University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

# ECE 101: Exploring Digital Information Technologies

## Authentication and Physical Security

# A Gesture of Agreement …

# How do you do this Online?

**Would you make a deal with this "person"?**

# Online Security: Authentication to Reliably Identify an Entity

**Authentication**: the process or action of verifying the identity of a user or process.

By extension, the process to associate a computer program with a person or a company (an entity).

Example:
**Usernames** and
**Passwords**

When **authenticating** a user,

- a machine or a website compares the **password** entered by the user
- with the one that it already **knows**.

# Problems with Password Storage

- Many early systems (and, unfortunately, some to this day) kept the passwords for all of their accounts in a "**password file**" that contained the passwords in **clear text**.

- Normally, by design, the password storage **would only be accessed by an administrator** user (root, admin, superuser) and operating system utilities.

- But under **unusual circumstances**, caused by software implementation errors or deliberate misuse, the contents of the password storage file almost inevitably can become available to **adversaries**.

# Encryption

- **Encryption** is the process of encoding information.
- Encryption is part of the broader field of **Cryptography**, which is the **practice and study of techniques for secure communication** i.e. communication in the presence of an adversary.
- Today, cryptography is used as a tool for informatics, business, finance, politics, human rights—any sector that deals with **personal information** or **requires communication**.

# Encrypted Passwords

Since storing passwords in the clear has clearly proven itself to be a bad idea, one option was to **store passwords in an encrypted form** instead—store them in a coded form.

- In the 70's Unix operating systems would encrypt a block of 0 bytes with the user password and store the result in an `/etc/passwd` file (Wolfram Notebook example)
- To counter brute-force-attack, **hash value of password** could be stored, using for example **cryptographic hash functions**, such as **SHA256**

# Attacks on Encrypted Passwords

| | |
|---|---|
| 123456 | O2lURwjNgQjBwIVsL1D/uQ |
| 12345678 | FjIolUT/tpP47leYswBl9w |
| password | rzbhDUtxz53ChwzA2C+eZg |
| password1 | cmBkNwQSH2z5QbcPEYEBuQ |
| 11111 | RklJD+e/T2v3XEqUwtUyTw |
| qwerty | gaoeHZm7d9KTeLfOq3kljw |
| 123321 | /46FF5WIAbZC9ZUNy+isDg |
| ... | ... |

A **rainbow table attack** is conducted

- by choosing thousands of common passwords,
- encrypting (or hashing) them,
- storing the precomputed values in a table (dictionary),
- looking for any matches with the values stored in /etc/passwd.

Precomputed dictionary attacks have been successful because people:

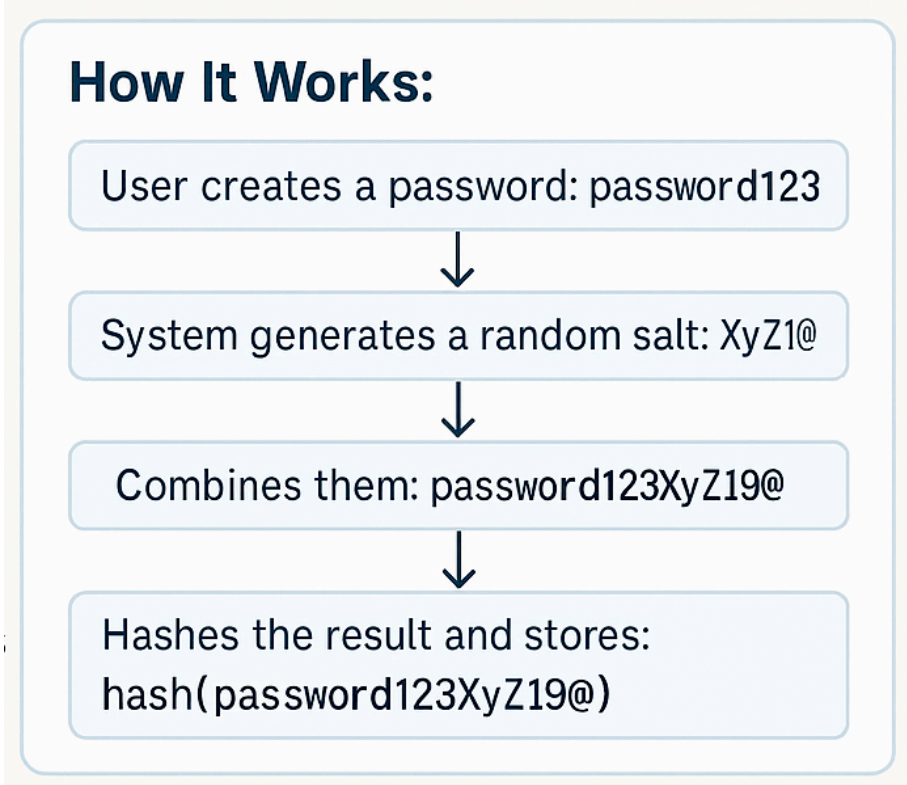
- are not good at creating and remembering long random passwords and
- they tend to use as passwords common words, phrases or dates. PwnedPasswordsTop100k.txt (you can read the archived blog post here)

