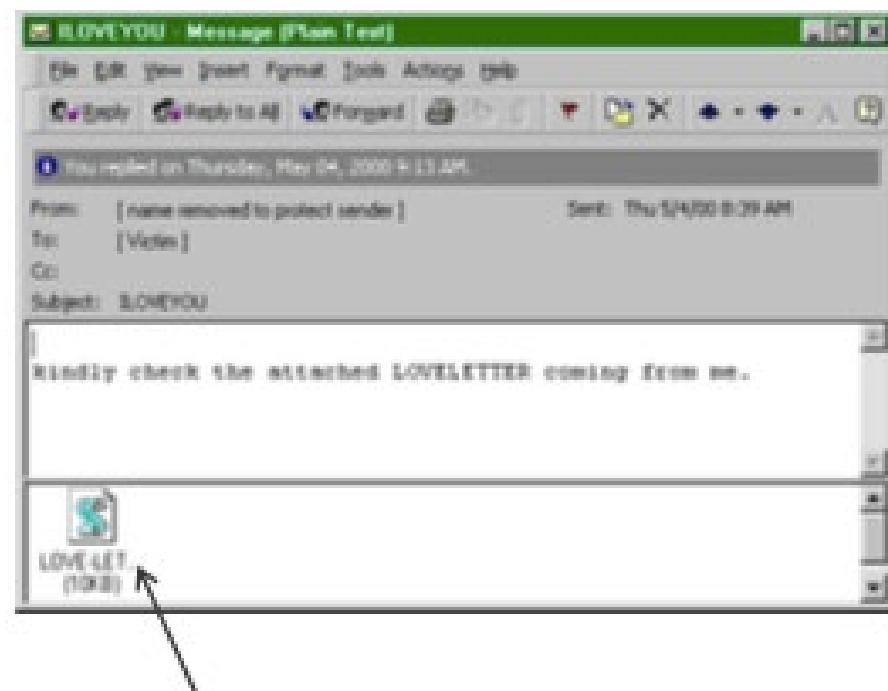
- Viruses traditionally require a human to spread
- Worms are self-replicating and stand-alone programs
  - Do not require human intervention
- Scanning worms or email worms
- Exploit known software vulnerabilities in order to spread



# Notable Worms

---

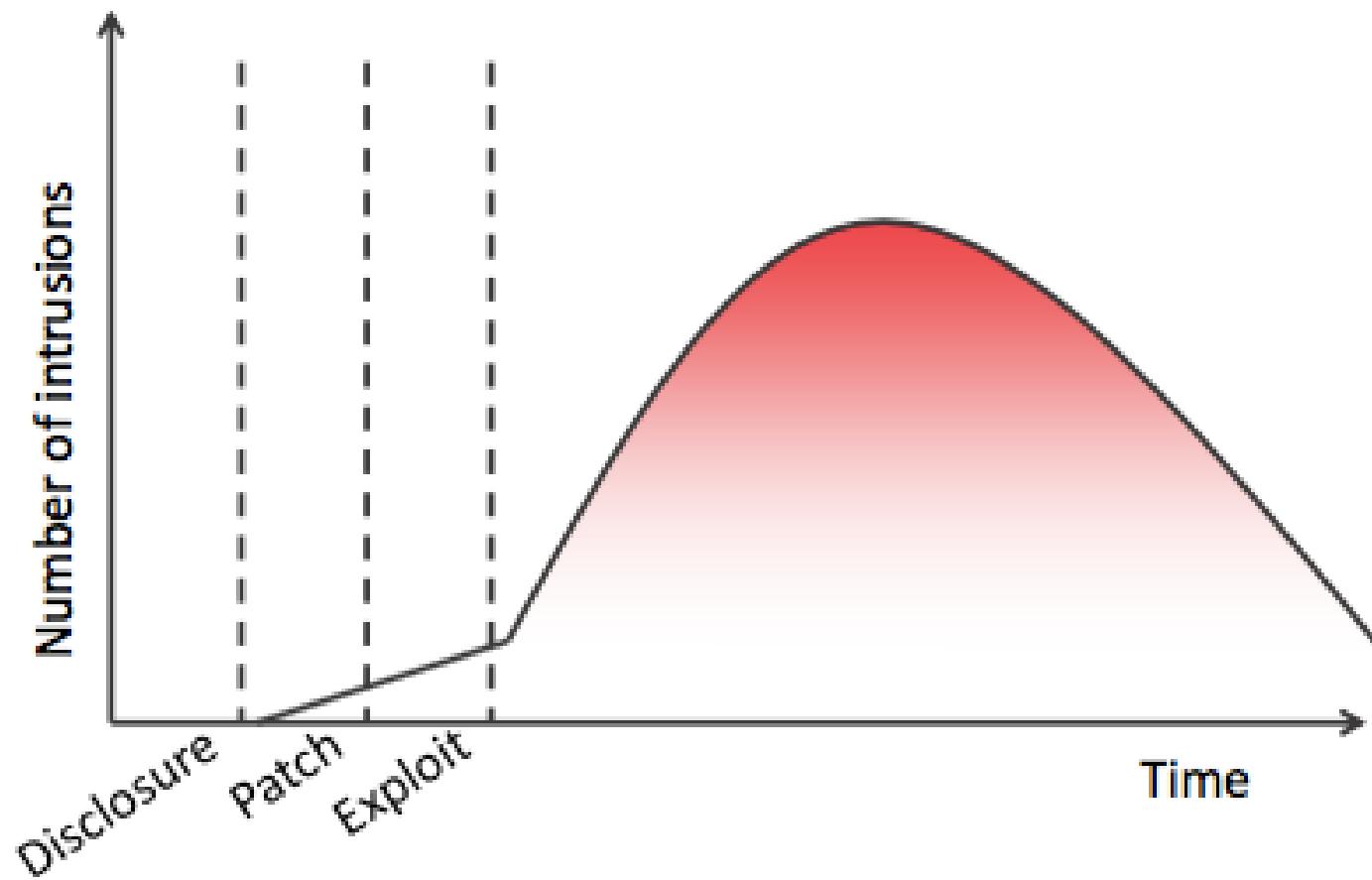
- 1988: The Morris Worm, affects BSD Unix machines. One of the first known buffer overruns
- 2000: The ILOVEYOU worm, one of the most damaging worms ever, shows how powerful social engineering can be



LOVE-LETTER-FOR-YOU.txt.vbs

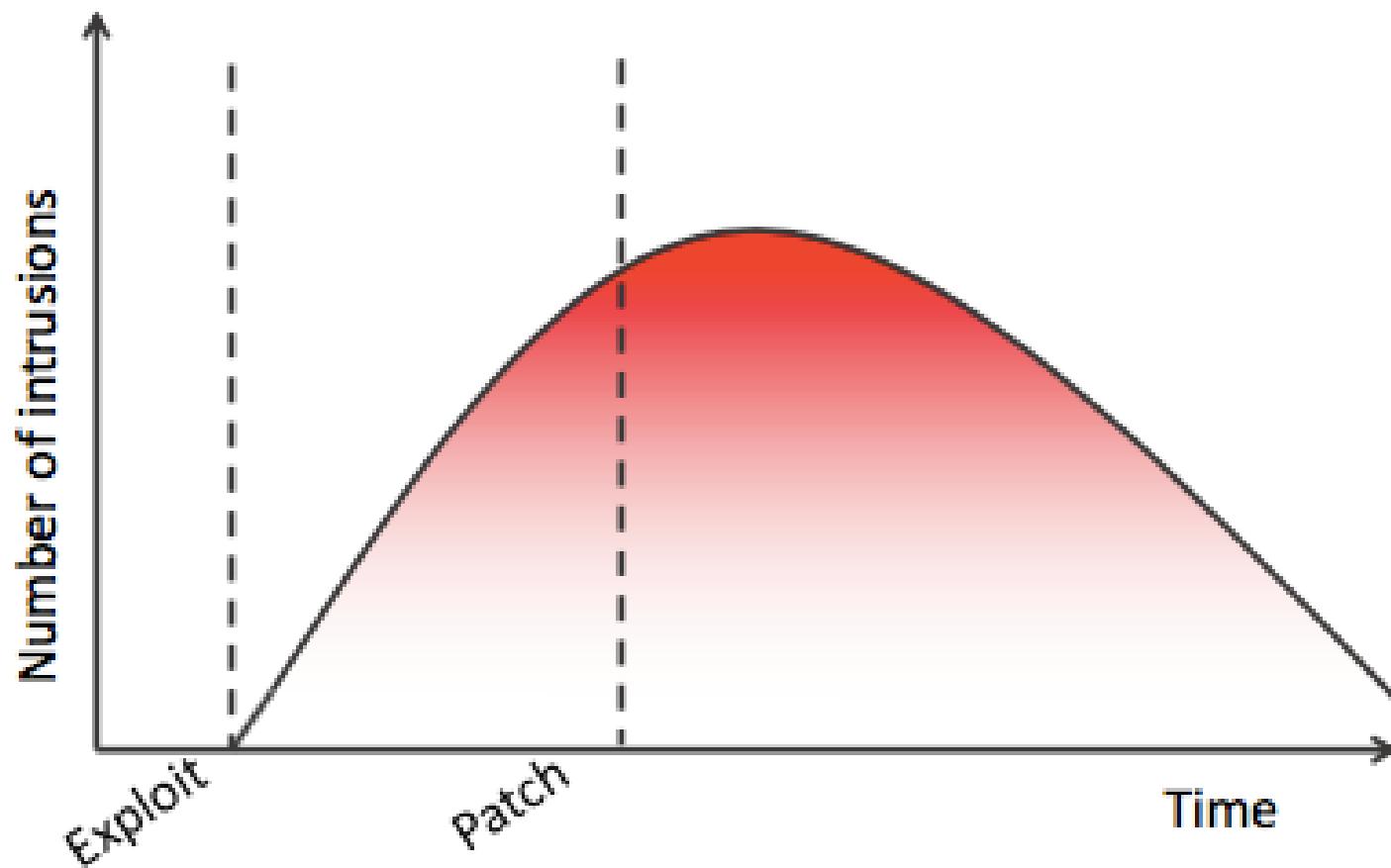
# Exploit Life Cycle

- Many exploits are reversely engineered from patches, or developed simultaneously to patches



# Zero-Day Exploits

- An exploit that is previously unknown – by far the most dangerous



# Trojans

---

- A malicious program pretending to be a legitimate application
  - For example, an application to “clean” hardware, but instead infects the host with adware and spyware
- Often obtained in email attachments or on malicious websites
- Don’t replicate themselves – user error



# Ransomware

- Since about 2013, the rise of Ransomware



# Ransomware

---

- Usually encrypts or blocks access to files and demands a ransom
- It is a clever attack, because if an anti-virus (AV) system removes it, it is often too late
- Usually distributed on malicious websites, or to already infected machines
- The file decryption keys are protected by encrypting using the public key of a C&C server

# Ransomware Variants

---

- Most of the challenge in successfully using ransomware is tricking a user into running it, and bypassing AV and browser protections
  - Fake emails
  - Malicious web pages
  - Obfuscated Javascript (JS) attachments
  - Deployed using exploit kits

From: c.makee@duvalschools.com  
To: yourmail@nottingham.edu.cn

Subject: My Resume

Hi, my name is Carlos Mckee  
Please find my resume in the attachment

Thank you,

Carlos

# Backdoors

---

- Once you have access to a system, a backdoor can be used to provide easy access
- These often work in a client-server architecture, awaiting commands from a central server
  - Remote Administration Tool (RAT)
- More discrete methods will involve subtly changing existing software, or the kernel itself
- Reverse shells – connect back for instructions

# Rootkits

---

- Hide the presence of malicious code by hiding it from the operating system's process table
- Often activated before or during a boot
- Used to hide other malware from standard countermeasures
- Often installs inside the kernel itself, hooking into system call tables, or loads from the master boot record

# Exploits

---

- The easiest way of getting access to a machine is having the user to install something for you – not always possible!
- Failing this, we need an exploit that defeats the security perimeter put in place by the operating system

# How is this allowed to happen?!

---

- There are numerous reasons that exploits can be found in software
  - Programming errors
  - Unchecked user input
  - Incorrect assumptions
  - Weak APIs
  - Bad protocols
- Damage is often worse due to the widespread use of some libraries

# Bad Software

---

- Bad software is everywhere:
  - NASA Mars Lander (\$165m) – Conversion of units
  - Ariane-5 Rocket (\$500m) – Integer overflow
  - Cancer radiation treatment (6 patients died) – Radiation measure malfunction
- Just search google for “worst software bugs”!
  - ... or look back at some of our slides in SQA, FSE,...

# Memory Management

---

- In C and C++, the programmer performs memory management
- Flexible, powerful, fast, but dangerous
  - Buffer Overruns
  - Stack Overruns
  - Heap Overruns
- Memory-managed languages avoid this, but of course may have their own vulnerabilities

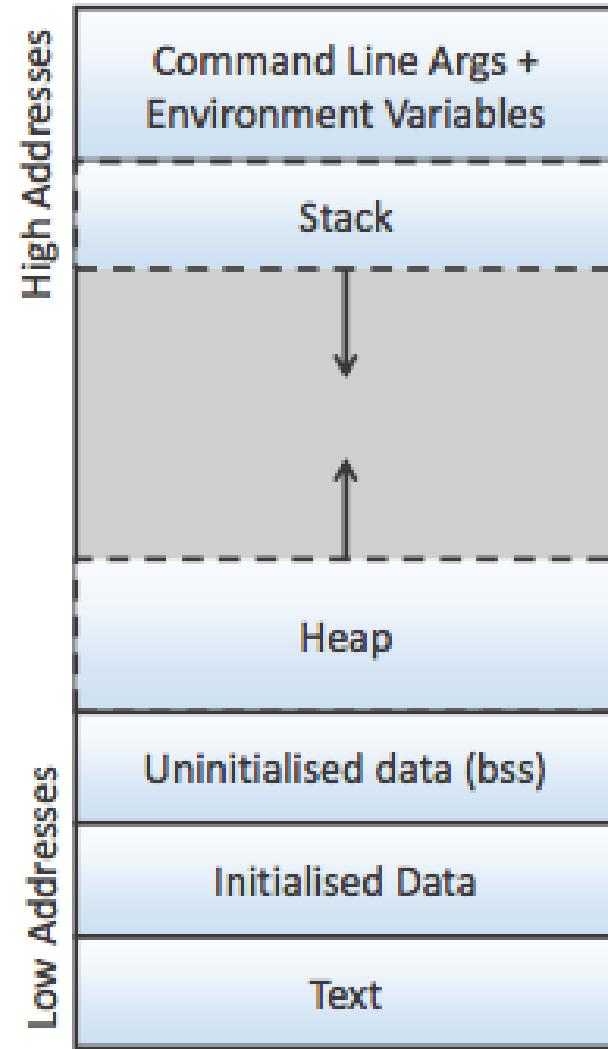
# Buffer Overflows

---

- When a program is executed, contiguous blocks of memory can be allocated to store arrays (buffers)
- If data is written into a buffer that exceeds its size, an overflow occurs
- The data will overwrite the memory beyond the buffer – Bad!

# Program Memory

- Memory is stored in a virtual address space from 0x00000000 to 0xFFFFFFFF (32-bit)
- Parts of the program are held in different regions by convention
- Different restrictions are placed on these regions

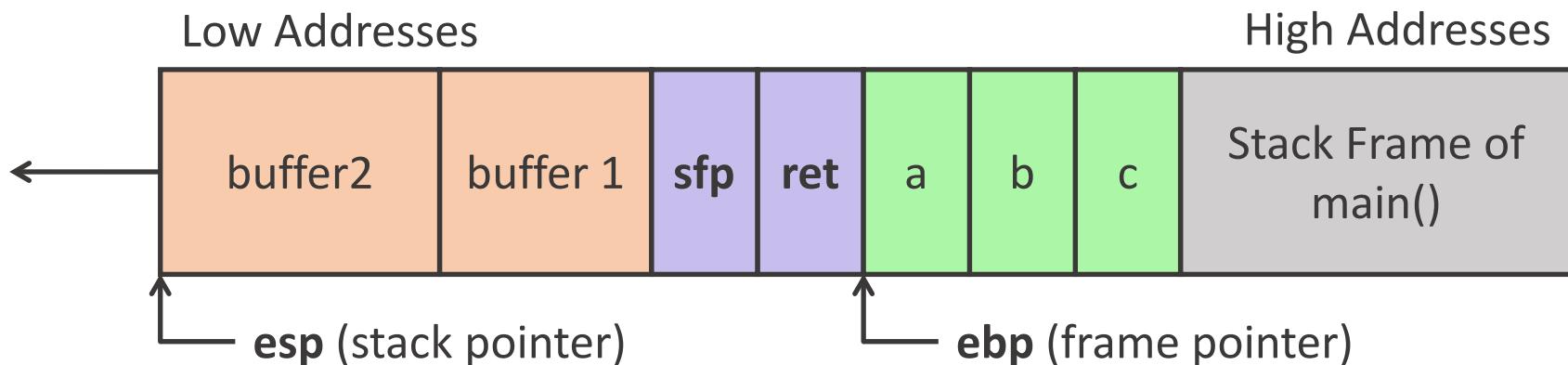


# The Stack

- The stack holds information on local variables and functions calls (stack frames)
- A function call will push a new frame onto the stack
- A return will pop it off, and go to **ret**

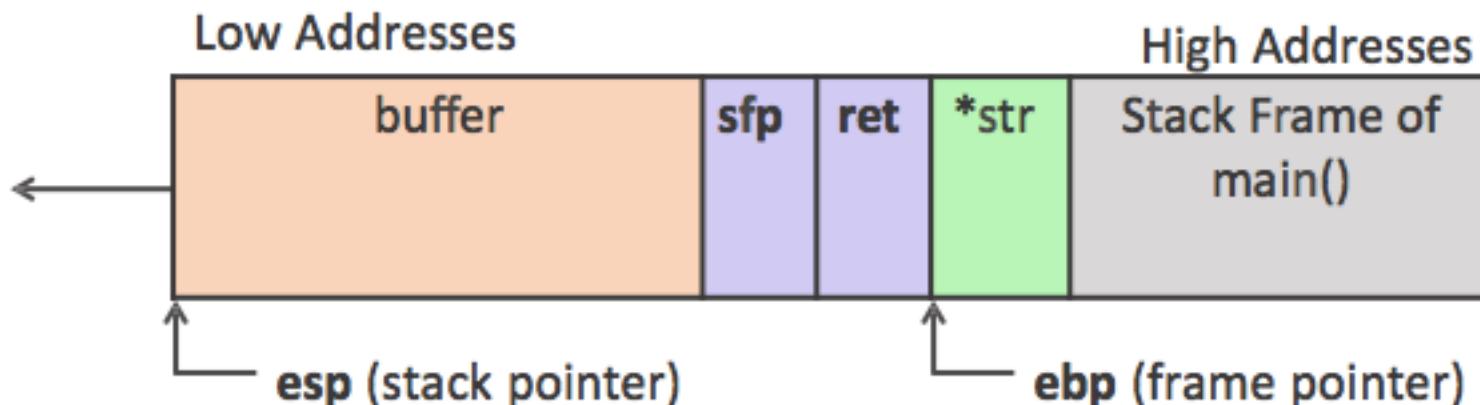
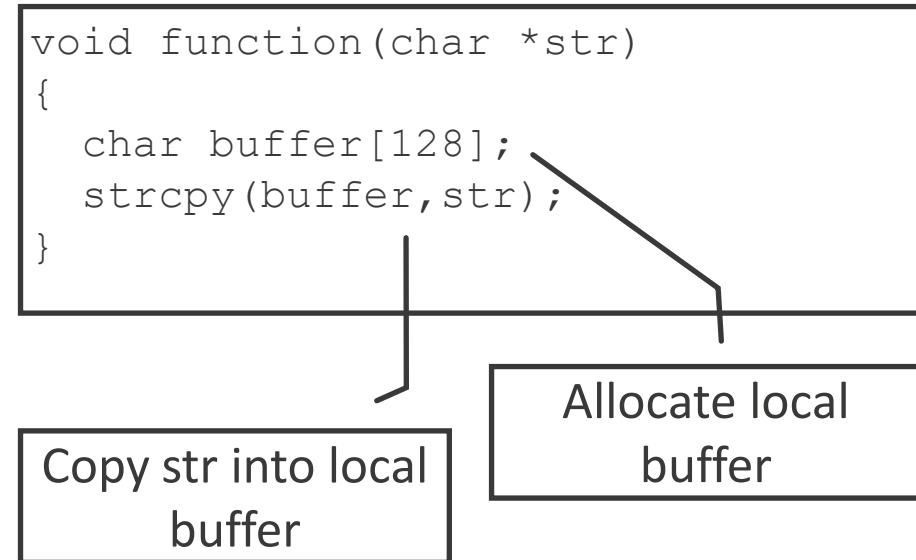
```
void function(int a, int b, int c)
{
    char buffer1[5];
    char buffer2[10];
}

void main()
{
    function(1,2,3);
}
```



# Stack Smashing

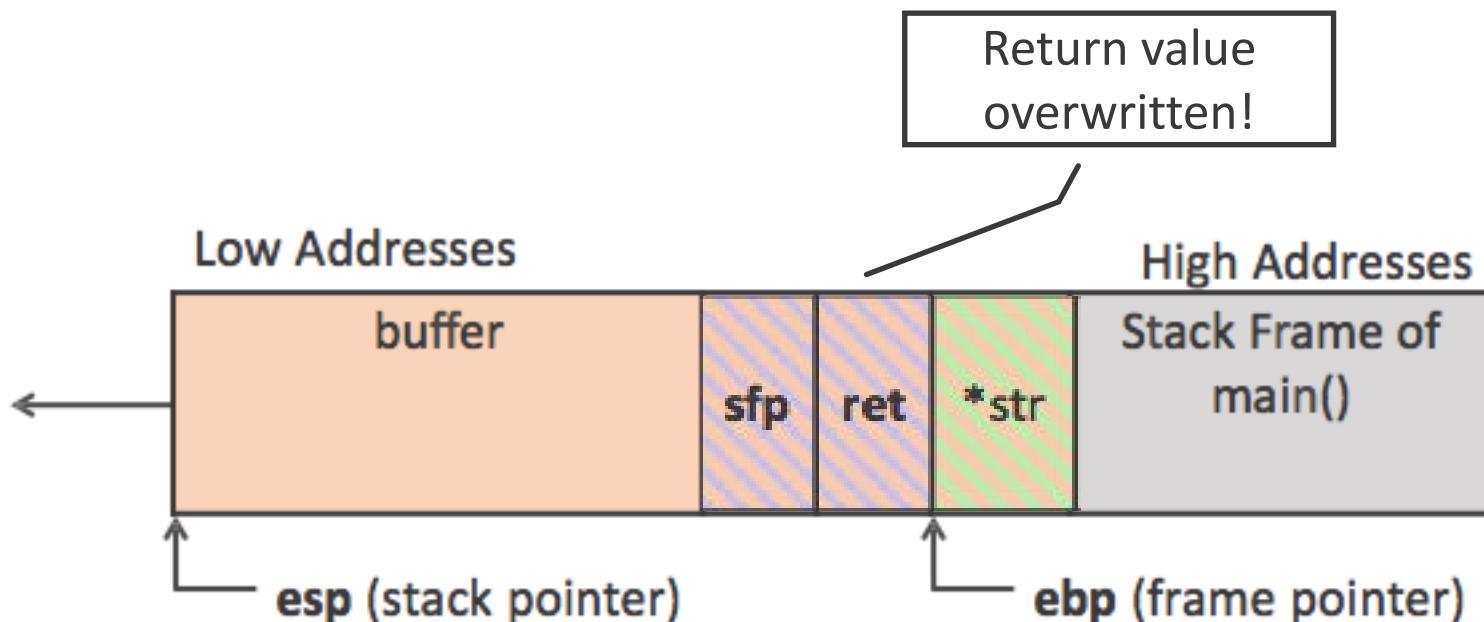
- In C and C++, low level functions like strcpy perform no bounds checking at all
- If str is long, we can write into other memory



# Stack Smashing

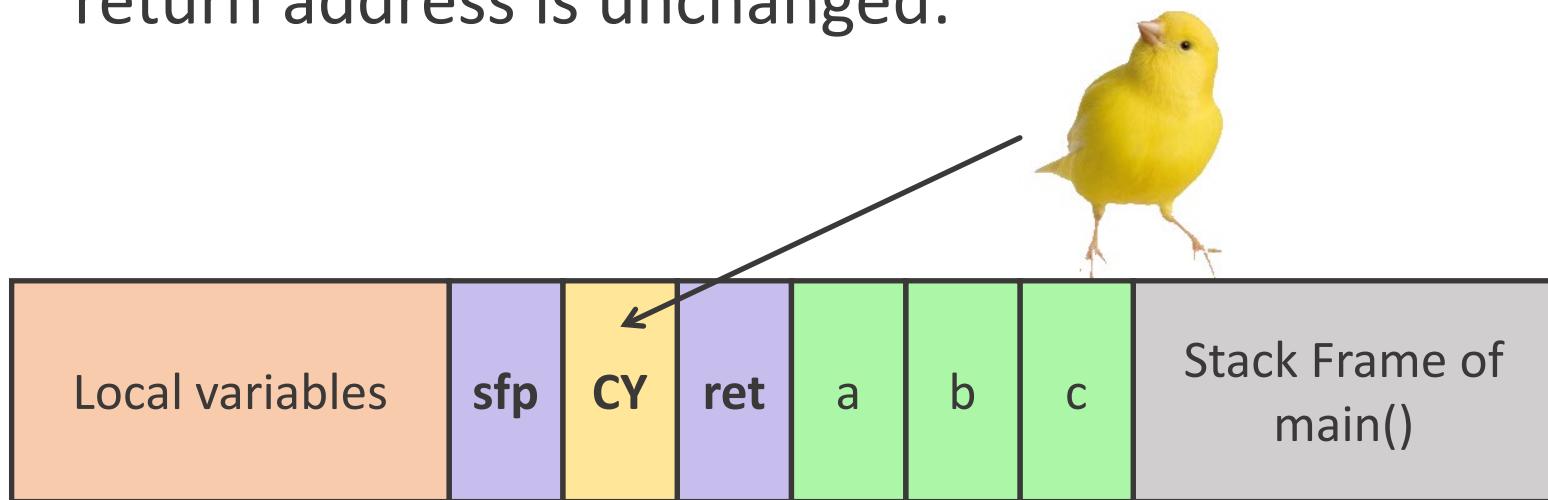
- By crafting the string str, we can overwrite the buffer and the return address with custom exploit code!

```
void function(char *str)
{
    char buffer[128];
    strcpy(buffer, str);
}
```



# Protection: Canaries

- Stack canaries modify the prologue and epilogue of all functions to check a value in front of the return address is unchanged:



- If you can work out the canary value, there is no issue. You could also corrupt the Structured Exception Handler (SEH)

# Data Execution Prevention (NX)

---

- Modern operating systems (where possible) will mark the stack as non-executable.
  - NX on AMD, XD on Intel, XN on arm
- An NX stack means that adding in our exploit code won't work
- We can circumvent this using a return-to-libc attack

# Further Protection

---

- To defeat ret2libc, various 0x0 null bytes are inserted into standard libraries
- Developers also restrict access to obvious system calls
- Address Space Layout Randomisation (ASLR) moves the address of library and programs around
- They don't have to move too much before your hand-crafted **ret** addresses will break

# Race Conditions

---

- With concurrent threads or processes, timing can lead to security vulnerabilities:

**Victim**

```
if (!access("file", W_OK))  
{  
    exit(1);  
}  
  
fd = open("file", O_WRONLY);  
write(fd, buffer,  
      sizeof(buffer));
```

**Attacker**

# Race Conditions

- With concurrent threads or processes, timing can lead to security vulnerabilities:

Victim

```
if (!access("file", W_OK))  
{  
    exit(1);  
}  
  
fd = open("file", O_WRONLY);  
write(fd, buffer,  
      sizeof(buffer));
```

Attacker

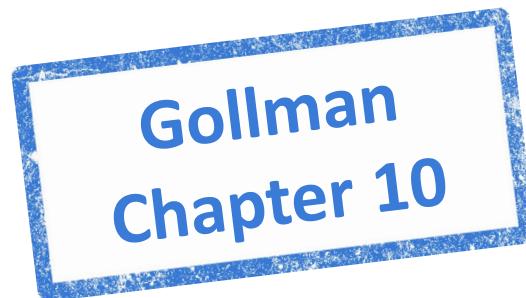
```
...  
symlink("/etc/passwd", "file");  
...
```

r00t::0:0:Owned:/root:/bin/bash

# Summary

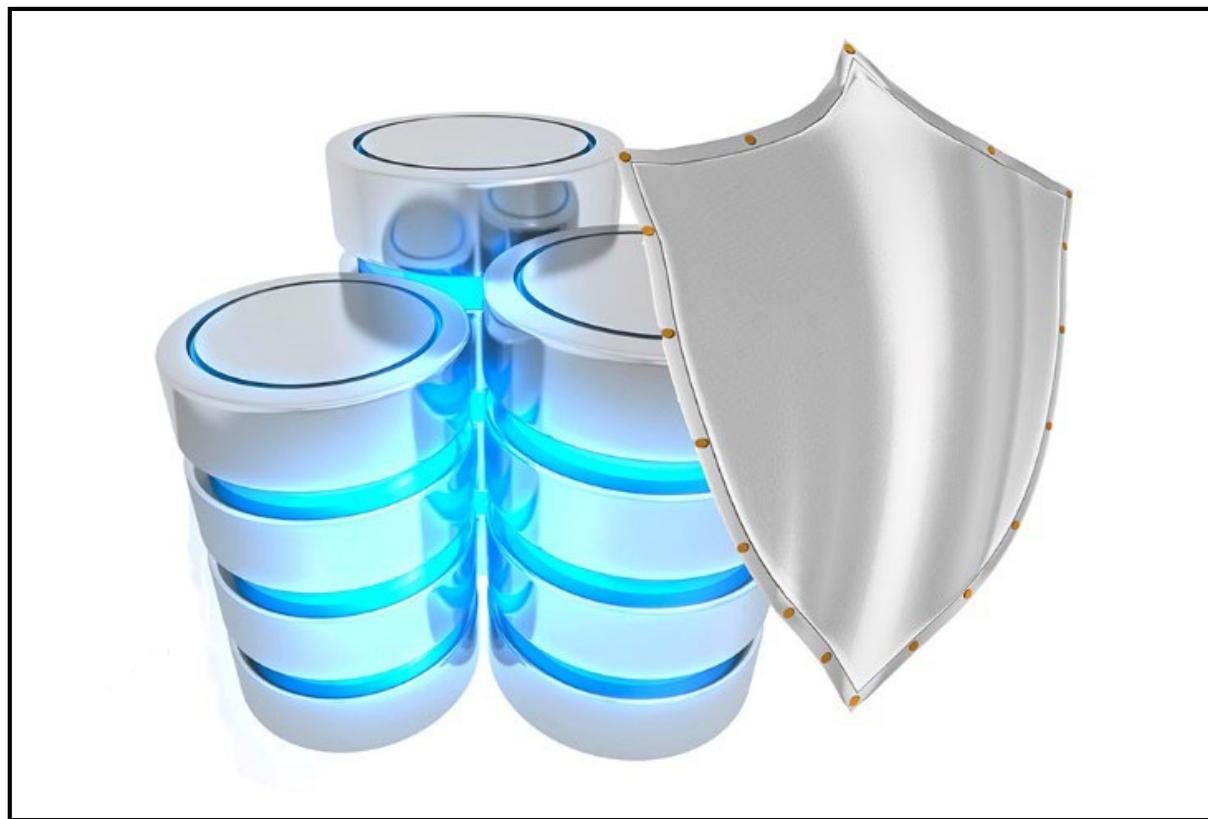
---

- Malware
- Viruses
- Worms
- Trojans
- Ransomware
- Rootkits and Backdoors
- Exploits
- Coding Flaws
- Common Vulnerabilities
- Buffer overflows
- ...



# COMP3052.SEC Computer Security

## Session 14: Database Security



# Acknowledgements

---

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Michael Pound, Dave Towey...

# This Session

---

- Database Security
  - Privileges
  - SQL Security
  - Views
- Statistical Database Trackers
- SQL Injection

# Introduction

---

- Database security is concerned with information
  - Can look at the content
  - More man (or woman) than machine oriented

# Protecting Information

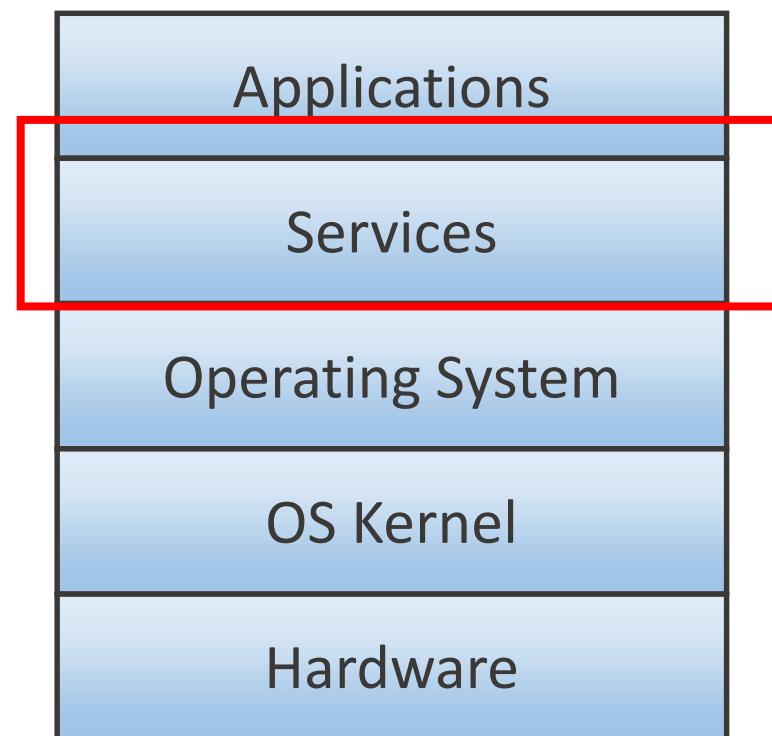
---

- Protecting sensitive information is hard
- Attackers may be interested in many types of information:
  - Exact data
  - Bounds
  - Existence / Negative results
  - Probable value
- Balance protecting sensitive information and utility

# Security Model

---

- DBMS security is defined at the services layer, above the kernel and OS
- Some security may be in the application layer, e.g., view-based policies
- DBMS enforces access control policies and maintains consistency



# SQL Security

---

- Three Entities
  - Users
  - Actions
  - Objects
- Users invoke actions on objects
- Newly created objects are owned by the creator
- Privileges can be granted:
  - Granter, Grantee, Object, Action, Grantable

# Privilege Granting / Revoking

---

```
GRANT SELECT, UPDATE (Day, Flight)  
ON TABLE Diary  
TO Sam, Zoe  
WITH GRANT OPTION
```

```
REVOKE UPDATE  
ON TABLE Diary  
FROM Sam
```

- Grant revocation cascades to all grantees of revoked grantee – safer than not doing this

# View-based Security

---

- Views are derived relations:

```
CREATE VIEW pharm_order AS  
    SELECT DrugDB.Name, SUM(Total)  
        FROM Patients, DrugDB  
        GROUP BY (DrugDB.Name)  
    WITH CHECK OPTION
```



# Why use views?

---

- Views are a flexible way of creating policies closer to application requirements
- Views can enforce context-dependent and data-dependent policies
- Views can implement **controlled invocation**
- Data can be easily reclassified

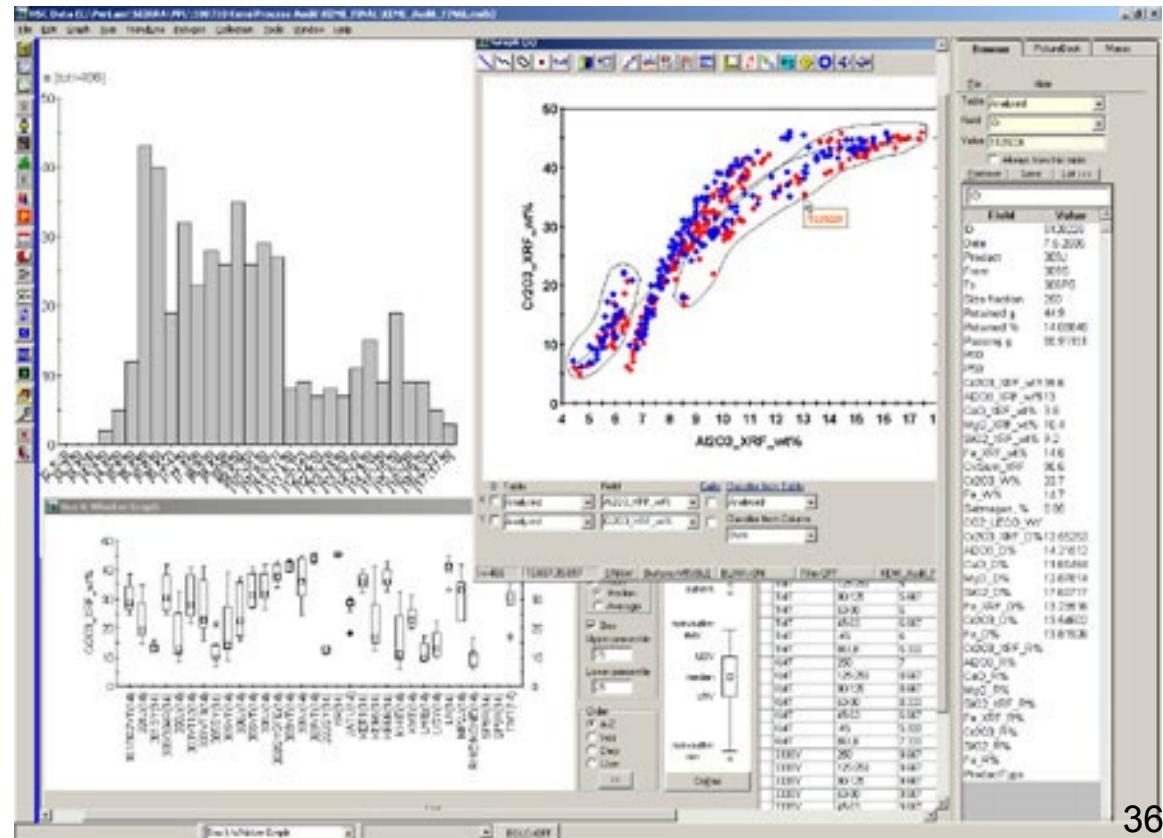
# Why not?

