IEMS5710 Cryptography, Info. Security & Privacy

Sherman Chow
Chinese University of Hong Kong
2nd Trimester, 2024-25
Lecture 1: Overview & Sampling

Rundown

- 0. Prologue:
  - Contacts and Platforms
  - (Science of) Cryptography
  - Motivation
- I. Security Goals
  - Modern Applications
  - Cryptographic Primitives
- II. Overview/Sampling
- III. What is information security?
- IV. Unique Aspects of Security
  - Adversarial Thinking
  - Interdisciplinary Nature
  - Security vs. Usability
  - Long-term Thinking
  - Precision and Formality
  - Threat Model
- V. How cryptography helps information security

What is Cryptography?

- From Greek: “kryptos” (secret) and “grapho” (writing)
- Originally, the “art” of “secret writing”
- You don’t know how to read
  - exemplified by historical methods like the Caesar cipher
  - often lacked a systematic scientific foundation
- You don’t know how to write
- Control access (learning & influencing) to “information”
- So, only cipher/encryption and (digital) signature?
- Much more!

Cryptography as Science

- Modern cryptography, in contrast, is based on *mathematical* principles and rigorously tested algorithms, making it a scientific discipline.
  - involves the systematic study and development of techniques to protect ***information*** and ensure ***secure communication***.

- Evolving with practical applications in the digital age

Why study Cryptography?

- Data, or information, is always under transmission
- Internet/cloud storage
  - Outsourcing storage -> Outsourcing computation
- ~3 billion Facebook users
  - was 500 million when I drafted this slide
- ~5 billion Internet users
  - was 2 billion a decade ago
- Everyone's data is digitalized!
  - personal info., credit card, health record, *etc*.

Data Confidentiality

- Many massive security breaches
- *E.g.*, PlayStations got hacked (April 2011)
  - Sony said that the credit card numbers were *encrypted*, but the hackers might have made it into the *main database* [CNN]
- It is as secure as its **weakest link**.

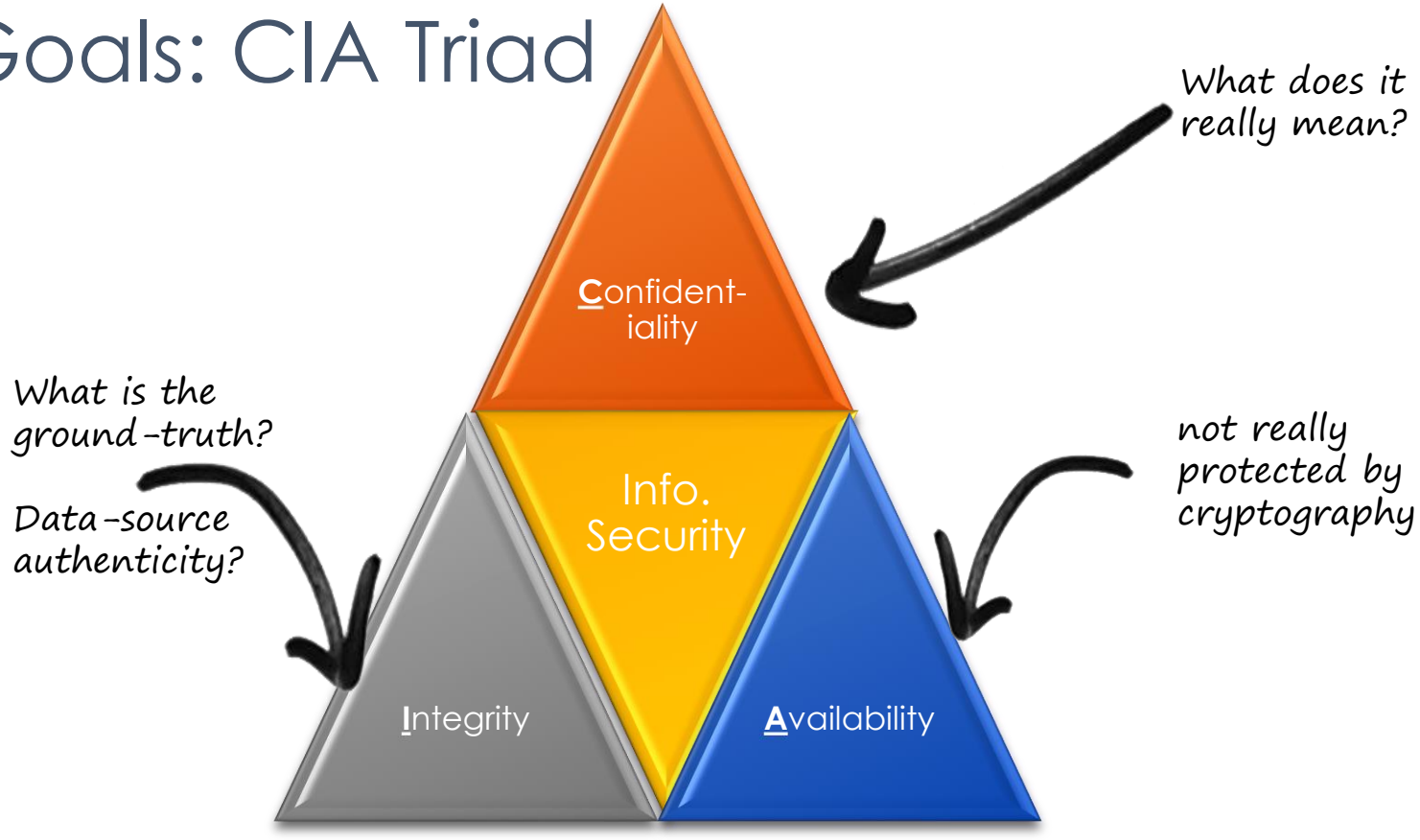
I have faith. Why can't I trust in them?