# Seasoning the Passwords?

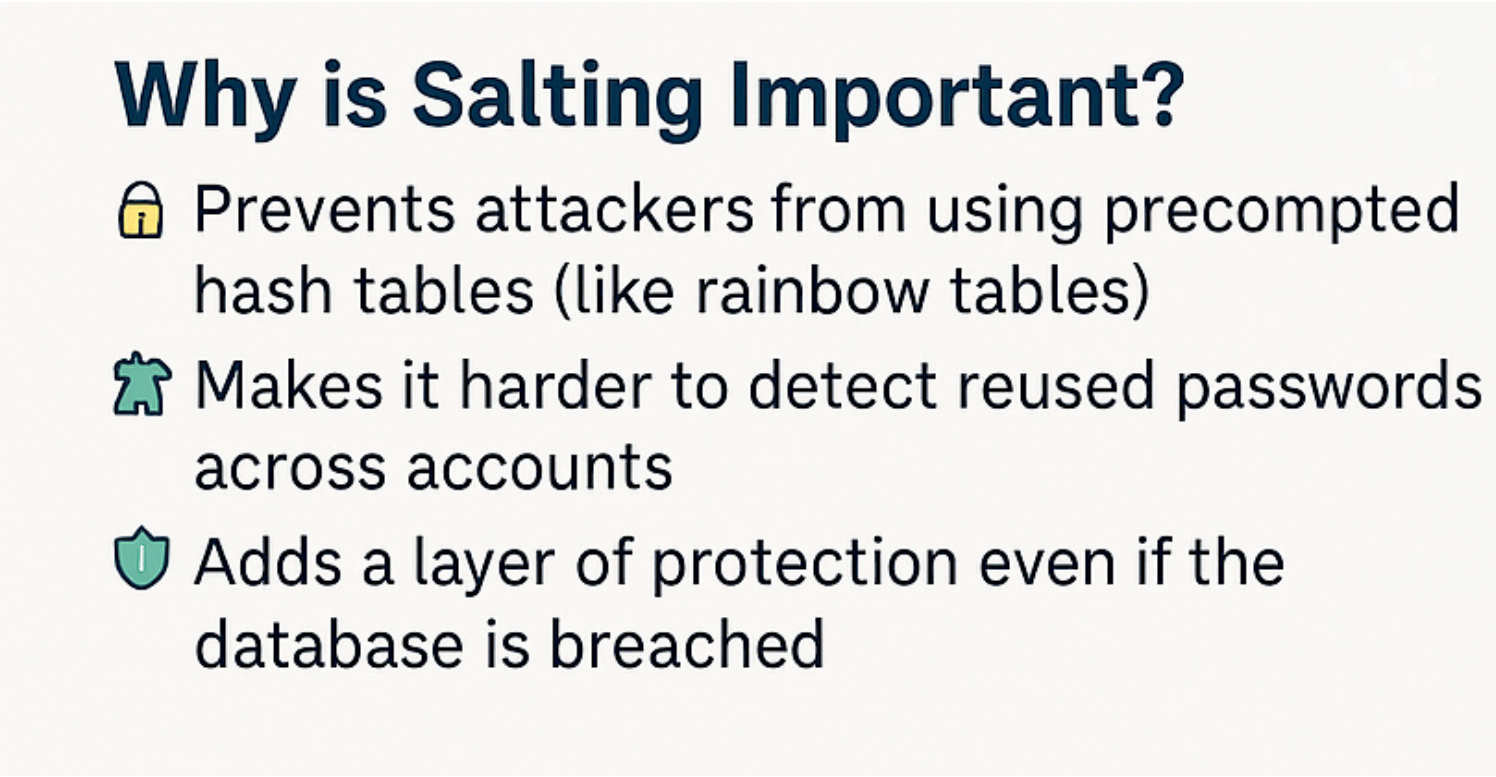
**Salt** is random data that is used alongside the password as additional input to a password storage safeguard algorithm.