---

- INSERT / UPDATE actions depend on the CHECK options, else might be **blind inserts**
- Definitions must be correct in order to capture intended security policy
- Completeness and consistency are not achieved automatically
- Can quickly become very inefficient

# Statistical Databases

- Where access to data is restricted, access to aggregates might still be permitted:
  - COUNT
  - SUM
  - AVG
  - MAX
  - MIN



# Inference

---

- Since individual items are sensitive, we cannot permit access
- Statistical queries are useful, but by definition refer to the data
- Some queries can reveal information on the underlying data – **Covert Channel**

# Tracking

---

- Direct attack
  - Aggregate is computed to capture information of individual data elements
- Indirect
  - Combines information from several aggregates
- Tracker Attack
  - Generalised indirect attack

# Salaries

---

- $S$  = The sum of all salaries in the department
- $T$  = The sum of all salaries for the department except those that have “Head of School” as “Position”
- Boss’ salary =  $S - T$

*Do not allow sets of just one*

# Salaries

---

- $S$  = Sum of all salaries
- $T$  = Sum of all salaries of women, and anyone whose first name is Albert
- $U$  = The sum of all men's department salaries
- Albert's salary =  $T + U - S$

*Do not allow conditions that refer to just one*

# Salaries

---

- $S$  = Sum of all salaries
- Number of department heads named Albert is not allowed
- $T$  = sum of all salaries for those named Albert
- $U$  = The sum of all salaries for department heads
- $V$  = The sum of all salaries for those who are not department heads, and not named Albert
- Salaries of DHs named Albert =  $V + T - S$

# Further Defences

---

- Data swapping – Swap records but keep stats the same
- Noise addition – Alter aggregate output (a little)
- Table splitting – Separate data completely
- User tracking – Log queries

# SQL Injection Attacks

- It's common for user input to be read (e.g., in a web form) and then used within an SQL query:

<https://insecure-website.com/products?category=Gifts>

```
$query = "SELECT * FROM products WHERE category =  
'Gifts' AND released = 1";
```

- Unexpected user input can completely rewrite the query. An attacker can construct an attack like:

<https://insecure-website.com/products?category=Gifts'-->

```
$query = "SELECT * FROM products WHERE category =  
'Gifts'--' AND released = 1";
```

- It no longer includes AND released = 1. This means that all products are displayed, including unreleased products

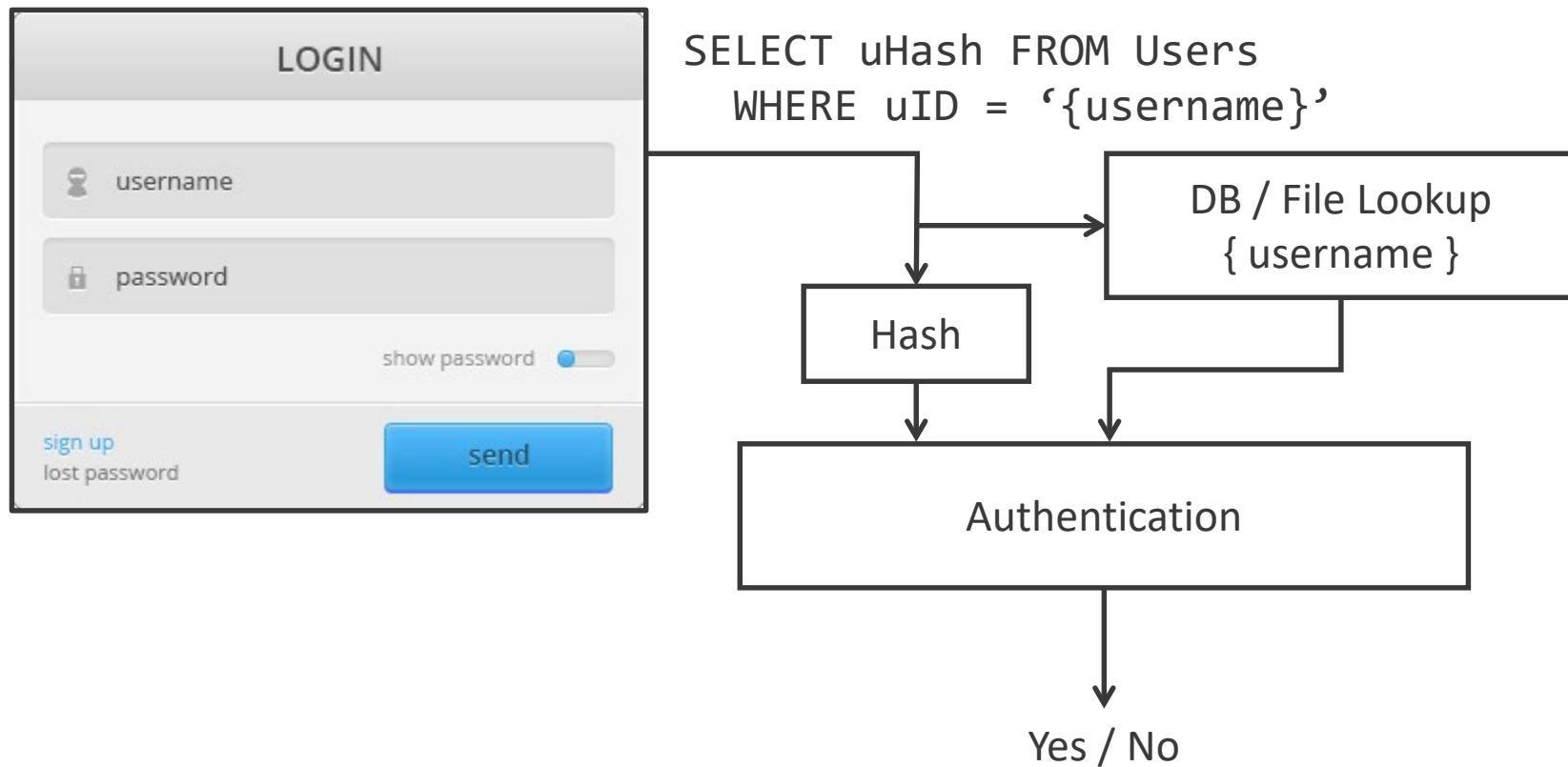
# SQL Injection Attacks

---

- An application or website is vulnerable to an injection if it doesn't **filter SQL control characters**:
  - ' represents the beginning or end of a string
  - ; represents the end of a command
  - /\*...\*/ represent comments
  - -- represents a comment for the rest of the line

# SQL Injection Attacks

- Login pages will request hashes from the database



# Retrieving from other DB Tables

---

- An attacker can leverage a SQL injection vulnerability to retrieve data from other tables within the database
- This is done using the **UNION** keyword

```
$query = "SELECT name, description FROM products  
WHERE category = 'Gifts' UNION SELECT username,  
password FROM users--";
```

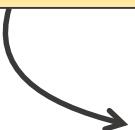
- This will cause the application to return all usernames and passwords along with the names and descriptions of products

# UNION

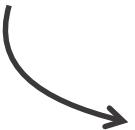
---

- UNION appends (not joins) two tables together
  - They must have the **same number of columns**

```
http://shop.com/search.php?terms=hammers+nails
```

 Returns a table of items and prices and quantity of any items matching the terms hammers and nails

```
http://shop.com/search.php?terms=hammers+' UNION SELECT  
1,ids,hashes FROM users;--
```

 Appends the user table!

# Blind SQL Injection

- Most servers won't directly output SQL errors to the screen
- A **blind** SQL injection performs database analysis without any actual output

```
http://shop.com/items.php?id=
```

```
http://shop.com/items.php?id=2 and 1=1
```

→ Returns item #2

```
http://shop.com/items.php?id=2 and 1=2
```

→ Returns no items found

# Fingerprinting the DB

---

- Some commands are specific to an individual DBMS:

```
http://shop.com/items.php?id=2; waitfor delay '0:0:10'--
```

→ Waits for a while on MS SQL Server but not MySQL

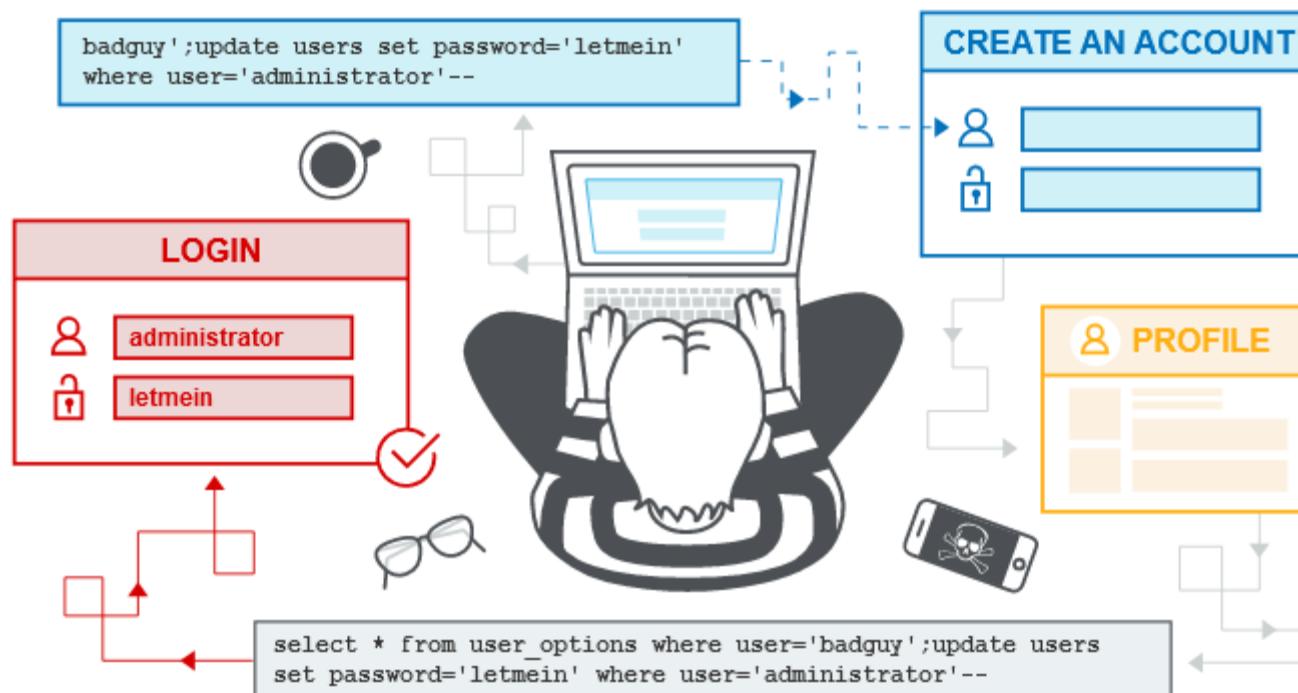
- Once you know the DB, access the system tables:

```
http://shop.com/items.php?id=2 AND 1=(SELECT COUNT(*)  
FROM information_schema.tables WHERE TABLE_NAME='users')
```

→ If an item returns, there is a table named 'users' in MySQL

# Second Order SQL Injection

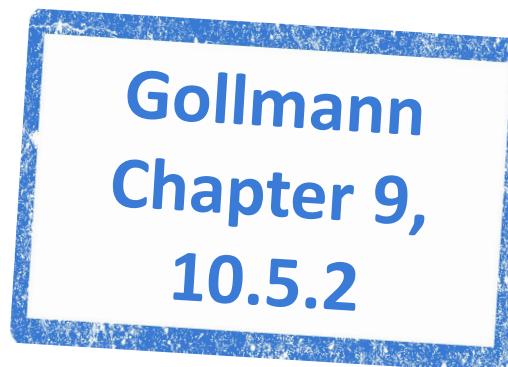
- Entry points may be checked for special characters, but internal functions?
- Store the exploit in one pass, then have it executed later



# Summary

---

- Database Security
  - Privileges
  - SQL Security
  - Views
- Statistical Database Trackers
- SQL Injection (<https://portswigger.net/web-security/sql-injection>)



# COMP3052 SEC Computer Security

## Session 15-1 Crypto I

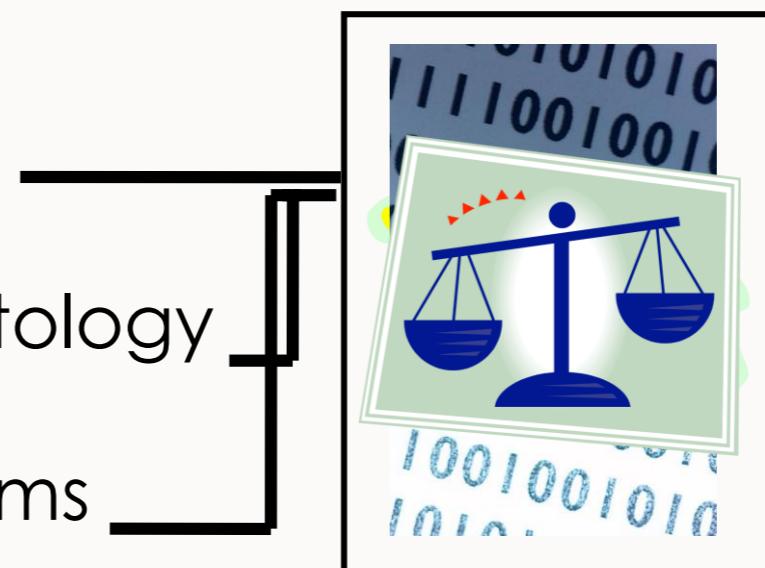
$$\begin{aligned} & \frac{x^2(yf(x) + x^0(x)^2)y_3 + x_2(x)y_2 + x_3(x)y_3}{(x+1)} \\ &= \left(\frac{x(x-2)}{2}\right)1 + (x(x-1))0 + \left(\frac{x(x-1)}{2}\right) \\ &= \left(\frac{(x-1)(x-2)}{2}\right)1 + (x(x-1))0 + \frac{x+1}{2}\left(\frac{x(x-1)}{2}\right) \\ &= \frac{(x+6x^2+1)(y^4-2y^3+8y^2-6y+9)}{(x+1)(x+6)^4(x+9)^4} \\ &= \frac{-9b+\sqrt{3}\sqrt[3]{4a^3+27b^2})y^3+6x^2(y+10x+4\sqrt{3})x+1}{2^{17/3}3^{2/3}} \\ &= \frac{x(x+6)^2}{2^{17/3}3^{2/3}} + \frac{(y+9x+1)(y+8x)^2}{(y+8x)^2(y+7x+4)^4(y+9)} \\ &= \frac{(1-i\sqrt{3})(-9b+\sqrt{3}\sqrt[3]{4a^3+27b^2})}{4/3\cdot 2^{1/3}x+9} \end{aligned}$$

# ACKNOWLEDGEMENTS

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towey ...

# TOPICS COVERED

- Concepts of Cryptology
- The Mathematics of Cryptology
- Algorithms and Mechanisms
  - Integrity Checking
  - Digital Signatures
  - Encryption
- Assessment of Algorithms and Mechanisms



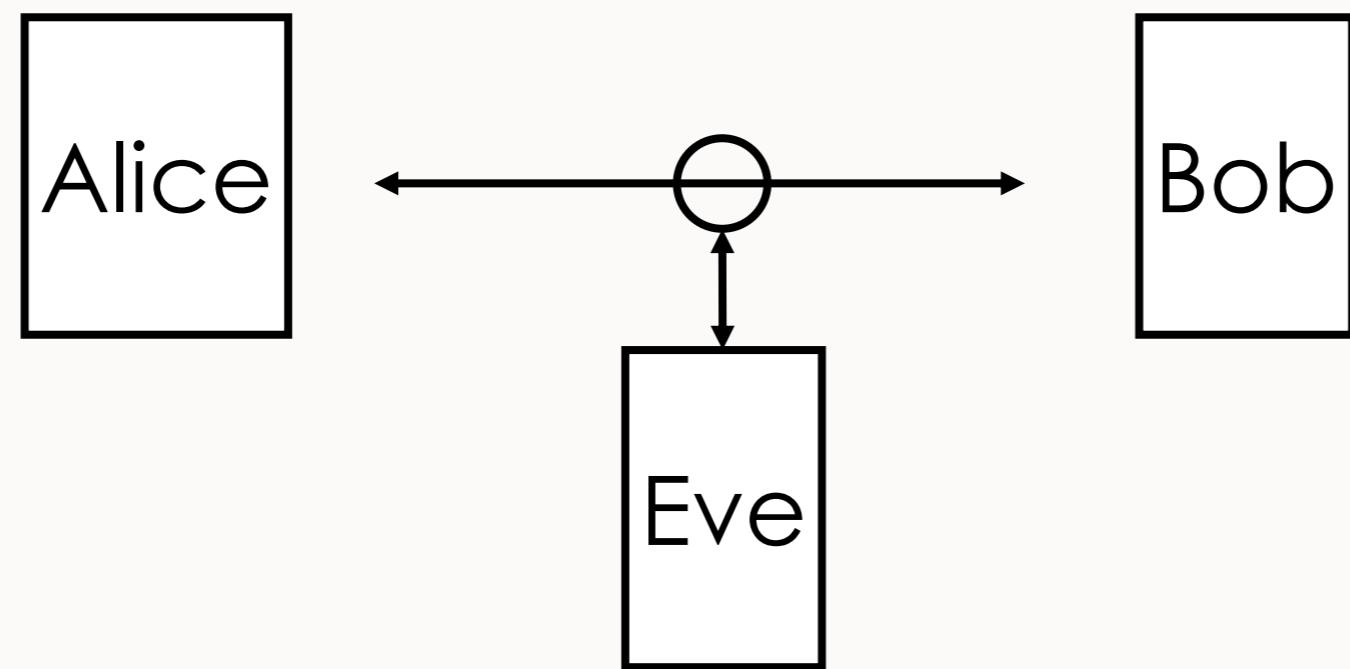
# DEFINITIONS

- **Cryptography**

- **Cryptanalysis**

- **Cryptology**

# THE COMMUNICATION MODEL



# ALL ABOUT EVE

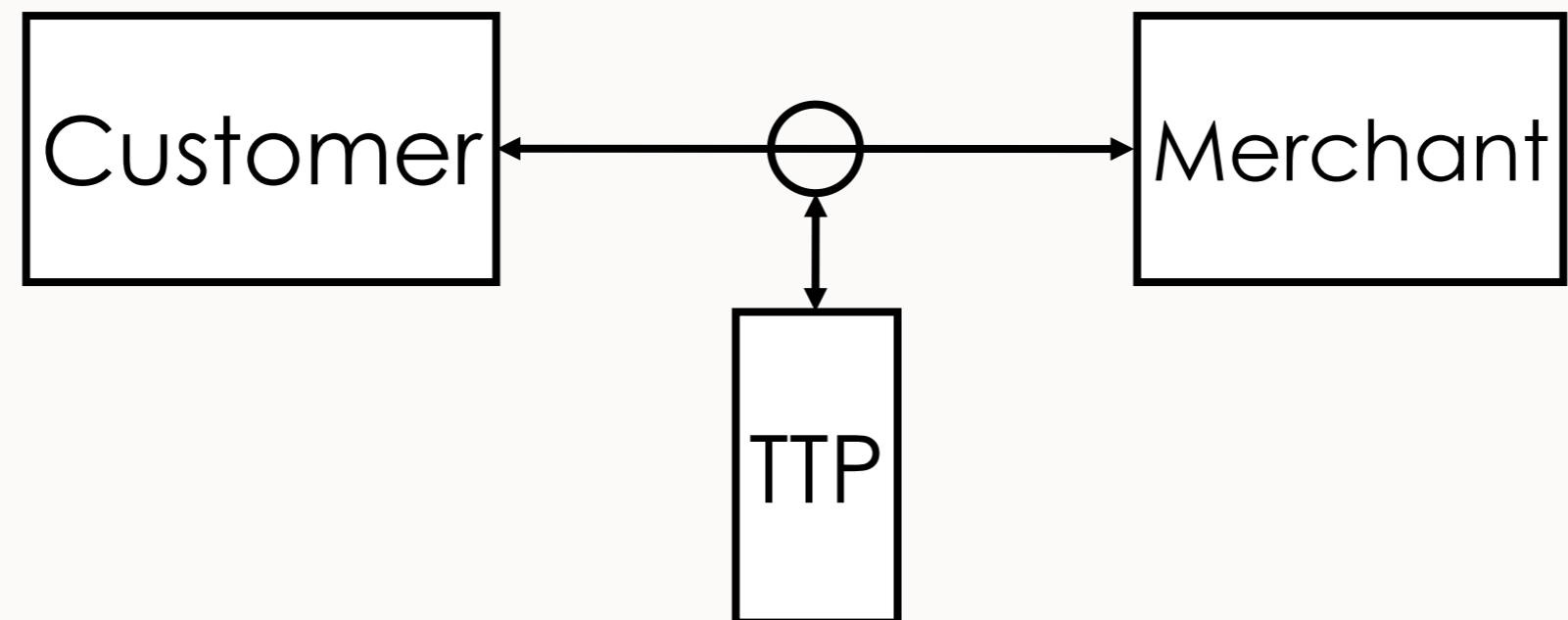
## Objectives

- Read messages
- Edit Messages
- Write messages

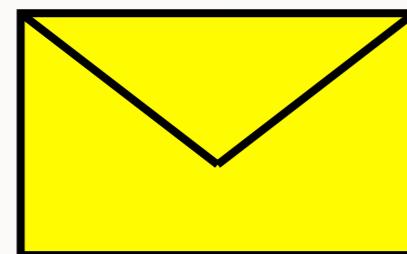
## Countermeasures

- Encryption
- Integrity Checking
- Origin Checking

# THE COMPUTER SECURITY MODEL



# CRYPTOGRAPHY PARADIGMS

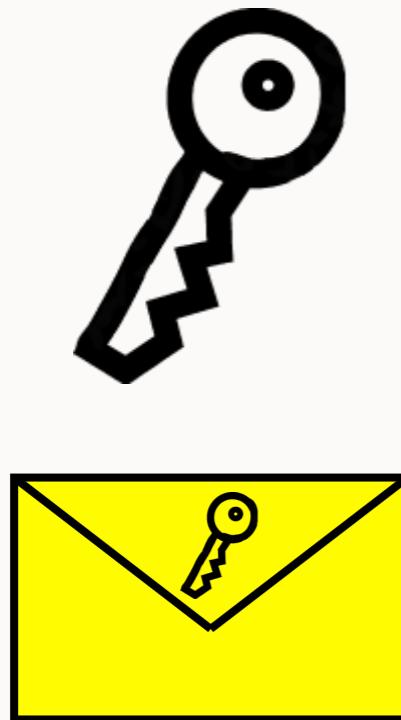


SECRET  
PROCESS

# ACTIVITY ...

- Please answer:
  - What happens if someone reversely engineers/sells your process?
  - Who verifies that your technique is strong enough?

# THE NEW PARADIGM



Known  
process



# KEYS AND STRENGTH

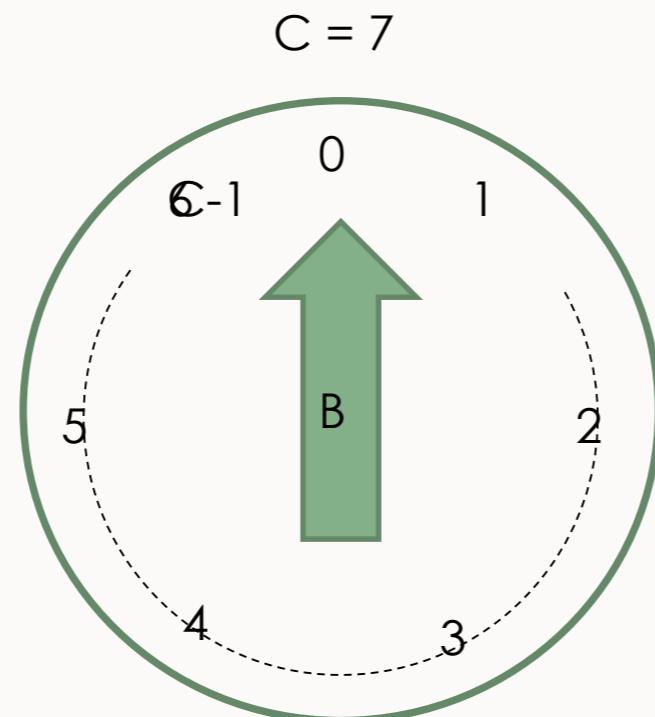
- “Security is as strong as its weakest link”
  - Bruce Schneier
- Keys vary in strength
- Algorithms vary in strength
- “Brute Force” attacks

# KEY MANAGEMENT

- Where do we generate keys?
- How do we generate keys?
- Where are keys stored?
- How do we transport keys?
- Where are keys used?
- How are keys revoked and replaced?

# MODULUS MATHEMATICS

**A = 80**



# MODULUS MATHEMATICS

$$A \equiv B \pmod{C}$$

# PROPERTIES OF THE MODULUS

Let

$$a_1 \equiv b_1 \pmod{C}$$

$$a_2 \equiv b_2 \pmod{C}$$

Then

$$a_1 + a_2 \equiv (b_1 + b_2) \pmod{C}$$

$$a_1 a_2 \equiv (b_1 b_2) \pmod{C}$$

# PROPERTIES OF THE MODULUS

- Constrained to integers so division is complex
- Let's define division as the opposite of multiplication
  - If  $f = (1/e) \text{ mod } C$ 
    - then  $(e^*f) \text{ mod } C = 1$
  - If multiple prime solutions exist for  $1/e$ , the division is **undefined** like  $X/0$  in regular mathematics

# ACTIVITY ...

- Calculate the value of f:

$$f = (1/3) \bmod 7$$

# ACTIVITY ...

- Calculate the value of f:

$$f = (1/3) \text{ Mod } 7$$

- $f = (1/3) \text{ Mod } 7$
- So  $(3f) \text{ Mod } 7 = 1$
- Possible values of f:
  - $0, 0 \text{ Mod } 7 = 0$
  - $1, 3 \text{ Mod } 7 = 3$
  - $2, 6 \text{ Mod } 7 = 6$
  - $3, 9 \text{ Mod } 7 = 2$
  - $4, 12 \text{ Mod } 7 = 5$
  - **5, 15 Mod 7 = 1**
  - $6, 18 \text{ Mod } 7 = 4$
- So  $f = (1/3) \text{ Mod } 7 = 5$

# UNDEFINED MODULAR DIVISION

- What about
  - $f = (5/5) \text{ Mod } 10$
  - $(5f) \text{ Mod } 10 = 5$

# UNDEFINED MODULAR DIVISION

- What about
  - $f = (5/5) \text{ Mod } 10$
  - $(5f) \text{ Mod } 10 = 5$
  - $0, 0 \text{ Mod } 10 = 0$
  - $1, 5 \text{ Mod } 10 = 5$
  - $2, 10 \text{ Mod } 10 = 0$
  - $3, 15 \text{ Mod } 10 = 5$
  - $4, 20 \text{ Mod } 10 = 0$
- Etc

Undefined – too many prime answers!

There are cases where there are no answers!

# PRIMES AND MODULUS

- Let  $p$  be a prime number
- Let  $a$  be an integer
- iff  $a \not\equiv 0 \pmod{p}$

There is always another integer  $d$  such that:

$$a * d = 1 \pmod{p}$$

# FERMAT'S LITTLE THEOREM

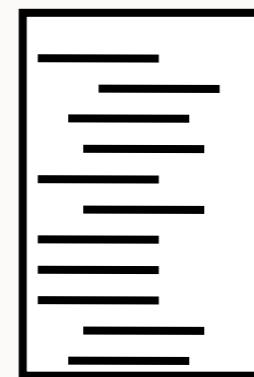
- iff  $a \not\equiv 0 \pmod{p}$

$$a^{p-1} = 1 \pmod{p}$$

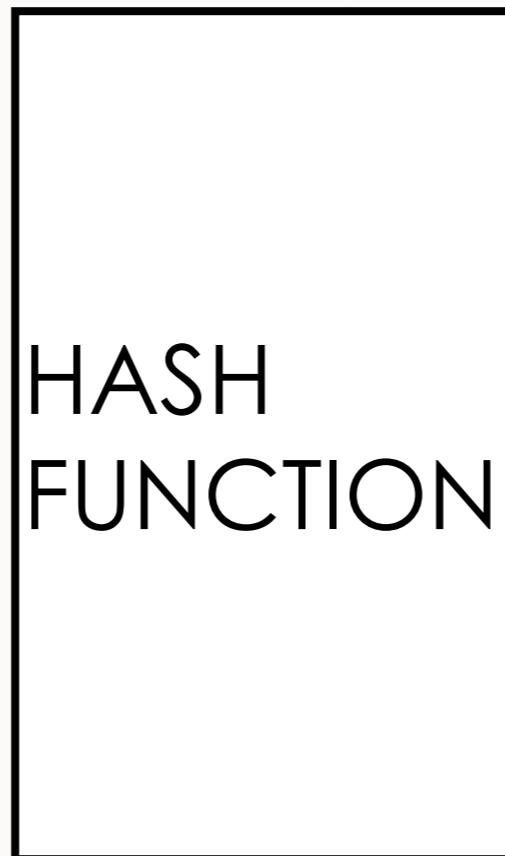
- This yields a good way of testing if a number is prime
  - If you calculate  $a^{p-1} \pmod{p}$  for a series of numbers if there are no 1's it probably isn't prime!

# HASH FUNCTIONS

X



Any number of bits



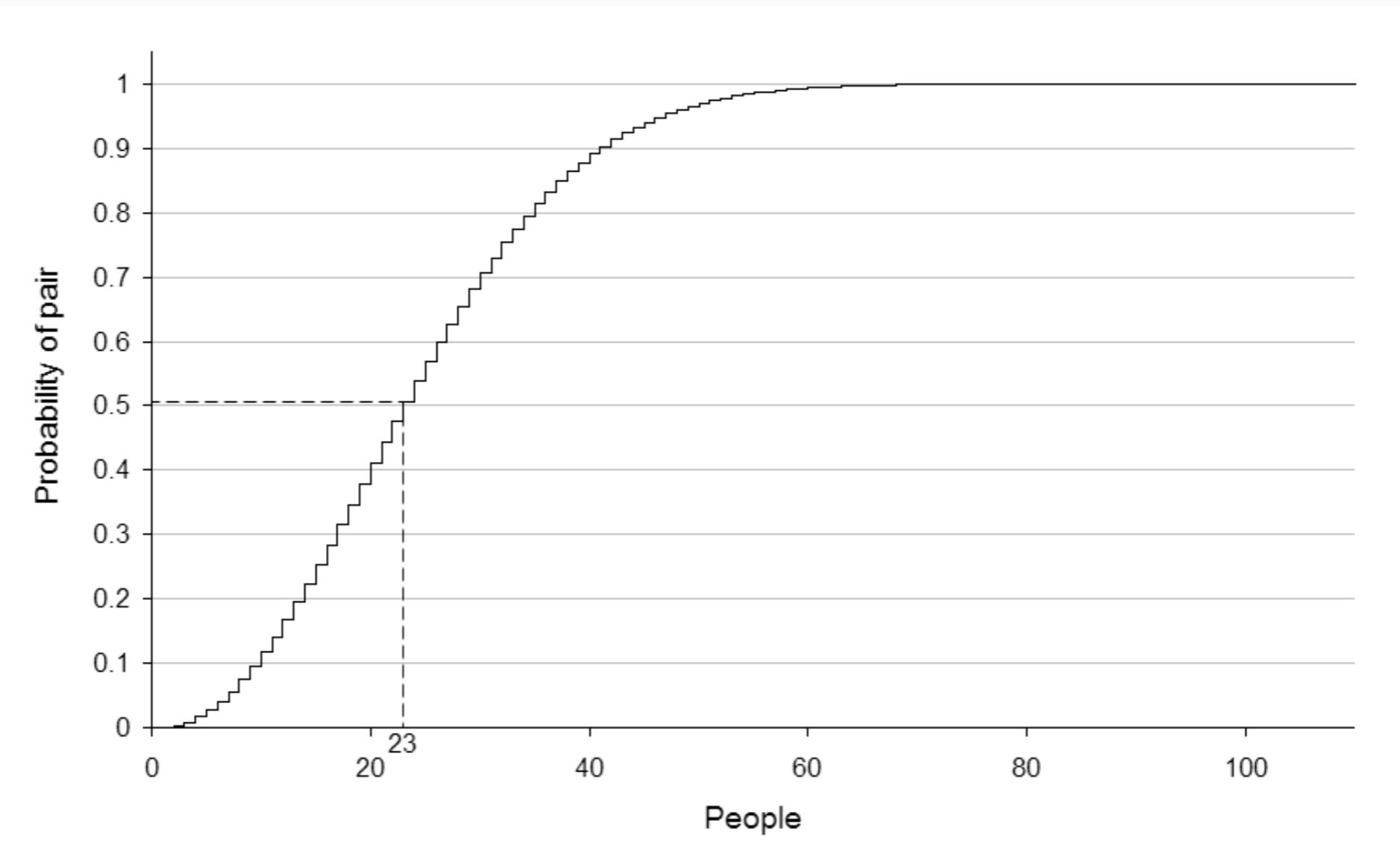
# HASH FUNCTION PROPERTIES

- Compression
  - No matter how long the input is, the output has the same length
- Ease of computability
  - Given  $x$ , it should be easy to find  $h(x)$
- Collision Avoidance
  - It should be “computationally infeasible” to find collisions

# QUESTION/ACTIVITY ...

## THE BIRTHDAY PARADOX

- How many of you share a birthday?



