Introduction to Computer Security

CE Bootcamp

Nader Sehatbakhsh
Department of Electrical and Computer Engineering
University of California, Los Angeles

What is computer security?

- A set of *policies* for maintaining three properties:
  - Confidentiality
  - Integrity
  - Availability

UCLA Samueli School of Engineering

Video

Security vs. Privacy vs. Safety

- Security is often about protecting data from unauthorized access.
- Privacy is about making sure that the data is either not collected in the first place or, if collected, not misused.
- Safety (also called resiliency or robustness) is about making sure that systems still work as expected...
    - But the "adversary" is mother nature rather than deliberate human action.

UCLA Samueli School of Engineering

Video

Why it matters?

Video

Why it matters?

- Our lives are increasingly dependent on computers
  - Protecting our assets
  - Controlling critical tasks
  - Productibility
  - Safety
    - Flights, cars, medical devices, etc.
  - …

UCLA Samueli School of Engineering

Video

Why it matters?

- Our lives are increasingly dependent on computers
  - Protecting our assets
  - Controlling critical tasks
  - Productibility
  - Safety
    - Flights, cars, medical devices, etc.
  - …

- *Even beyond individual users – government scale*

UCLA Samueli School of Engineering

Video

$6

Trillion

Cybercrime Cost
by 2021

11

Seconds

average time
for an attack
by 2021

$1 Trillion

Cybersecurity Spending
by 2021

NotPetya

- Attacking critical businesses in Ukraine.
  - ranging from media outlets to railway firms
- The attackers released a malicious "worm"
  - A program which self-propagates: spreads from computer to computer in an institution.
- And then disabled all the infected computers with a fake "ransomware" payload.
  - Ransomware is a program that "encrypts" the computer's hard-drive.

THE UNTOLD STORY OF NOTPETYA, THE MOST DEVASTATING CYBERATTACK IN HISTORY

Crippled ports. Paralyzed corporations. Frozen government agencies. How a single piece of code crashed the world.

BY ANDY GREENBERG

IT WAS A perfect sunny summer afternoon in Copenhagen when the world's largest shipping conglomerate began to lose its mind.

Video

UCLA Samueli School of Engineering

NotPetya

- Attackers asked for Ransom to "decrypt" each drive.

- According to the White House estimates, this attack has $10B in damage globally (mainly to Ukraine).

Video

What is in danger?

- Everything is hackable – especially if they are connected to the internet.

Video

Security: Status Quo

- For a long time we (mostly) didn't care…
  - We didn't "design for security."
  - Not much "knowledge" about security.
  - Security was a secondary objective.
  - *(What were the main objectives?)*

Video

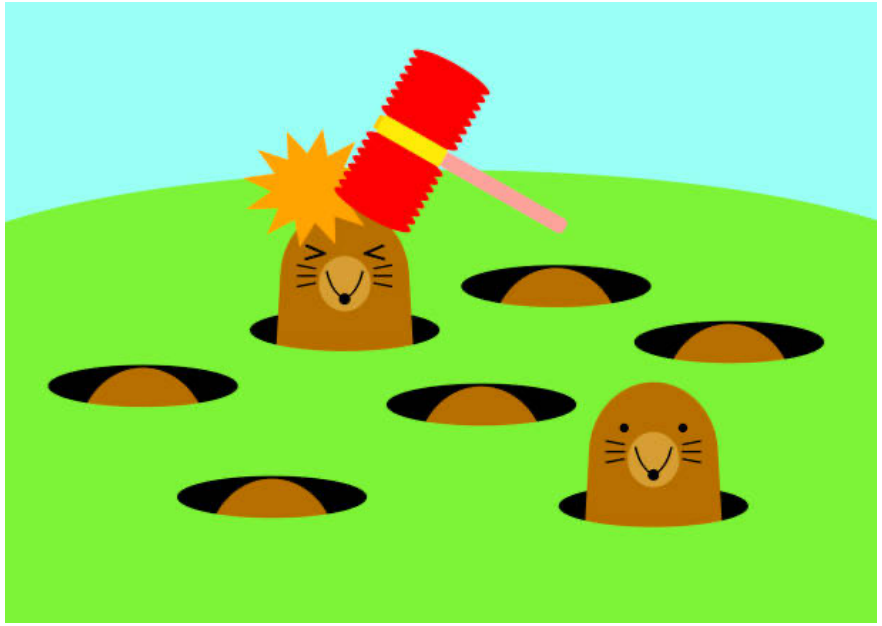
Security: Status Quo

- For a long time we (mostly) didn't care…
  - We didn't "design for security."
  - Not much "knowledge" about security.
  - Security was a secondary objective.
  - *(What were the main objectives?)*

-- We have recently realized that wasn't good enough!

Video

Current Strategy



UCLA Samueli School of Engineering

Video

Why security is critical in the next decade?

Video

UCLA Samueli School of Engineering

Why security is critical in the next decade?