A new salt is randomly generated for each password,

- so even if the **passwords** used were the **same**,
- the resulting **hash** stored would be different.



# Seasoning the Passwords?



# Two-Factor Authentication (2FA)

**What is it?**

Two-Factor Authentication adds an extra layer of security to your online accounts by requiring **two forms of verification**:

1. **Something you know** – like a password
2. **Something you have** – like a phone or security key

**Why use 2FA?**

🔐 Protects against stolen passwords

📵 Prevents unauthorized access

💬 Alerts you to suspicious login attempts

# Two-Factor Authentication (2FA)

**Common 2FA Methods:**

- **SMS codes** sent to your phone
- **Authenticator apps** (e.g., Google Authenticator, Authy)
- **Hardware tokens** (e.g., YubiKey)
- **Biometrics** (e.g., fingerprint, face recognition)

**Best Practices:**

✅ Enable 2FA on all important accounts (email, banking, school portals)
✅ Use an authenticator app instead of SMS when possible
✅ Keep backup codes in a safe place

# Hash Functions

**Hash functions** take an arbitrary long, but finite, input and produce a fixed-size output based on that input.

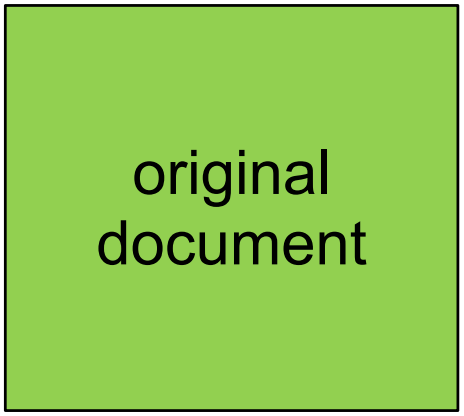
Applying a hash function to the input data creates a **digest** that contains (check notebook)

- hash value,
- hash code,
- hash checksum

You check against the digest to see if the input has been **accidentally** or **intentionally** changed.

# Application: Large Documents Easier to Distribute through Others

Imagine that I've
- written an important document (or a large program) and
- want to make it publicly available,
- but don't want to pay for bandwidth for downloads.



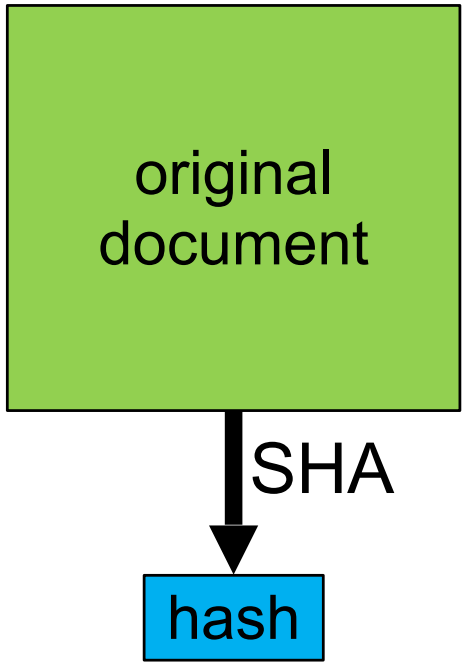

Instead, I **allow other people to provide copies** of the document.

# Use a SHA Hash to Check the Document's Authenticity

**How can you tell whether someone has tampered with the copy you obtain?**

To allow you to check,

- I **compute a SHA** value (a **hash**)
- for my document (say **64 Bytes**—small)
- and **give the hash to everyone directly**.




**Common Hashing Algorithms:**

- **SHA-256** (Secure Hash Algorithm)
- **MD5** (older, less secure)
- **BLAKE2** (modern, fast, secure)