# HASH FUNCTION PROPERTIES

- Given a hash function that produces  $N$  bit hashes
  - If you generate around  $2^{N/2}$  random inputs, you are likely to find a collision

# HASH FUNCTION PROPERTIES

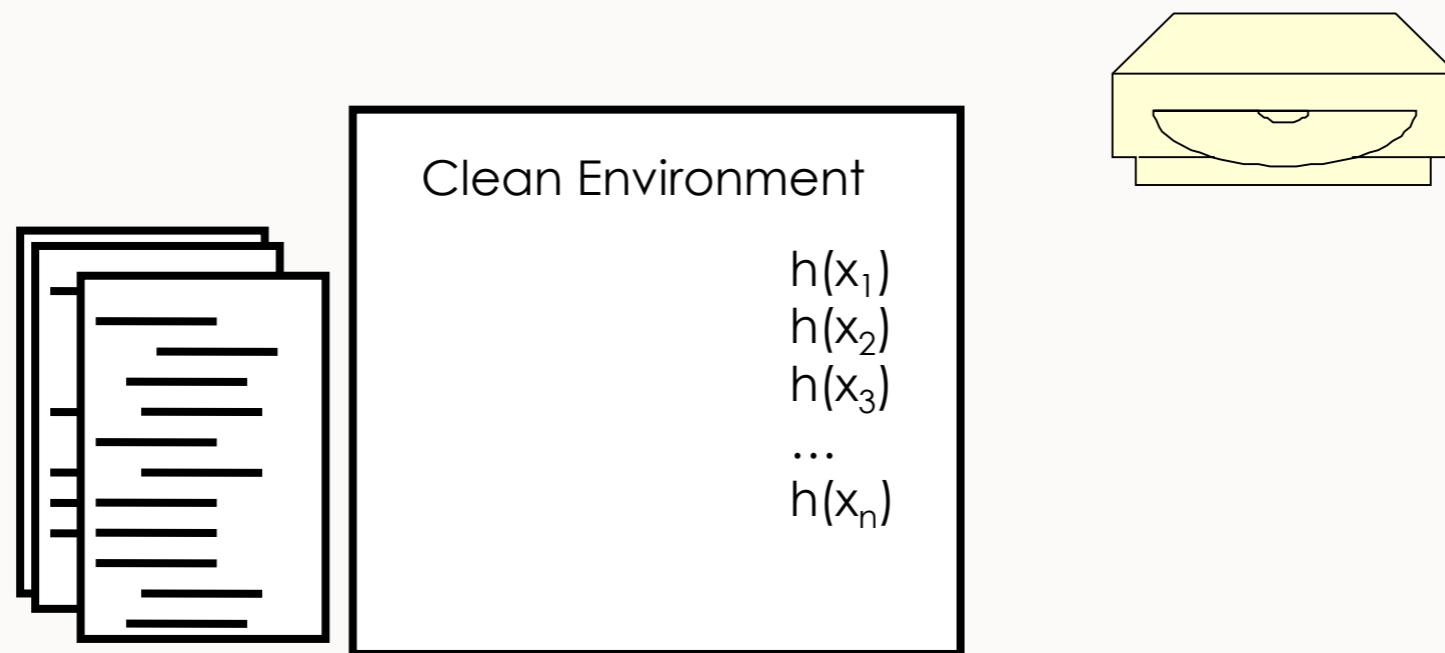
- Preimage Resistance
  - Given  $y$ , it should be “computationally infeasible” to find  $x$  to satisfy:
    - $h(x) = y$
    - Expected number of tries is  $2^{n-1}$  for an  $n$ -bit hash
- Second Preimage Resistance
  - (weak collision resistance)
  - Given  $x$  and  $h(x)$ , it should be “computationally infeasible” to find  $x'$  to satisfy
    - $h(x) = h(x')$

# HASH FUNCTION PROPERTIES

- Collision Resistance
  - (strong collision resistance)
  - It should be “computationally infeasible” to find any  $x$  and  $x'$  that satisfy:
    - $h(x) = h(x')$

# MDC-2

- Modification Detection Codes 2
- A cryptographic hash function



# SUMMARY

- What is cryptology?
- Communication/encryption paradigms
- Modulus Mathematics
- Hash functions
- MDCs

Read:

- Gollman: Chapter 14
- Anderson: Chapter 5

# COMP3052.SEC Computer Security

## Session 15-2 Crypto II: EXTRA MATHS MATERIALS

$$\begin{aligned} & \frac{x^2(yf(2) + 10y^2)y_3 + e_2(x)y_2 + e_3(x)y_3}{(x+1)} \\ &= \left(\frac{x(x-2)}{2}\right)1 + (x(x-1))0 + \left(\frac{x(x-1)}{2}\right) \\ &= \left(\frac{(x-1)(x-2)}{2}\right)1 + (x(x-1))\cancel{0} + \cancel{\left(\frac{x(x-1)}{2}\right)} \\ &= f_p(x, y) \\ & \frac{y^2(y+6x+7)^4(2x^2+y^2+8x)^2(y+9x+5)^4(y+1)}{(x+1)(x+6)^4(x+9)^4} \\ &= \frac{x(x+5)(x+2)^4}{(y+8x+10)^2(x+1)} \\ &= \frac{-9b+\sqrt{3}\sqrt[3]{4a^3+27b^2}(y^3+6x)^2(y+10x+7)^2}{2^{1/3}3^{2/3}} \\ &= \frac{(y+8x)^2}{x(x+6)^2} \frac{(y+9x+5)^4}{(y+8x+10)^2} \\ &= \frac{(1-i\sqrt{3})(-9b+\sqrt{3}\sqrt[3]{4a^3+27b^2})^{4/3}}{2^{4/3}3^{2/3}x+9} \frac{(y+8x+10)^2}{(y+8x)^2(y+7x+4)^4(y+9)^2} \end{aligned}$$

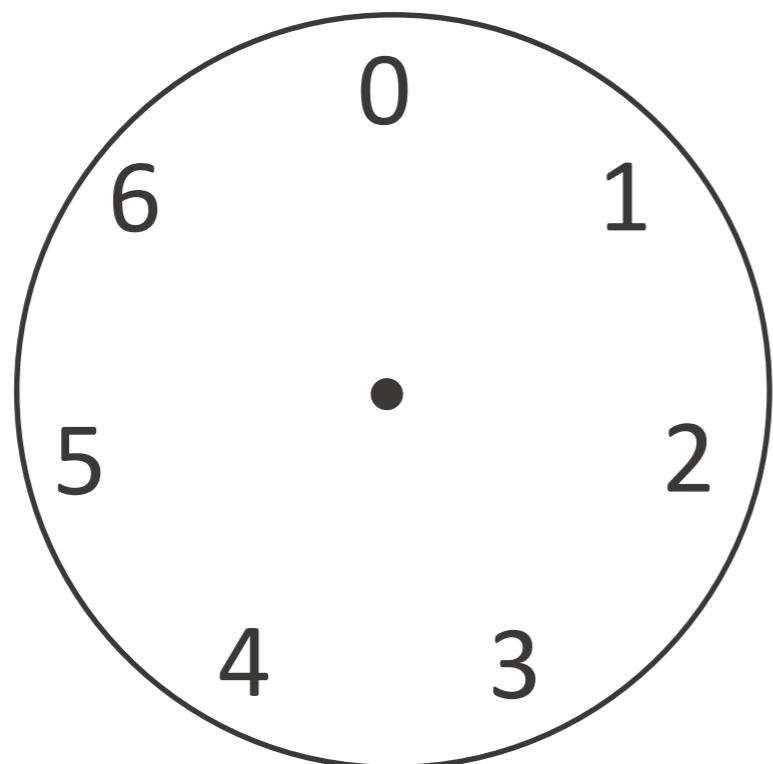
# Acknowledgements

---

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Michael Pound, Dave Towey ...

# Modular Arithmetic

- A system of arithmetic based around cycles of numbers
  - Numbers modulo n are a **finite field**



The set of numbers  
modulo 7

# The Congruence Relation

---

$$a \equiv b \pmod{n}$$

$$a \pmod{n} = b \pmod{n}$$

# Equivalences

---

$$((a \bmod n) + (b \bmod n)) \bmod n = (a + b) \bmod n$$

$$((a \bmod n) \cdot (b \bmod n)) \bmod n = (a \cdot b) \bmod n$$

# Multiplication Example

---

Rule:

$$((a \bmod n) \cdot (b \bmod n)) \bmod n = (a \cdot b) \bmod n$$

Example:  $(29013 \cdot 1123) \bmod 7$

$$32581599 \bmod 7$$

# Multiplication Example

---

**Rule:**

$$((a \bmod n) \cdot (b \bmod n)) \bmod n = (a \cdot b) \bmod n$$

**Example:**  $(29013 \cdot 1123) \bmod 7$

$$32581599 \bmod 7$$

**Or:**

$$((29013 \bmod 7) \cdot (1123 \bmod 7)) \bmod 7$$

$$(5 \cdot 3) \bmod 7$$

$$15 \bmod 7 = 1$$

# Exponentiation

---

Example:  $13^{11} \bmod 7 = ?$

# Exponentiation

---

Example:  $13^{11} \bmod 7 = ?$

$$((13^2 \bmod 7) \cdot (13^9 \bmod 7)) \bmod 7$$



$$169 \bmod 7 = 1$$

# Exponentiation

---

Example:  $13^{11} \bmod 7 = ?$

$$((13^2 \bmod 7) \cdot (13^9 \bmod 7)) \bmod 7$$

$$(1 \cdot (13^9 \bmod 7)) \bmod 7$$

$$(1 \cdot (13^2 \bmod 7) \cdot (13^7 \bmod 7)) \bmod 7$$

etc.

# Exponentiation

---

Example:  $13^{11} \bmod 7 = ?$

$$((13^2 \bmod 7) \cdot (13^9 \bmod 7)) \bmod 7$$

$$(1 \cdot (13^9 \bmod 7)) \bmod 7$$

$$(1 \cdot (13^2 \bmod 7) \cdot (13^7 \bmod 7)) \bmod 7$$

etc.

$$(1 \cdot (13^1 \bmod 7)) \bmod 7$$

$$13 \bmod 7 = \mathbf{6}$$

# Logarithms

---

- A logarithm is the inverse function to exponentiation:

$$a^b = c$$

$$b = \log_a(c)$$

- This is easy to compute even for large numbers

# Discrete Logarithms

---

- When operating mod n, we call the operation a **discrete logarithm**:

$$a^b = c \pmod{n}$$

$$b = \text{dlog}_{a,n}(c)$$

Example:

$$7^2 = 4 \pmod{9}$$

$$2 = \text{dlog}_{7,9}(4)$$

# Discrete Logarithms

- Discrete logs are much harder to compute

$$3^? \equiv 1 \pmod{7}$$

$$? = \text{dlog}_{3,7}(1)$$

# Discrete Logarithms

- Discrete logs are much harder to compute

$$3^? \equiv 1 \pmod{7}$$

$$? = \text{dlog}_{3,7}(1)$$

Brute force:

$$3^1 \equiv 3 \pmod{7}$$

$$3^2 \equiv 2 \pmod{7}$$

$$3^3 \equiv 6 \pmod{7}$$

$$3^4 \equiv 4 \pmod{7}$$

$$3^5 \equiv 5 \pmod{7}$$

$$3^6 \equiv 1 \pmod{7}$$

# Summary

---



- This very brief session was really just to give you a taste (hopefully a reminder) of the kind of maths to be ready to do

# COMP3052.SEC Computer Security

## Session 15-3 Crypto III



# ACKNOWLEDGEMENTS

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towey...

# TOPICS COVERED

- What is encryption?
- Primitive types
- Historic ciphers
- The One Time Pad
- Stream ciphers

# WHAT DOES THIS MEAN?

Security Overview



This page is secure (valid HTTPS).

- Valid Certificate

The connection to this site is using a valid, trusted server certificate.  
[View certificate](#)
- Secure Connection

The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE\_RSA with P-256), and a strong cipher (AES\_128\_GCM).
- Secure Resources

All resources on this page are served securely.

# ENCRYPTION

- Encryption: We encode a message such that only **authorised users** may read it
- Cipher: takes a string of **plaintext**, and converts it into a string of **ciphertext**
- Encryption can provide:
  - Confidentiality
  - Integrity
  - Authenticity



# NOTATION

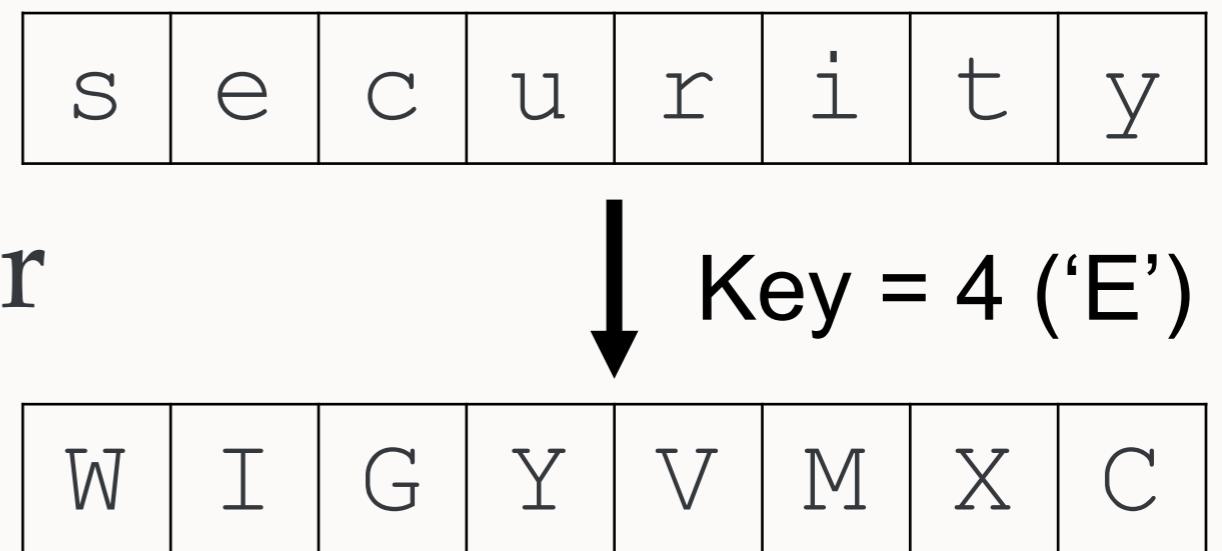
- A cipher converts a **plaintext** message  $M$  into a **ciphertext**  $C$  under the control of a **key**  $K$
- $C$  is not a secret, but without knowledge of the key, it should be impossible to reconstruct  $M$
- Comes in two forms:
  - Symmetric – same key for encryption / decryption
  - Asymmetric – separate keys

# PRIMITIVE TYPES

- Stream Ciphers
  - Operate on a stream of input data
- Block Ciphers
  - Operate on a fixed sized block
- Hash Functions
  - Take data of any size and output a block of fixed size

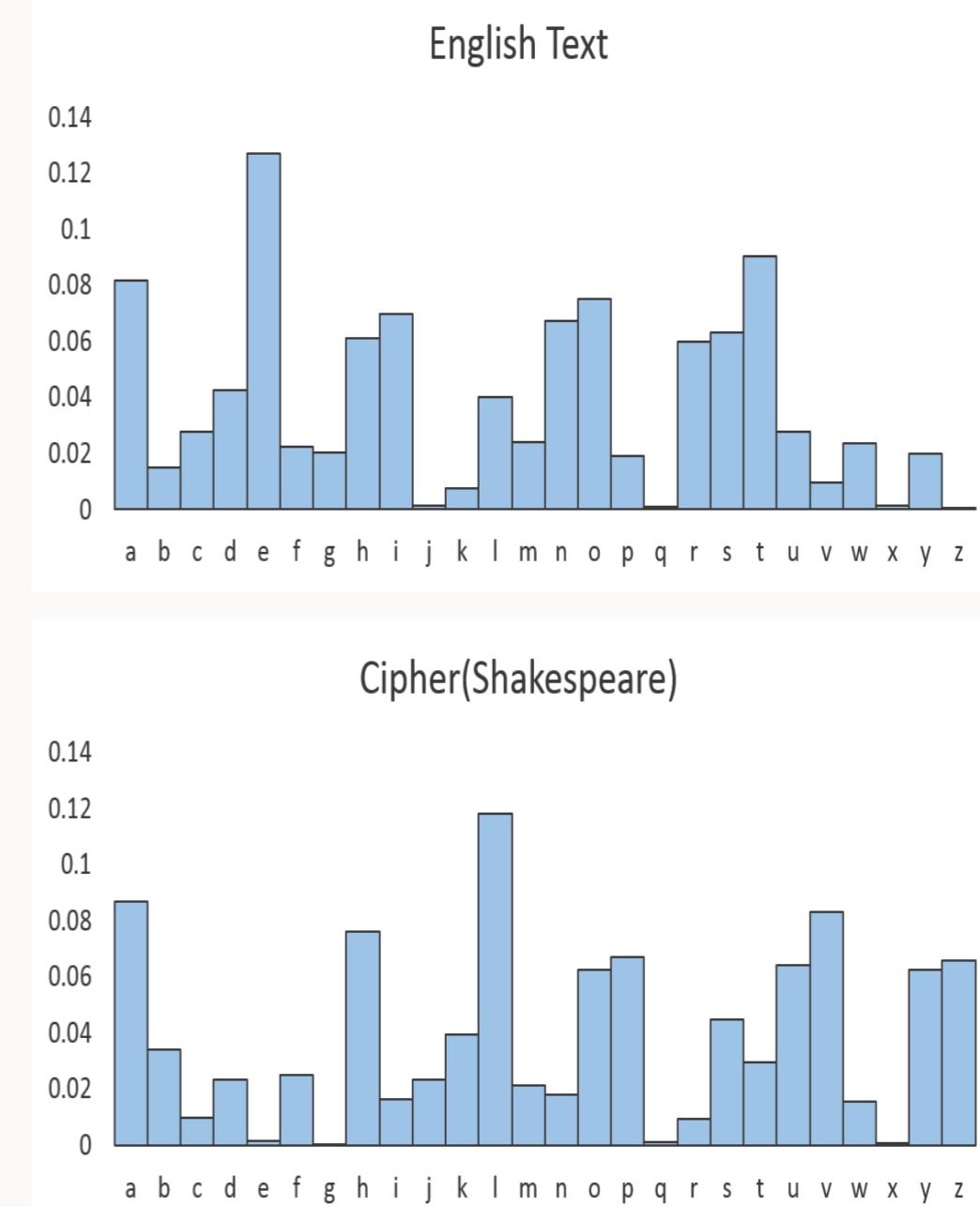
# THE CAESAR CIPHER

- An early **substitution** cipher, we replace each letter of plaintext with a shifted letter further down the alphabet
- Vulnerable to **frequency analysis**



# FREQUENCY ANALYSIS

- The frequency of occurrences of each character are very consistent across the same language
- The longer the ciphertext, the easier this becomes



# MONOALPHABETIC SUBSTITUTION

- A *slightly* better approach, a key is used to alter the cipher alphabet:

**Abcdefghijklmnopqrstuvwxyz**

**SECRTKYABDFGHIJLMNOPQUVWXZ**

- Still extremely vulnerable to frequency analysis

# THE VIGENÈRE CIPHER (POLYALPHABETIC SUBSTITUTION)

- An early stream cipher, the Vigenère cipher adds the plaintext to a key modulo 26:
- Unlike the Caesar cipher, the key is repeated for as long as is required

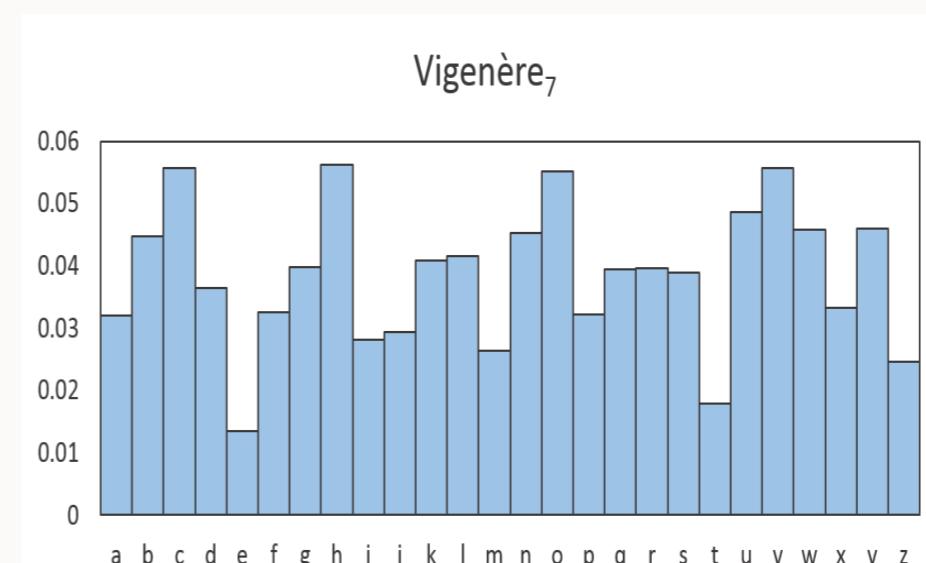
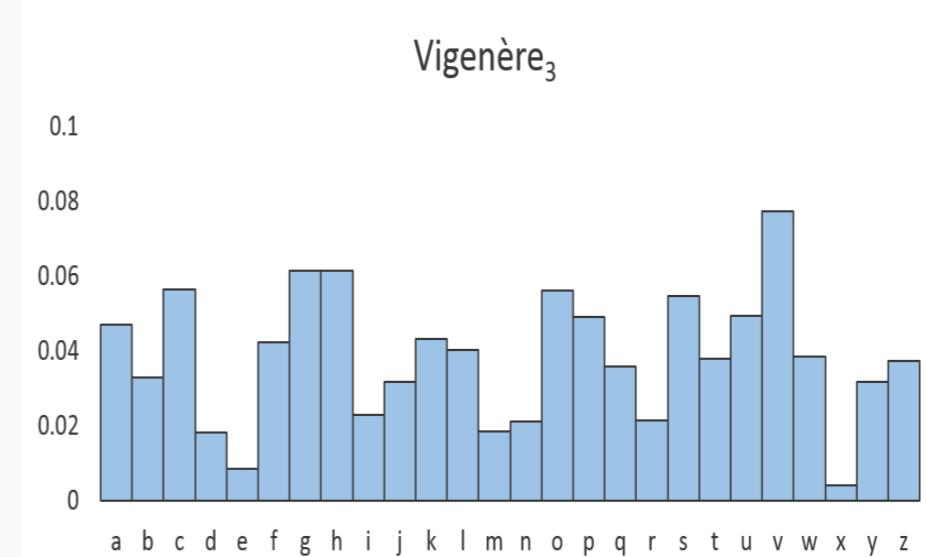
**tobeornottobethatisthequestion**

**keykeykeykeykeykeykeykeykeykeykeykeykey**

**DSZOSPXSRDSZOXFKXGCXFOUSOWRSSI**

# THE VIGENÈRE CIPHER

- Equivalent to multiple interleaved Caesar ciphers
- Spreads out the occurrences of characters making frequency analysis hard



# THE VIGENÈRE CIPHER

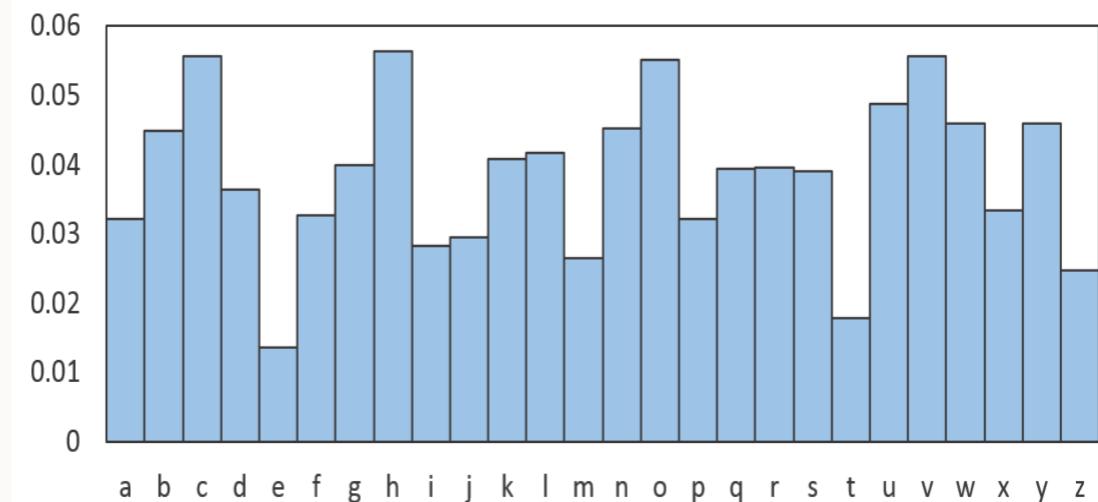
- This cipher is weak to Kasiski examination:
  - Repeated phrases in the ciphertext give away clues as to the length of the running key
  - Once the length of the key is known, simple frequency analysis can be performed

**tobeornottobethatisthequestion  
keykeykeykeykeykeykeykeykeykey  
DSZOSPXSRD**SZOX**EKXGC**XF**OUSOWRSSL**

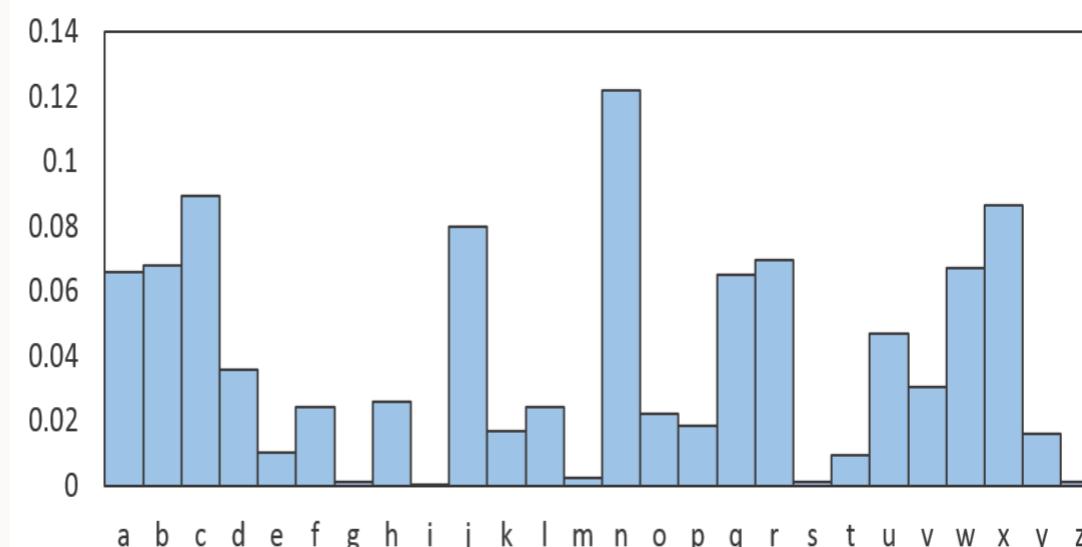
# THE VIGENÈRE CIPHER

- Frequency analysis of the Shakespeare text:
- After determining the key length, frequency analysis is straightforward

Key length 7, all ciphertext



Every 7<sup>th</sup> Character



# PLAYFAIR

- Playfair is an early **block cipher**
- By using blocks of two characters (digrams) we increase the number of mappings to 650 (=26x25)
- This makes frequency analysis much harder
- Nice example on Wikipedia:  
[https://en.wikipedia.org/wiki/Playfair\\_cipher](https://en.wikipedia.org/wiki/Playfair_cipher)

# PLAYFAIR

- Playfair is an early **block cipher**
- By using blocks of two characters (digrams), we increase the number of mappings to 650
- This makes frequency analysis much harder

S	E	C	U	R
I	T	Y	A	B
D	F	G	H	K
L	M	N	O	P
Q	V	W	X	Z

**attackatzerofourforty**

↓  
at ta ck at ze ro fo ur fo rt yz

BY YB RG BY VR UP HM RS HM EB BW

# PLAYFAIR

- Block size not sufficient – can piece together table
- • Reverse plaintext leads to reversed ciphertext
- • Probable plaintext can be guessed
- • Changes often propagate to only one character

S	E	C	U	R
I	T	Y	A	B
D	F	G	H	K
L	M	N	O	P
Q	V	W	X	Z

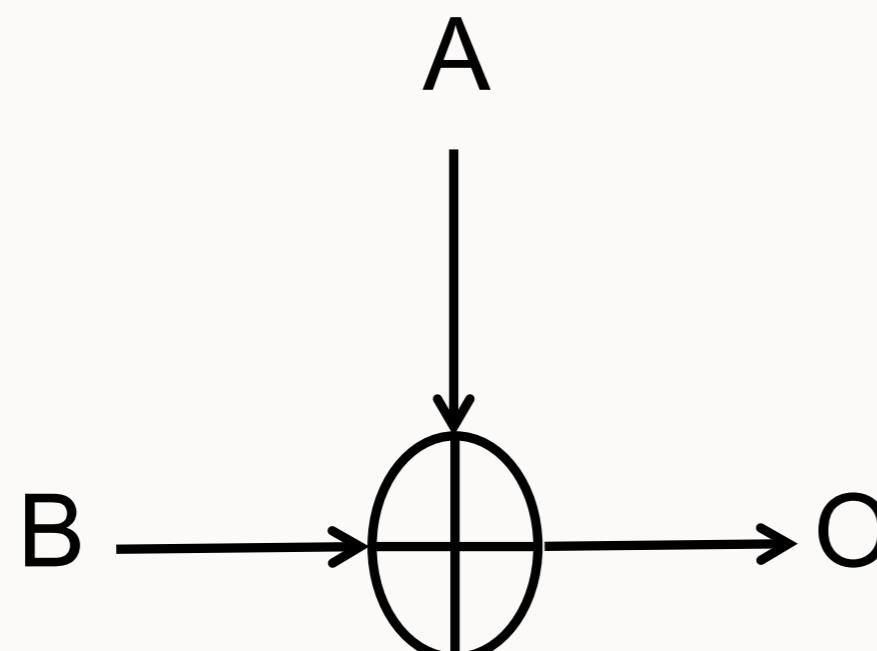
**attackatzerofourforty**

at ta ck at ze ro fo ur fo rt yz

BY YB RG BY VR UP HM RS HM EB BW

# XOR

“XOR is a binary operator between two values that returns true if either one input or the other is true, but not both”



A	B	O
0	0	0
0	1	1
1	0	1
1	1	0

Think of A determining if B flips when output

# WHY XOR IS COOL!

- Applying XOR twice, reverses its effect:

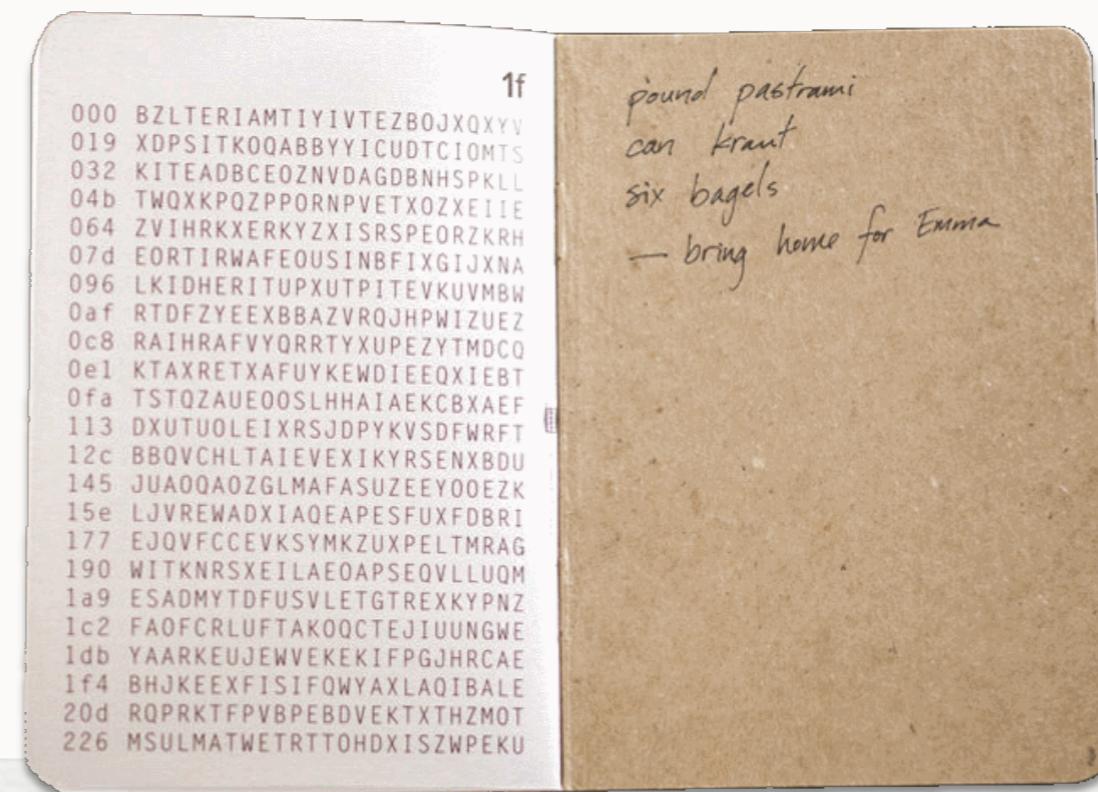
$$\begin{aligned} A \oplus B \oplus A &= ((A \oplus A) \oplus B) \\ &= 0 \oplus B \\ &= B \end{aligned}$$

- A “encrypts” B, and then “decrypts” it again

# THE ONE-TIME PAD

- Is there such a thing as a perfect cipher?
- Use a key that is the same length as the message
  - The one-time pad is the only example of **perfect secrecy**

M iloveyou  
K emrqytpn  
C MXFLCRDH



# THE ONE-TIME PAD

Can we design a cipher that uses XOR to encrypt and decrypt a message?