1. More Devices (CPS and industry 4.0)

UCLA Samueli School of Engineering

Video

200

BILLION

BY 2020

$11.1

TRILLION

BY 2025

Video

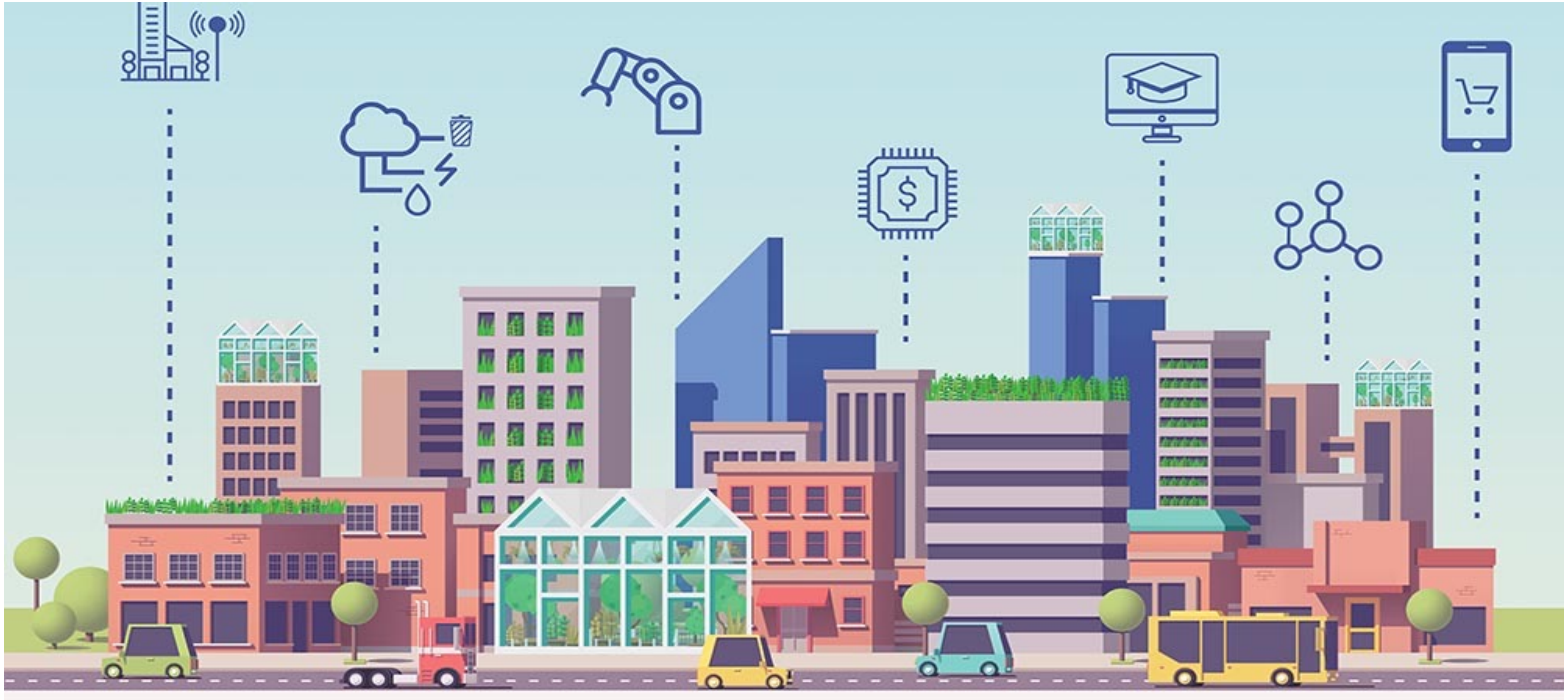
UCLA Samueli School of Engineering

Why security is critical in the next decade?

1. More Devices (CPS and industry 4.0)

2. More Security Critical Applications

Video



Computers control many *critical* tasks!

UCLA Samueli School of Engineering

Video

Different Areas in Security

- Hardware Security
- Memory safety
- Operating System Security
- Network Security
- Web/Internet Security
- Software Security
- Cryptography
- Privacy
- …

UCLA Samueli School of Engineering

Video

How do we enforce security in computers today?

Video

How do we enforce security in computers today?

1- "Principle of Least Privilege"

Users/Process should only have access to the data and resources needed to perform routine, authorized tasks.

UCLA Samueli School of Engineering

Video

How do we enforce security in computers today?

1- "Principle of Least Privilege"

Users/Process should only have access to the data and resources needed to perform routine, authorized tasks.

-- This leads to "privilege separation". Typically "user" and "root" level access.

Modern systems have multiple privilege levels.

Video

UCLA Samueli School of Engineering

How do we enforce security in computers today?

2- Isolation

A process can not access (read or write) the memory content of any other process.

Isolation is typically enforced by OS through address translation.

Video

UCLA Samueli School of Engineering

How do we enforce security in computers today?

3- Trusted Computing Base (TCB)

Trust something (e.g., hardware), build everything on top/around that.

- *Only need to verify the TCB.*
- *Keep it simple and small so it can be easily(!) verified.*

Video

Does it work?

UCLA Samueli School of Engineering

Video

Why?

- Because of design bugs mainly
  - A computer system is very complex with so many components.
  - Hard to verify everything *(known unknown)*

UCLA Samueli School of Engineering

Video

Why?

- Because of design bugs mainly
  - A computer system is very complex with so many components.
  - Hard to verify everything *(known unknown)*

  - Further, information can be leaked through additional channels called side-channel *(unknown unknown)*

Video

UCLA Samueli School of Engineering

How do computers work?

Video

How do computers work?

- Theoretical and Historical Points of View

UCLA Samueli School of Engineering

Video

How do computers work?

- Theoretical and Historical Points of View
    - Turing machines and history of electronics and computers

UCLA Samueli School of Engineering

Video

How do computers work?

- Theoretical and Historical Points of View
  - Turing machines and history of electronics and computers

Watch This:

- Theory of Computation: https://www.youtube.com/watch?v=PLVCscCY4xI
- History of Computers: https://www.youtube.com/watch?v=pBiVyEfZVUU

UCLA Samueli School of Engineering

Video

How do computers work?

- We can see a computer as a *box* that runs our programs and shows us the results (display, print, etc.)

UCLA Samueli School of Engineering

Video

How do ~~computers work~~ we see computers?

- We can see a computer as a *box* [with multiple layers] that runs our programs and shows us the results (display, print, etc.).

Video

How do ~~computers work~~ we see computers?

- We can see a computer as a *box* [with multiple layers] that runs our programs and shows us the results (display, print, etc.).

We define these layers as ***abstraction*** layers.

UCLA Samueli School of Engineering

Video

Using Abstraction

- We see/define a computer as a *box* with multiple layers of **abstraction.**
  - Depending on which layer we want to work on, we *abstract away* the irrelevant layers.

UCLA Samueli School of Engineering

Video

Using Abstraction

- We see/define a computer as a *box* with multiple layers of **abstraction.**

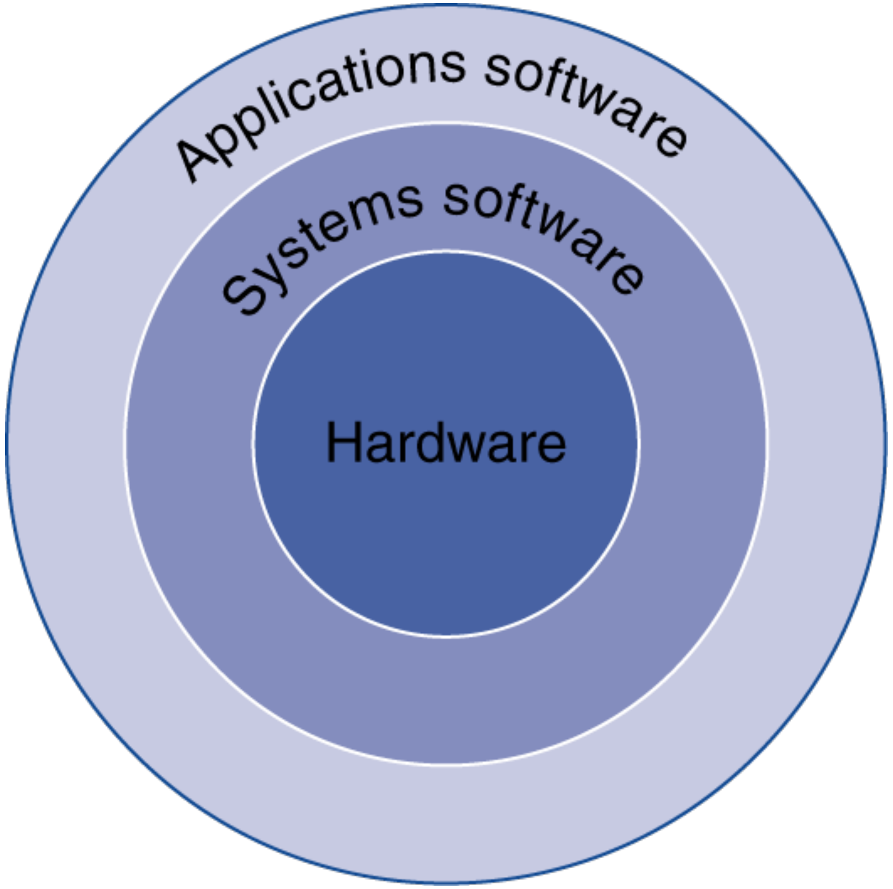
  - Depending on which layer we want to work on, we *abstract away* the irrelevant layers.