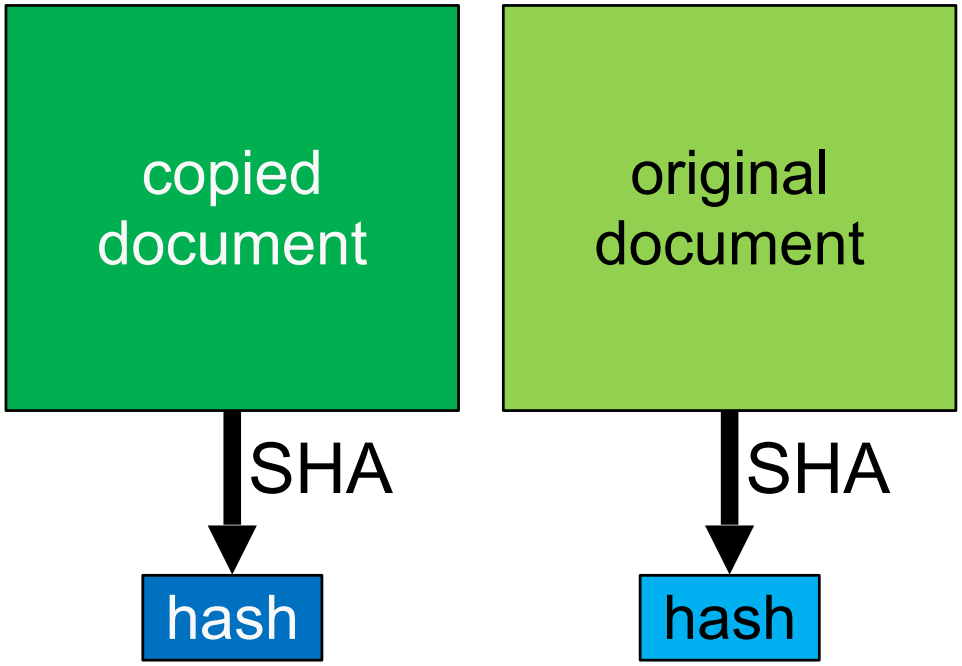
# Matched Hashes Imply Authentic Documents

After downloading a copy,
**you compute a** SHA **hash**.

**If** the **hashes match**, either:

1. **your copy is accurate**,
2. or someone produced a different document with the same SHA hash.

The second case is assumed to be impossible.




Ref: Notebook

# Hash Space is Effectively Infinite

**Why is it so hard to guess?**

If hash bits are random,

- the **chance** that a forged document
- **has the same** 64-Byte (512-bit) **hash**
- is 1 in $2^{512}$, or about **1 in $10^{155}$**.

**One-way hash functions**

- are thus **considered unbreakable**
- … **for now**.

only $10^{80}$ particles in the observable universe

# An Indispensable Tool of Cryptography

Cryptographic hash functions are an indispensable tool of cryptography, as they are:

- efficient to compute for any finite input
- same input to the same hash function will always produce same output
- hard to “trick”; a slight change in the input results in a drastically different output
- hard to guess or work back to an input that will produce a given output

# Encryption beyond Passwords

There is always someone malicious to listen in on your conversation …

- Modern communications, especially the Internet, operate under the assumption that the world is hostile and for anything you say there is always someone malicious to listen
- Same reason why people would put handwritten letters in an envelope before sending, but scaled for billions of people and devices.
- Cryptography, in turn, is one of the major instruments in the arsenal of information security, a digital protective envelope for communications.

# Secure Communication

Today people are used to most of their connections to the web being secure, but that was not always the case.

- HTTP (Hypertext Transfer Protocol), the foundation of data communication over the World Wide Web (WWW), is plaintext.
- All of the data in HTTP requests and responses is sent in the clear, under a risk of all sorts of intrusions and fraud.

Providing security of communication over computer networks is the continuing challenge.

# How can Cryptography Help?

Cryptography has grown to be not only about encryption anymore, but includes a group of special-purpose algorithms to sustain the wider infrastructure of information security, such as:

- user and message authentication,
- protection from illegitimate changes to messages,
- **protection from eavesdropping**, etc.

# Ciphers

**Ciphers** operate on a lower level of message structure—letters historically, bits nowadays—and utilize some defined set of mathematical operations.

A **key** is used to encrypt the message text.

Encryption produces an illegible sequence of letters (or bits)—called a **ciphertext**.