- Use a key that has the same length as the message
- XOR each message bit with each key bit

$$\begin{array}{rcl} \textcolor{red}{M} & 01011010 & 00110101 \\ & \oplus & \\ \textcolor{red}{K} & 01001011 & 10111001 \\ & = & \\ \textcolor{red}{C} & 00010001 & 10001100 \end{array}$$

Perfect Secrecy

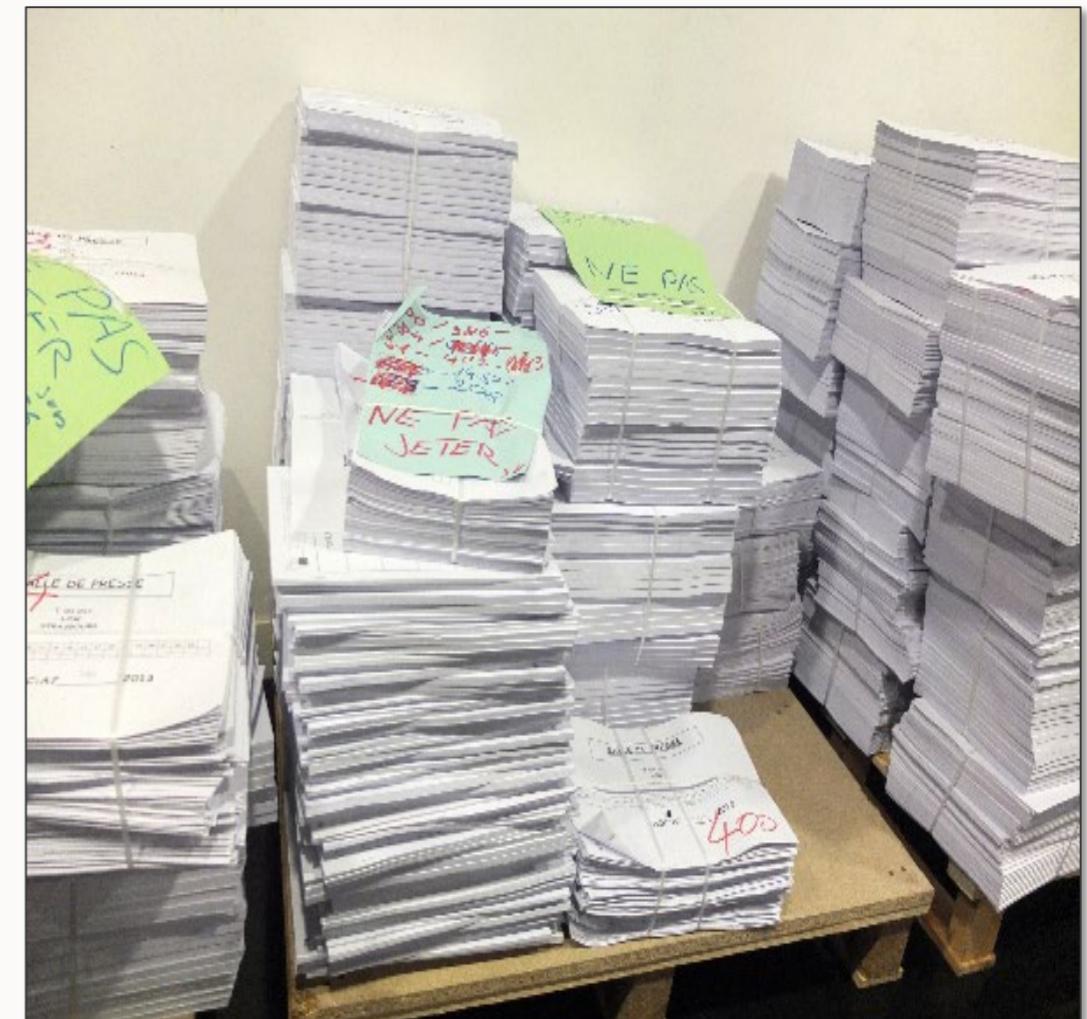
# THE ONE-TIME PAD

- Any possible plaintext can be recovered depending on the key
- Brute force is pointless
- Example with  $M = (C - K) \bmod 26$

C	<b>MXFLCRDH</b>	<b>MXFLCRDH</b>
K	<b>emrqytpn</b>	<b>eqfsytpn</b>
M	<b>iloveyou</b>	<b>ihatetyou</b>

# BUT...

- The one time pad is **not practical**:
  - A 1GB file would need a 1GB key!
  - How are we transporting these keys? Or storing them?
  - If you ever **reuse a key**, the entire cipher is **broken**

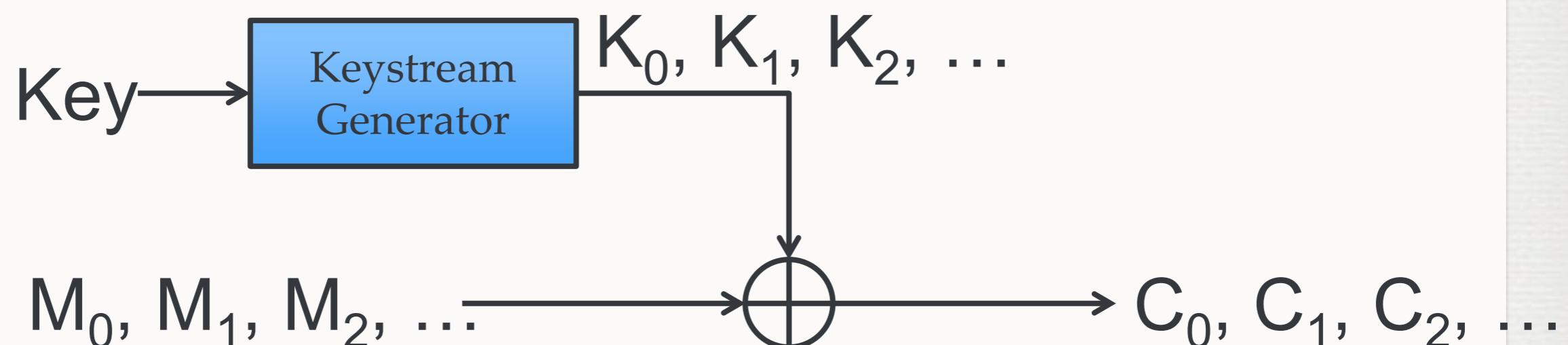


# THE ONE-TIME PAD

- What the one-time pad gives us in secrecy, it lacks in:
  - Portability – the key has the same size as the message
  - Convenience – you must *never* reuse a key
  - Modern stream ciphers attempt to approximate the one-time pad with a pseudorandom **key stream**

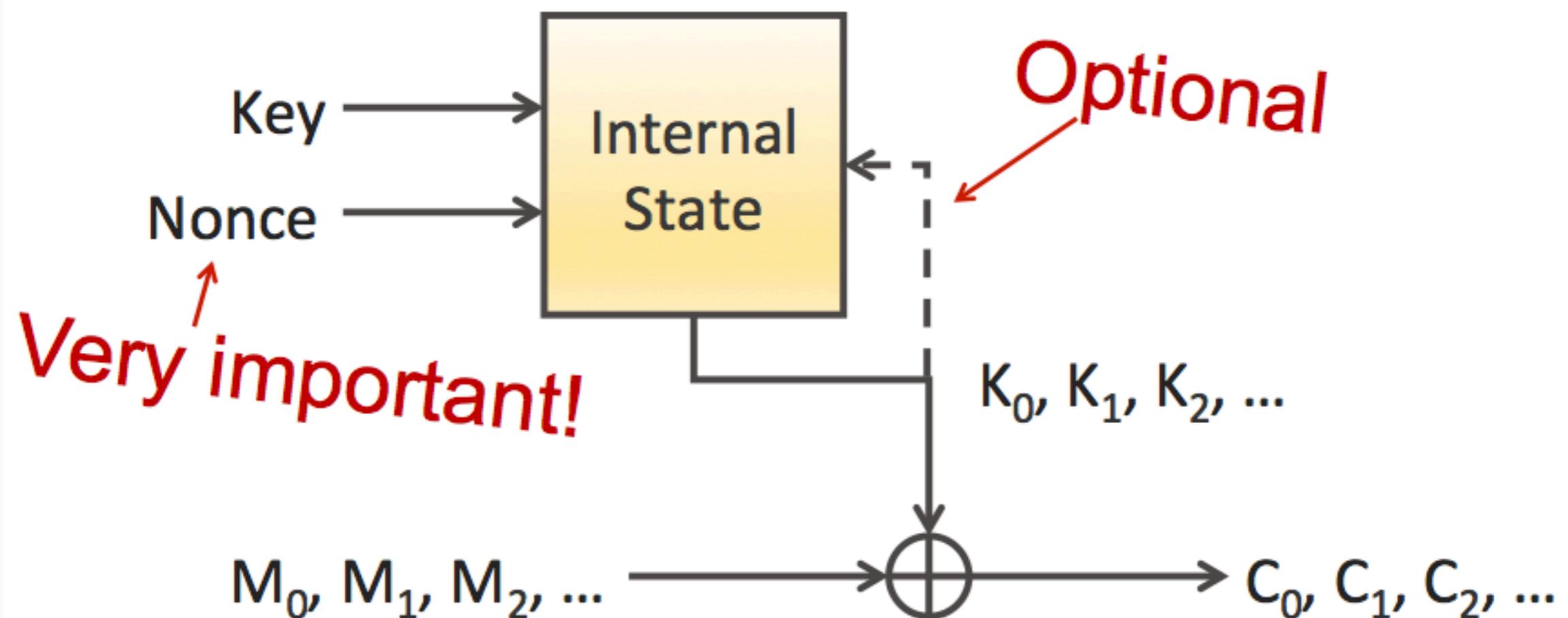
# MODERN STREAM CIPHERS

- Modern stream ciphers use an initial **seed** key to generate an infinite pseudorandom keystream
- The message and keystream are usually combined using **XOR** -  $\oplus$  - which is **reversible** if applied twice



# MODERN STREAM CIPHERS

- Common to seed an initial state using a key, then update this state for as long as needed:

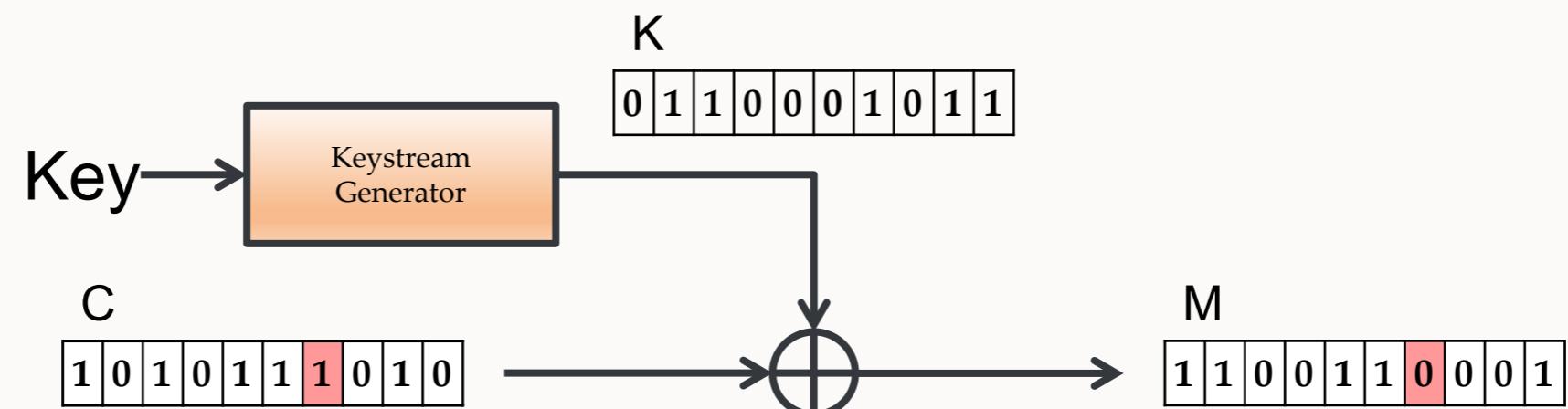


# MODERN STREAM CIPHERS

- Stream ciphers are usually reserved for situations where:
  - Hardware resources might be limited
  - The message stream is of unknown length, but long and continuous
    - For example, GSM mobile communications, Bluetooth, ...
  - Extremely fast with a low memory footprint, ideal for low-power devices
  - If designed well, can seek to any location in the stream
    - E.g. Streaming video with DRM

# MODERN STREAM CIPHERS

- Stream ciphers give us **confidentiality**, but not **integrity**
- We must include another mechanism



# KERCKHOFFS' PRINCIPLE

加密系统的安全性不应依赖于系统的保密性，而应仅依赖于密钥的保密性

“A cryptographic system must be secure even if everything is known about the system with the exception of the secret key”

- Algorithms can be published, but will work providing that the key remains secret - rather than security through obscurity

# CRYPTOGRAPHIC ATTACK MODELS

- If we know an algorithm inside-out,  
what can we do to attack it?
  - Brute-force
  - Ciphertext-only
  - Known-plaintext
  - Chosen-plaintext
  - Chosen-ciphertext
  - Related-key attack

Attack Strength



# BUT ...



# SUMMARY

- What is encryption?
- Primitive types
- Historic ciphers
- The One Time Pad
- Stream ciphers

Read:

- Gollman: Chapter 14
- Anderson: Chapter 5

# COMP3052.SEC Computer Security

## Session 15-4: Cryptology IV



# ACKNOWLEDGEMENTS

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towe...  
...

# TOPICS COVERED

- Block Ciphers
  - DES -> AES
  - Modes of Operation
- Public Key Crypto
- Maths ...but don't panic!

# WHAT DOES THIS MEAN?

Security Overview

��色锁图标  

---

This page is secure (valid HTTPS).

---

- Valid Certificate

The connection to this site is using a valid, trusted server certificate.  
[View certificate](#)
- Secure Connection

The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE\_RSA with P-256), and a strong cipher **(AES\_128\_GCM)**.
- Secure Resources

All resources on this page are served securely.

# BLOCK CIPHERS

- Block ciphers use a key to encrypt a fixed-size block of plaintext into a **fixed-size block** of ciphertext
- They are usually more computationally expensive than stream ciphers, but have numerous benefits
- If you're careful, you can convert between block and stream ciphers using modes of operation

# BLOCK CIPHERS

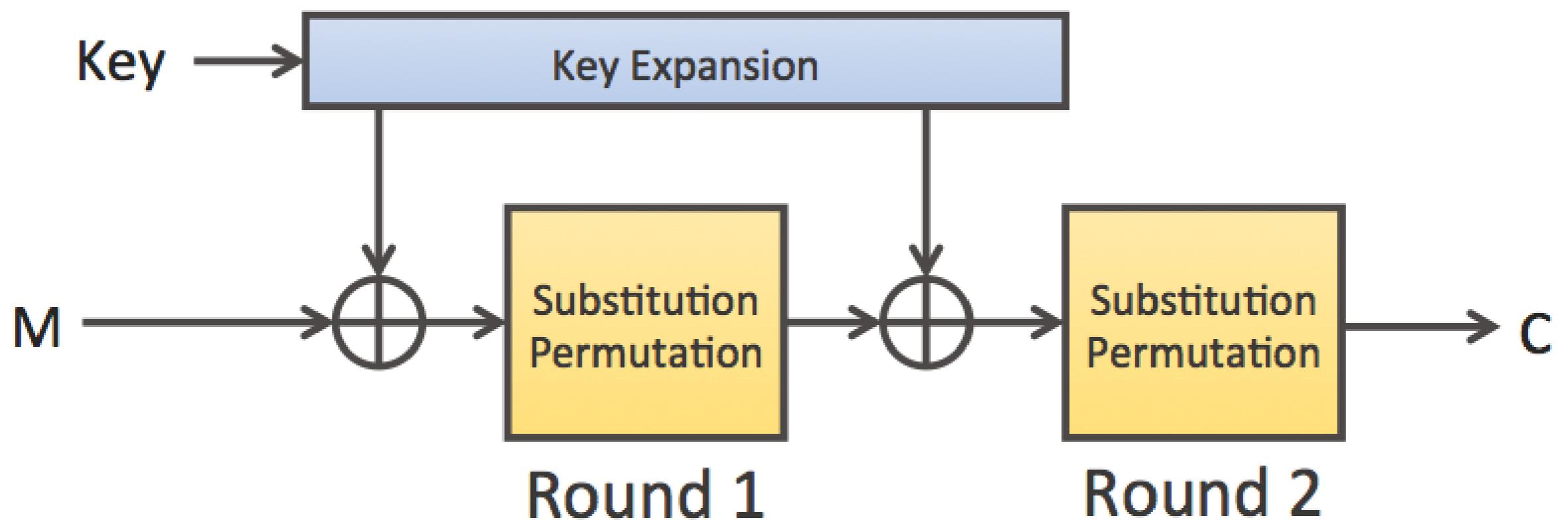
- If we change one bit of plaintext in a traditional cipher, one bit of output will change
  - Vulnerable to known and chosen plaintext attacks
- Modern block ciphers are designed to **diffuse** changes throughout each block

# SP-NETWORKS

- Claude Shannon suggested that all that was required for a strong cipher was repeated **substitution** and **permutation**
- SP-Networks combine a substitution process with a permutation into a single **round**
- Rounds are then **repeated** enough times to ensure the algorithm is secure

# SP-NETWORKS

- Can add a key using XOR:



# FEISTEL CIPHERS

- Developed by Horst Feistel in the 1970s at IBM
- Allows us to chain multiple rounds together, using **any round function**
- The basis of many block ciphers
  - Blowfish, DES, CAST-128, GOST 28147-89

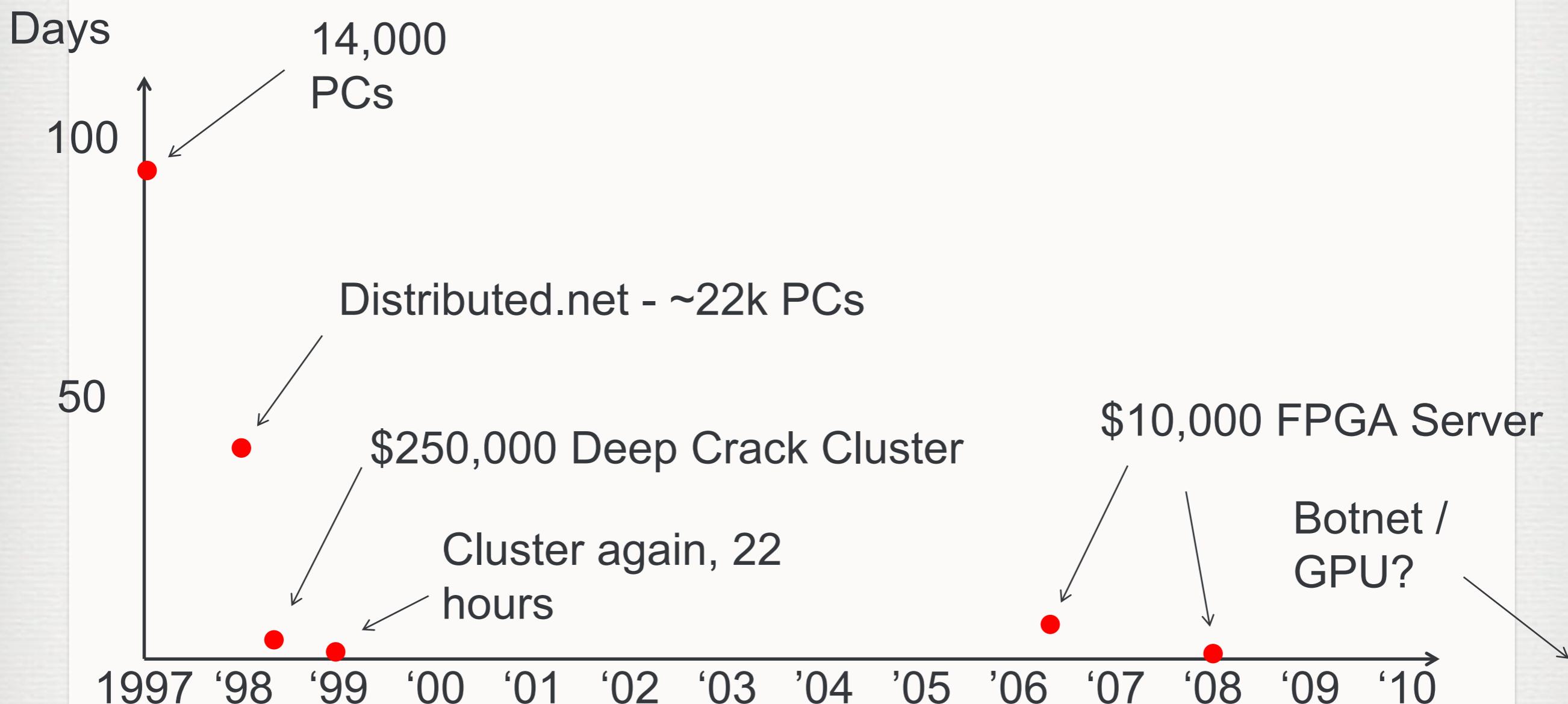
# LUBY-RACKOFF

- Mike Luby and Charlie Rackoff showed that if you have well-designed rounds (appear statistically random) in your Feistel cipher:
  - 3 rounds is sufficient to appear so random that a chosen plaintext attack won't work
  - 4 rounds is sufficient to beat a chosen plaintext *and* ciphertext attack

# DATA ENCRYPTION STANDARD

- DES was, until quite recently, one of the most used symmetric ciphers
- It is a 64-bit blocklength, 16 round Feistel cipher, with a 56-bit key
- Developed by IBM in 1970s, with a bit of interference from the NSA
- Once released, researchers were suspicious of some secrecy regarding the design, and the short key

# CRACKING DES



# ADVANCED ENCRYPTION STANDARD

- Superseded DES as a standard in 2002
- A standard built around the **Rijndael** algorithm
- Rijndael is an SP-Network with a 128-bit block size, and a key length of 128, 192 or 256-bits
- Round count depends on key length
  - 10, 12 or 14 cycles

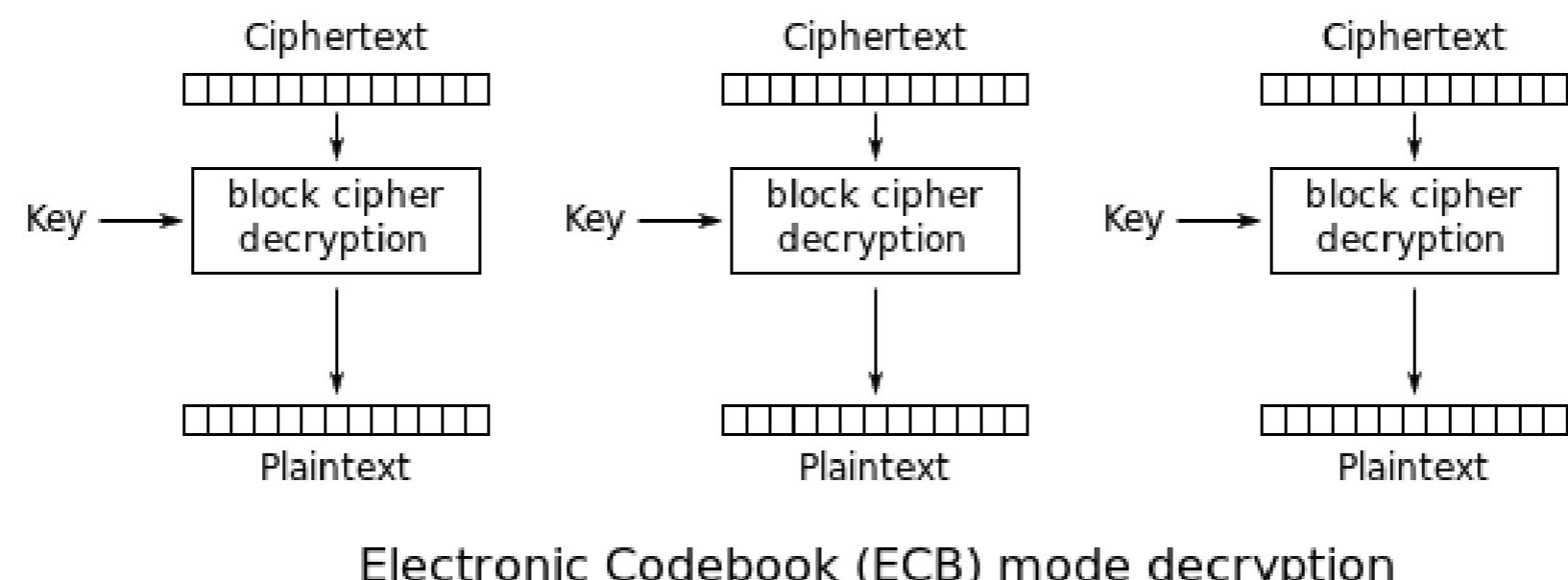
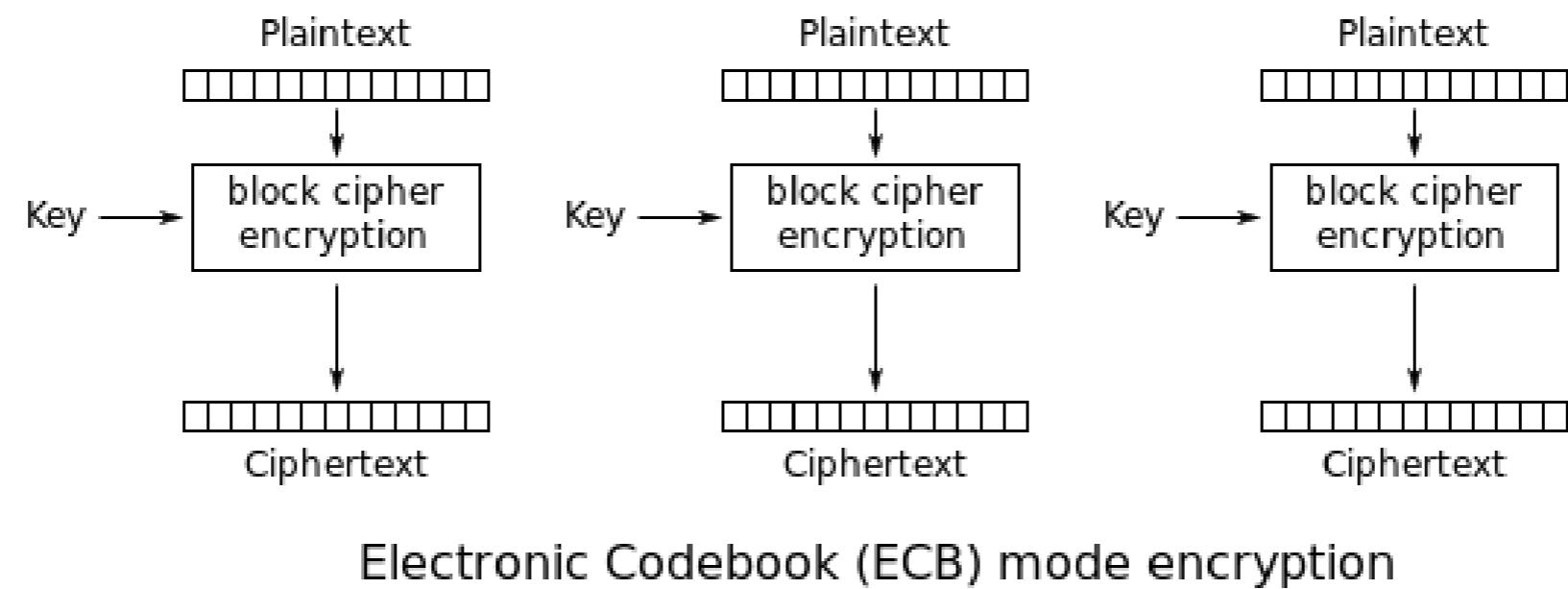
# AES SECURITY

- Is currently the standard algorithm for symmetric encryption, and is likely to remain that way
- $2^{128}$  keys, let's say we get lucky and crack it at  $2^{127}$
- Let's say 1000 flops per key test, we'd break an AES key in about 30 trillion years

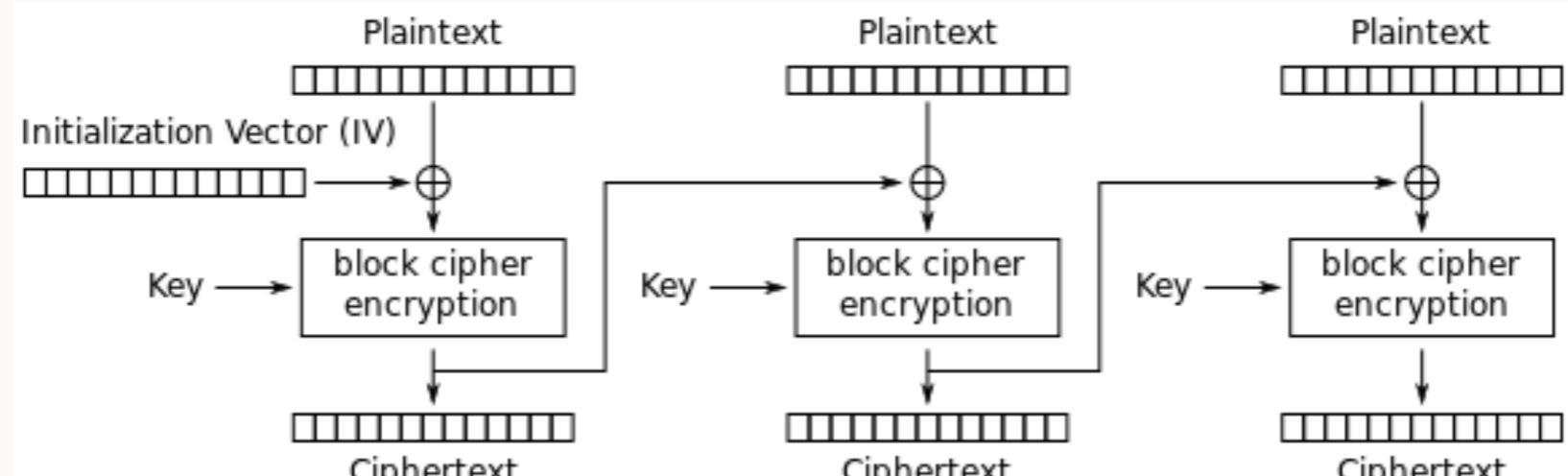
# BLOCK CIPHER MODES

- Most messages aren't in convenient 128-blocks
  - Run a block cipher repeatedly on consecutive blocks
    - Electronic Code Book (ECB)
      - encrypt each block one after another
    - Cipher Block Chaining (CBC)
      - XOR the output of each cipher block with the next input
    - Counter Mode (CTR)
      - Encrypting a counter to produce a stream cipher
    - Galois Counter Mode (GCM)
      - Extension of CTR