  - The ***main benefit*** is that we don't need to know the unnecessary details of the other layers in order to be able to work on our layer.

UCLA Samueli School of Engineering

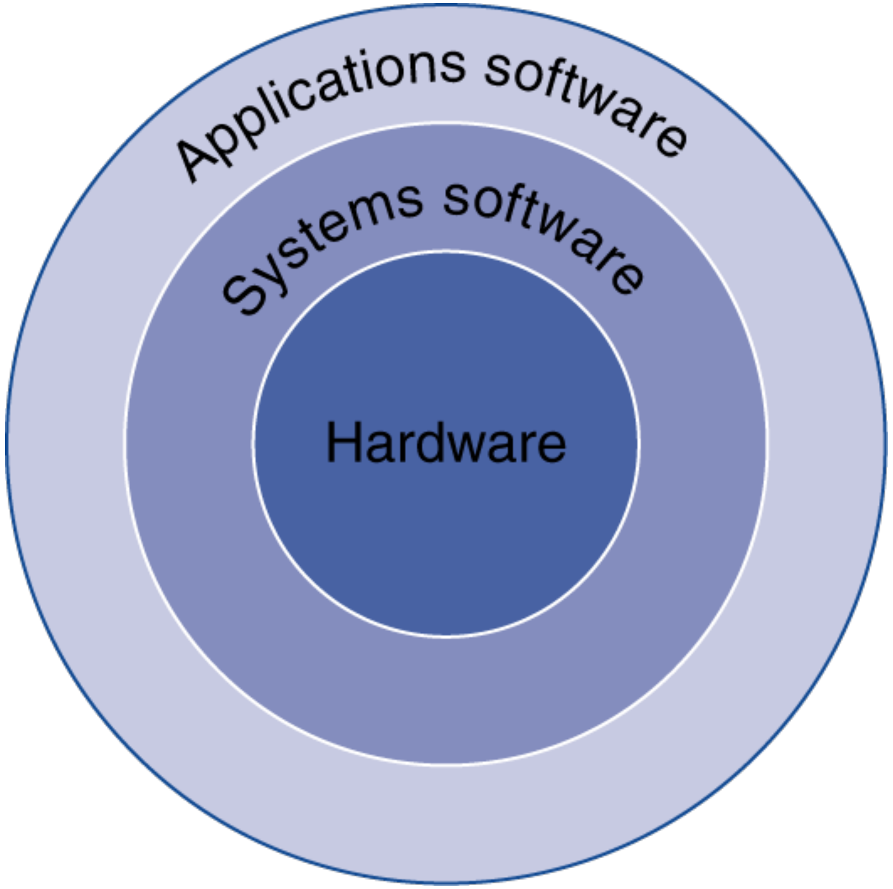
Video

Computer Abstractions



Video

Computer Abstractions



- Application software
  - Translation from *algorithm* to code
  - Written in high-level language (e.g., C, JAVA)
- System software
  - ***Compiler***: translates HLL code to machine code
  - ***Operating System***: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks & sharing resources
- Hardware
  - Processor, memory, I/O controllers

UCLA Samueli School of Engineering

Video

Computer Abstractions

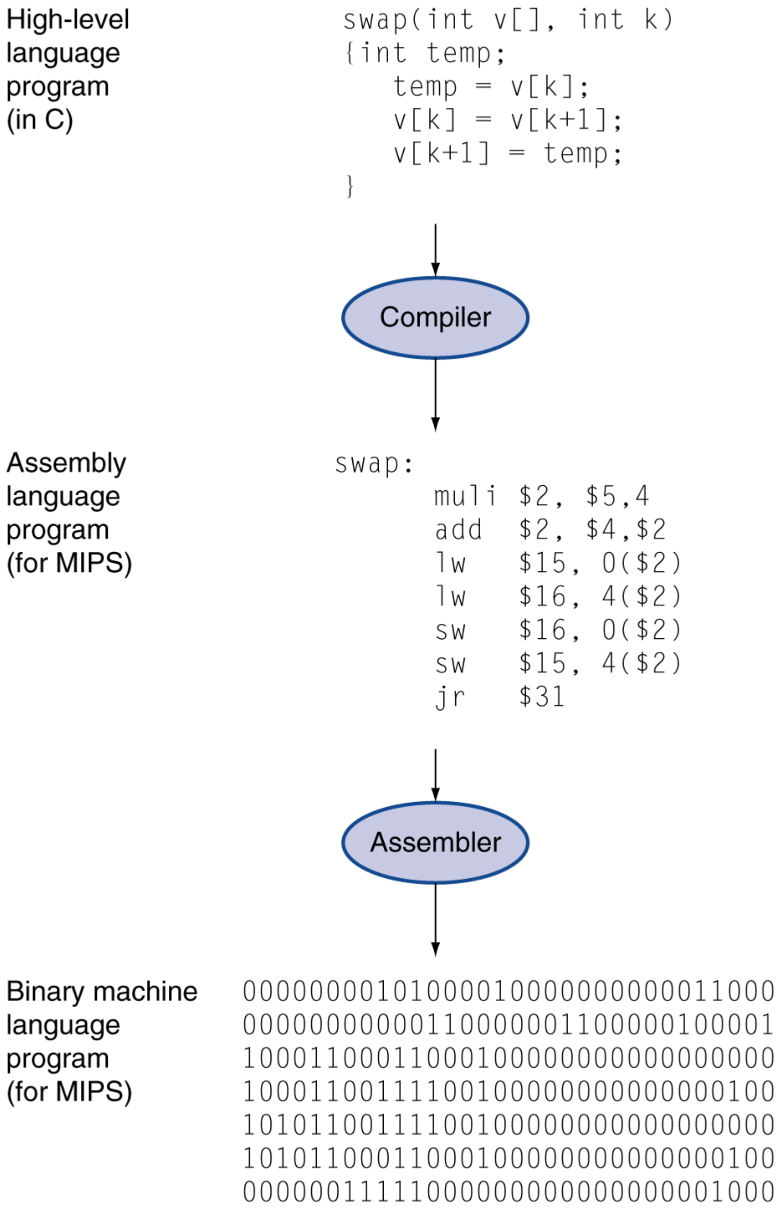
Watch This:
https://www.youtube.com/watch?v=_y-5nZAbgt4

- Application software
  - Translation from *algorithm* to code
    - Managing memory and storage
    - Scheduling tasks & sharing resources
- Hardware
  - Processor, memory, I/O controllers

Video

Level of Program Code

- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data



Video

UCLA Samueli School of Engineering

Level of Program Code

- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data

High-level language program (in C)

```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

Compiler

Assembly language program (for MIPS)

```
swap:
      muli $2, $5,4
      add  $2, $4,$2
      lw   $15, 0($2)
      lw   $16, 4($2)
      sw   $16, 0($2)
      sw   $15, 4($2)
      jr   $31
```

Assembler

Binary machine language program (for MIPS)

```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Compiler is a piece of software that translates HHL into a set of instructions based on a given hardware.