# Caesar Cipher

What does this mean?

khoor **hello**

How about the following?

qlfh wr phhw brx

**nice to meet you**

Try some examples

| | | | |
|---|---|---|---|
| a | + 3 | → | d |
| b | + 3 | → | e |
| c | + 3 | → | f |
| d | + 3 | → | g |
| e | + 3 | → | h |
| . | . | . | . |
| x | + 3 | → | a |
| y | + 3 | → | b |
| z | + 3 | → | c |

KEY

**h e l l o**

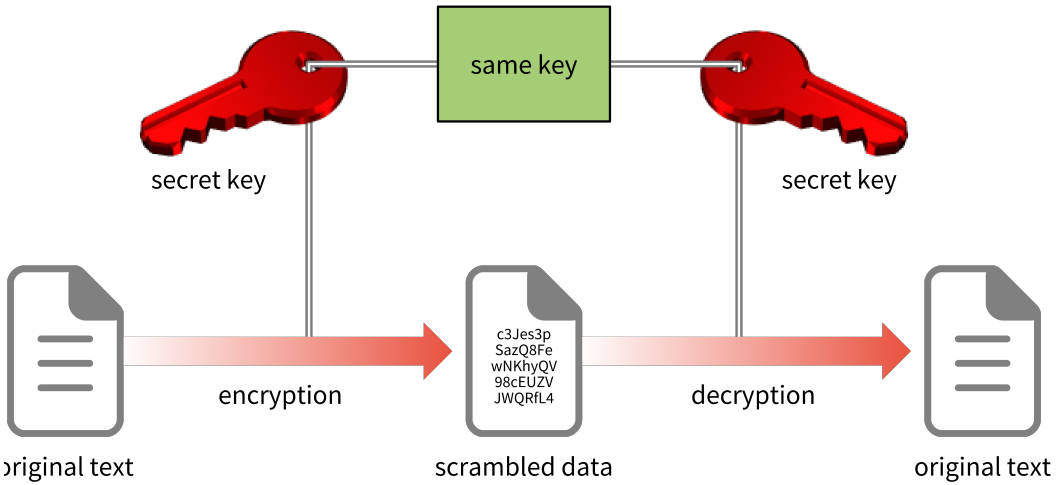
Encrypt

**k h o o r**

Decrypt

**h e l l o**

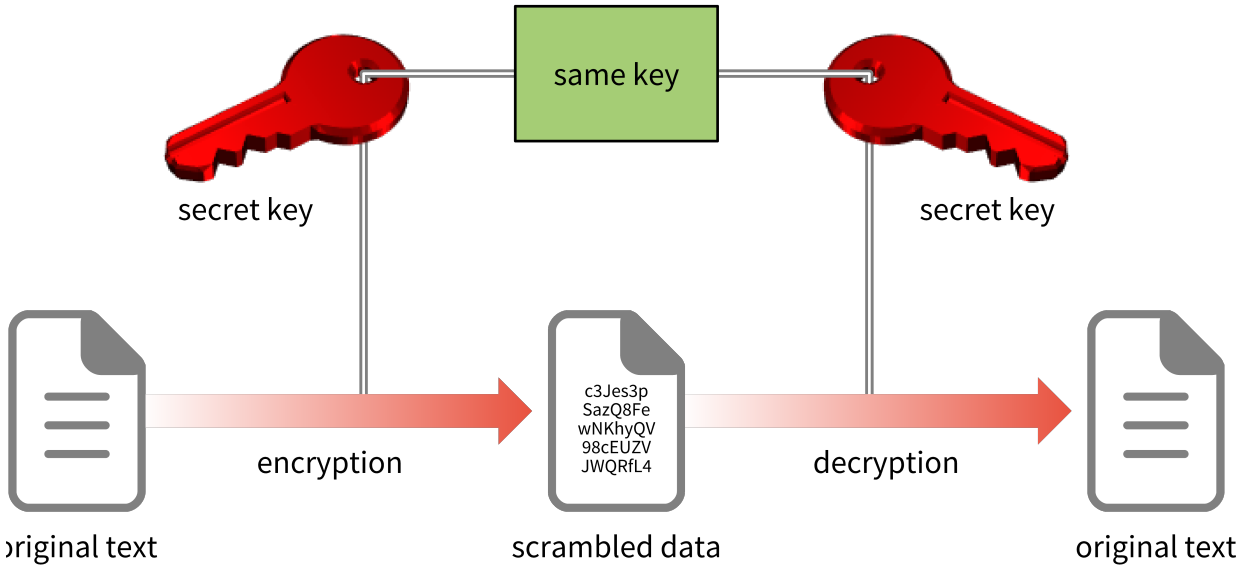
# Secret Key Cryptography




Try some examples

- Two parties agree on the **cipher** and the **key** to encrypt their future messages
- Alice uses the key to encrypt the plaintext and sends the ciphertext to Bob
- Bob has the same key and uses it to decrypt the message
- A single key is used to both encrypt and decrypt
- Also called **symmetric key cryptography**