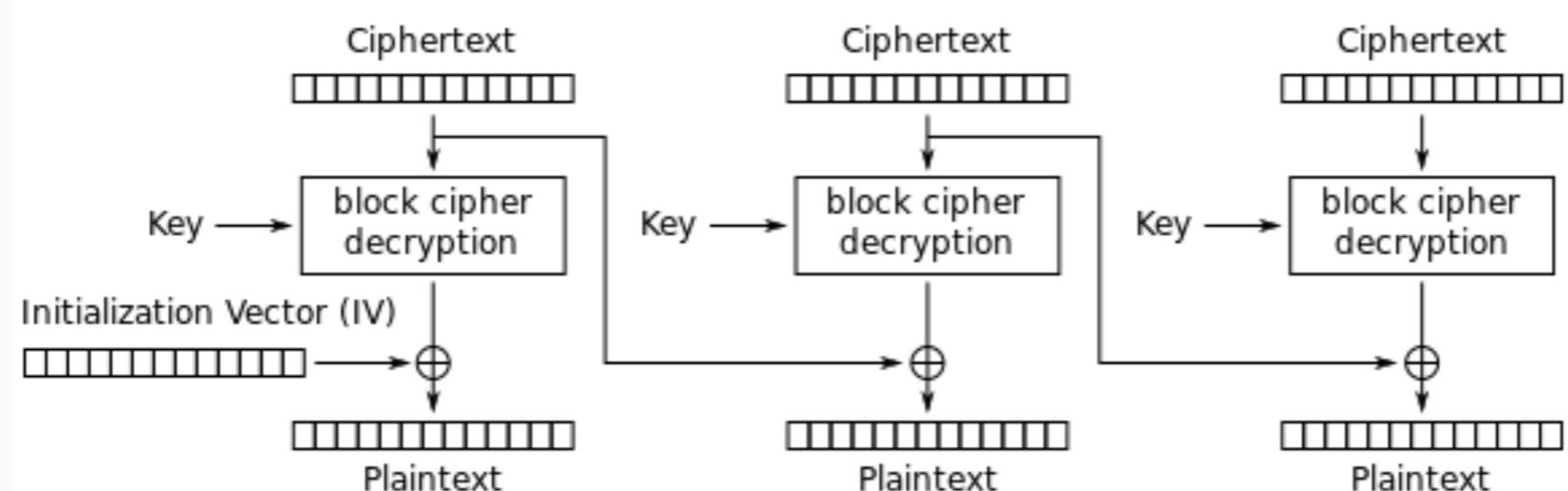
# ELECTRONIC CODE BOOK



# CIPHER BLOCK CHAINING

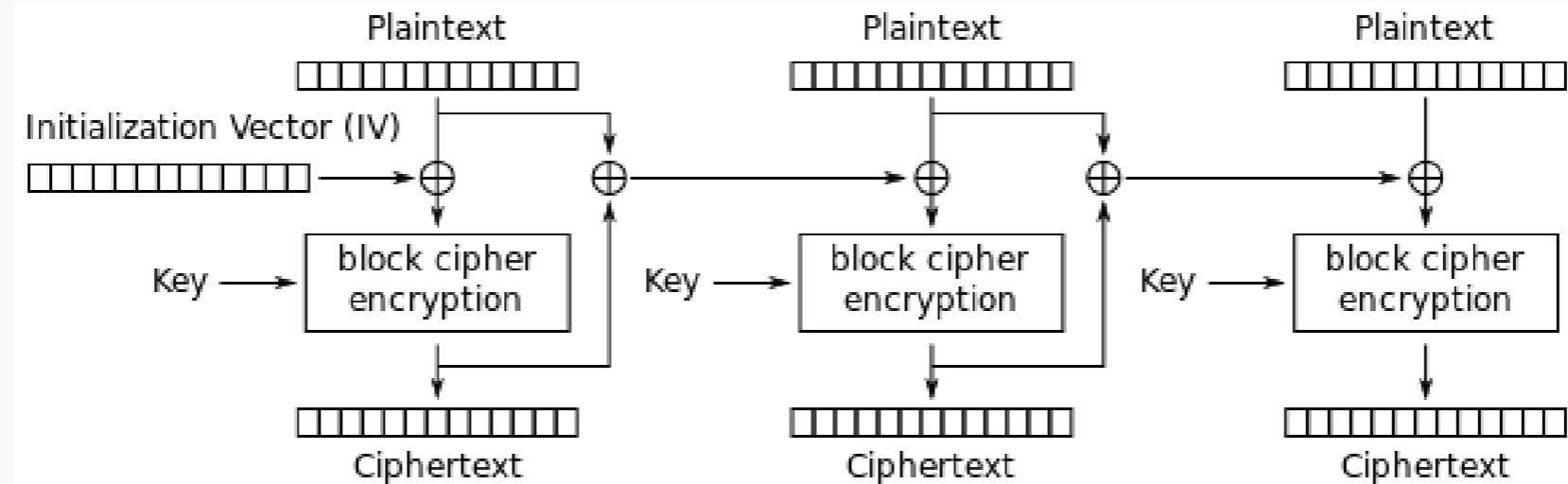


Cipher Block Chaining (CBC) mode encryption

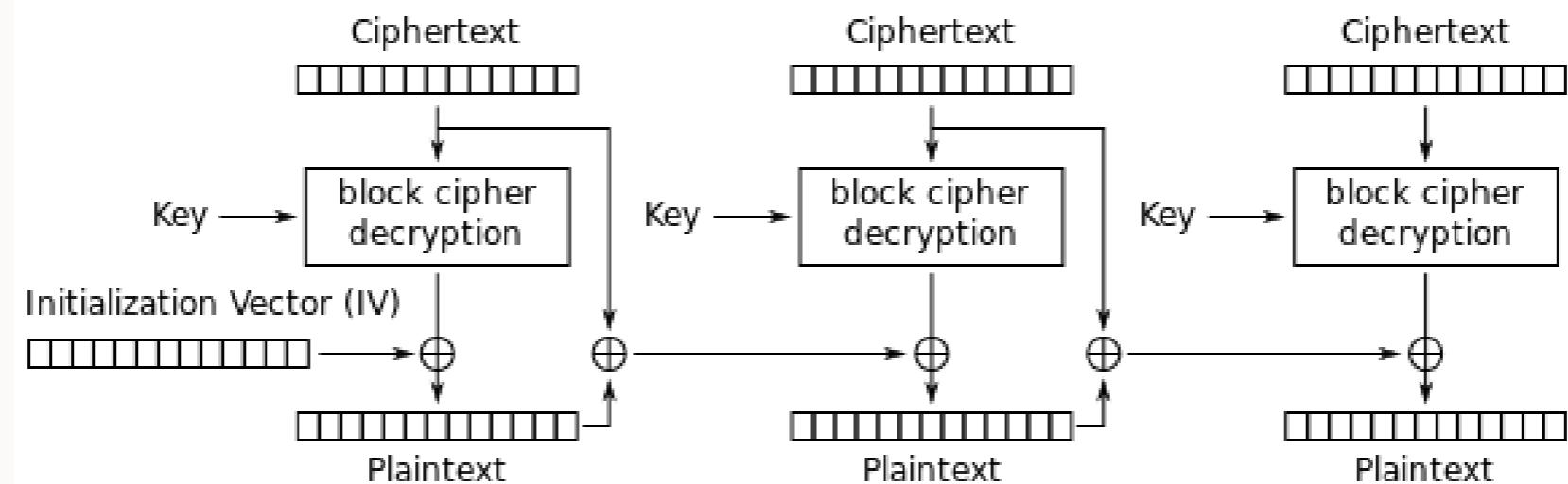


Cipher Block Chaining (CBC) mode decryption

# PROPAGATING CBC

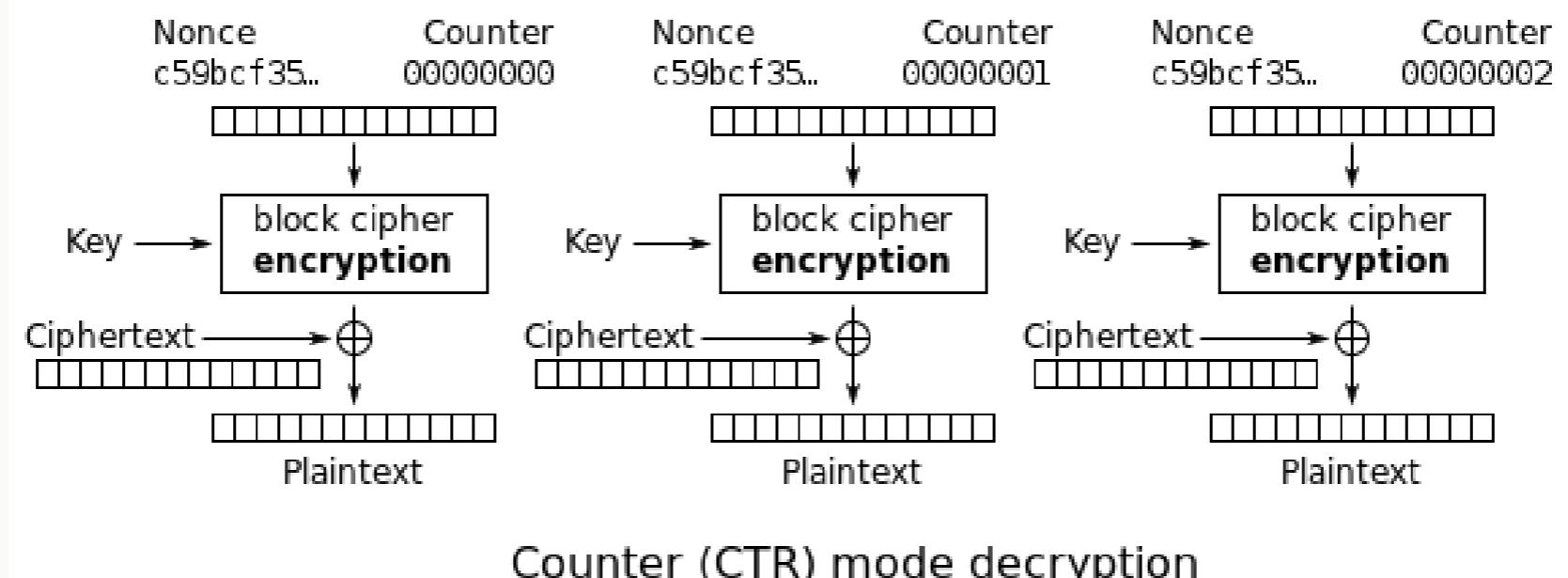
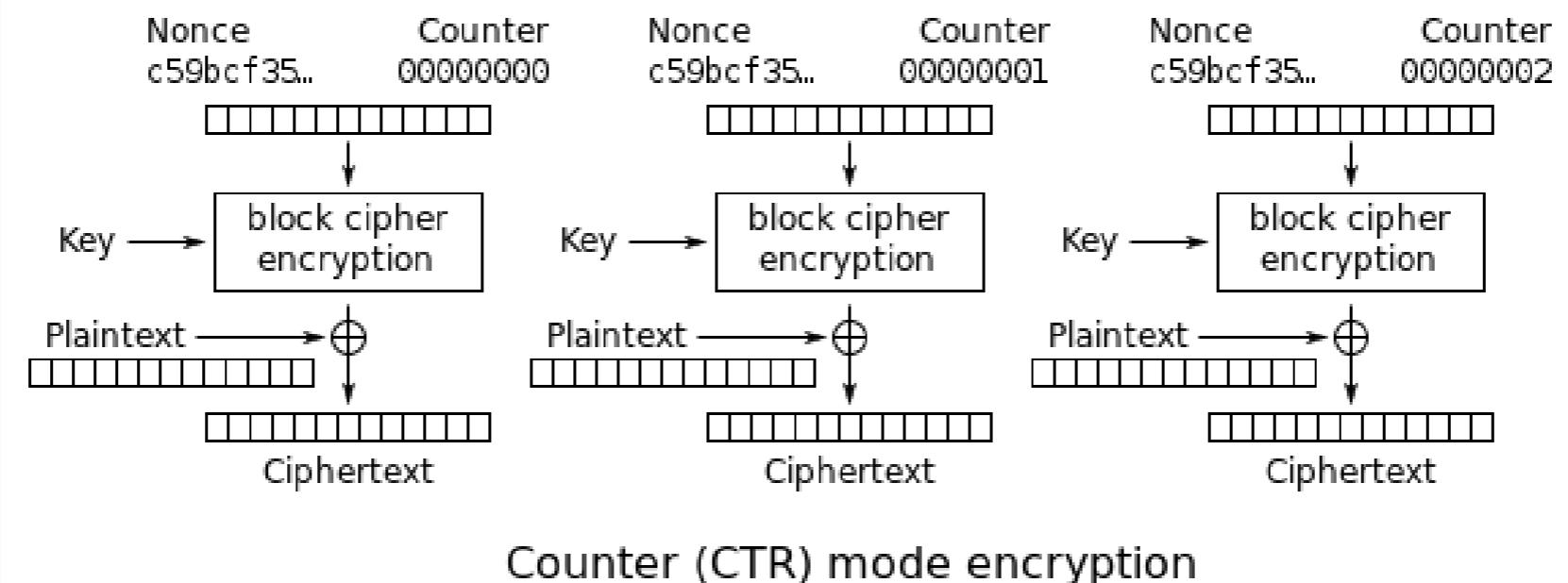


Propagating Cipher Block Chaining (PCBC) mode encryption



Propagating Cipher Block Chaining (PCBC) mode decryption

# COUNTER MODE



# WHAT DOES THIS MEAN?

Security Overview

---

This page is secure (valid HTTPS).

---

- Valid Certificate

The connection to this site is using a valid, trusted server certificate.  
[View certificate](#)
- Secure Connection

The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE\_RSA with P-256), and a strong cipher (AES\_128\_GCM).
- Secure Resources

All resources on this page are served securely.

Advanced Encryption Standard  
(Rjindael)  
128-bit Key Size  
Galois Counter Mode

# WHAT DOES THIS MEAN?

Security Overview

This page is secure (valid HTTPS).

- Valid Certificate  
The connection to this site is using a valid, trusted server certificate.  
[View certificate](#)
- Secure Connection  
The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE RSA with P-256), and a strong cipher (AES\_128\_GCM).
- Secure Resources  
All resources on this page are served securely.

**Advanced Encryption Standard  
(Rjindael)  
128-bit Key Size  
Galois Counter Mode**



# PUBLIC-KEY CRYPTOGRAPHY

- Two keys, a public key and a private key
- Public-key (asymmetric) cryptography hinges upon the premise that:

*It is computationally infeasible to calculate a private key from a public key*

- In practice, this is achieved through **intractable mathematical problems**

# WHY?

- Public-key cryptography gains us a few important abilities:
  - We can exchange a private symmetric key “in the open”
  - We can verify the sender of a message
  - Non-repudiation – you can’t deny you did something

# SYMMETRIC VS. ASYMMETRIC

<u>Symmetric</u>	<u>Asymmetric</u>
One Key	Two Keys
Keys are usually 128 or 256-bits	Keys are much longer, 2048 or 4096-bits
Usually extremely fast	Computationally slower
Longer term communication	Key exchange, verification and authentication only
Based on circuits of permutation and substitution	Based entirely on mathematical principles

# SOME MATHS ...

- Modular Arithmetic
- Exponentiation
- Logarithms
- Discrete Logarithms
- Primitive Roots
- Elliptic Curve Cryptography
- ...

# Primitive Roots

- The number that is raised to a certain power, is called the **generator**  $g$

$$\mathbf{g = 9}$$

$$9^1 = 2 \pmod{7}$$

$$9^2 = 4 \pmod{7}$$

$$9^3 = 1 \pmod{7}$$

$$9^4 = 2 \pmod{7}$$

$$9^5 = 4 \pmod{7}$$

$$9^6 = 1 \pmod{7}$$

$$\mathbf{g = 3}$$

$$3^1 = 3 \pmod{7}$$

$$3^2 = 2 \pmod{7}$$

$$3^3 = 6 \pmod{7}$$

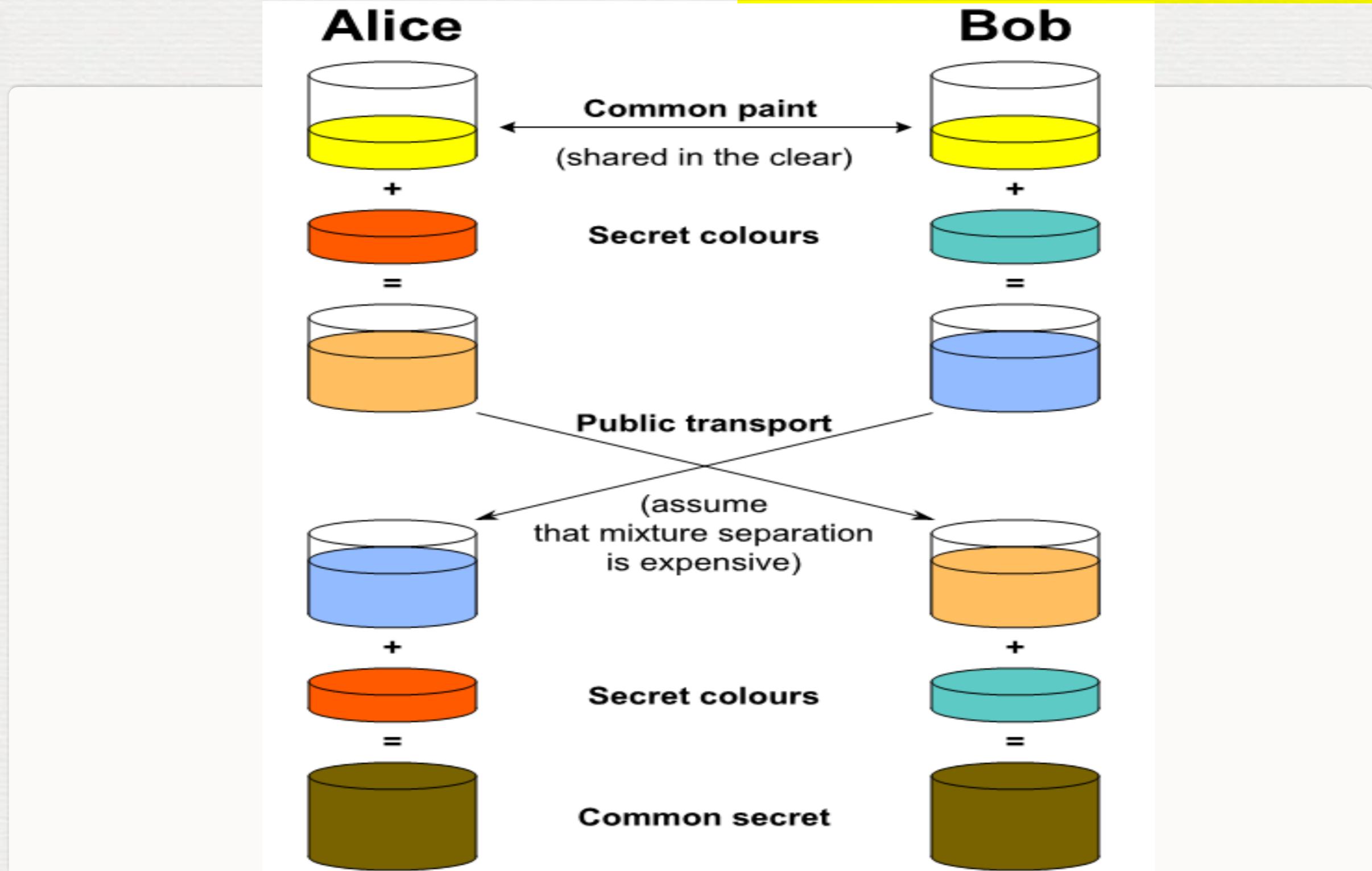
$$3^4 = 4 \pmod{7}$$

$$3^5 = 5 \pmod{7}$$

$$3^6 = 1 \pmod{7}$$

primitive root

# DIFFIE-HELLMAN KEY EXCHANGE



By Lorddota - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=62609302>

# DIFFIE-HELLMAN KEY EXCHANGE

- Diffie-Hellman uses a public-key protocol to **exchange a symmetric key in private**
- Relies on the difficulty of finding discrete logs

# DIFFIE-HELLMAN

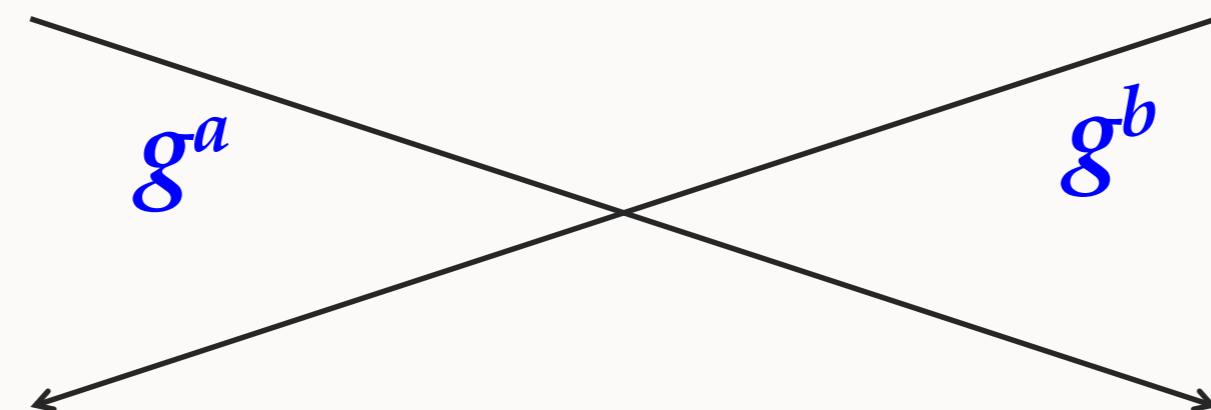
1. Alice and Bob agree on a large prime  $p$ , and a generator  $g$  that is a primitive root of  $p$
2. Alice chooses a private value  $a$  at random, then sends Bob a public  $g^a \text{ mod } p$
3. Bob chooses a private value  $b$  at random, then sends Alice a public  $g^b \text{ mod } p$
4. Alice computes  $(g^b)^a \text{ mod } p$ , which is actually  $g^{ab}$
5. Bob computes  $(g^a)^b \text{ mod } p$ , which is also  $g^{ab}$

# EXAMPLE

Alice and Bob agree on  $g = 3$  and  $p = 29$

Alice chooses  $a = 23$ ,  
then  $g^a = 3^{23} \bmod 29 = 8$

Bob chooses  $b = 12$ , then  
 $g^b = 3^{12} \bmod 29 = 16$



Alice calculates:  
 $(g^b)^a \bmod 29 =$   
 $16^{23} \bmod 29 = 24$

Bob calculates:  
 $(g^a)^b \bmod 29 =$   
 $8^{12} \bmod 29 = 24$

# WHY IS DH KEX SECURE?

- The secret shared key is  $g^{ab}$
- Yet, only  $g$ ,  $p$ ,  $g^a$  and  $g^b$  have been transmitted and are public
- The only way to calculate  $g^{ab}$  is either  $(g^a)^b$  or  $(g^b)^a$
- The only way to find  $a$  or  $b$  is solve:

$$a = \text{dlog}_{g,p}(g^a)$$

$$b = \text{dlog}_{g,p}(g^b)$$

# VULNERABILITIES

## ■ Man-in-the-middle

- A third party could intercept the initial communication from Alice, then create two separate key exchanges with both Alice and Bob
- Simply re-encrypting each message would allow them to sit in the middle of the conversation
- We can avoid this by combining DH with another cryptographic protocol

# PERFECT FORWARD SECRECY

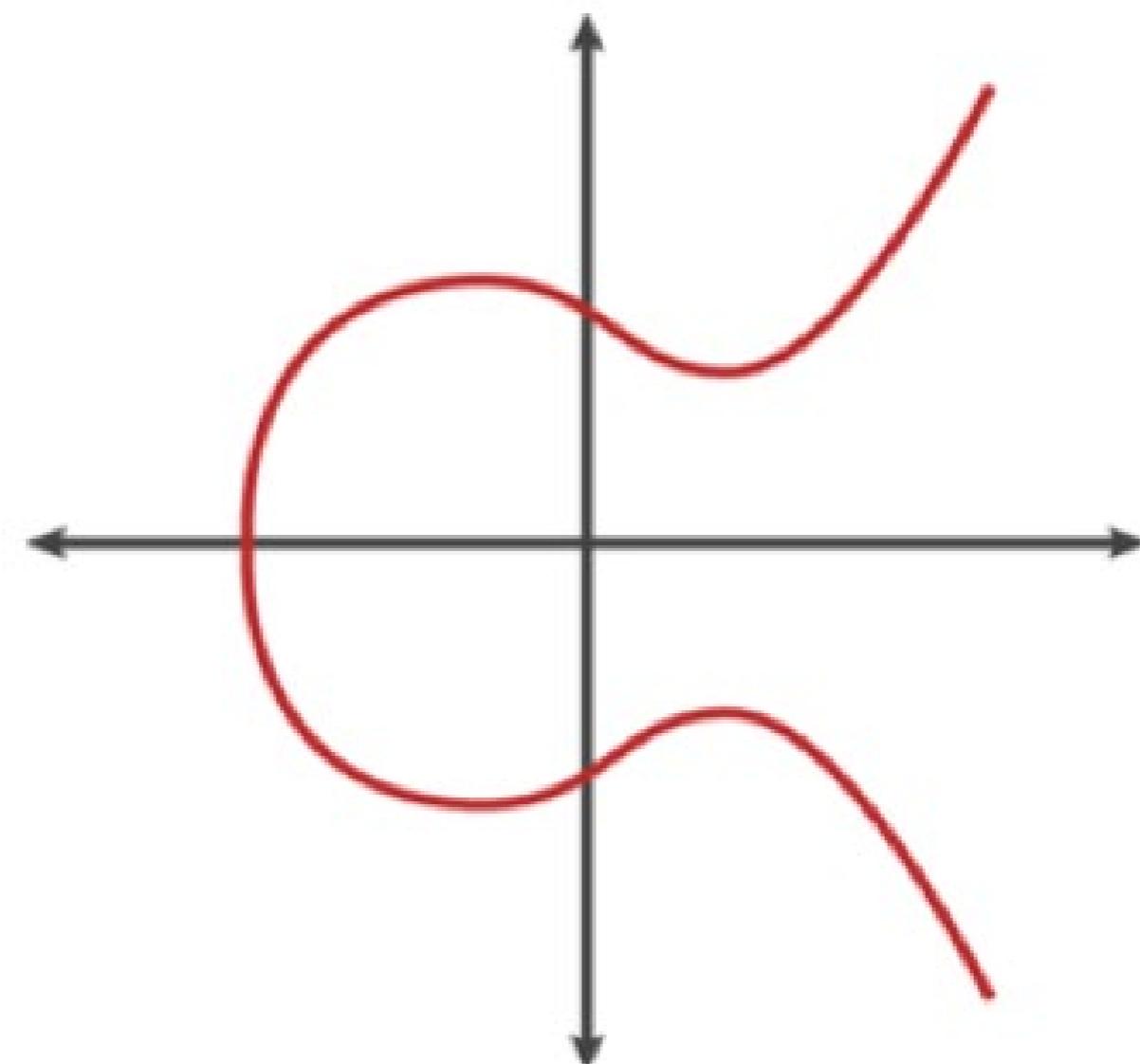
- There's always a chance a DHKEX key might be broken
- If we establish a symmetric key, how long should we use it for?
- Perfect forward secrecy means we generate **new keys for each session**, rather than persistent keys

# EPHEMERAL MODE

- In protocols like TLS, running Diffie-Hellman in ephemeral mode forces a new key exchange every time
- The recommended settings for TLS are now 2048-bit DH keys, in ephemeral mode

# ELLIPTIC CURVE CRYPTOGRAPHY

- Elliptic curves, of the form  $y^2 = x^3 + ax + b$  can be used in place of mod arithmetic in DHKEX



# ELLIPTIC CURVE CRYPTOGRAPHY

- Elliptic curve Diffie-Hellman is very similar to the regular version, but instead of  $g^a \bmod p$  we calculate  $aG \bmod p$ , so  $G + G + G \dots a$  times
- Given another point on the curve  $P$ , and the generator  $G$ , it is extremely hard to calculate  $a$  where

$$P = aG$$

- This is the **elliptic curve discrete logarithm problem**

# WHAT DOES THIS MEAN?

## Security Overview



This page is secure (valid HTTPS).

- Valid Certificate

The connection to this site is using a valid, trusted server certificate.

[View certificate](#)

Elliptic Curve  
Diffie-Hellman  
Ephemeral

- Secure Connection

The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE\_RSA with P-256), and a strong cipher (AES\_128\_GCM).

Advanced Encryption Standard  
(Rjindael)  
128-bit Key Size  
Galois Counter Mode

- Secure Resources

All resources on this page are served securely.

# SUMMARY

- Block Ciphers
  - DES -> AES
  - Modes of Operation
- Public Key Crypto
- (some) maths

Read:

- Gollman: Chapter 14
- Anderson: Chapter 5

# COMP3052.SEC Computer Security

## Session 15-5: Cryptology V



# ACKNOWLEDGEMENTS

- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towe...  
...

# TOPICS COVERED

- More maths ... (still no need to panic)
- RSA
- Message Authentication Codes
- Digital Certificates

# WHAT DOES THIS MEAN?

Security Overview

This page is secure (valid HTTPS).

- Valid Certificate  
The connection to this site is using a valid, trusted server certificate.  
[View certificate](#)
- Secure Connection  
The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE\_RSA with P-256), and a strong cipher (AES\_128\_GCM).
- Secure Resources  
All resources on this page are served securely.

Elliptic Curve Diffie-Hellman Ephemeral

Advanced Encryption Standard (Rjindael)  
128-bit Key Size  
Galois Counter Mode

# WHAT DOES THIS MEAN?

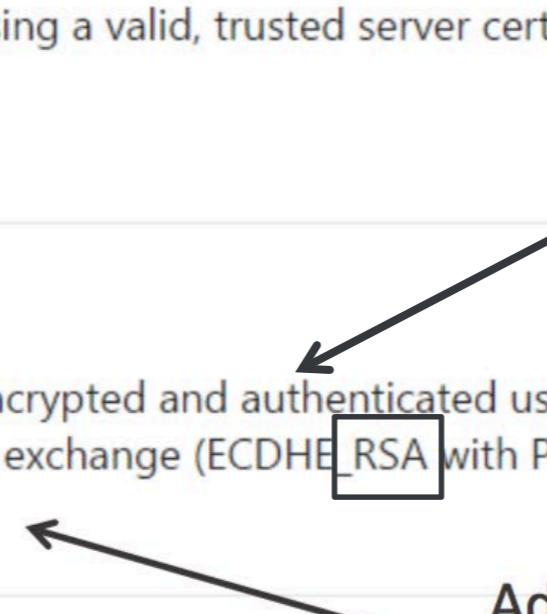
Security Overview

This page is secure (valid HTTPS).

- Valid Certificate  
The connection to this site is using a valid, trusted server certificate.  
[View certificate](#)
- Secure Connection  
The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE\_RSA with P-256), and a strong cipher (AES\_128\_GCM).
- Secure Resources  
All resources on this page are served securely.

Elliptic Curve Diffie-Hellman Ephemeral

Advanced Encryption Standard (Rjindael)  
128-bit Key Size  
Galois Counter Mode



# INTEGER FACTORISATION

- Any integer can be expressed as the multiplication of a list of prime numbers:

Example: 103284720

$$\begin{aligned} &= 2 \times 2 \times 2 \times 2 \times 3 \times 3 \times 5 \\ &\quad \times 7 \times 9 \times 9 \times 11 \times 23 \end{aligned}$$

- The longer the value, the harder (and slower) this gets

# INTEGER FACTORISATION

- Any integer can be expressed as the multiplication of a list of prime numbers:

Example: 103284720

$$\begin{aligned} &= 2 \times 2 \times 2 \times 2 \times \\ &3 \times 3 \times 3 \times 3 \times 3 \times 3 \times \\ &5 \times 7 \times 11 \times 23 \end{aligned}$$

- The longer the value, the harder (and slower) this gets

# INTEGER FACTORISATION

- Semi-primes are the **hardest** numbers to factor:
  - Product of two primes,

Example:  $n = 1522605027922533360535618378$   
 $1326374297180681149613806886$   
 $5790849458012296325895289765$   
 $4000350692006139$

What are  $p$  and  $q$ ?

# INTEGER FACTORISATION

- Semi-primes are the **hardest** numbers to factor:
  - Product of two primes,

Example:  $n = 1522605027922533360535618378$   
 $1326374297180681149613806886$   
 $5790849458012296325895289765$   
 $4000350692006139$

What are  $p$  and  $q$ ?

$p = 37975227936943673922808872755445627854565536638199$   
 $q = 40094690950920881030683735292761468389214899724061$

# EULER TOTIENT FUNCTION

- Integers  $a$  and  $b$  are **relatively prime** if they do not share a divisor (except 1)
- The **Euler totient**  $\Phi$  is the <sup>number of</sup> integers from 1 to  $n-1$  that are relatively prime with  $n$ 
  1. What is  $\Phi(9)$ ?
  2. What about  $\Phi(11)$ ?

# EULER TOTIENT FUNCTION

- The <sup>Euler</sup> totient value of a prime  $p$  is simply  $p-1$
- For two primes multiplied together it's  $(p-1)(q-1)$
- Euler's Theorem:  $a$  and  $n$  are relatively prime

$$a^{\Phi(n)} \equiv 1 \pmod{n}$$

$$a^{\Phi(n)+1} \equiv a \pmod{n}$$

# RSA

- Developed by Rivest, Shamir and Adleman
  - Based on integer factorisation
  - Can provide both **encryption** and **authentication**
  - The most common PK algorithm in the world

# RSA

1. Choose two large primes,  $p$  and  $q$ , then calculate  $n = pq$
2. Select a value  $e$  that is relatively prime with the totient of  $n$ .
  - (Remember, we know  $\Phi(n)$  as  $(p-1)(q-1)$ )

Example:  $p = 17, q = 11$

$$n = p \cdot q = 187$$

$$\Phi(n) = 160$$

$$e = \text{one of } 3, 6, 7, 11, \dots = 7$$

Public, Private

# RSA

3. Calculate a multiplicative inverse to  $e$ :  $d$ , where:

$$e \equiv d^{-1} \pmod{\Phi(n)}$$

Or:  $(e \cdot d) \bmod \Phi(n) = 1$

4. This is easily achieved if we know  $\Phi(n)$ , but not otherwise, we won't show the formula here

Example:  $e = 7, d = ?$

$$(7 \cdot ?) \bmod 160 = 1$$

Public, Private

# RSA

3. Calculate a multiplicative inverse to  $e$ :  $d$ , where:

$$e \equiv d^{-1} \pmod{\Phi(n)}$$

Or:  $(e \cdot d) \bmod \Phi(n) = 1$

4. This is easily achieved if we know  $\Phi(n)$ , but not otherwise, we won't show the formula here

Example:  $e = 7, d = 23$

$$(7 \cdot 23) \bmod 160 = 1$$

Public, Private

# ENCRYPTION WITH RSA

- Now we have a public key  $e, n$  and a private key  $d$
- Recall encryption is performed by:

$$M^e = C \pmod{n}$$

$$C^d = M \pmod{n}$$

Example:       $M = 74$

$$C = 74^7 \pmod{187} = 167$$
$$M' = 167^{23} \pmod{187} = ?$$

Public, Private

# ENCRYPTION WITH RSA

- Now we have a public key  $e, n$  and a private key  $d$
- Recall encryption is performed by:

$$M^e = C \pmod{n}$$

$$C^d = M \pmod{n}$$

Example:       $M = 74$   
 $C = 74^7 \pmod{187} = 167$   
 $M' = 167^{23} \pmod{187} = \mathbf{74}$

Public, Private

# WHY IS RSA SECURE

- We'd like the message  $M$  based on some ciphertext  $C$ , given the public key  $e$ :

$$C = ?^e \pmod{n}$$

Equivalent to:  $M = C^d \pmod{n}$

Remember:  $(e \cdot d) \bmod \Phi(n) = 1$

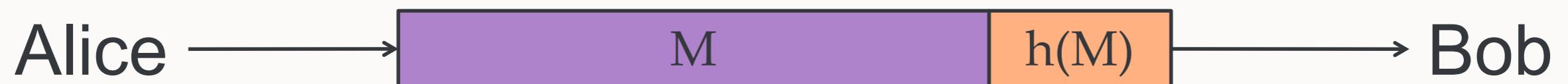
- Calculating  $d$  can only be achieved by knowing the totient  $\Phi$  of  $n$ . Finding this is extremely hard, for example we could factor  $n$  into  $p$  and  $q$

# USING RSA

- The keys ( $e, n$ ) and ( $d$ ) are reversible – either can be used for encryption, and the other used for decryption
- Everyone knows the public key, only the owner knows the private key
- This leads us to two very useful use cases for RSA:
  - Encryption only the owner can read
  - Signing that must have been performed by the owner

# MESSAGE AUTHENTICATION CODES

- Provide **integrity** and **authenticity**, not confidentiality
  - Protecting system files
  - Ensuring messages haven't been altered
- Calculate a hash of the message, then append this to the end of the message



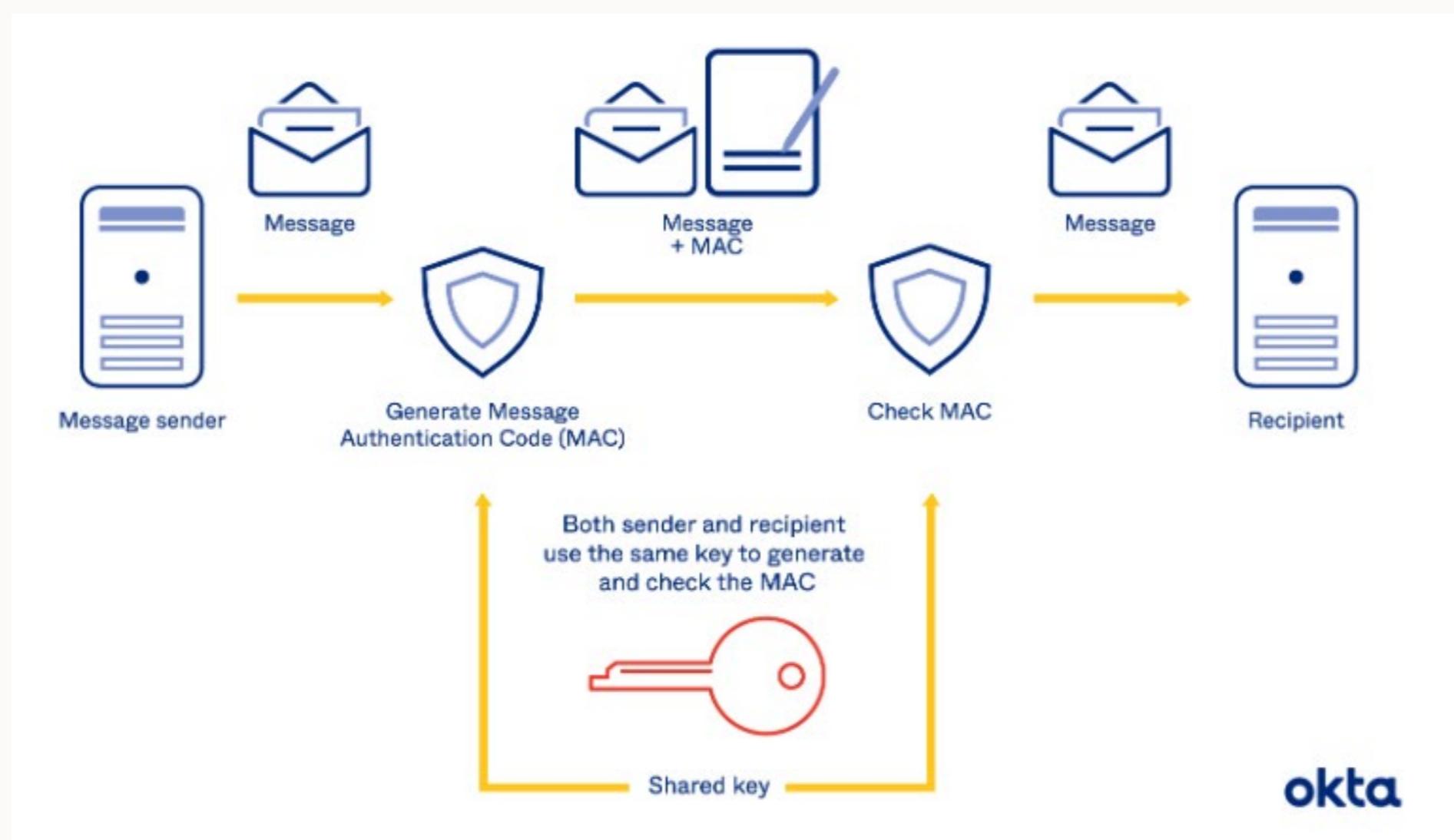
# KEYED HASHING

- Use a shared key to preserve message integrity
- Only those who know the key K can alter the message or hash



# HMAC (HASH-BASED MAC)

- <https://www.okta.com/identity-101/hmac/>



# HMAC ALGORITHMS

$$\text{HMAC}(K, m) = \text{H} \left( (K' \oplus opad) \parallel \text{H} \left( (K' \oplus ipad) \parallel m \right) \right)$$

$$K' = \begin{cases} \text{H}(K) & \text{if } K \text{ is larger than block size} \\ K & \text{otherwise} \end{cases}$$

where

$\text{H}$  is a cryptographic hash function.