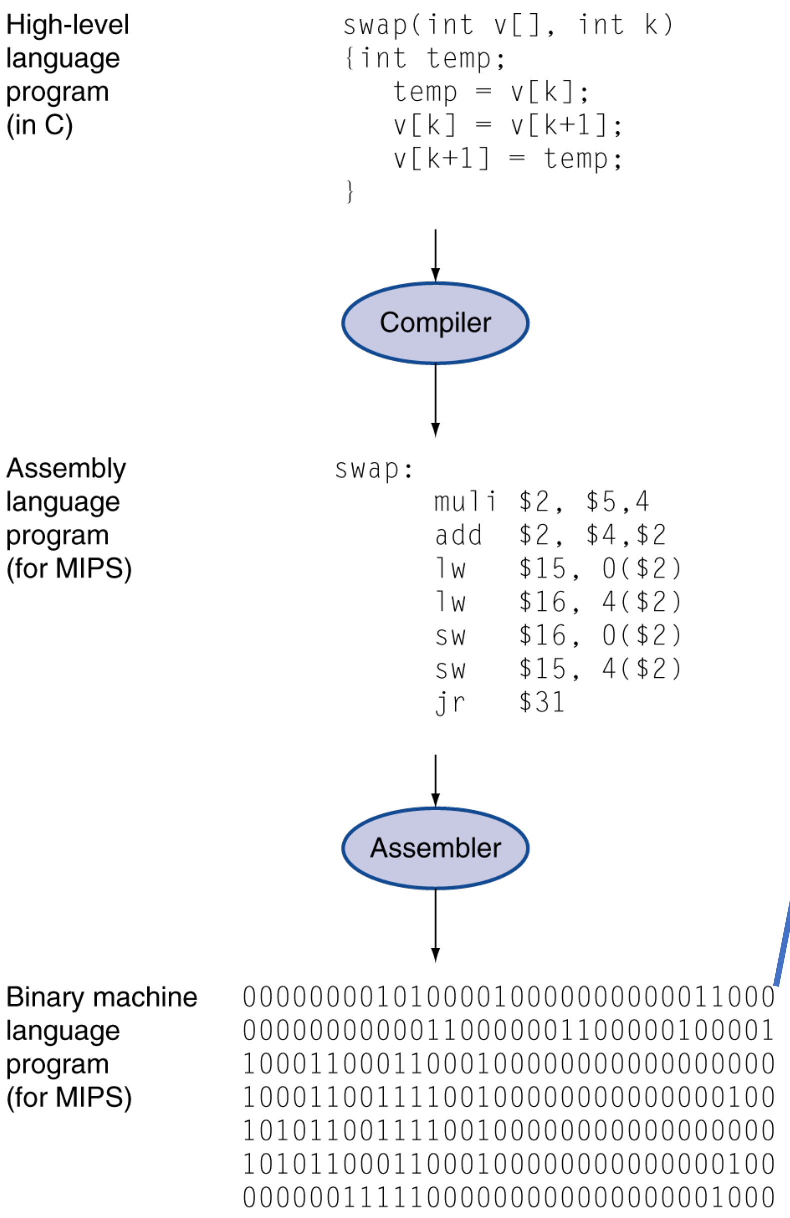
We will talk about this more!

Video

UCLA Samueli School of Engineering

Level of Program Code

- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data



Think about the processor as a *machine* that reads these bits and executes the instructions.

Video

UCLA Samueli School of Engineering

Computer Abstraction and Security

- For each layer set of rules can be enforced to ensure security
  - Hardware, operating systems, software

- Finding the interaction between layers is hard to quantify (hence new vulnerabilities).

UCLA Samueli School of Engineering

Video

Why?

- Because of design bugs mainly
  - A computer system is very complex with so many components.
  - Hard to verify everything *(known unknown)*

  - Further, information can be leaked through additional channels called side-channel *(unknown unknown)*

UCLA Samueli School of Engineering

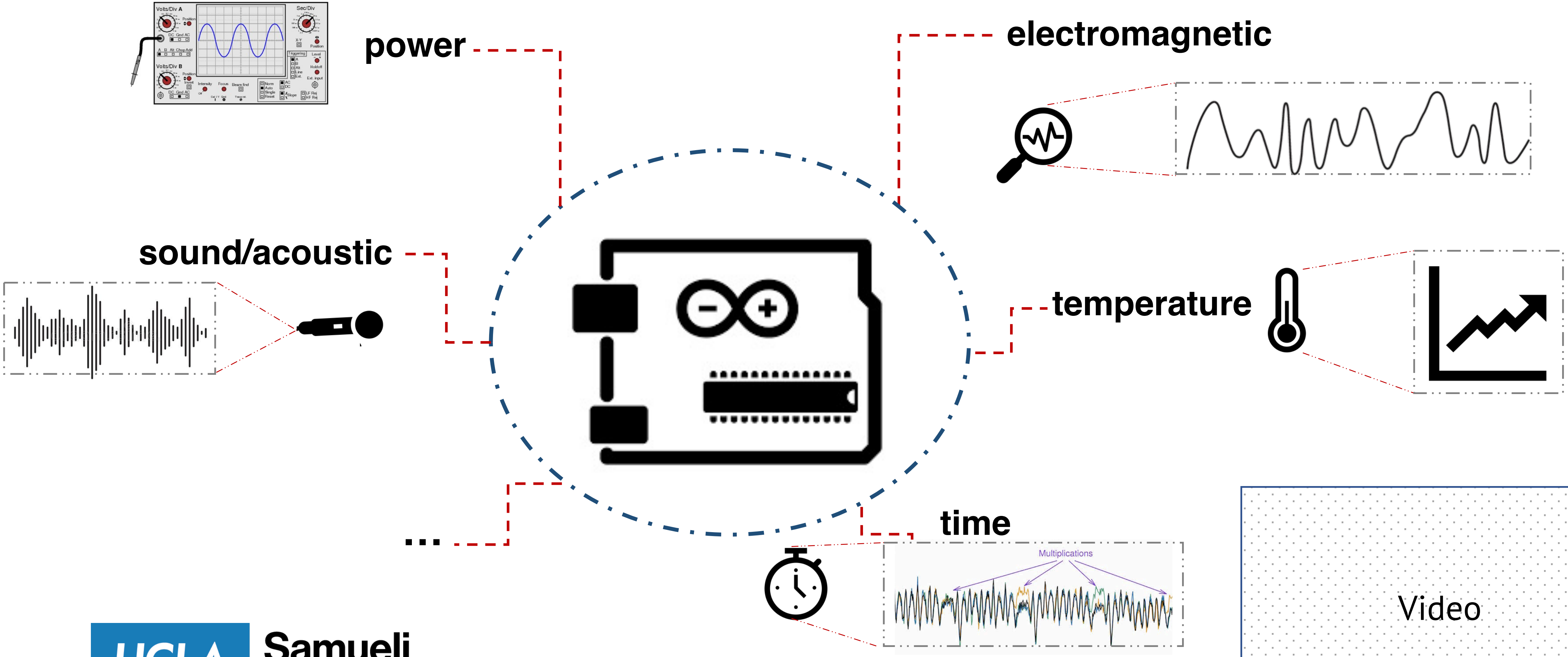
Video

So, what is “side-channels”?