# Pros and Cons



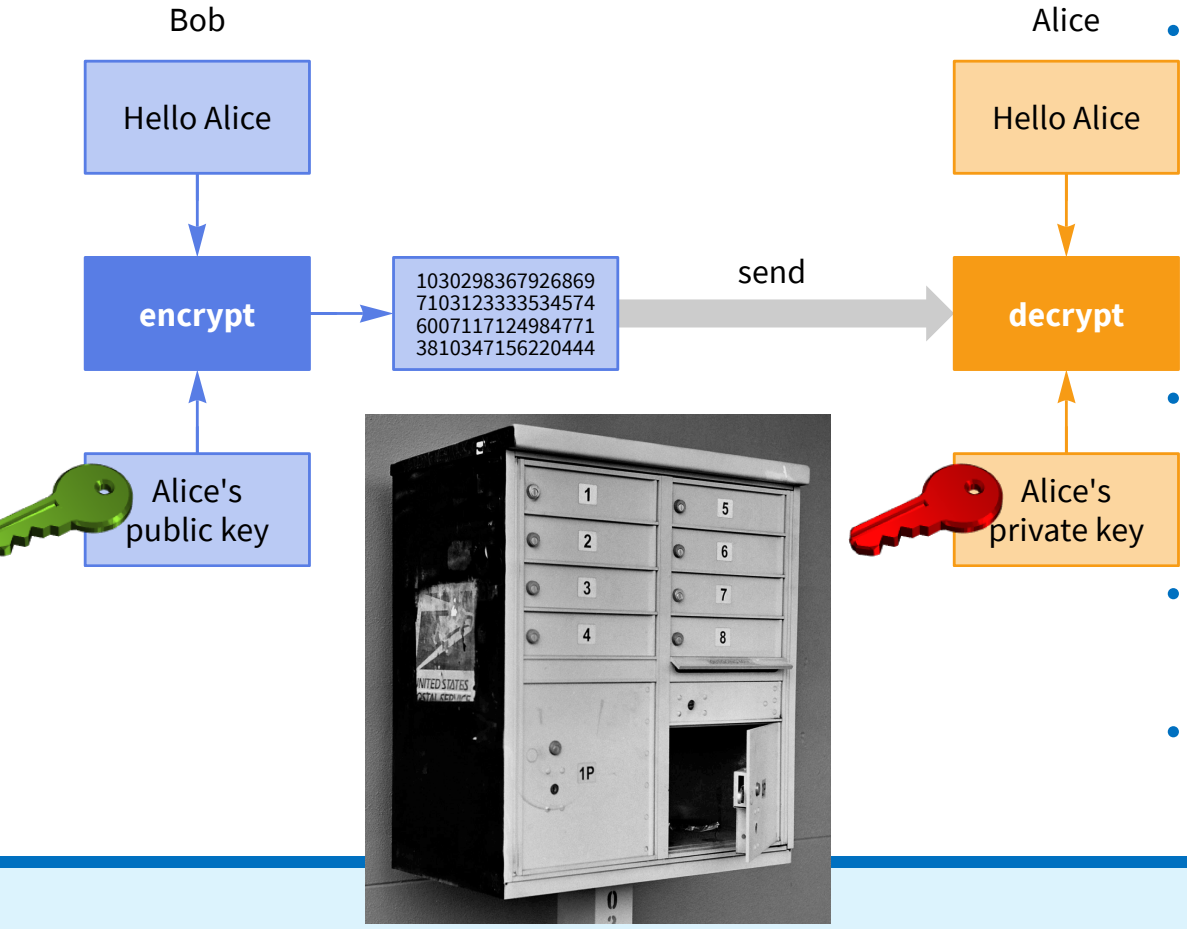

## Pros

- they can encrypt arbitrary big data
- they are very efficient at it

## Cons

- the exchange of the key

# Public Key Cryptography




- A public key, which, as its name suggests, does not need to be kept protected and may be shared with other parties over any open channel or even posted freely on your website.
- Its corresponding private key—the secret that has to be kept hidden
- One key is used to encrypt and the other to decrypt
- Also called **Asymmetric key cryptography**

Try some examples

# Cryptographic Objectives

What is one trying to achieve by employing cryptographic algorithms in information security?