- Conflict of interests
  - R&D, insider info, strategic plan
  - Government agencies
- The Law
  - **S**arbanes–**Ox**ley Act: Financial records
  - **H**ealth **I**nsurance **P**ortability & **A**ccountability **A**ct: Medical data
  - **C**alifornia **C**onsumer **P**rivacy **A**ct: Consumer records
  - **G**eneral **D**ata **P**rotection **R**egulation

What are you trusting?

- Data is stored in more than one server
  - Trusting all servers / insiders / other tenants
- Relying on the server for access control
  - Horizontal or vertical privilege escalation
- A company have many employees
  - Careless/Cheating employees
- Encryption (number-theoretic assumptions?)

I. Goals: CIA Triad



Confidentiality

- Protect information from unauthorized access
- with historical roots in the need for wartime secrecy.

- Encryption/Cipher converts *plaintext* into *ciphertext*
- making it unreadable w/o an appropriate decryption key

- The information can be sensitive personal information
- or it can be a secret key for some other functionality
  - like a *decryption key*
  - or an *authentication key*, see next slide

Variants of Confidentiality

- Anonymity: ability to use a resource without disclosing identity/location
- Copy protection: ability to control the use of information
- Information flow control: ability to control the flow of information
- Privacy: fair collection and use of personal data

More Variant of Confidentiality

- Subliminal encoding:
  - Subliminal audio cues in stores, like background music with hidden messages
  - Subtly influencing customers, *e.g.*, discouraging theft or encouraging purchases.
- Transmission concealment:
  - Steganography hides the very existence of message
  - Messages are embedded within other messages or media
  - Copyright protection, *e.g.*, digital watermarking, to identify the "traitor"
- Existence hiding:
  - Unobservability: ability to use a resource w/o revealing this activity to third parties
- General differences:
  - Cryptography (or encryption) protects the content of messages
  - *Steganography* conceals their very existence

Variants of Anonymity

- Pseudonymity : anonymity with accountability for actions
- Unlinkability: ability to use a resource multiple times without others being able to link these uses together
  - *e.g.*, is that guy/girl you saw on the shuttle also taking 5710?
  - *cf.*, HTTP "cookies" were introduced to provide linkability

Integrity and Authentication

- A sealed letter provides confidentiality and **integrity**
- Altering the content requires tampering with the seal ($P \Rightarrow Q$)
- If the seal looks fine, the content hasn't been altered ($\neg Q \Rightarrow \neg P$)

logically equivalent

implication/inference

- **Authentication**: authentic information, so, like integrity
- What if the whole envelope is replaced?
  - The content has not been altered, literally
  - but it's not authentic
- Authentication links to the entity originated the communication
- Entity authentication: just to make sure the identity of the entity

Cryptographic Primitives / Building Blocks

- **Encryption** provides **confidentiality**, *e.g.*,:
  - *One-time* pad (OTP): the only perfectly secure scheme, the key is *as long as* the message
  - Pseudorandom (sequence) generator (PRG), pseudorandom function (PRF: "many PRGs")
  - Advanced Encryption Standard (AES) for **secret-key** encryption
  - Rivest-Shamir-Adleman (RSA) for **public-key** encryption

- (Cryptographic) **Hash functions** provides **integrity**
  - generates fixed-length hash values or "*digests*" (digital fingerprints) of input data
  - widely used in *password* storage: passwords remain hidden even if the database is stolen

- Authentication mechanisms/tools provide **authenticity**
- **Message authentication code** (MAC) for secret-key approaches, *e.g.*, HMAC
- **Digital signatures** for public-key approaches, *e.g.*, RSA, digital signature algorithm

- After the course, you know a suite of cryptographic tools for your problems.