Video

UCLA Samueli School of Engineering

Side-Channels

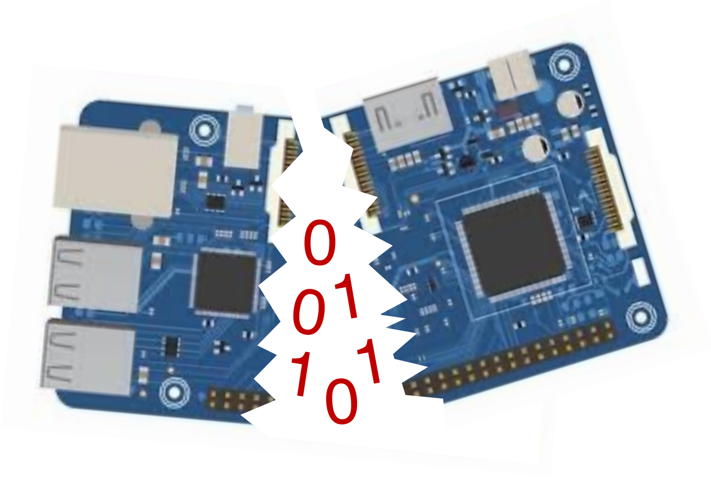


Video

UCLA Samueli
School of Engineering

Side-Channels can *leak* secrets!

An adversary can leverage the existing ***correlation*** between the side-channel signals and critical information in the application to discover the secrets.



Video

UCLA Samueli School of Engineering

Types of Side-Channels

Digital/Microarchitectural

Analog/Physical

- cache
- memory
- u-arch
- TLB

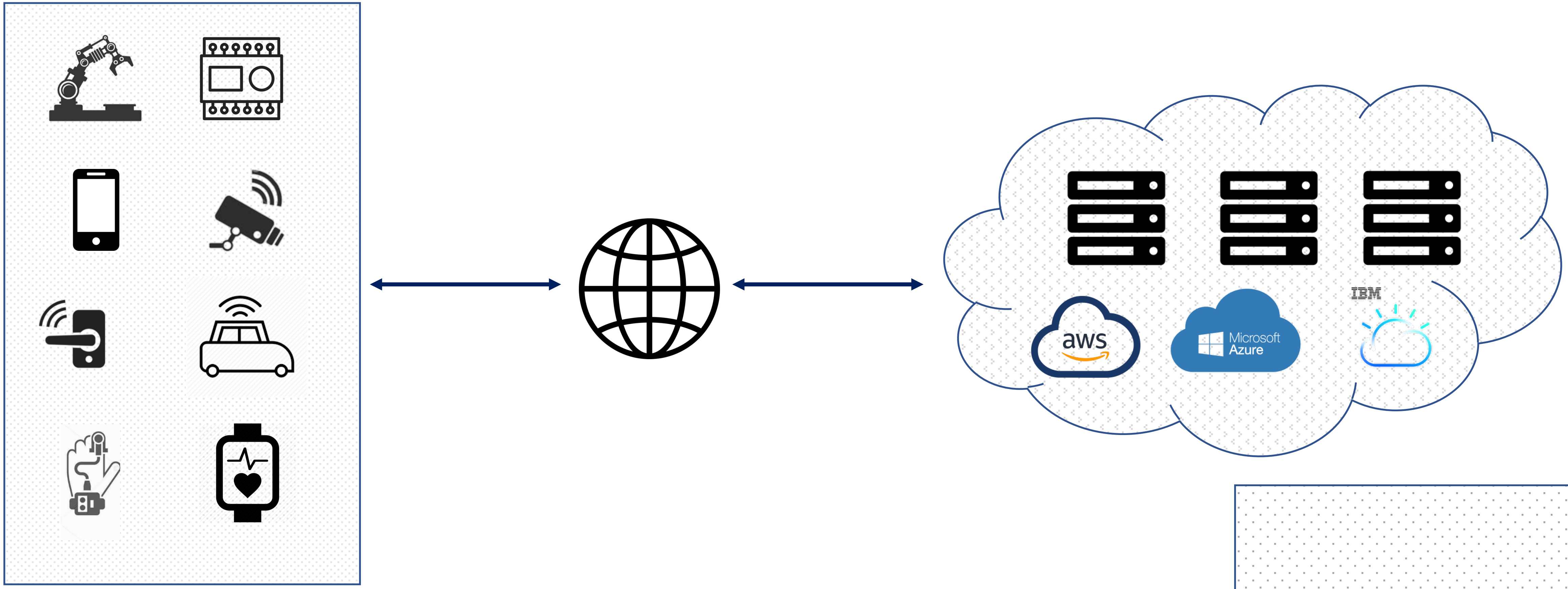
…

- electromagnetic
- power
- temperature
- sound

…

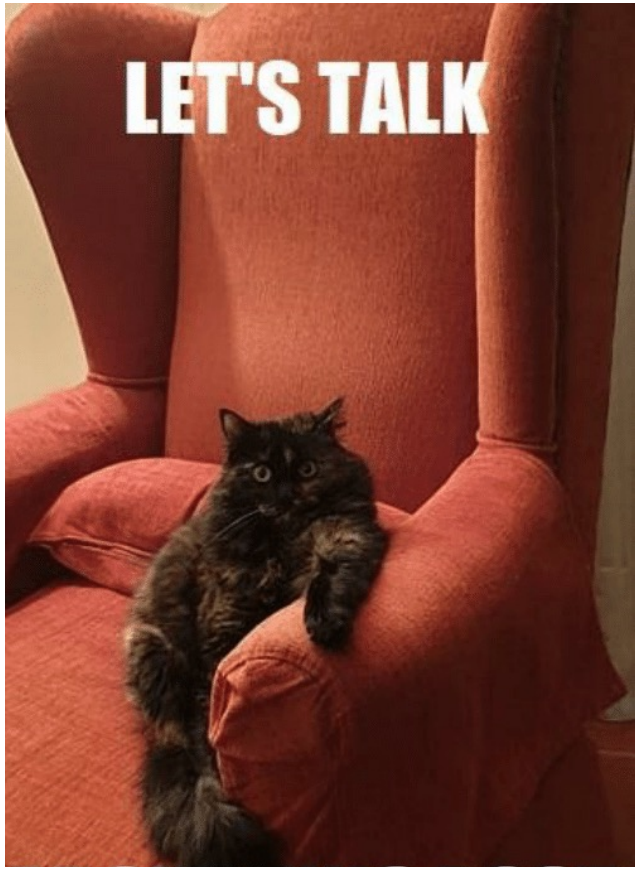
Video

Main issues



Video

Now let's talk a little bit about cryptography…



Video

Why do we need cryptographic primitives?

-- *How to ensure communication are secured?*

-- *How to trust a user?*

-- *How to make sure a file is not modified?*

…

Video

Foundations of Computer Security

- We make some *fundamental assumptions and definitions* (e.g., usually based on some hard problems).

- Using these basic assumptions/definitions, we should be able to ***mathematically*** prove the statement in question.

UCLA Samueli School of Engineering

Video

Foundations of Computer Security

- We make some *fundamental assumptions and definitions* (e.g., usually based on some hard problems).

- Using these basic assumptions/definitions, we should be able to ***mathematically*** prove the statement in question.