- **Confidentiality**—obligation to keep the information secret from all eyes except those of the owner or intended receiver
- **Authentication**—communication over a network does not happen in physical proximity; parties do not see or talk to one another in person, so they need to be confident of each other's identity and the data exchanged.
  - **Entity authentication**—mutual identification of parties in a communication
  - **Data origin authentication**—binding the information (message) to its source, author, date of origin, data content, time sent, etc.

# Cryptographic Objectives

- **Data integrity**—ensuring prevention and detection of the unauthorized alteration of data
- **Non-repudiation**—capability to determine whether a party has performed a certain action, such as:
  - creating information,
  - digitally signing information,
  - sending a message or transaction,
  - approving information or receiving a message,

# How Do We Know that Keys are Valid?

**Cryptography is pretty hard to break.**

Fooling the humans who use it … not so hard.

**How do your browser and a server**

- **know that no one in the middle of the Internet**
- **is "helping" to agree on a key?**

**How do you know that the "public key for Abrita" is really mine?**

**Trust has to start somewhere.**

For most people, it's not with the government.

And it's not with most companies.

# Certification Authorities Provide the Root of Trust

Instead, we create
- trust or **certification authorities (CAs)**,
- organizations independent enough of both government and corporate influence
- that they can be trusted.

For example, ssl.com,
- a group named for the Secure Socket Layer
- used between browsers and web servers.

# Role of Businesses Built on CAs

Companies such as Verisign can also play other roles:

- Help companies use computing to replace signatures.
- Serve as a witness or notary for a signature on an agreement between authenticated users.
- Ensure that loss of a single secret key doesn't allow an entity to back out of their signature.

# Beyond the Algorithms: Security needs Planning

- Security of communications **cannot be achieved solely by developing mathematical algorithms and protocols**.
- Strong cryptography is necessary for secure communications, but not sufficient. Reliable security of information goes beyond cryptography alone.
- It also requires **carefully planned procedures, operation and establishment of laws**.

# Beyond the Algorithms: The Tool is as Good as How you Use It

- Cryptography is a powerful tool that needs to be utilized properly.
- A tough bank vault will not protect the gold inside if the lock combination is written on a sticky note next to it. Like seat belts, cryptography will not completely protect us, but is indispensable nevertheless.
- **Many systems fail because they were designed to protect the wrong things—or the right things, but in the wrong way**.

# Terminology You Should Know from These Slides

- Encryption
- Salt
- 2FA
- authentication
- hash function
- Secure Hash Algorithm (SHA)
- cipher
- cryptographic key
- symmetric-key cryptography
- asymmetric-key cryptography
- certification/certificate authorities (CAs)

# Physical Security

# Humans: When Do You Need Help?

**When these images appear on your home camera, do you call the police?**



**Never?**

**That's not my cat!  Always?**

# Security: Determining Whether Our Property is in Danger

**(physical) security:**
How can I **decide whether my home** (for example) **is in danger?**

# Role of Computing in Security

- improves and coordinates sensors
- learns habits and preferences
- mimics human presence
- integrates with personal computing
- preserves data

# The Security Company Business Model

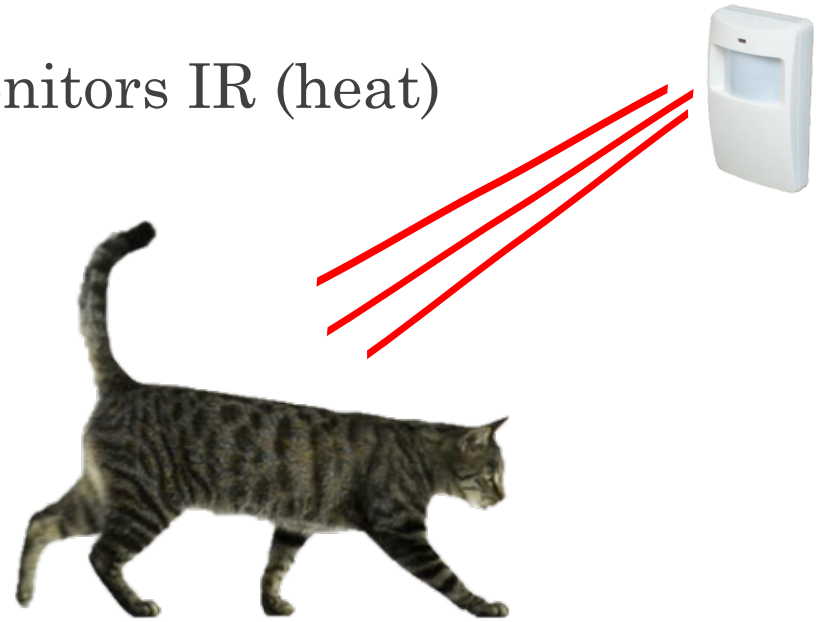
In return for purchase and installation costs and a monthly service fee, **home security firms offered**…