Real-World Applications

- Secure Messaging Apps: End-to-end **encryption** ensures that only the intended recipient can decrypt and read messages.
- Online Banking: **Digital certificates** authenticate the bank's website and secure transactions through encryption.
- E-commerce: **Digital signatures** guarantee the authenticity of legal documents in online transactions.
- Data Encryption in Healthcare:
  - **Encryption** safeguards sensitive patient records,
  - making them accessible only to authorized medical professionals

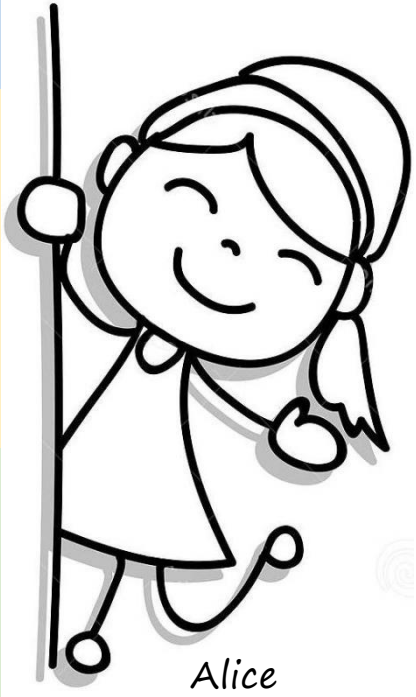
Modern Relevance

- Emerging *Applications*:
  - from online communications to financial transactions
- Evolving *Threats* and *Challenges*:
  - Ransomware, bad use of encryption!
  - Metadata leakage in communication, *e.g.*, who talks to whom
- Emerging *Technologies* in Cryptography:
  - Post-quantum cryptography addresses the potential threat from quantum computers to current encryption methods.
- Emerging *Platforms*: cloud, edge, meta-verse?
- Emerging *Needs*: machine learning over encrypted data?

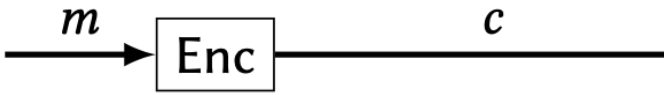
II. Overview of Crypto Topics

- Symmetric-Key Encryption
  - One-Time Pad (OTP)
  - Stream cipher
    - Pseudorandom generator (PRG)
  - Block cipher
    - Pseudorandom function (PRF)
- (Symmetric-Key) Authentication
  - Hash (and password storage)
  - Message authentication Code (MAC)
- Public-Key Cryptography
  - Digital Signatures
  - Key Agreement Protocol
  - Public-Key Encryption

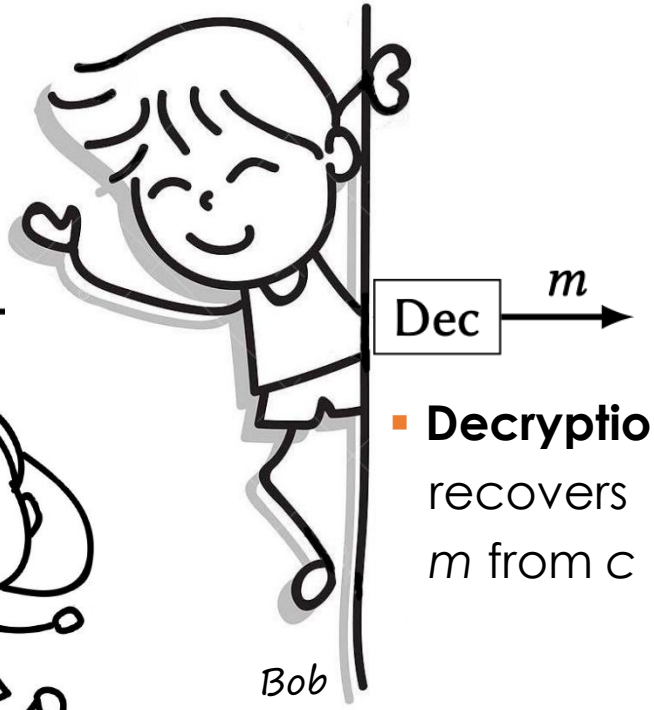
i. “Private” (Confidential) Communication