(if assumptions are correct → statement is correct)

- Assumptions may be proven incorrect!
- Assumptions may not be always correct!
- *Hardware implementation can change things!*
- ***Side-Channels!***
- Security can not be ad-hoc!

Video

UCLA Samueli School of Engineering

Why do we need cryptographic primitives?

- Confidentiality/Privacy
  - Only the communicating parties should know the message.

UCLA Samueli School of Engineering

Video

Why do we need cryptographic primitives?

- Confidentiality/Privacy
  - Only the communicating parties should know the message.
- Integrity
  - Guaranteeing that the message is not comprised/modified.

UCLA Samueli School of Engineering

Video

Why do we need cryptographic primitives?

- Confidentiality/Privacy
  - Only the communicating parties should know the message.
- Integrity
  - Guaranteeing that the message is not comprised/modified.
- Authentication and Trust
  - proving or showing something (e.g., identity, computation) to be true, genuine, or valid.

Video

A (very) High-Level View of a *Cryptosystem*

- For any cryptosystem we have:
  - A secret value (called key)
  - A cryptographic algorithm (e.g., encryption, MAC, Hash, etc.)
  - An input: usually called a message (code, data, etc.)
  - An output: ciphertext, hash, …

Video

A (very) High-Level View of a *Cryptosystem*

- For any cryptosystem we have:
  - A secret value (called key)
  - A cryptographic algorithm (e.g., encryption, MAC, Hash, etc.)
  - An input: usually called a message (code, data, etc.)
  - An output: ciphertext, hash, …

-- Kerckhoffs's principle / Shannon : A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.*

Video

Main Cryptographic Modes

- Symmetric Key

- Asymmetric Key

UCLA Samueli School of Engineering

Video

Main Cryptographic Modes

- Symmetric Key
  - *A value (called secret key) that is secretly shared among trusted parties.*
- Asymmetric Key

Video

UCLA Samueli School of Engineering

Main Cryptographic Modes

- Symmetric Key
  - *A value (called secret key) that is secretly shared among trusted parties.*

- Asymmetric Key
  - A pair of values: *(public key, private key)*.
  - *Public key* is unique to each user but shared to everyone publicly.
  - *Private key* is unique to every user and should be held secret.

UCLA Samueli School of Engineering

Video

Encryption/Decryption

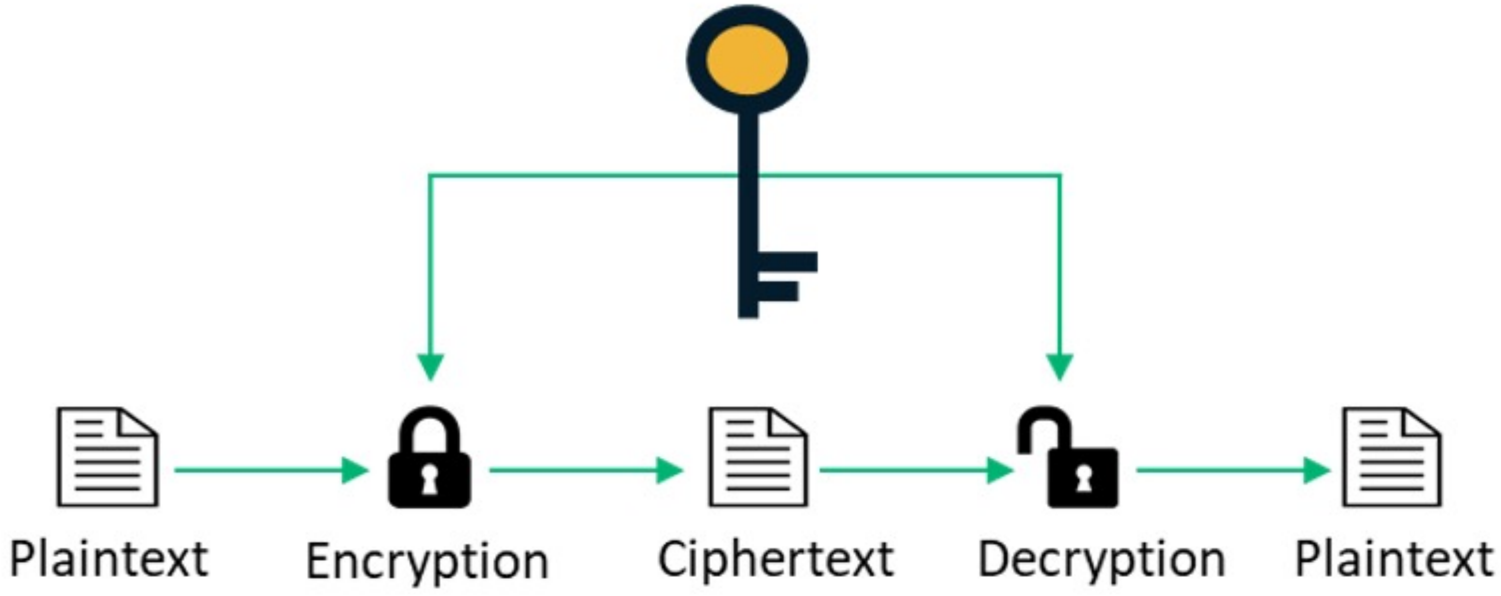


*Image was taken from: https://sectigostore.com/blog/5-differences-between-symmetric-vs-asymmetric-encryption

Correctness:

- M = Dec(Enc(M))

Video

UCLA Samueli School of Engineering

What is a secure encryption algorithm?

$$c = E(m, k)$$

$$m = D(c, k)$$

UCLA Samueli School of Engineering

Video

What is a secure encryption algorithm?

- *It should be hard to completely determine m from c, without knowledge of k.*

Video

What is a secure encryption algorithm?

- *It should be hard to completely determine m from c, without knowledge of k.*

-- Is this enough?

UCLA Samueli School of Engineering

Video

What is a secure encryption algorithm?

- *It should be hard to completely determine m from c, without knowledge of k.*

-- Is this enough?

  - Say we have only two messages. Seeing C, if there is not equal chance between: $C = E(m_1,k_1)$ and $C = E(m_2,k_2)$, *then the adversary can guess the correct answer with more than 50%.*

UCLA Samueli School of Engineering

Video

What is a secure encryption algorithm?

- *It should be hard to completely determine m from c, without knowledge of k.*

-- Is this enough?

- Say we have only two messages. Seeing C, if there is not equal chance between: $C = E(m_1,k_1)$ and $C = E(m_2,k_2)$, *then the adversary can guess the correct answer with more than 50%.*

--- We need perfect secrecy!

UCLA Samueli School of Engineering

Video

Perfect Secrecy

- If for all $m_0, m_1 \in M$, and all $c \in C$, and random variable k is *uniformly* distributed over K:

$$\Pr[E(k, m_0) = c] = \Pr[E(k, m_1) = c];$$

UCLA Samueli School of Engineering

Video

Can we achieve perfect secrecy?

- Yes!

-- One-Time Pad:

$$m \oplus k = c$$

Video

UCLA Samueli School of Engineering

Issue with One-Time-Pad?

- For perfect secrecy we need:

$$|m| = |k|$$

(key should be as large as the message!)