- an **array of sensors** to detect intrusion,
- **centralized control** with a PIN to turn the alarm system on and off,
- **remote 24/7 human oversight** to minimize false positives
- **automatic notification** of both emergency services (police/fire) as well as the homeowner

They could also help with timer-based **control of lights and sounds** within the home to trick an observer into thinking that someone was in the home.

# Historical Motion Sensing

Passive sensor monitors IR (heat) in field of view.



Rapid changes register as “motion.”

# Historical Sensing of Perimeter State

When doors are closed, electrical current flows across connection made between edges of doors (red line).

When a door opens, the circuit breaks (black line), detecting the change.

# Historical Sensing of Broken Glass

A similar approach, involving a thin strip of foil, was used to detect broken windows.

When a window is broken, the foil tears, breaking the circuit.

# Historical Signage to Discourage Intrusion

Companies provided signs to warn potential intruders that a building was under 24/7 surveillance.

Generally, they didn't explain why such tactics can be effective.

# Sensing and Processing Have Advanced Dramatically

**How does ubiquitous computing change this business?**

Let's start with sensors.

Semiconductor optics technology has enabled **high-resolution, low-power cameras**.

Powerful **image processing**

- can now be applied to sensor output
- in conjunction with other sensor data (called **sensor fusion**).

# Audio and Other Sensing Technologies Also Useful

**Audio processing**

- can also be **much more sophisticated**,
- thanks to improvements in computation power (per dollar).
- Now we **can "hear" glass breaking** instead of checking every pane in the house.

Other **sensing media**

- useful for **identify**ing and differentiating **human occupants and pets** from intruders,
- including radio, IR, and visible light.

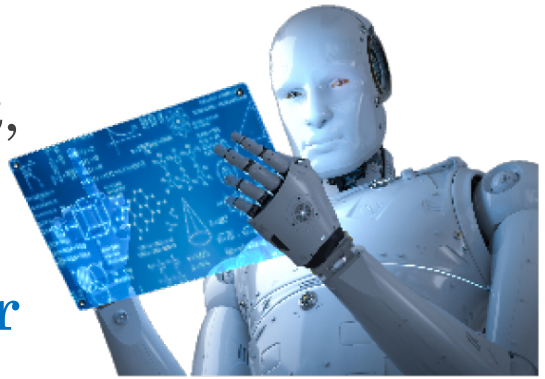
Not my cat!


(I don't own a cat.)

# Many More Advances as Well

Computation has also enabled other improvements:

- **automated mapping** of home environment,
- **understanding habits** and personal preferences
- control systems to **mimic normal behavior**
- **integration with personal computing** (wifi, mobile phones)
- **preservation of data** (protected in tamperproof storage and in the cloud)



# But Do We Still Need Security Companies?

But **detection technologies**—
- even whole "home security" packages—
- **are readily available in stores**.

Given the ubiquity
- of sensors, wifi, mobile phones,
- why do companies like ADT still exist?


**Can't you just put a system together yourself?**

**Don't many people already do so?**

# Evolution of the Sales Pitch

A home security system
- also monitors your home
- when you're at home.

**What if someone breaks in and attacks you?**

**What if a stray / wild animal gets in?**

# Not Everyone Wants to Do It Themselves

Being able
- to **differentiate** these situations
- **from your throwing a party** may be difficult.

**Security companies**
- **have the most data**
- as well as humans online 24/7
- to make the right call.

That may be nicer than having your security system call the cops to bust up your party…

# Evolution of the Sales Pitch

**So the pitch is still relevant.**

**These companies provide**

- **expert integrators**—avoid human error in integrating devices and software, and
- **expert overseers**—humans (or ML systems, or both) monitor actual events before reporting, without involving the homeowner.

**And** of course there's still the **signage**…

# Terminology

- Physical security
- Sensors: motion, video, audio
- Sensor fusion

# Concepts You Should Know from These Slides

- roles of computing in security
- timer-based **control of lights and sounds**
- How computing can incorporate data from multiple sensors
- how computing enhances detection of security events and differentiates the unusual from the usual
- role of experts: integration and oversight