$m$  is the message to be authenticated.

$K$  is the secret key.

$K'$  is a block-sized key derived from the secret key,  $K$ ; either by padding to the right with 0s up to the block size, or by hashing down to less than or equal to the block size first and then padding to the right with zeros.

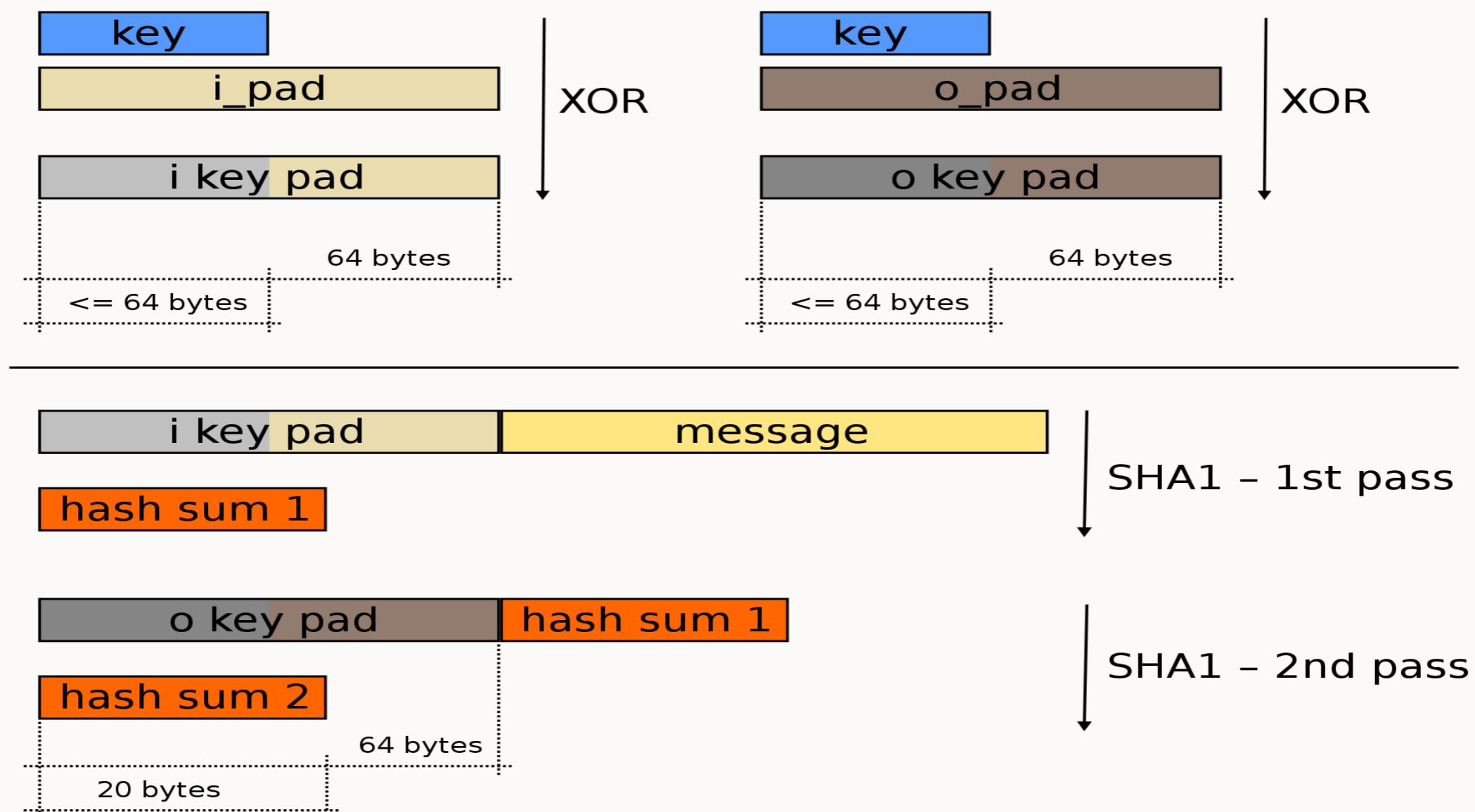
$\parallel$  denotes concatenation.

$\oplus$  denotes bitwise exclusive or (XOR).

$opad$  is the block-sized outer padding, consisting of repeated bytes valued 0x5c.

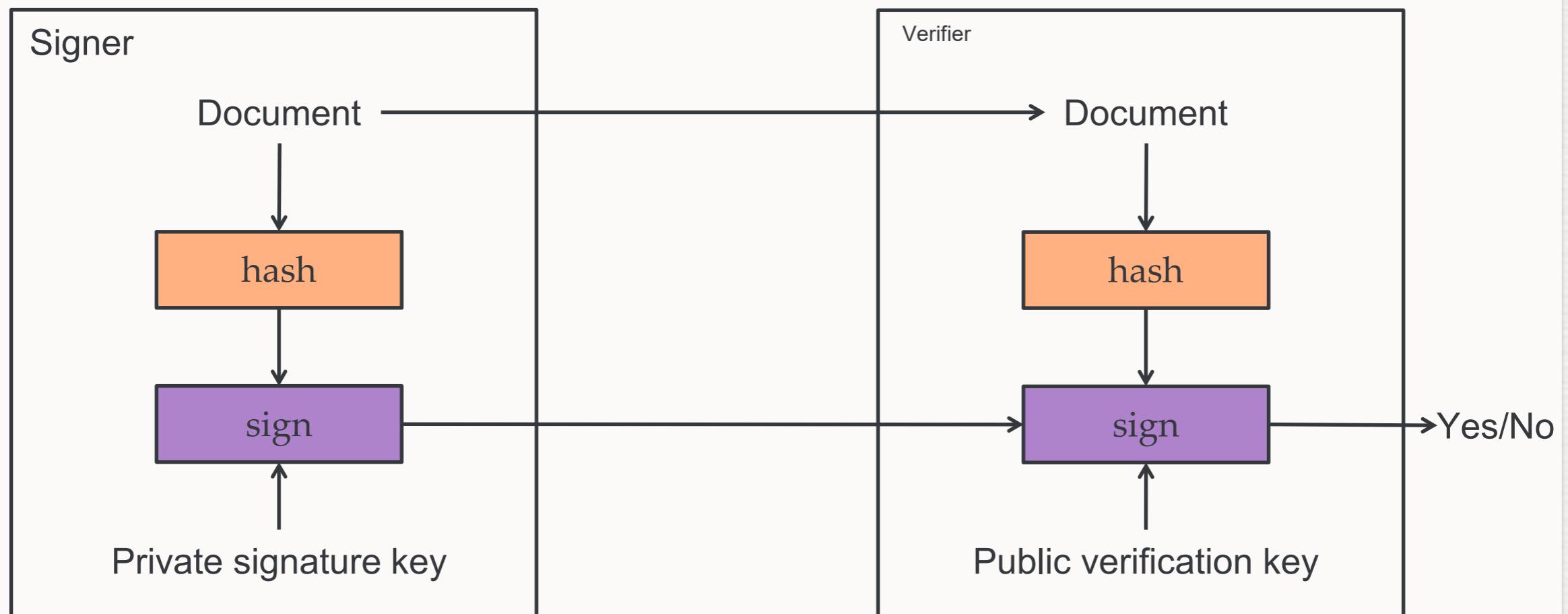
$ipad$  is the block-sized inner padding, consisting of repeated bytes valued 0x36.<sup>[3]</sup>

# HMAC-SHA1



# DIGITAL SIGNATURES

- Authentication codes provide integrity, but don't guarantee the sender
- Public-key encryption allows us to sign documents



# DIGITAL SIGNATURES

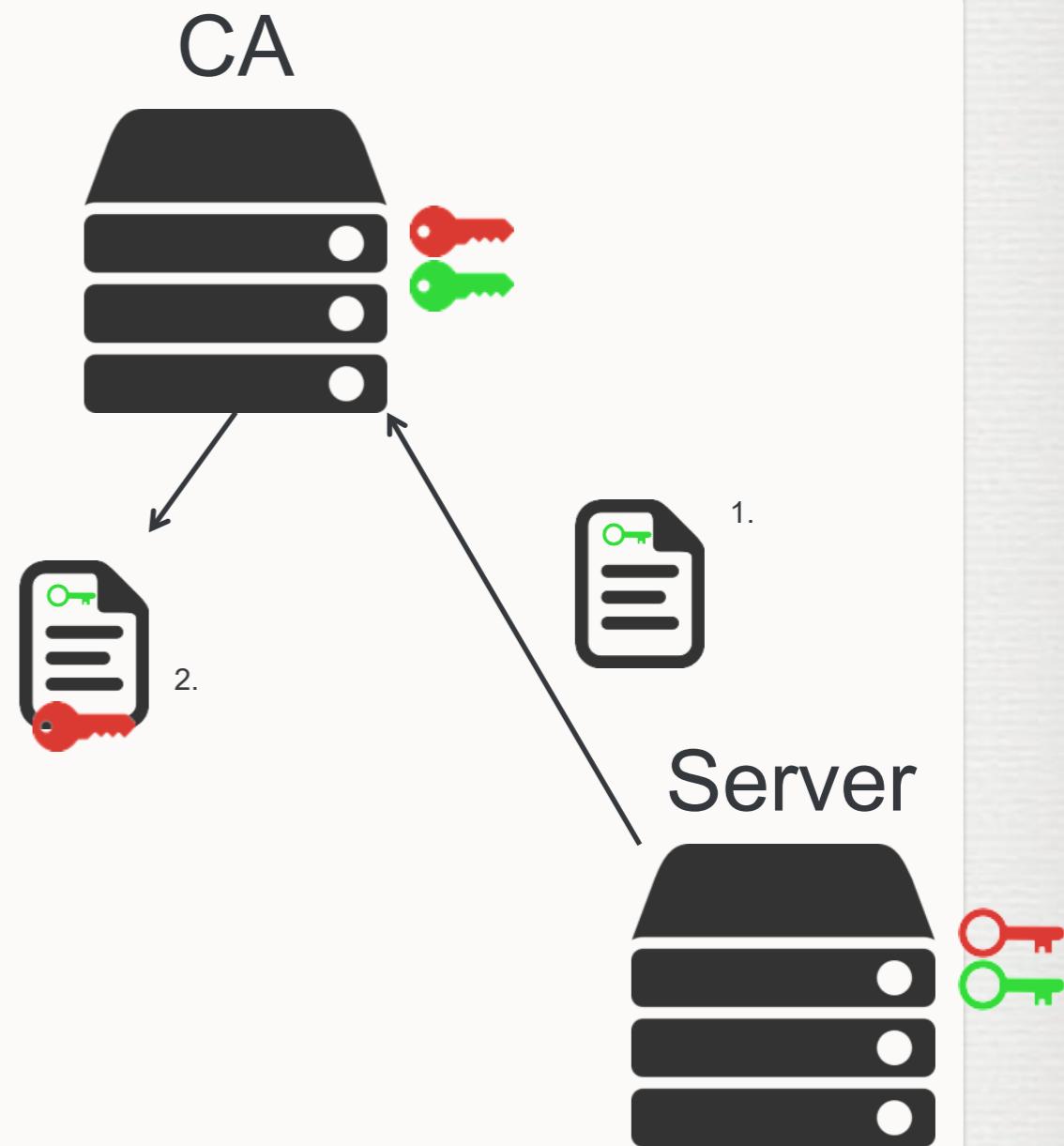
- Several digital signature algorithms in use, but majority of modern protocols use:
  - The Digital Signature Algorithm (DSA)
    - Based on large exponents in modulus arithmetic, much like Diffie-Hellman
  - **RSA** Signing
    - A variant of RSA, based on the problem of factoring large composite primes

# DIGITAL CERTIFICATES

- We can use a trusted third party (TTP) in order to verify the ownership of a public key
- Bob then knows he has Alice's genuine key, not an imposter's
- Can also be 'self signed'
- An important part of Transport Layer Security (TLS)

# USING DIGITAL CERTIFICATES

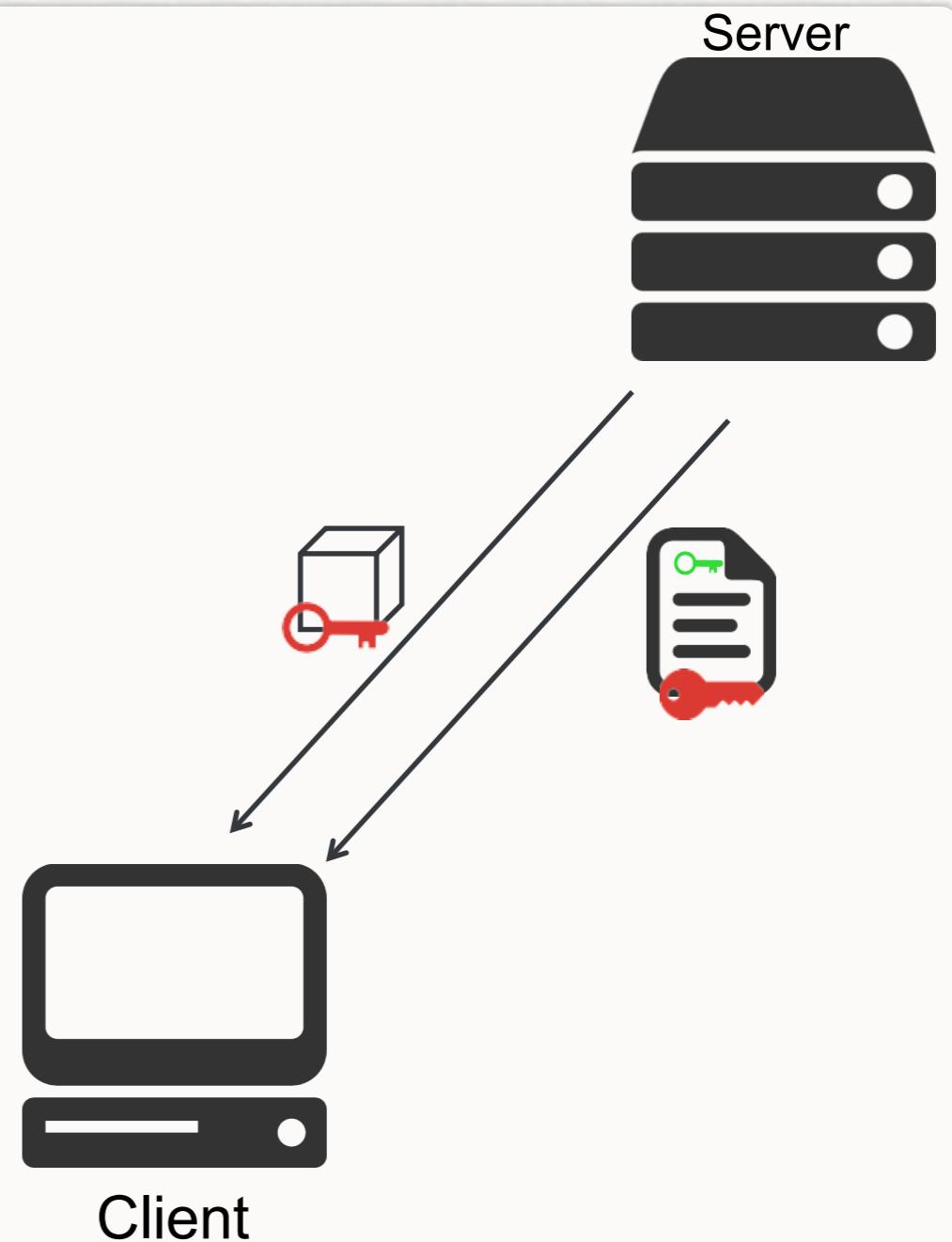
- Server produces a certificate containing its public key, which wants to be trusted
- The certificate goes to a certificate authority (CA), who, after doing ID checks, signs the certificate with CA's private key



Private, Public

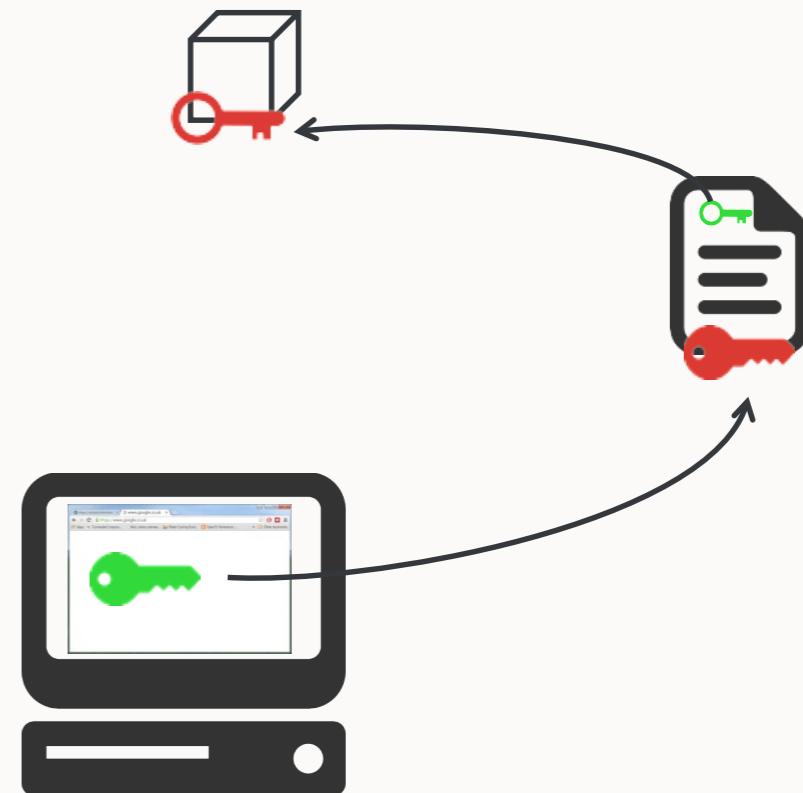
# USING DIGITAL CERTIFICATES

- Client wants to use the server, who sends client something (e.g., DHKEX parameters) signed with its private key
- Server sends client a certificate, which has been signed by a CA for verification



# USING DIGITAL CERTIFICATES

- Client has the CA public key stored in the browser. Client trusts the browser, so client trusts the CA. Client uses the CA public key to verify the certificate.
- When the certificate is verified, it is trusted, and the public key is used to verify the object



# WHAT DOES THIS MEAN?

Security Overview

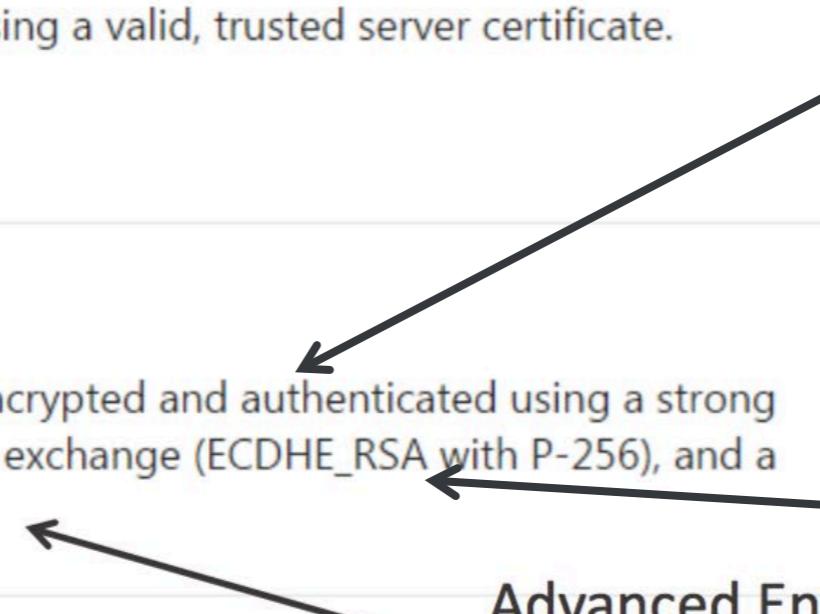
This page is secure (valid HTTPS).

- Valid Certificate  
The connection to this site is using a valid, trusted server certificate.  
[View certificate](#)
- Secure Connection  
The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE\_RSA with P-256), and a strong cipher (AES\_128\_GCM).
- Secure Resources  
All resources on this page are served securely.

Elliptic Curve Diffie-Hellman Ephemeral

RSA for message authentication

Advanced Encryption Standard (Rjindael)  
128-bit Key Size  
Galois Counter Mode



# SUMMARY

- More maths
- RSA
- Message Authentication Codes
- Digital Certificates

Read:

- Gollman: Chapter 14
- Anderson: Chapter 5

# CONCLUSIONS

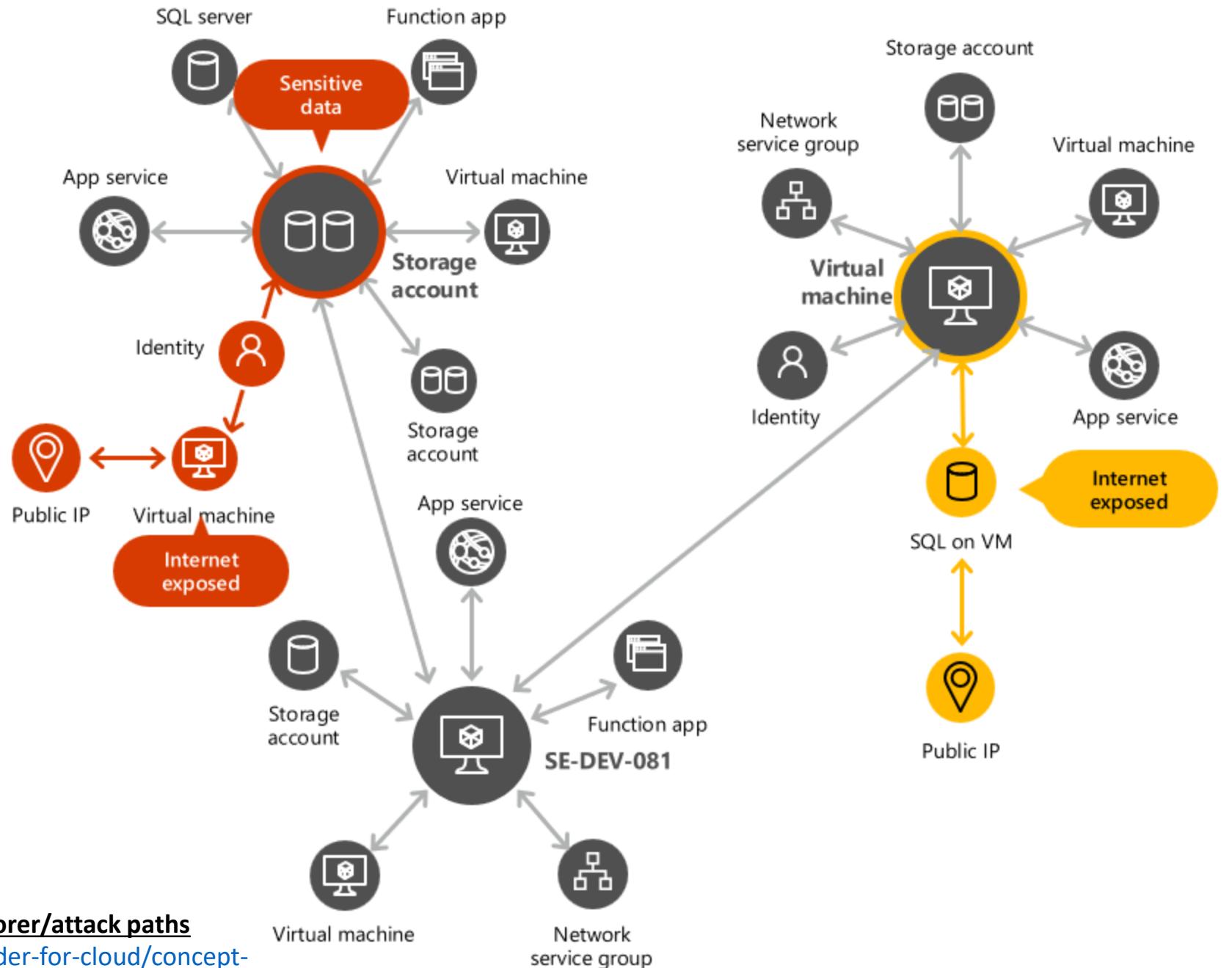
- You should now have a working knowledge of:
  - The basic principles and terminology of cryptology
  - The mathematical foundations that cryptology is based on
  - The key factors to consider when choosing a security solution for a network

Read:

- Gollman: Chapter 14
- Anderson: Chapter 5

# COMP3052 Computer Security

Session 01: Introduction to COMP3052.SEC



Reference: Investigating risk with security explorer/attack paths

<https://learn.microsoft.com/en-us/azure/defender-for-cloud/concept-attack-path>

## Assets:

The assets could include customer data (e.g., personal information, account details), financial transactions, reputation and trust of the company. Let's say the total value of the assets is **\$10 million.**

## Vulnerabilities:

Let's assume the following probabilities for the 3 identified vulnerabilities:

- (1) Unpatched software on the web server: Probability = 0.6
- (2) Weak password policy for employee accounts: Probability = 0.4
- (3) Lack of encryption for stored customer data: Probability = 0.3

The combine vulnerabilities =  $(0.6+0.4+0.3)/3 = 1.3/3 = \text{probability of } 0.433.$

## Threat:

Likelihood of a cyberattack by hackers targeting the system is moderate, with a **probability of 0.3.**

Now, let's calculate the risk:

**Risk = Assets \* Vulnerabilities \* Threat**

Substituting the values:

$$\text{Risk} = \$10,000,000 * 0.433 * 0.3$$

$$\text{Risk} = \$1,299,000$$

(A) Hacker

(B) Attacker

1. \_\_\_\_\_ refers to someone who proactively explores, identifies and alerts organizations to vulnerabilities that an attacker could use for malicious purposes. They seek to disclose in good faith by alerting organizations that may or may not have vulnerability disclosure policies.
  
2. \_\_\_\_\_ refers to someone who gains unauthorized access to someone else's network and computers for malicious purposes without permission or without warning the organization. This can be for monetary gain such as in ransomware attacks.

Reference: Hacker vs Attacker

<https://www.tripwire.com/state-of-security/hackers-vs-attackers-different-animals>

(A) Accidental Failures   (B) Operating Failures   (C) Intentional Failures

1. Security measures are primarily focused on protecting the computer system from \_\_\_\_\_, such as malicious attacks or unauthorized access.
  
2. Usability contributes to the prevention of \_\_\_\_\_ by making the computer system more user-friendly and intuitive, thus reducing the likelihood of errors made by users.
  
3. Reliability addresses \_\_\_\_\_ by ensuring the consistent and dependable performance of the computer system under normal operating conditions. This includes minimizing downtime, preventing crashes or system freezes, and maintaining data integrity.

# (A) Reliability   (B) Usability   (C) Security

1. \_\_\_\_\_ measures include redundant power supplies and backup systems that can prevent system downtime due to hardware failures, while error-checking and error-correction codes help detect and correct data corruption or transmission errors.
  
2. \_\_\_\_\_ measures include antivirus software to protect against malicious software that can compromise system integrity, while strong authentication mechanisms like biometrics or two-factor authentication prevent unauthorized users from gaining access to sensitive information.
  
3. \_\_\_\_\_ measures include error messages that provide clear explanations and suggested actions help users troubleshoot issues effectively, reducing the risk of data loss or system misconfiguration.

# COMP3052 Computer Security

Session 02: Motivating Example

Videoclip: Key Exchange

<https://www.youtube.com/watch?v=U62S8SchxX4>

Videoclip: How Encryption Keys Work - with Chris Bishop

<https://www.youtube.com/watch?v=EDTx3meleT0>

File Machine View Input Devices Help



sec@kali: ~/lab4

02:33 AM □ 🔔 | 🔒 G

sec@kali: ~/lab4

File Actions Edit View Help

```
drwxr-xr-x 2 sec sec 4096 Nov 17 2020 lab6  
drwxr-xr-x 2 sec sec 4096 Nov 17 2020 lab7
```

sec@kali:~\$

sec@kali:~\$ cd lab4

Change directory to lab4

sec@kali:~/lab4\$

sec@kali:~/lab4\$ ls -l

```
total 32  
-rwxr-xr-x 1 sec sec 371 Nov 17 2020 checkpasswd  
drwxr-xr-x 2 sec sec 4096 Nov 17 2020 dicts  
drwxr-xr-x 2 sec sec 4096 Nov 17 2020 hashes  
drwxr-xr-x 3 sec sec 4096 Nov 17 2020 pack  
-rwxr-xr-x 1 sec sec 118 Nov 17 2020 reset  
drwxr-xr-x 3 sec sec 4096 Nov 17 2020 rules  
-rwxr-xr-x 1 sec sec 431 Nov 17 2020 status  
-rwxr-xr-x 1 sec sec 108 Nov 17 2020 submitpasswd
```

sec@kali:~/lab4\$

sec@kali:~/lab4\$ echo -n "password" | openssl md5

Print "password" to standard output without newline character.

(stdin)= 6d1ae5cf1c748c9125a422b9cf53c9d0

Encrypt the string "password" with the hashing algorithm (md5).

Hash value of 32 characters long.

sec@kali:~/lab4\$

sec@kali:~/lab4\$ echo -n "password" | openssl sha512

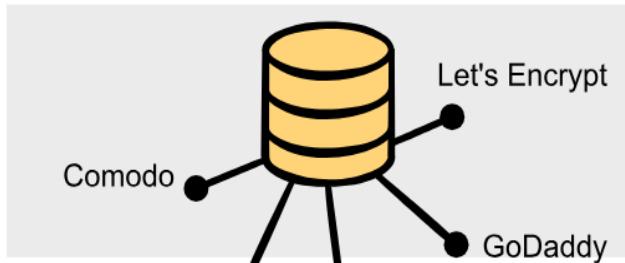
Encrypt the string "password" with the hashing algorithm (sha512).

(stdin)= 06195409b5f92bc9f64a2cfedec634cbc3f8e9c4929f402d2eee05723606119e967c4e28a3822bfe47f94c683f3d02afb7a794daf284b1b1866fa829f33741

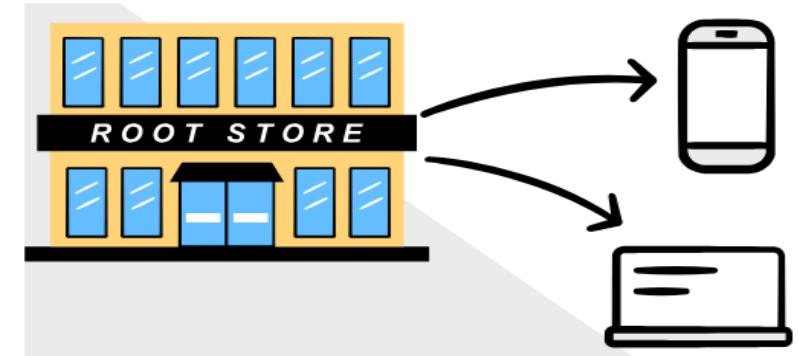
sec@kali:~/lab4\$

sec@kali:~/lab4\$

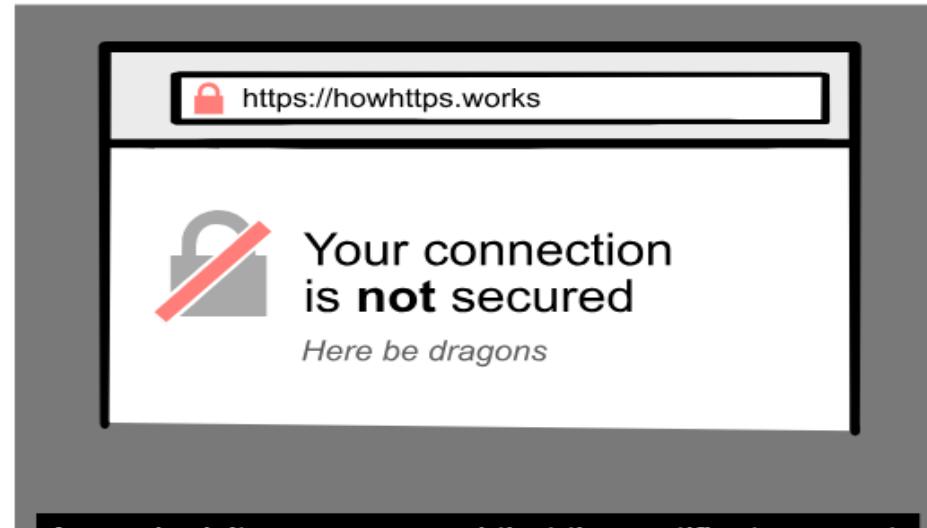
Hash value of 128 characters long.



A root store is basically a database of trusted CAs.



Apple, Windows, and Mozilla run their own root stores that they pre-install in your computer or device.



## Reference: Certificate Authorities

<https://howhttps.works/certificate-authorities/>

Kali-server [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

sec@kali: ~

sec@kali: ~

sec@kali: ~\$ Home

sec@kali: ~\$ sudo -i

[sudo] password for sec:

root@kali: ~#

root@kali: ~#

root@kali: ~# wget -O keyfile https://archive.kali.org/archive-key.asc

-- 2023-01-26 08:20:23-- https://archive.kali.org/archive-key.asc

Resolving archive.kali.org (archive.kali.org) ... 192.99.45.140

Connecting to archive.kali.org (archive.kali.org)|192.99.45.140|:443 ... connected.