Video

Semantic Security

- Given two messages and their encryptions, the chance that the attacker finds out which encryption belongs to which message should be *negligible*.

Video

UCLA Samueli School of Engineering

Message Recovery Attack

- User/challenger encrypts $\hat{m}$ and sends c to the adversary. The chance that the adversary can guess the message better than $1/|M|$ should be *negligible*.

(Semantic Security guarantees this.)

UCLA Samueli School of Engineering

Video

One time vs. many time encryption

- Add this ….

Video

PRF and PRP

Video

Semantically Secure Symmetric Encryption

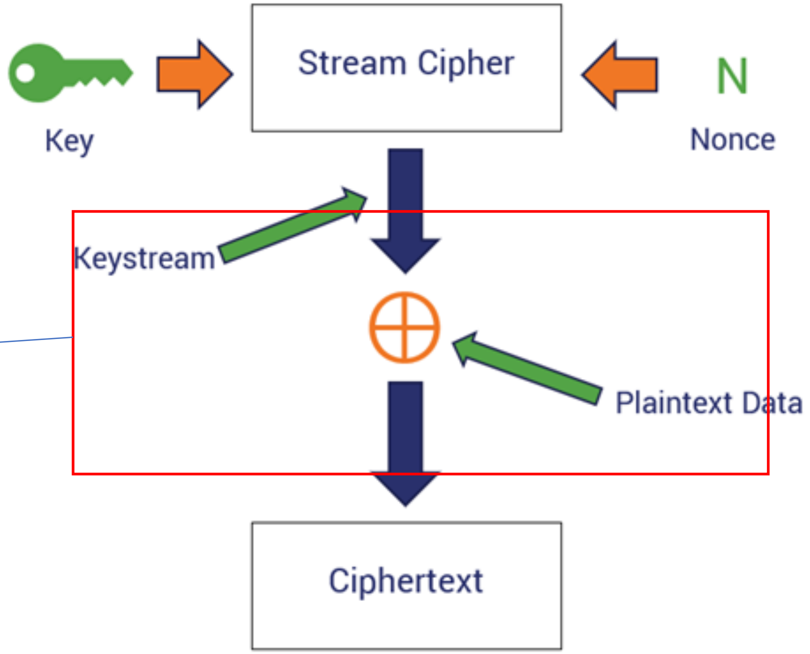
- Two Options:

  1 - Stream Ciphers

  2 - Block Ciphers

Video

Stream Cipher

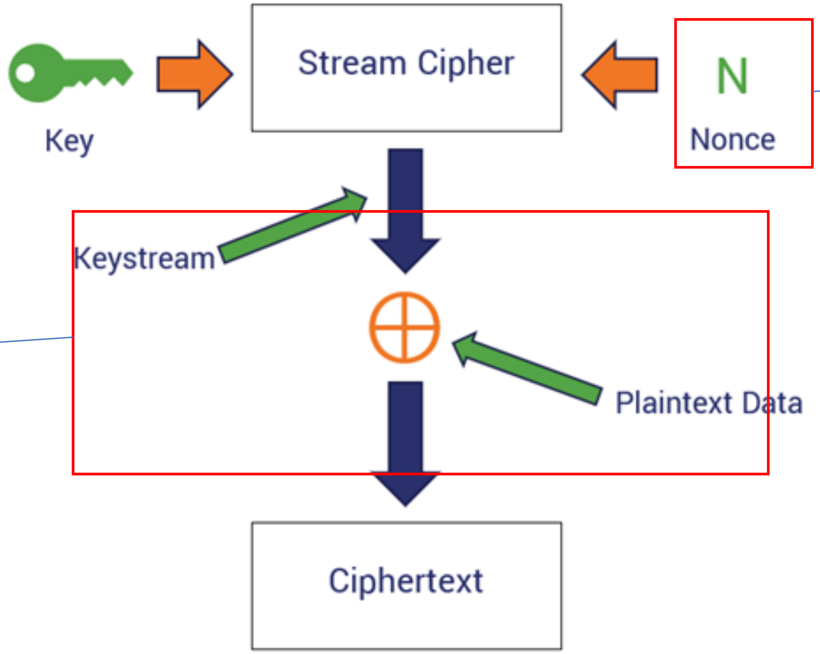


One-time pad

UCLA Samueli School of Engineering

Video

Stream Cipher

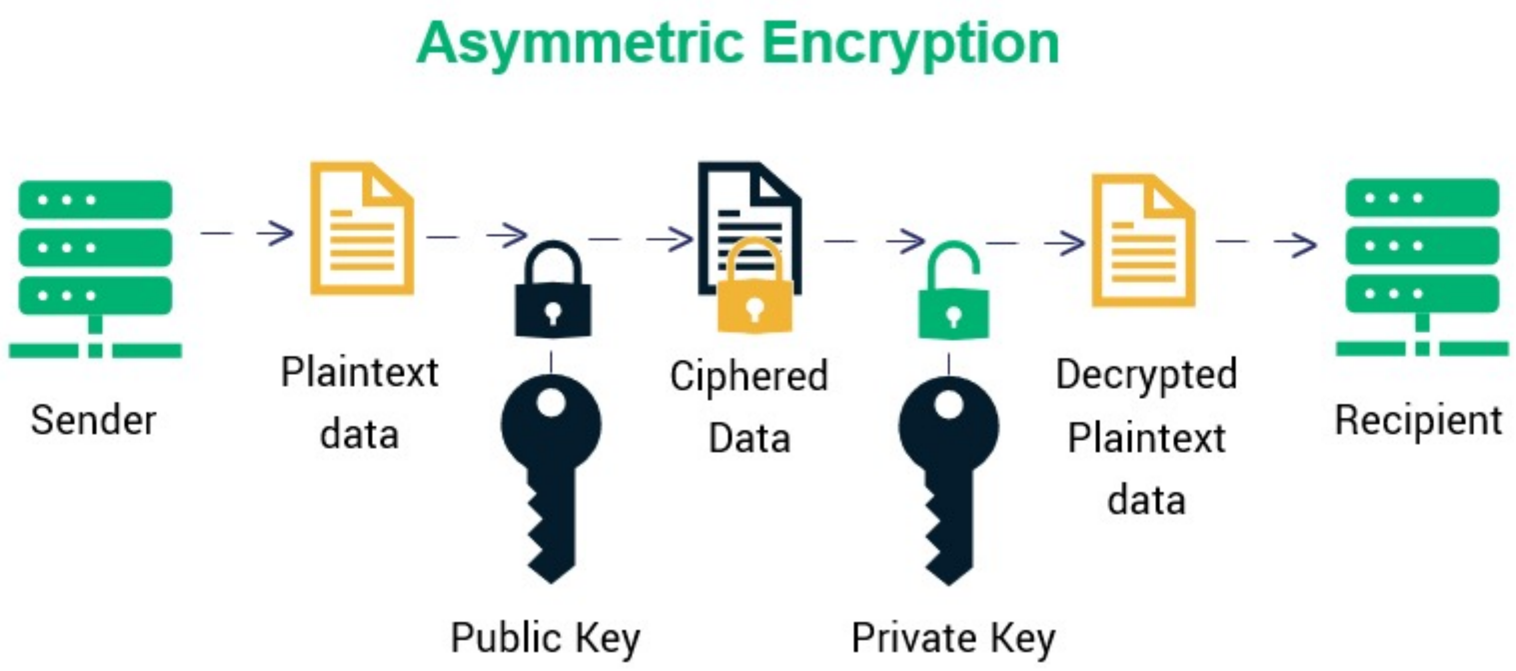


Adding randomness

One-time pad

Video

Asymmetric Encryption



different

Video

*Image was taken from: https://sectigostore.com/blog/5-differences-between-symmetric-vs-asymmetric-encryption

Differences with Symmetric Encryption

- No need to agree on the same key.
- No need to secure others' keys.