HTTP request sent, awaiting response ... 200 OK

Length: 3155 (3.1K) [application/octet-stream]

Saving to: 'keyfile'

keyfile 100%[=====] 3.08K --KB/s in 0s

2023-01-26 08:20:24 (88.6 MB/s) - 'keyfile' saved [3155/3155]

root@kali: ~#

root@kali: ~#

root@kali: ~# cat keyfile

-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBEU1CgBEAChen9+cvBS8ioHoCU6wBbL9jaIk5P7ZkPpjDsovMvimgZaozS8  
fEAZM23gJlFratc+rRllV9hPZmGqhtT50RLDzC3yF0vFnJqAPvpVD02ipQCVnJDX  
0eWDhT62RDwk+FhjksEDwP7Yc4CgohdGDYQuizTBSLL5qen3rckCnHF20nSiKnYM  
8YCIKAYMt4VRArAvivjOMspN+1xy2S8GYXX2felsu3Ir1DXvUIE7b/9sdK6MzBcq  
joDH340qX6isqAW0+k93lmVN+U4yFmzfEB74UMQNWKg39mCB0K/VfQ89ih4zvF9a  
zENbFzfF000h09oHF4ZTaUfeI8JImp/x9FC+CveUyJot9t/xv0HVUd08Y4Pg048C  
iXOTqqqm/DPF0AbHJGpTuonOsKy3/dYhk7Fvsfn02DMds+RksukBEzypTIIIzMBF  
Uwiq/GaaNRWw6ln0yE4wMmpwRa11QVDDWkMpuOr1tPV7M+EMAhZY7cyHDmRTOFL3  
H0CxYnInis+k1NQikqtLxNrzWdxsXU25BMbEsAqq7aTs7wpOnUK+yY+qTG/V5nl  
J6II+/CtWJThIef22r8EYX1BQOXRggamy0nxViC3S6kjuU2Lb1qnDb/c7T9hb723  
4T9yrRHjbygTvQD8BJBAuDgRy7+XInCp01V4nAJZSu20qopEg082SdWK9FwARAQAB  
tCZLYWxpIExpbnV4IFJlcG9zaXRvcnkpgGRldmVsQghbGkub3JnPokCVQQTAAQoA  
PwIBAwYLCQgHAwIGFQgCCQoLBByCAwECHgECF4AWIQRExlE6jk+z0wh191jtRE/w  
fY0L9gUCYe/xtgUJG63pgAKCRDtRE/wfY0L9ofUD/9zichMeQ5+XxHnpHTSmUNR  
9eCcis6NGIbW+hTfsxSttfCQBrxKGFYXwVcq9GRIisDp9FcfcqH5UsgPOzgJzmlBLV  
iOHfMRDpcCWJzk8VfWde/5Hv0P0Xsx/+vR7PMJV0zWkf7o0kM4bITx3MSBYY08s  
leoZMNzo9SSR0GdzBxoIXkuBwMG+nw0qjfsyI8Qcsii52Uk8YhdCkgaAb8vYyMw0  
qKJ6SIIENT6ycyZw4ueFZXZY/RYPJFvWZEYB0cF0MbNwlrXFUZyr4fDdPruaPlZW  
zBIU8jwEE372frdRuXilJ20+Me9jhSxvnJegJsHDdgeU6IDEaiaaTzvufr3KmA  
YifZ1Q20xDzdN5RZ5+3vN9cihB3qizSUe0yLfppPeIPoaa+ENzT4WqhP6rfMMhF  
2KcwhNFP+faB+qs79YqIWnChrJNiE5HQDgP6DY1ETmjj6Bo+DC9RyasWT5psJYA  
8rOVJ+1bE4sxlqtvDvlAdR1LwIpHeskz4o/lAMPHSNTX2WBDJdYvhT28xTC3cYLA  
kcdpvBbjj7C4VqgMvsPLg13lfR/A28AD06AxY323+OpPv6crsDbY0cIBpwBgHb96  
AYyott1EfNnLBcysR/m34TdlYvFW0KDYsbMYLcJH/uZJK318TCdVBfkzDMMctSFV  
nFY/xHdfWeArXNhru2CJLkCDQRPVNQoARAAoMOct6yDngNUawaFLqFzzkQ2UDt1  
LywMM5qRusYmW7DbMqRgL816Ahw3qGXlpET2QDK/C7np8kiwx22CwK2W7e877  
bKGX1jH0k8jIZWxE15pBCBkTk+zb6elc263qiw36jxAlEnwd4eP00AY6SxD9xi5H  
fJ7zX0+2hF4bVgofRGNmr5IA9SPL0yRJo+dNm3Sh+MhdNMpvJp0dk0PWqq1ZP9LC

1. Can someone spy on your data if your connection is not secured?

- (A) Yes                    (B) No

2. Why do we need HTTPS?

- (A) For privacy and identification                    (B) For faster websites  
(C) For privacy, integrity, and identification                    (D) For identification only

3. In the context of HTTPS, what does integrity mean?

- (A) That my browser has ethics  
(B) That communication is not being tampered with  
(C) That the website I am visiting is honest  
(D) That the internet is strong and durable

Reference: The differences between HTTPS, SSL, and TLS

<https://howhttps.works/https-ssl-tls-differences/>

1. Can someone spy on your data if your connection is not secured?

- (A) Yes                    (B) No

2. Why do we need HTTPS?

- (A) For privacy and identification                    (B) For faster websites  
(C) For privacy, integrity, and identification                    (D) For identification only

3. In the context of HTTPS, what does integrity mean?

- (A) That my browser has ethics  
(B) That communication is not being tampered with  
(C) That the website I am visiting is honest  
(D) That the internet is strong and durable

Reference: The differences between HTTPS, SSL, and TLS

<https://howhttps.works/https-ssl-tls-differences/>

1. In \_\_\_\_\_ cryptography, the same key is used by the sender (for encryption) and the receiver (for decryption).
- (A) Symmetric-key                          (B) Asymmetric-key  
(C) Public-key                              (D) None of the above
2. In \_\_\_\_\_ cryptography, everyone has access to everyone's public key.
- (A) Symmetric-key                          (B) Asymmetric-key  
(C) Both (A) and (B)                      (D) None of the above

Reference: Chapter 28 Quiz

[https://highered.mheducation.com/sites/0072967722/student\\_view0/chapter\\_28\\_quiz.html](https://highered.mheducation.com/sites/0072967722/student_view0/chapter_28_quiz.html)

1. In \_\_\_\_\_ cryptography, the same key is used by the sender (for encryption) and the receiver (for decryption).  

(A) <u>Symmetric-key</u>	(B) Asymmetric-key
(C) Public-key	(D) None of the above
2. In \_\_\_\_\_ cryptography, everyone has access to everyone's public key.  

(A) Symmetric-key	(B) <u>Asymmetric-key</u>
(C) Both (A) and (B)	(D) None of the above

Reference: Chapter 28 Quiz

[https://highered.mheducation.com/sites/0072967722/student\\_view0/chapter\\_28\\_quiz.html](https://highered.mheducation.com/sites/0072967722/student_view0/chapter_28_quiz.html)

## 1. What is hashing in the context of security?

Answer: A hash function is any function that can be used to map data of arbitrary size to fixed-size values. The values returned by a hash function are called hash values, hash codes, digests, or simply hashes.

## 2. Why are hash functions used to store passwords in DB?

Answer: Since hash is a one-way compression function, instead of storing the password itself, its hash value is stored. The user enters their password at the time of login. Then the hash value of the entered password will be compared with the stored hash value in DB and if both matched then the user will be logged in.

Reference: 20 Web Security Questions

<https://circuit.bcit.ca/repository/islandora/object/repository%3A1362/datarstream/PDF/view>

1. A \_\_\_\_\_ serves as the trusted third-party agency that is responsible for issuing the digital certificates.

Answer: Certificate Authority (CA)

Explanation: A CA is an entity that is responsible for issuing digital certificates. These certificates are used to verify the authenticity and integrity of digital communications and transactions. The CA acts as a trusted third-party agency, ensuring that the certificates are issued to the correct entities and that they can be trusted by relying parties. The CA uses cryptographic algorithms to generate and sign these certificates, providing a secure and reliable mechanism for establishing trust in the digital world.

Reference: How Much Do You Know About Digital Certificate? Chapter 12 Quiz

[https://www.proprofs.com/quiz-school/story.php?title=chap-12\\_4](https://www.proprofs.com/quiz-school/story.php?title=chap-12_4)

Reference: Digital Certificates Explained - How digital certificates bind owners to their public key

<https://www.youtube.com/watch?v=5rT6fZUwhG8>

# COMP3052 Computer Security

## Session 03: Foundations of Security

Videoclip: DDoS Attack Explained

<https://www.youtube.com/watch?v=ilhGh9CEIwM>

Videoclip: Smurf Attack Explained

[https://www.youtube.com/watch?v=u\\_75dJCQFGg&t=57s](https://www.youtube.com/watch?v=u_75dJCQFGg&t=57s)

1. What is the CIA triad?
  - (A) Ongoing validation processes involving all employees in an organization
  - (B) A mandatory security framework involving the selection of appropriate controls
  - (C) A foundational security model used to set up security policies and systems
  - (D) A set of security controls used to update systems and networks
  
2. Which element of the CIA triad specifies that only authorized users can access specific information?
  - (A) Confirmation
  - (B) Confidentiality
  - (C) Integrity
  - (D) Access

Reference: Test your knowledge: The CIA triad Quiz Answers

<https://niyander.com/test-your-knowledge-the-cia-triad-quiz-answers/>

Reference: What is the CIA Triad?

<https://www.youtube.com/watch?v=kPPFNrlN3zo>

1. What is the CIA triad?
  - (A) Ongoing validation processes involving all employees in an organization
  - (B) A mandatory security framework involving the selection of appropriate controls
  - (C) A foundational security model used to set up security policies and systems
  - (D) A set of security controls used to update systems and networks

Answer: The CIA triad is a foundational security model used to set up security policies and systems. The core principles of the model are confidentiality, integrity, and availability.

2. Which element of the CIA triad specifies that only authorized users can access specific information?
  - (A) Confirmation
  - (B) Confidentiality
  - (C) Integrity
  - (D) Access

Answer: Confidentiality specifies that only authorized users can access specific information.

Reference: Test your knowledge: The CIA triad Quiz Answers

<https://niyander.com/test-your-knowledge-the-cia-triad-quiz-answers/>

Reference: What is the CIA Triad?

<https://www.youtube.com/watch?v=kPPFNrlN3zo>

3. A security analyst discovers that certain data is inaccessible to authorized users, which is preventing these employees from doing their jobs efficiently. The analyst works to fix the application involved in order to allow for timely and reliable access. Which element of the CIA triad does this scenario describe?

- (A) Availability
- (B) Capacity
- (C) Applicability
- (D) Integrity

4. According to the CIA triad, \_\_\_\_\_ refers to ensuring that an organization's data is verifiably correct, authentic, and reliable.

- (A) Availability
- (B) Accuracy
- (C) Integrity
- (D) Credibility

Reference: Test your knowledge: The CIA triad Quiz Answers

<https://niyander.com/test-your-knowledge-the-cia-triad-quiz-answers/>

3. A security analyst discovers that certain data is inaccessible to authorized users, which is preventing these employees from doing their jobs efficiently. The analyst works to fix the application involved in order to allow for timely and reliable access. Which element of the CIA triad does this scenario describe?

(A) Availability

(B) Capacity

(C) Applicability

(D) Integrity

Answer: This scenario describes availability. Availability specifies that data is accessible to authorized users.

4. According to the CIA triad, \_\_\_\_\_ refers to ensuring that an organization's data is verifiably correct, authentic, and reliable.

(A) Availability

(B) Accuracy

(C) Integrity

(D) Credibility

Answer: According to the CIA triad, integrity refers to ensuring that an organization's data is verifiably correct, authentic, and reliable.

Reference: Test your knowledge: The CIA triad Quiz Answers

<https://niyander.com/test-your-knowledge-the-cia-triad-quiz-answers/>

5. Which of the following statements accurately describes a Smurf attack?
- (A) A DoS attack that is caused when an attacker pings a system by sending it oversized ICMP packet that is bigger than the maximum size
  - (B) A network attack performed when an attacker sniffs an authorized user's IP address and floods it with packets
  - (C) A DoS attack performed by an attacker repeatedly sending ICMP packets to a network server
  - (D) A network attack performed when an attacker intercepts a data packet in transit and delays it or repeats it at another time

Reference: Course 3 – Connect and protect: networks and network security

<https://quitztudy.com/coursera-google-courses/google-cybersecurity/secure-against-network-intrusions-course-3-module-3/>

Reference: Smurf attack: What it is and how it works

<https://nordvpn.com/blog/what-is-smurf-attack/>

5. Which of the following statements accurately describes a Smurf attack?
- (A) A DoS attack that is caused when an attacker pings a system by sending it oversized ICMP packet that is bigger than the maximum size
  - (B) A network attack performed when an attacker sniffs an authorized user's IP address and floods it with packets
  - (C) A DoS attack performed by an attacker repeatedly sending ICMP packets to a network server
  - (D) A network attack performed when an attacker intercepts a data packet in transit and delays it or repeats it at another time

Explanation: A Smurf attack is a network attack performed when an attacker sniffs an unauthorized user's IP address and floods it with packets. It is a combination of a DDoS attack and an IP Spoofing attack.

Reference: Course 3 – Connect and protect: networks and network security

<https://quitztudy.com/coursera-google-courses/google-cybersecurity/secure-against-network-intrusions-course-3-module-3/>

Reference: Smurf attack: What it is and how it works

<https://nordvpn.com/blog/what-is-smurf-attack/>

6. What term means that a user cannot deny a specific action because there is positive proof that he or she performed it?
- (A) Accountability
  - (B) Auditing
  - (C) Nonrepudiation
  - (D) Validation

Reference: Identity and Access Management

<https://www.pearsonitcertification.com/articles/article.aspx?p=2738310>

6. What term means that a user cannot deny a specific action because there is positive proof that he or she performed it?
- (A) Accountability
  - (B) Auditing
  - (C) Nonrepudiation
  - (D) Validation

Reference: Identity and Access Management

<https://www.pearsonitcertification.com/articles/article.aspx?p=2738310>

7. What is the difference between the security and safety aspects of a system? Explain with an example.

Answer: Safety in the context of a nuclear power plant disaster refers to the impact of the event on the environment and human health. The explosion and subsequent release of radioactive material from the nuclear power plant has devastating consequences for the surrounding area. The safety concerns revolve around the radiation exposure to the population, the contamination of soil and water, and the long-term health effects on individuals.

Security, on the other hand, refers to the system's failures or vulnerabilities that may lead to a disaster. There may be several security failures that may contribute to the accident. These include design flaws in the reactor, inadequate operating protocols, human errors, and a lack of proper training for the operators. The security failures in this case may be the underlying causes that may allow the disaster to occur.

So, in summary, safety concerns the impact on the environment and human health following a disaster, while security focuses on the failures within the system that may lead to a disaster.

# COMP3052 Computer Security

## Session 04: Authentication

Videoclip: This is How Hackers Crack Passwords!

<https://www.youtube.com/watch?v=YiRPT4vrSSw>

Videoclip: Brute Force Password Cracking with Hashcat

[https://www.youtube.com/watch?v=fIU\\_8pxNQ3g](https://www.youtube.com/watch?v=fIU_8pxNQ3g)

Videoclip: how to HACK a password // password cracking with Kali Linux and HashCat

[https://www.youtube.com/watch?v=z4\\_oqTZJqCo](https://www.youtube.com/watch?v=z4_oqTZJqCo)

1. (Spoofing / Phishing) An email asking the user to confirm personal information – for example, ‘we couldn’t verify your information – click on the link to confirm the same’.
2. (Spoofing / Phishing) Hackers break into a website and change the IP address of the site.
3. (Spoofing / Phishing) Phone calls or emails from your bank requesting an OTP or your bank PIN.

Reference: Phishing vs Spoofing

<https://www.educba.com/phishing-vs-spoofing/>

4. (Spoofing / Phishing) An email indicating that an Amazon payment had failed.
5. (Spoofing / Phishing) When the user login into a website that appears to be a banking website, the user discovers that the user's account has been stolen.
6. (Spoofing / Phishing) An email that encourages the user with the promise of tax refunds.

Reference: Phishing vs Spoofing

<https://www.educba.com/phishing-vs-spoofing/>

1. (Spoofing / Phishing) An email asking the user to confirm personal information – for example, ‘we couldn’t verify your information – click on the link to confirm the same’.
2. (Spoofing / Phishing) Hackers break into a website and change the IP address of the site.
3. (Spoofing / Phishing) Phone calls or emails from your bank requesting an OTP or your bank PIN.

Reference: Phishing vs Spoofing

<https://www.educba.com/phishing-vs-spoofing/>

4. (Spoofing / Phishing) An email indicating that an Amazon payment had failed.
  
5. (Spoofing / Phishing) When the user login into a website that appears to be a banking website, the user discovers that the user's account has been stolen.
  
6. (Spoofing / Phishing) An email that encourages the user with the promise of tax refunds.

Reference: Phishing vs Spoofing

<https://www.educba.com/phishing-vs-spoofing/>

1. What is the term used when a malicious party sends a fraudulent email disguised as being from a legitimate, trusted source?
  - (A) Keylogging
  - (B) Spoofing
  - (C) Phishing
  - (D) Trojan
  
2. A user receives a phone call from a person who claims to represent IT services and then asks that user for confirmation of username and password for auditing purposes. Which security threat does this phone call represent?
  - (A) Anonymous Keylogging
  - (B) Social Engineering
  - (C) DDoS
  - (D) Spamming

Reference: Module 2: Quiz – Network Threats (Answers) Network Security

<https://itexamanswers.net/module-2-quiz-network-threats-answers-network-security.html>

1. What is the term used when a malicious party sends a fraudulent email disguised as being from a legitimate, trusted source?  
(A) Keylogging  
(B) Spoofing  
(C) Phishing  
(D) Trojan
2. A user receives a phone call from a person who claims to represent IT services and then asks that user for confirmation of username and password for auditing purposes. Which security threat does this phone call represent?  
(A) Anonymous Keylogging  
(B) Social Engineering  
(C) DDoS  
(D) Spamming

Reference: Module 2: Quiz – Network Threats (Answers) Network Security

<https://itexamanswers.net/module-2-quiz-network-threats-answers-network-security.html>

3. What is keylogging?

- (A) The act of recording a user's computer screen.
- (B) It is when keystrokes do not register correctly on a computer.
- (C) The act of recording which keys a user presses on their keyboard.
- (D) It is when a malicious actor hacks into someone's social media accounts.

4. Which statement regarding a keylogger is NOT true?

- (A) Software keyloggers can be designed to send captured information automatically back to the attacker through the Internet.
- (B) Software keyloggers are generally easy to detect.
- (C) Hardware keyloggers are installed between the keyboard connector and computer keyboard USB port.
- (D) Keyloggers can be used to capture passwords, credit card numbers, or personal information.

Reference: Quesba

<https://www.quesba.com/questions/statement-regarding-keylogger-not-true-software-keyloggers-designed-send-823751>

3. What is keylogging?

- (A) The act of recording a user's computer screen.
- (B) It is when keystrokes do not register correctly on a computer.
- (C) The act of recording which keys a user presses on their keyboard.
- (D) It is when a malicious actor hacks into someone's social media accounts.

4. Which statement regarding a keylogger is NOT true?

- (A) Software keyloggers can be designed to send captured information automatically back to the attacker through the Internet.
- (B) Software keyloggers are generally easy to detect.
- (C) Hardware keyloggers are installed between the keyboard connector and computer keyboard USB port.
- (D) Keyloggers can be used to capture passwords, credit card numbers, or personal information.

Reference: Quesba

<https://www.quesba.com/questions/statement-regarding-keylogger-not-true-software-keyloggers-designed-send-823751>

5. Technicians are testing the security of an authentication system that uses passwords. When a technician examines the password tables, the technician discovers the passwords are stored as hash values. However, after comparing a simple password hash, the technician then discovers that the values are different from those on other systems. What are two causes of this situation? (Choose two.)

- (A) Both systems scramble the passwords before hashing.
- (B) The systems use different hashing algorithms.
- (C) One system uses hashing and the other uses hashing and salting.
- (D) Both systems use MD5.
- (E) One system uses symmetrical hashing and the other uses asymmetrical hashing.

6. What is a feature of a cryptographic hash function?

- (A) Hashing requires a public and a private key.
- (B) The hash function is a one-way mathematical function.
- (C) The output has a variable length.
- (D) The hash input can be calculated given the output value.

Reference: Cybersecurity Essentials FINAL Quiz Answers Full Questions

<https://itexamanswers.net/cybersecurity-essentials-final-quiz-answers-full-questions.html>

5. Technicians are testing the security of an authentication system that uses passwords. When a technician examines the password tables, the technician discovers the passwords are stored as hash values. However, after comparing a simple password hash, the technician then discovers that the values are different from those on other systems. What are two causes of this situation? (Choose two.)

- (A) Both systems scramble the passwords before hashing.
- (B) The systems use different hashing algorithms.
- (C) One system uses hashing and the other uses hashing and salting.
- (D) Both systems use MD5.
- (E) One system uses symmetrical hashing and the other uses asymmetrical hashing.

6. What is a feature of a cryptographic hash function?

- (A) Hashing requires a public and a private key.
- (B) The hash function is a one-way mathematical function.
- (C) The output has a variable length.
- (D) The hash input can be calculated given the output value.

Reference: Cybersecurity Essentials FINAL Quiz Answers Full Questions

<https://itexamanswers.net/cybersecurity-essentials-final-quiz-answers-full-questions.html>

7. A user has created a new program and wants to distribute it to everyone in the company. The user wants to ensure that when the program is downloaded that the program is not changed while in transit. What can the user do to ensure that the program is not changed when downloaded?
- (A) Create a hash of the program file that can be used to verify the integrity of the file after it is downloaded.
  - (B) Turn off antivirus on all the computers.
  - (C) Distribute the program on a thumb drive.
  - (D) Encrypt the program and require a password after it is downloaded.
  - (E) Install the program on individual computers.
8. Alice and Bob use the same password to login into the company network. This means both would have the exact same hash for their passwords. What could be implemented to prevent both password hashes from being the same?
- (A) Peppering
  - (B) Pseudo-random generator
  - (C) Salting
  - (D) RSA

Reference: Cybersecurity Essentials Chapter 5 Quiz Questions Answers

<https://itexamanswers.net/cybersecurity-essentials-chapter-5-quiz-questions-answers.html>

7. A user has created a new program and wants to distribute it to everyone in the company. The user wants to ensure that when the program is downloaded that the program is not changed while in transit. What can the user do to ensure that the program is not changed when downloaded?
- (A) Create a hash of the program file that can be used to verify the integrity of the file after it is downloaded.
- (B) Turn off antivirus on all the computers.
- (C) Distribute the program on a thumb drive.
- (D) Encrypt the program and require a password after it is downloaded.
- (E) Install the program on individual computers.
8. Alice and Bob use the same password to login into the company network. This means both would have the exact same hash for their passwords. What could be implemented to prevent both password hashes from being the same?
- (A) Peppering
- (B) Pseudo-random generator
- (C) Salting
- (D) RSA

Reference: Cybersecurity Essentials Chapter 5 Quiz Questions Answers

<https://itexamanswers.net/cybersecurity-essentials-chapter-5-quiz-questions-answers.html>

9. Which method tries all possible passwords until a match is found?
- (A) Brute force
  - (B) Rainbow tables
  - (C) Dictionary
  - (D) Cryptographic
10. A user is evaluating the security infrastructure of a company and notices that some authentication systems are not using best practices when it comes to storing passwords. The user is able to crack passwords very fast and access sensitive data. The user wants to present a recommendation to the company on the proper implementation of salting to avoid password cracking techniques. What are three best practices in implementing salting? (Choose two.)
- (A) A salt should not be reused.
  - (B) A salt should be unique for each password.
  - (C) The same salt should be used for each password.
  - (D) Salts should be short.

Reference: Cybersecurity Essentials Chapter 5 Quiz Questions Answers

<https://itexamanswers.net/cybersecurity-essentials-chapter-5-quiz-questions-answers.html>

9. Which method tries all possible passwords until a match is found?

- (A) Brute force
- (B) Rainbow tables
- (C) Dictionary
- (D) Cryptographic

10. A user is evaluating the security infrastructure of a company and notices that some authentication systems are not using best practices when it comes to storing passwords. The user is able to crack passwords very fast and access sensitive data. The user wants to present a recommendation to the company on the proper implementation of salting to avoid password cracking techniques. What are three best practices in implementing salting? (Choose two.)

- (A) A salt should not be reused.
- (B) A salt should be unique for each password.
- (C) The same salt should be used for each password.
- (D) Salts should be short.

Reference: Cybersecurity Essentials Chapter 5 Quiz Questions Answers

<https://itexamanswers.net/cybersecurity-essentials-chapter-5-quiz-questions-answers.html>

# COMP3052 Computer Security

## Session 05: Access Control

# Videoclip: Access Control Models: An Overview of the Four Main Types

<https://www.youtube.com/watch?v=7PJdQGALG2k>

1. What are three access control security services? (Choose three.)

- (A) Access
- (B) Authorisation
- (C) Repudiation
- (D) Authentication
- (E) Availability
- (F) Accountability

2. Which component of AAA is used to determine which resources a user can access and which operations the user is allowed to perform?

- (A) Authorisation
- (B) Authentication
- (C) Accountability
- (D) Auditing

Reference: Module 7: Quiz – Authentication, Authorization, and Accounting (AAA) (Answers) Network Security

<https://itexamanswers.net/module-7-quiz-authentication-authorization-and-accounting-aaa-answers-network-security.html>

1. What are three access control security services? (Choose three.)

- (A) Access
- (B) Authorisation
- (C) Repudiation
- (D) Authentication
- (E) Availability
- (F) Accountability

2. Which component of AAA is used to determine which resources a user can access and which operations the user is allowed to perform?

- (A) Authorisation
- (B) Authentication
- (C) Accountability
- (D) Auditing

Reference: Module 7: Quiz – Authentication, Authorization, and Accounting (AAA) (Answers) Network Security

<https://itexamanswers.net/module-7-quiz-authentication-authorization-and-accounting-aaa-answers-network-security.html>

3. Which term describes the ability of a web server to keep a log of the users who access the server, as well as the length of time they use it?
  - (A) Assigning permissions
  - (B) Authorisation
  - (C) Authentication
  - (D) Accountability
4. Because of implemented security controls, a user can only access a server with FTP. Which AAA component accomplishes this?
  - (A) Authorisation
  - (B) Authentication
  - (C) Accountability
  - (D) Auditing

Reference: Module 7: Quiz – Authentication, Authorization, and Accounting (AAA) (Answers) Network Security

<https://itexamanswers.net/module-7-quiz-authentication-authorization-and-accounting-aaa-answers-network-security.html>

3. Which term describes the ability of a web server to keep a log of the users who access the server, as well as the length of time they use it?
- (A) Assigning permissions
  - (B) Authorisation
  - (C) Authentication
  - (D) Accountability
4. Because of implemented security controls, a user can only access a server with FTP. Which AAA component accomplishes this?
- (A) Authorisation
  - (B) Authentication
  - (C) Accountability
  - (D) Auditing

Reference: Module 7: Quiz – Authentication, Authorization, and Accounting (AAA) (Answers) Network Security  
<https://itexamanswers.net/module-7-quiz-authentication-authorization-and-accounting-aaa-answers-network-security.html>

5. When a security audit is performed at a company, the auditor reports that new users have access to network resources beyond their normal job roles. Additionally, users who move to different positions retain their prior permissions. What kind of violation is occurring?
- (A) Password
  - (B) Network Policy
  - (C) Least Privilege
  - (D) Audit
6. Which access control model assigns security privileges based on the position, responsibilities, or job classification of an individual or group within an organization?
- (A) Discretionary
  - (B) Mandatory
  - (C) Group Based
  - (D) Role Based

Reference: 3.5.2 Module 3 – Access Control Quiz Answers

<https://itexamanswers.net/3-5-2-module-3-access-control-quiz-answers.html>

5. When a security audit is performed at a company, the auditor reports that new users have access to network resources beyond their normal job roles. Additionally, users who move to different positions retain their prior permissions. What kind of violation is occurring?
- (A) Password
  - (B) Network Policy
  - (C) Least Privilege
  - (D) Audit
6. Which access control model assigns security privileges based on the position, responsibilities, or job classification of an individual or group within an organization?
- (A) Discretionary
  - (B) Mandatory
  - (C) Group Based
  - (D) Role Based

Reference: 3.5.2 Module 3 – Access Control Quiz Answers

<https://itexamanswers.net/3-5-2-module-3-access-control-quiz-answers.html>

7. Which type of access control applies the strictest access control and is commonly used in military or mission critical applications?
- (A) Discretionary Access Control
  - (B) Mandatory Access Control
  - (C) Attribute Based Access Control
  - (D) Identity Based Access Control
8. After a security audit for an organization, multiple accounts were found to have privileged access to systems and devices. Which three best practices for securing privileged accounts should be included in the audit report? (Choose three.)
- (A) Secure password storage.
  - (B) Reduce the number of privileged accounts.
  - (C) Only the CIO should have privileged access.
  - (D) No one should have privileged access.
  - (E) Enforce the principle of least privilege.
  - (F) Only managers should have privileged access.

Reference: 3.5.2 Module 3 – Access Control Quiz Answers

<https://itexamanswers.net/3-5-2-module-3-access-control-quiz-answers.html>

7. Which type of access control applies the strictest access control and is commonly used in military or mission critical applications?
- (A) Discretionary Access Control
  - (B) Mandatory Access Control
  - (C) Attribute Based Access Control
  - (D) Identity Based Access Control
8. After a security audit for an organization, multiple accounts were found to have privileged access to systems and devices. Which three best practices for securing privileged accounts should be included in the audit report? (Choose three.)
- (A) Secure password storage.
  - (B) Reduce the number of privileged accounts.
  - (C) Only the CIO should have privileged access.
  - (D) No one should have privileged access.
  - (E) Enforce the principle of least privilege.
  - (F) Only managers should have privileged access.

Reference: 3.5.2 Module 3 – Access Control Quiz Answers

<https://itexamanswers.net/3-5-2-module-3-access-control-quiz-answers.html>

# COMP3052 Computer Security

## Session 06: Firewalls

Videoclip: What is Firewall? Simplilearn

<https://www.youtube.com/watch?v=9GZlVOafYTg>

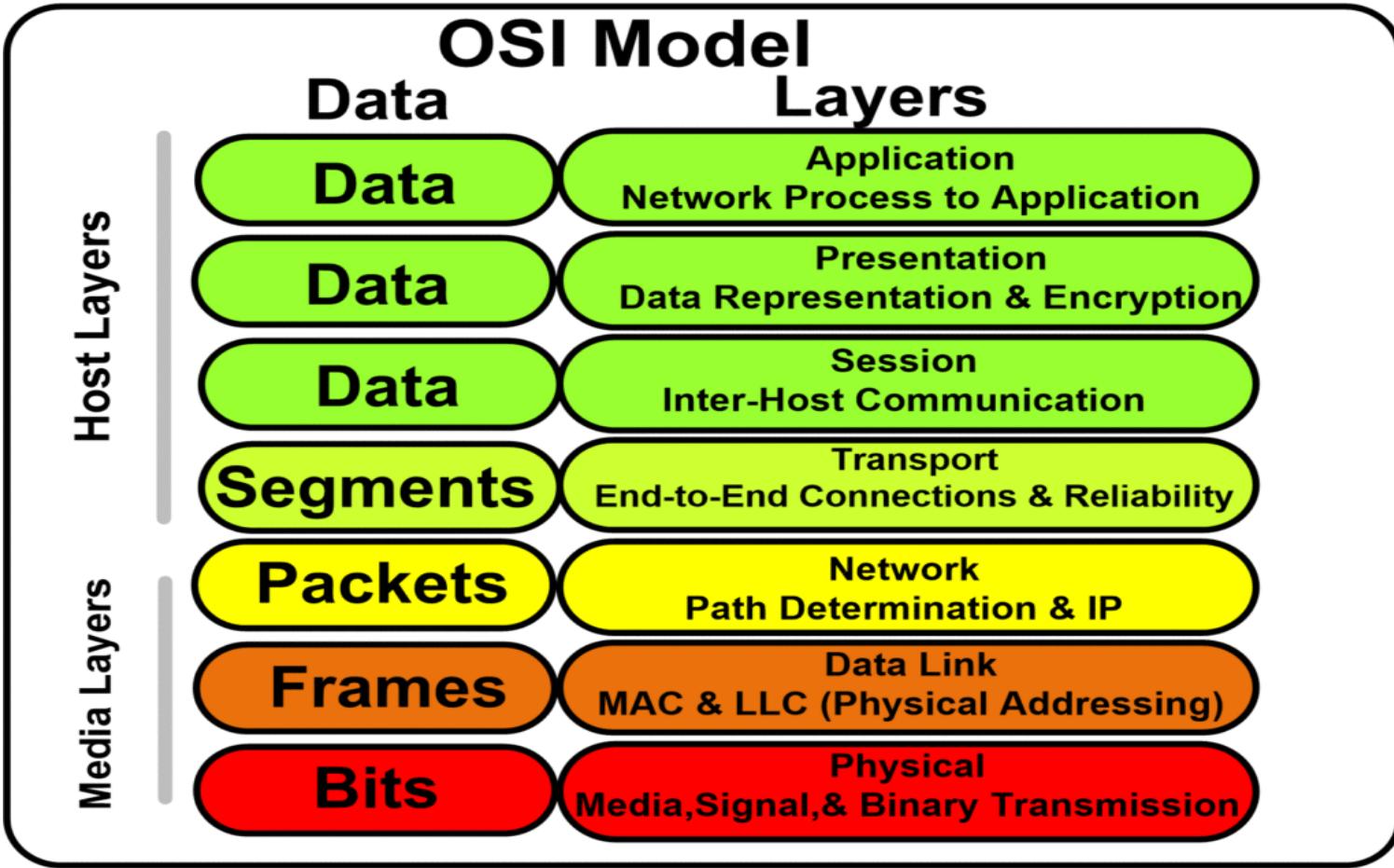
Videoclip: What is a Firewall? PowerCert Animated Videos

<https://www.youtube.com/watch?v=kDEX1HXybrU>

```
214.1.211.251 - - [15/Apr/2011:09:40:17 -0700] "GET /global.asa HTTP/1.0" 404 315 "-" "-"
214.1.211.251 - - [15/Apr/2011:09:40:17 -0700] "GET /~root HTTP/1.0" 404 310 "-" "-"
214.1.211.251 - - [15/Apr/2011:09:40:18 -0700] "GET /~apache HTTP/1.0" 404 312 "-" "-"
219.167.17.173 - - [17/Apr/2011:17:55:40 -0700] "POST /sony/mmr HTTP/1.1" 200 130 "-" "PS
218.41.54.67 - - [17/Apr/2011:18:20:18 -0700] "POST /sony/mmr HTTP/1.1" 200 130 "-" "PS3A
10.132.93.114 - - [18/Apr/2011:11:05:39 -0700] "POST /sony/mmr HTTP/1.1" 200 61 "-" "Ledi
10.132.93.114 - - [18/Apr/2011:11:07:07 -0700] "POST /sony/mmr HTTP/1.1" 200 61 "-" "Ledi
10.132.93.114 - - [18/Apr/2011:11:13:52 -0700] "POST /sony/mmr HTTP/1.1" 200 61 "-" "Ledi
218.41.54.67 - - [20/Apr/2011:17:42:37 -0700] "POST /sony/mmr HTTP/1.1" 200 100 "-" "PS3A
60.34.131.229 - - [20/Apr/2011:18:22:32 -0700] "POST /sony/mmr HTTP/1.1" 200 100 "-" "PS3
202.213.251.245 - - [21/Apr/2011:21:16:45 -0700] "POST /sony/mmr HTTP/1.1" 200 100 "-" "F
202.213.251.245 - - [21/Apr/2011:21:24:43 -0700] "POST /sony/mmr HTTP/1.1" 200 100 "-" "F
178.202.110.92 - - [22/Apr/2011:18:59:05 -0700] "GET / HTTP/1.1" 200 315 "-" "Mozilla/5.0
178.202.110.92 - - [22/Apr/2011:18:59:05 -0700] "GET /favicon.ico HTTP/1.1" 404 333 "-" "
178.202.110.92 - - [22/Apr/2011:18:59:05 -0700] "GET /favicon.ico HTTP/1.1" 404 333 "-" "
178.202.110.92 - - [22/Apr/2011:18:59:07 -0700] "GET /access-navigator-media HTTP/1.1" 20
178.202.110.92 - - [22/Apr/2011:19:05:00 -0700] "GET /admin/cdr/counter.txt HTTP/1.1" 404
178.202.110.92 - - [22/Apr/2011:19:05:41 -0700] "GET //help/readme.nsf?OpenAbout HTTP/1.1
178.202.110.92 - - [22/Apr/2011:19:05:54 -0700] "GET /catinfo?A HTTP/1.1" 404 329 "-" "Mc
178.202.110.92 - - [22/Apr/2011:19:06:08 -0700] "GET /errors-navigator-media HTTP/1.1" 20
178.202.110.92 - - [22/Apr/2011:19:27:04 -0700] "GET / HTTP/1.1" 200 315 "-" "Mozilla/5.0
```

## Reference: Network Logging: Definition & Tools

<https://study.com/academy/lesson/network-logging-definition-tools.html>

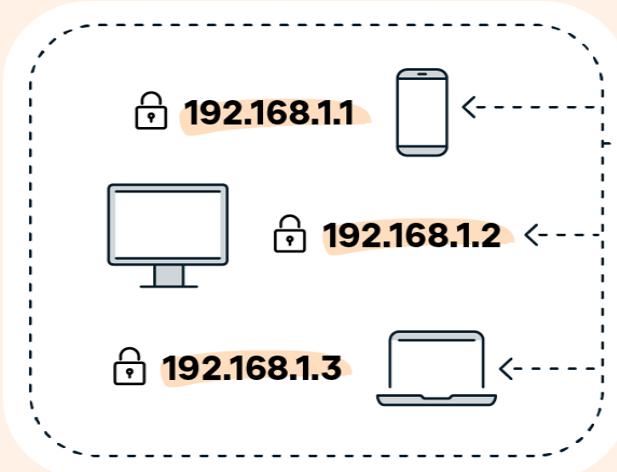


Reference: 7 Layers of OSI Model and Their Functions

<https://electricala2z.com/cloud-computing/osi-model-layers-7-layers-osi-model/>

# Public vs. Private IP Addresses

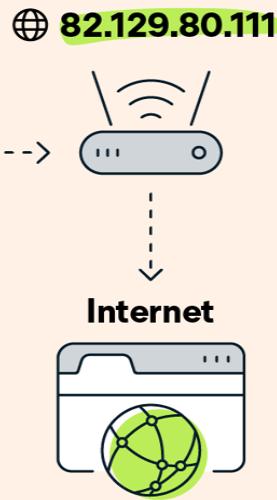
**Private / Local / Internal**  
- automatically generated



Found via internal  
device settings



**Public / External**  
- assigned by ISP



Found by Googling:  
"What is my IP address?"