- Correctness:

$$D(sk, E(pk, m)) = m$$

- Same security guarantees.

UCLA Samueli School of Engineering

Video

Differences with Symmetric Encryption

- *Chosen Cyphertext Attack:*
  - In symmetric key cryptography, the adversary cannot send chosen messages (doesn't have access to the key to encrypt).
  - In public-key cryptography, the attacker can actively choose a desired cyphertext (the public key is known).

UCLA Samueli School of Engineering

Video

Symmetric and Asymmetric Cryptosystems

- AES
- RSA
- …

(how and why secure?)

UCLA Samueli School of Engineering

Video

Other Applications

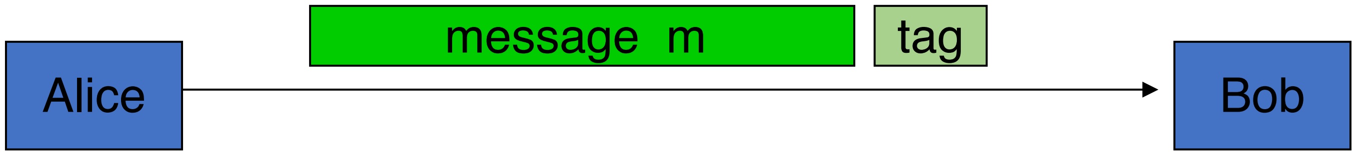
- Message Integrity

Video

Message Integrity

-- Goal:

Provide proof that the message/data is not modified.



Generate/Sign tag

G(m) = tag

Verify tag:

if (V(m) == tag)

accept …

*Image was taken from: https://crypto.stanford.edu/~dabo/courses/OnlineCrypto/

Video

Other Applications

- Message Integrity

- Hash Functions

Video

UCLA Samueli School of Engineering

Hash Function

- 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 *digests*, or simply hashes.

-- Saves space, time, computation, etc.

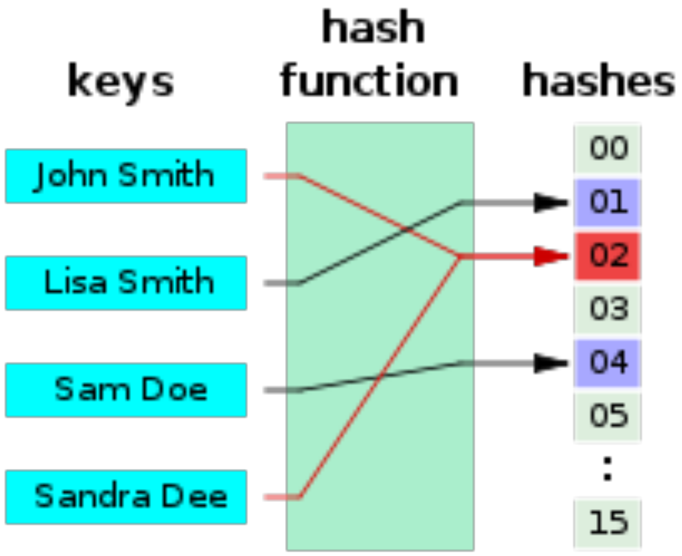


*Image was taken from Wikipedia.

Video

Secure Hash Function

-- Collision Resistance:

Let $H: M \rightarrow T$ be a hash function ($|M| >> |T|$)

A *collision* for H is a pair $m_0, m_1 \in M$ such that:

$$H(m_0) = H(m_1) \quad \text{and} \quad m_0 \neq m_1$$

A function H is *collision resistant* if for all (explicit) "eff" algs. A:

Pr[A outputs collision for H] is "neg".

Video

Secure Hash Function

-- *Collision Resistance*

-- *Pre-Image Resistance:*

Given a hash value h, it should be difficult to find any message m such that h = hash(m).

-- *Second pre-image Resistance:*

Given an input m_1, it should be difficult to find a different input m_2 such that hash(m_1) = hash(m_2)

UCLA Samueli School of Engineering

Video

Other Applications

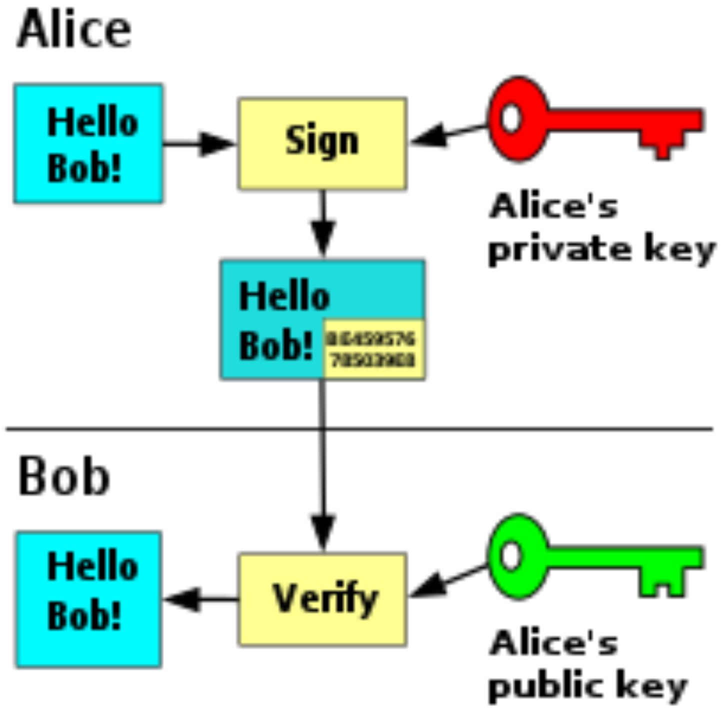
- Message Integrity
- Hash Functions
- Digital Signature

UCLA Samueli School of Engineering

Video

Digital Signature

- A valid digital signature, gives a recipient ensures:
  - The message was created by a known sender (authentication),
  - The message was not altered in transit (integrity).



Video

Recap

UCLA Samueli School of Engineering

Video

Summary

- Computer security is a very important topic in our modern era which is receiving a growing attention.

- It crosscuts various disciplines in computer science and engineering.

Video

Summary

- Security is not just about systems but also about the users.
  - Good practices
  - Following the protocols
  - Being cautions
  - …

UCLA Samueli School of Engineering

Video

What’s next?

Video

UCLA Samueli School of Engineering

End of Presentation

- Contact Info:

Email: nsehat@ucla.edu

Twitter: @SehatNader

Linkedin: Nader Sehatbakhsh

UCLA Samueli School of Engineering

Video