Your private IP address exists within specific private IP address ranges reserved by the Internet Assigned Numbers Authority (IANA) and should never appear on the internet. There are millions of private networks across the globe, all of which include devices assigned private IP addresses within these ranges:

- Class A: 10.0.0.0 – 10.255.255.255
- Class B: 172.16.0.0 – 172.31.255.255
- Class C: 192.168.0.0 – 192.168.255.255

These might not seem like wide ranges, but they don't really need to be. Because these IP addresses are reserved for private network use only, they can be reused on different private networks all over the world — without consequence or confusion.

Reference: Public vs. Private IP Addresses: What's the Difference?

<https://www.avast.com/c-ip-address-public-vs-private>

1. What is an advantage of UDP over TCP?

- (A) UDP communication requires less overhead.
- (B) UDP communication is more reliable.
- (C) UDP reorders segments that are received out of order.
- (D) UDP acknowledges received data.

2. When is UDP preferred to TCP?

- (A) When a client sends a segment to a server.
- (B) When all the data must be fully received before any part of it is considered useful.
- (C) When an application can tolerate some loss of data during transmission.
- (D) When segments must arrive in a very specific sequence to be processed successfully.

Reference: 15.3.3 Transport Layer Quiz Answers

<https://itexamanswers.net/15-3-3-transport-layer-quiz-answers.html>

1. What is an advantage of UDP over TCP?

(A) UDP communication requires less overhead.

(B) UDP communication is more reliable.

(C) UDP reorders segments that are received out of order.

(D) UDP acknowledges received data.

2. When is UDP preferred to TCP?

(A) When a client sends a segment to a server.

(B) When all the data must be fully received before any part of it is considered useful.

(C) When an application can tolerate some loss of data during transmission.

(D) When segments must arrive in a very specific sequence to be processed successfully.

Reference: 15.3.3 Transport Layer Quiz Answers

<https://itexamanswers.net/15-3-3-transport-layer-quiz-answers.html>

## 1. What are iptables?

Answer: Iptables is a command line utility used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel.

## 2. What a chain is in context with iptables?

Answer: A chain is a set of rules that determine how a packet should be handled. When a packet arrives, it is compared against the rules in each chain until a match is found. The packet is then handled according to the action specified in that rule. There are three built-in chains in iptables: INPUT, OUTPUT, and FORWARD.

Reference: 20 IPTables Interview Questions and Answers

<https://climbtheladder.com/iptables-interview-questions/>

### 3. What are the most commonly used chains in iptables?

Answer: The most commonly used chains in iptables are the INPUT, OUTPUT, and FORWARD chains. The INPUT chain is used to filter incoming traffic, the OUTPUT chain is used to filter outgoing traffic, and the FORWARD chain is used to filter traffic that is being forwarded through the system.

Reference: 20 IPTables Interview Questions and Answers

<https://climbtheladder.com/iptables-interview-questions/>

4. Can you explain how to set up an iptable rule for allowing traffic from any host on the network?

Answer: You can set up an iptable rule for allowing traffic from any host on the network by using the “**iptables -A INPUT -j ACCEPT**” command. This will allow all traffic from all hosts on the network to be accepted.

5. How can you allow access to a particular IP address using iptables?

Answer: You can allow access to a particular IP address using iptables by adding a rule that allows traffic from that IP address. For example, if you wanted to allow traffic from the IP address 1.2.3.4, you would add a rule that looks like this:  
“**iptables -A INPUT -s 1.2.3.4 -j ACCEPT**”

Reference: 20 IPTables Interview Questions and Answers

<https://climbtheladder.com/iptables-interview-questions/>

6. Can you explain how to open port 8080/tcp so that web servers running as non-root users can bind to it?

Answer: You can open port 8080/tcp by adding the following rule to your IPTables configuration: “**iptables -A INPUT -p tcp --dport 8080 -j ACCEPT**”

Reference: 20 IPTables Interview Questions and Answers

<https://climbtheladder.com/iptables-interview-questions/>

8. How do you limit the number of concurrent connections coming from a single source IP address to 100?

Answer: You can use the following rule in your iptables configuration:

**“iptables -A INPUT -p tcp –syn --dport 80 -m connlimit --connlimit-above 100 -j REJECT”**

This rule will limit the number of concurrent connections to port 80 (HTTP) from any single source IP address to 100. If more than 100 connections are attempted, the rule will reject the connection.

9. Can you tell me some disadvantages of iptables?

Answer: Some disadvantages of iptables include the fact that it can be difficult to configure, and it can be resource intensive if you are using it to filter a lot of traffic. Additionally, iptables can be bypassed if an attacker is able to gain access to the server itself, so it is not a perfect security solution.

Reference: 20 IPTables Interview Questions and Answers

<https://climbtheladder.com/iptables-interview-questions/>

# COMP3052 Computer Security

Session 07: Reference Monitor

Videoclip: Protection Mechanisms (CISSP Free by Skillset.com)

[https://www.youtube.com/watch?v=d-vhxg\\_j1kM&t=324s](https://www.youtube.com/watch?v=d-vhxg_j1kM&t=324s)

Videoclip: Linux Architecture 2/5: Kernel/Security/and more!

<https://www.youtube.com/watch?v=85eI NAowuMc>

Videoclip: Segmented, Paged and Virtual Memory

<https://www.youtube.com/watch?v=p9yZNLeOj4s>

1. The properties of a reference monitor are captured by the acronym NEAT. Which of the following is not a property of a reference monitor?
  - (A) The reference validation mechanism must be Non-bypassable, so that an attacker cannot bypass the mechanism and violate the security policy.
  - (B) The reference validation mechanism must be Evaluable, i.e., amenable to analysis and tests, the completeness of which can be assured (verifiable). Without this property, the mechanism might be flawed in such a way that the security policy is not enforced.
  - (C) The reference validation mechanism must be Always invoked. Without this property, it is possible for the mechanism to not perform when intended, allowing an attacker to violate the security policy.
  - (D) The reference validation mechanism must be Tamper-proof. Without this property, an attacker can undermine the mechanism itself and hence violate the security policy.
  - (E) The reference validation mechanism must be Tolerable to all kinds of security vulnerabilities and attacks.

Reference: Reference monitor. You know what it is, right?

<https://community.infosecinstitute.com/discussion/112586/reference-monitor-you-know-what-it-is-right>

1. The properties of a reference monitor are captured by the acronym NEAT. Which is the following is not a property of a reference monitor?
  - (A) The reference validation mechanism must be Non-bypassable, so that an attacker cannot bypass the mechanism and violate the security policy.
  - (B) The reference validation mechanism must be Evaluable, i.e., amenable to analysis and tests, the completeness of which can be assured (verifiable). Without this property, the mechanism might be flawed in such a way that the security policy is not enforced.
  - (C) The reference validation mechanism must be Always invoked. Without this property, it is possible for the mechanism to not perform when intended, allowing an attacker to violate the security policy.
  - (D) The reference validation mechanism must be Tamper-proof. Without this property, an attacker can undermine the mechanism itself and thence violate the security policy.
  - (E) The reference validation mechanism must be Tolerable to all kind of security vulnerabilities and attacks.

Reference: Reference monitor. You know what it is, right?

<https://community.infosecinstitute.com/discussion/112586/reference-monitor-you-know-what-it-is-right>

2. What are the essential characteristics of the reference monitor?
- (A)It is versatile, accurate, and operates at a very high speed.
  - (B)It is tamper-proof, can always be invoked, and must be small enough to test.
  - (C)It is restricted, confidential, and top secret

Reference: Cyber Security MCQs

<https://quizack.com/ecommerce-cyber-security/mcq/what-are-the-essential-characteristics-of-the-reference-monitor>

2. What are the essential characteristics of the reference monitor?
- (A)It is versatile, accurate, and operates at a very high speed.
- (B)It is tamper-proof, can always be invoked, and must be small enough to test.
- (C)It is restricted, confidential, and top secret

Reference: Cyber Security MCQs

<https://quizack.com/ecommerce-cyber-security/mcq/what-are-the-essential-characteristics-of-the-reference-monitor>

3. Which of the following statements pertaining to protection rings is false?
- (A) They provide strict boundaries and definitions on what the processes that work within each ring can access.
  - (B) Programs operating in inner rings are usually referred to as existing in a privileged mode.
  - (C) They support the CIA triad requirements of multitasking operating systems.
  - (D) They provide users with a direct access to peripherals

Reference: VCEguide

<https://vceguide.com/which-of-the-following-statements-pertaining-to-protection-rings-is-false-2/>

3. Which of the following statements pertaining to protection rings is false?
- (A) They provide strict boundaries and definitions on what the processes that work within each ring can access.
  - (B) Programs operating in inner rings are usually referred to as existing in a privileged mode.
  - (C) They support the CIA triad requirements of multitasking operating systems.
  - (D) They provide users with a direct access to peripherals

Reference: VCEguide

<https://vceguide.com/which-of-the-following-statements-pertaining-to-protection-rings-is-false-2/>

4. Which of the followings is malicious software that alters the regular functionality of an OS, takes full control on the targeted system, and acts as the system administrator on the victim's system?

- (A) Virus
- (B) Spyware
- (C) Trojan horse
- (D) Rootkit

Reference: Testbook

[https://testbook.com/question-answer/  
62b1a7ccaaad4a4ea8350814](https://testbook.com/question-answer/62b1a7ccaaad4a4ea8350814)

-is-malicious-software-that-alters-the-regul--

4. Which of the followings is malicious software that alters the regular functionality of an OS, takes full control on the targeted system, and acts as the system administrator on the victim's system?

- (A) Virus
- (B) Spyware
- (C) Trojan horse
- (D) Rootkit

Reference: Testbook

[https://testbook.com/question-answer/  
62b1a7ccaaad4a4ea8350814](https://testbook.com/question-answer/62b1a7ccaaad4a4ea8350814)

-is-malicious-software-that-alters-the-regul--

5. Which of the following statements about SIGSEGV or SIGBUS is incorrect?
- (A) SIGSEGV indicates an invalid access attempt to a valid physical address.
  - (B) SIGBUS indicates an access attempt to an invalid physical address.
  - (C) Stack overflow is a cause of SIGSEGV.
  - (D) An attempt to access a read-only location will cause a SIGBUS.

Reference: What Is Signal 11 SIGSEGV Error?

<https://phoenixnap.com/kb/sigsegv#:~:text=The%20main%20difference%20between%20the,inicates%20an%20invalid%20physical%20address.>

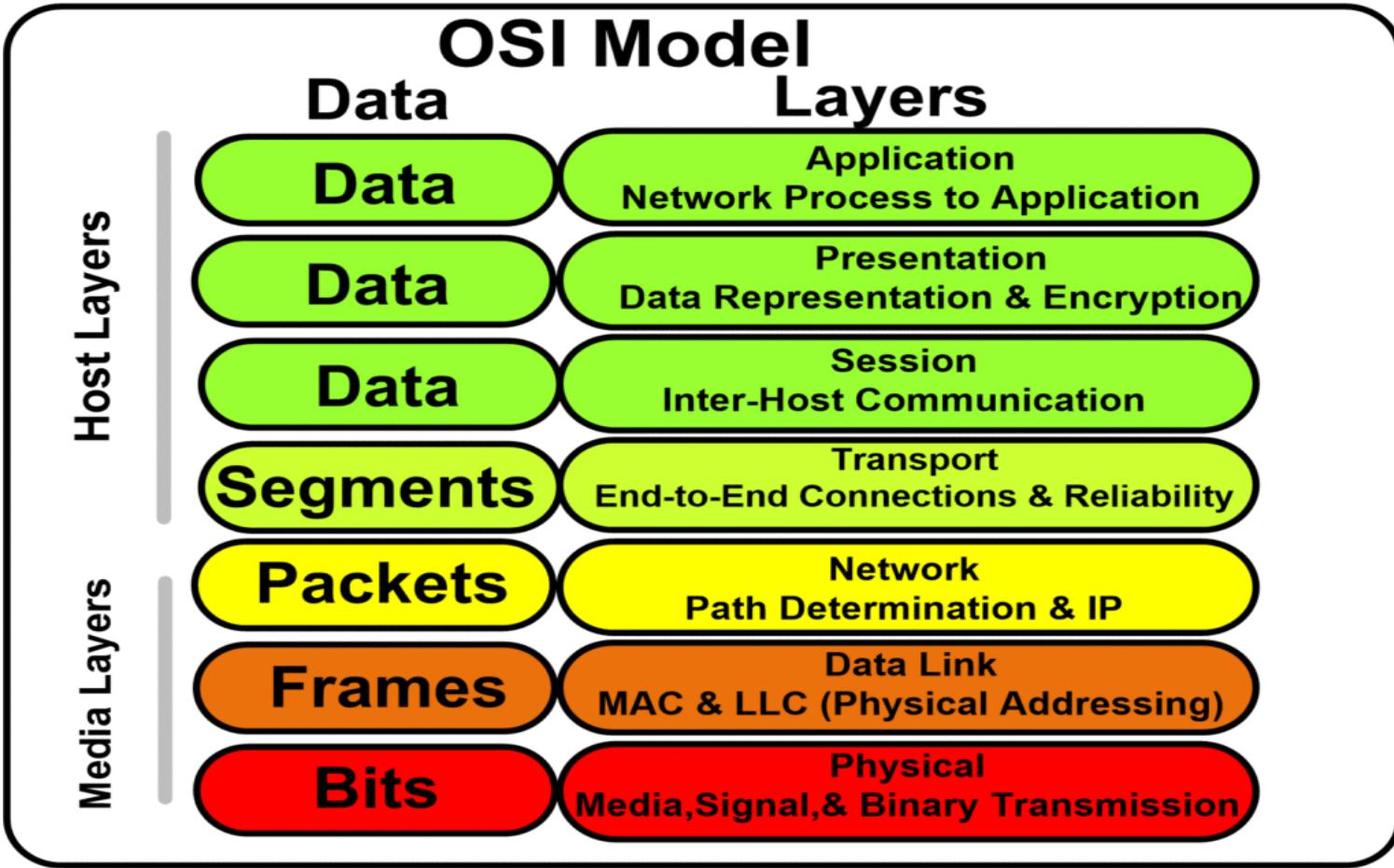
5. Which of the following statements about SIGSEGV or SIGBUS is incorrect?
- (A) SIGSEGV indicates an invalid access attempt to a valid physical address.
  - (B) SIGBUS indicates an access attempt to an invalid physical address.
  - (C) Stack overflow is a cause of SIGSEGV.
  - (D) An attempt to access a read-only location will cause a SIGBUS.

Reference: What Is Signal 11 SIGSEGV Error?

<https://phoenixnap.com/kb/sigsegv#:~:text=The%20main%20difference%20between%20the,inicates%20an%20invalid%20physical%20address.>

# COMP3052 Computer Security

## Session 08: Network Security

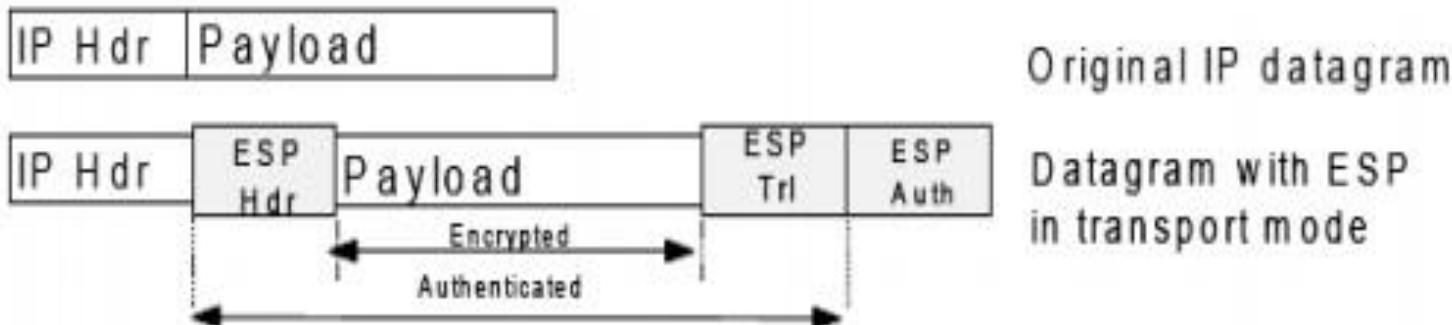


Reference: 7 Layers of OSI Model and Their Functions

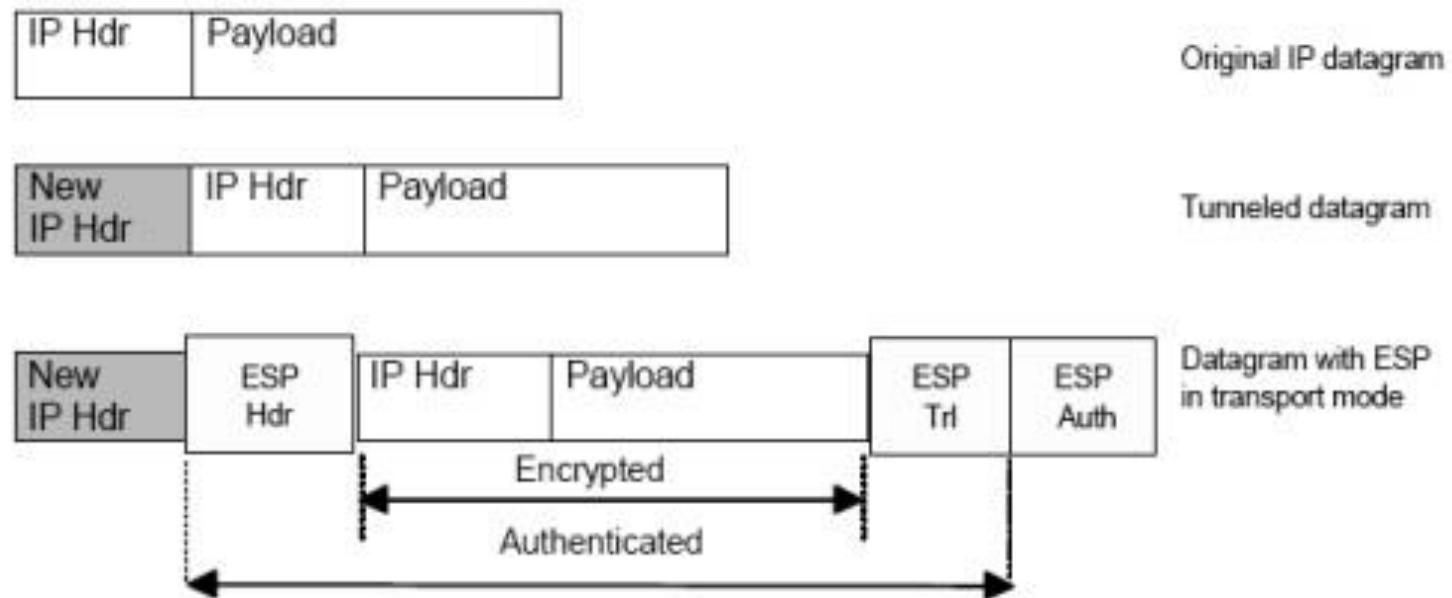
<https://electricala2z.com/cloud-computing/osi-model-layers-7-layers-osi-model/>

1. Which of the following diagrams describes ESP in tunnel mode?

(A)



(B)

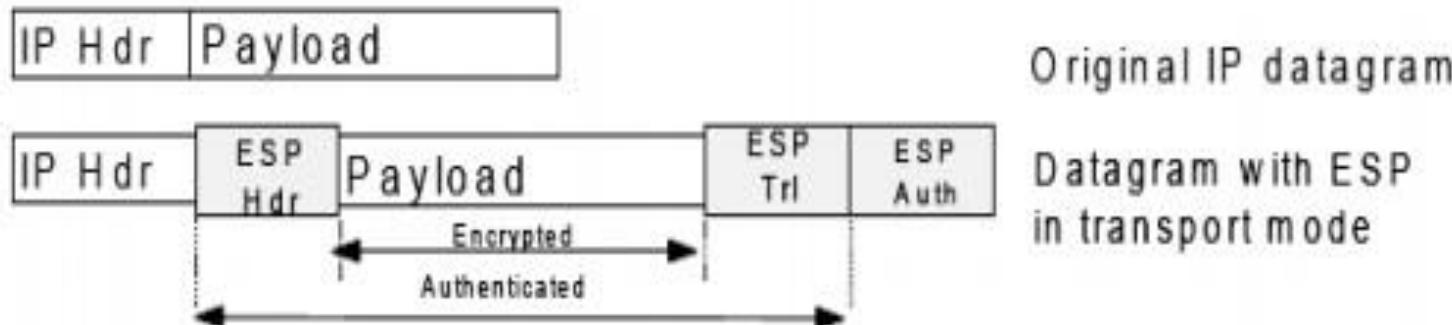


Reference: What is ESP in tunnel and transport mode and the difference between AH and ESP?

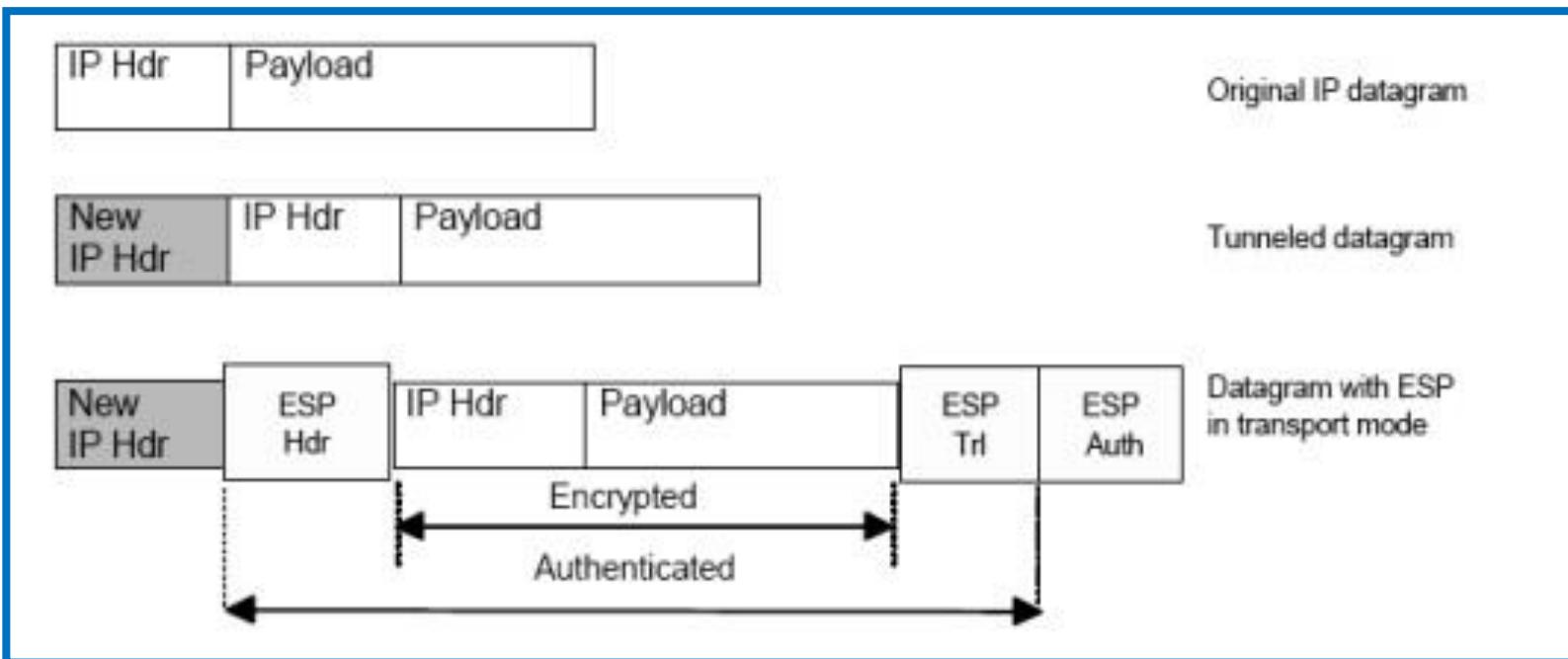
<https://www.tutorialspoint.com/what-is-esp-in-tunnel-and-transport-mode-and-the-difference-between-ah-and-esp>

1. Which of the following diagrams describes ESP in tunnel mode?

(A)



(B)



Reference: What is ESP in tunnel and transport mode and the difference between AH and ESP?

<https://www.tutorialspoint.com/what-is-esp-in-tunnel-and-transport-mode-and-the-difference-between-ah-and-esp>

2. Which protocol is used to discover the destination address needed to be added to an Ethernet frame?
- (A) ARP
  - (B) DNS
  - (C) DHCP
  - (D) HTTP

Reference: 7.2.3 Address Resolution Quiz Answers

<https://itexamanswers.net/7-2-3-address-resolution-quiz-answers.html>

3. Which protocol is used to discover the destination address needed to be added to an Ethernet frame?

- (A) ARP
- (B) DNS
- (C) DHCP
- (D) HTTP

Reference: 7.2.3 Address Resolution Quiz Answers

<https://itexamanswers.net/7-2-3-address-resolution-quiz-answers.html>

4. What is one function of the ARP protocol?
- (A) Obtaining an IPv4 address automatically
  - (B) Mapping a domain name to its IP address
  - (C) Resolving an IPv4 address to a MAC address
  - (D) Maintaining a table of domain names with their resolved IP addresses

Reference: 7.2.3 Address Resolution Quiz Answers

<https://itexamanswers.net/7-2-3-address-resolution-quiz-answers.html>

3. What is one function of the ARP protocol?
- (A) Obtaining an IPv4 address automatically
  - (B) Mapping a domain name to its IP address
  - (C) Resolving an IPv4 address to a MAC address
  - (D) Maintaining a table of domain names with their resolved IP addresses

Reference: 7.2.3 Address Resolution Quiz Answers

<https://itexamanswers.net/7-2-3-address-resolution-quiz-answers.html>

4. Refer to the exhibit below. What is occurring in this network?

Interface: 192.168.1.29 --- 0x11

Internet Address	Physical Address	Type
192.168.1.10	d8-a7-56-d7-19-ea	dynamic
192.168.1.67	d8-a7-56-d7-19-ea	dynamic
192.168.1.1	01-00-5e-00-00-16	static

- (A) ARP cache poisoning
- (B) DNS cache poisoning
- (C) MAC address table overflow
- (D) MAC flooding attack

Reference: Exam Topics

<https://www.examtopics.com/discussions/cisco/view/65956-exam-200-201-topic-1-question-63-discussion/>

4. Refer to the exhibit below. What is occurring in this network?

Interface: 192.168.1.29 --- 0x11

Internet Address	Physical Address	Type
192.168.1.10	d8-a7-56-d7-19-ea	dynamic
192.168.1.67	d8-a7-56-d7-19-ea	dynamic
192.168.1.1	01-00-5e-00-00-16	static

- (A) ARP cache poisoning
- (B) DNS cache poisoning
- (C) MAC address table overflow
- (D) MAC flooding attack

ARP poisoning, is a Man in the Middle (MitM) attack that allows attackers to intercept communication between network devices. The attacker must have access to the network. They scan the network to determine the IP addresses of at least two devices. The attacker uses a spoofing tool such as Arpspoof or Driftnet, to send out forged ARP responses. The forged responses advertise that the correct MAC address for both IP addresses, belonging to the router and workstation, is the attacker's MAC address. This fools both router and workstation to connect to the attacker's machine, instead of to each other. The two devices update their ARP cache entries and from that point onwards, communicate with the attacker instead of directly with each other. The attacker is now secretly in the middle of all communications.

Reference: Exam Topics

<https://www.examtopics.com/discussions/cisco/view/65956-exam-200-201-topic-1-question-63-discussion/>

5. Which of the following threats commonly relies on DNS poisoning and spoofing to exploit an unknowing victim?
- (A) Rainbow tables
  - (B) Brute force
  - (C) Man-in-the-middle
  - (D) Zero-day attacks
  - (E) Phishing

Reference: Exam Topics

<https://www.examtopics.com/discussions/comptia/view/75105-exam-220-1002-topic-1-question-492-discussion/>

5. Which of the following threats commonly relies on DNS poisoning and spoofing to exploit an unknowing victim?
- (A) Rainbow tables
  - (B) Brute force
  - (C) Man-in-the-middle
  - (D) Zero-day attacks
  - (E) Phishing

DNS spoofing is a type of attack in which a malicious actor intercepts DNS request and returns the address that leads to its own server instead of the real address. Hackers can use DNS spoofing to launch a man-in-the-middle attack and direct the victim to a bogus site that looks like the real one.

Reference: Exam Topics

<https://www.examtopics.com/discussions/comptia/view/75105-exam-220-1002-topic-1-question-492-discussion/>

# COMP3052 Computer Security

## Session 09: Internet Security

Videoclip: Digital Certificates Explained - How digital certificates bind owners to their public key

<https://www.youtube.com/watch?v=5rT6fZUwhG8>

# 1. What are HTTP cookies?

## Answer:

HTTP cookies are small data files stored on a user's computer by the web browser while browsing a website. They're designed to hold modest amounts of data specific to a client and server, enabling the server to deliver a page tailored to a particular user or carry information from one visit to another.

Reference: Top 25 HTTP Cookies Interview Questions and Answers  
<https://interviewprep.org/http-cookies-interview-questions/#:~:>

## 2. How cookies work?

### Answer:

Cookies work through a process initiated when a user visits a site. The server sends a Set-Cookie header with the response containing a unique ID. This cookie is stored in the user's browser and sent back to the server every time the user revisits the site. Cookies can be used for various purposes such as maintaining sessions, remembering user preferences, tracking user behavior, and implementing shopping cart functionality.

Reference: Top 25 HTTP Cookies Interview Questions and Answers

<https://interviewprep.org/http-cookies-interview-questions/#:~:>

### 3. How does the Secure attribute in a cookie affect its transmission?

#### Answer:

The Secure attribute in a cookie ensures that the cookie is only sent over HTTPS, not HTTP. This means it's transmitted only through an encrypted connection, preventing potential interception by unauthorized parties. If a website tries to send a secure cookie via an unsecured HTTP connection, the browser will block its transmission, enhancing user data protection.

Reference: Top 25 HTTP Cookies Interview Questions and Answers

<https://interviewprep.org/http-cookies-interview-questions/#:~:>

4. Describe a situation where you would use a session cookie instead of a persistent cookie?

Answer:

A session cookie would be used in an online shopping scenario. When a user adds items to their cart, the server creates a unique session ID for that user and stores it as a session cookie on the user's browser. This allows the server to keep track of the user's shopping cart contents while they browse the site. The session cookie is deleted once the user closes their browser, ensuring that their shopping cart doesn't persist beyond their current visit. A persistent cookie wouldn't be suitable here because we don't want the shopping cart data to remain after the user has left the site or closed their browser.

Reference: Top 25 HTTP Cookies Interview Questions and Answers  
<https://interviewprep.org/http-cookies-interview-questions/#:~:>

## 5. Discuss the differences between first-party and third-party cookies.

### Answer:

First-party cookies are created by the website a user is visiting. They enable site functionality, including remembering login details and products in shopping carts. These cookies are generally considered safe as they do not track users across multiple sites.

Third-party cookies, on the other hand, are created by domains different from the one visited by the user. They're often used for online advertising and tracking purposes, enabling advertisers to target ads based on browsing history. However, due to privacy concerns, their use is controversial and being phased out by many browsers.

Reference: Top 25 HTTP Cookies Interview Questions and Answers

<https://interviewprep.org/http-cookies-interview-questions/#:~:>

## 6. How would you implement HTTP-only cookies in a web application?

### Answer:

HTTP-only cookies can be implemented in a web application by setting the `HttpOnly` attribute in the `Set-Cookie` HTTP response header. This is done server-side, typically using a back-end language like PHP or Node.js.

In PHP, you would use the `setcookie()` function and set the seventh parameter to true. For example: `setcookie('name', 'value', time() + 3600, "/", "", false, true)`.

In Node.js with Express, you'd use `res.cookie()`, passing an options object with `httpOnly` set to true. Example: `res.cookie('name', 'value', {httpOnly: true})`.

This makes the cookie inaccessible via client-side scripting languages such as JavaScript, enhancing protection against cross-site scripting (XSS) attacks.

Reference: Top 25 HTTP Cookies Interview Questions and Answers

<https://interviewprep.org/http-cookies-interview-questions/#:~:>

7. What are some security risks associated with HTTP cookies? How would you mitigate these risks?

Answer:

HTTP cookies pose several security risks. The primary risk is unauthorized access to sensitive data, often through cross-site scripting (XSS) or cross-site request forgery (CSRF). XSS attacks exploit vulnerabilities in web applications to inject malicious scripts, while CSRF tricks the victim into submitting a malicious request.

To mitigate these risks, one should implement secure and HttpOnly flags. Secure flag ensures that cookies are sent over HTTPS, preventing interception during transmission. HttpOnly flag prevents client-side scripts from accessing cookies, mitigating XSS attacks.

Another mitigation strategy is implementing same-site attribute which restricts cookies to first-party contexts, reducing the risk of CSRF attacks. Additionally, setting short expiration times for cookies can limit the window of opportunity for an attack.

Lastly, proper validation and sanitization of inputs can prevent injection of malicious scripts. Regularly updating and patching systems also helps in keeping up with new threats.

Reference: Top 25 HTTP Cookies Interview Questions and Answers

<https://interviewprep.org/http-cookies-interview-questions/#:~:>