- **Plaintext**: *m*
- **Ciphertext**: *c*
- **Encryption** turns *m* into *c*



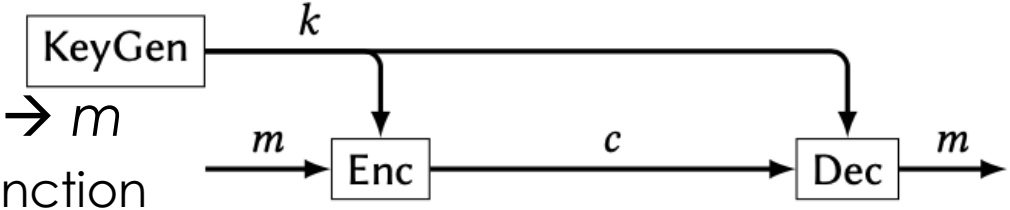
- **Eave**sdropper can (passively) observe the communication
- (easily doable in the real world)



- **Decryption** recovers *m* from *c*

What constitutes an encryption scheme?

- Key generation algorithm (KeyGen)
  - Input: security parameter λ (lambda)
  - Output: a key k
- $\text{Enc}_k(m) \rightarrow c$, $\text{Dec}_k(c) \rightarrow m$
  - *i.e.*, they are key-ed function
  - All these algorithms are supposed to be public



- A crypto scheme/construction is a collection of algorithms
- *Symmetric*-key encryption = (KeyGen, Enc, Dec)

One-Time Pad (OTP) based on XOR

- eXclusive OR (XOR): For $b_1 \oplus b_2$ (b_i is a bit), output as the table below
  - a logical operator that returns 1 (true) if the number of 1 (true) inputs is odd
- For bit-string operation $S_1 \oplus S_2$, just $\oplus$ in a bit-wise manner

- OTP = {KeyGen, Enc, Dec}
- KeyGen(1^λ):
  - *uniformly* sample *k* from the set of λ-bit strings
  - output *k*
- Enc(*k*, *m*) → *c* = *m* $\oplus$ *k*;
  - (*m* is λ-bit long)
- Dec(*k*, *c*) → *m* = *c* $\oplus$ *k*

b_2

XOR $\oplus$	0	1
0	0	1
1	1	0

b_1

$$\underline{\textbf{KeyGen:}}$$
$$k \leftarrow \{0,1\}^\lambda$$
$$\textbf{return } k$$

$$\underline{\textbf{Enc}(k, m \in \{0,1\}^\lambda):}$$
$$\textbf{return } k \oplus m$$

$$\underline{\textbf{Dec}(k, c \in \{0,1\}^\lambda):}$$
$$\textbf{return } k \oplus c$$

Crypto. Algorithms

- I have provided concrete examples, what did you learn?

$$\begin{array}{lll} & 11101111101111100011 & (m) \\ \oplus & 00011001110000111101 & (k) \\ \hline & 11110110011111011110 & (c = \mathrm{Enc}(k, m)) \end{array}$$

- You saw how Enc() (or Dec()) works for a particular input
- You get a *sense* of correctness (m = Dec(k, Enc(k, m)))
- But how can you argue about its security?

Why Cryptography is difficult?

- Security is a **global** property about the behavior of a system across **all possible** inputs.
  - You can't demonstrate security by example,
  - and there's nothing to see in a particular execution of an algorithm.

- Security is about a **higher level of abstraction**.
  - (and some students might not be comfortable with it)

- Most security definitions in this course are essentially:
  - "*the thing is secure if its outputs look like random junk.*"
  - *i.e.*, any example just look like meaningless garbage

ii. "Pseudo-"One-Time Pad

- How to encrypt a 2λ-bit plaintext m with only a *short* λ-bit secret k
  - Essentially, Alice and Bob need to "overuse" k (*e.g.*, reusing it "once more")
- Extending OTP('s Enc) from $c = m \oplus k$ to $c = m \oplus [pad]$
- $[pad]$ should be 2λ-bit long
- $[pad]$ should be uniformly distributed (we'll make it precise later)
- $[pad]$ should be derived by a *deterministic* function $G()$ from k
  - If $G(k)$ isn't unique given k, the decryptor recovers $[pad]$ wrongly
- How to formulate the requirements and design such $G()$?

How NOT to build a PRG

- Let's study a "simple" length-double PRG candidate
- $G(s) := s \,||\, s$
- Let's try to "run" a toy example of $\lambda = 4$ with different seeds
- `11011101, 01010101, 01110111, 01000100`

- Do they look like they were sampled uniformly from $\{0, 1\}^8$?
  - Can you see the "pattern"?
- Every string has its first half equal to its second half
  - not likely to hold true for a uniform distribution

How to build a PRG? How to apply it?

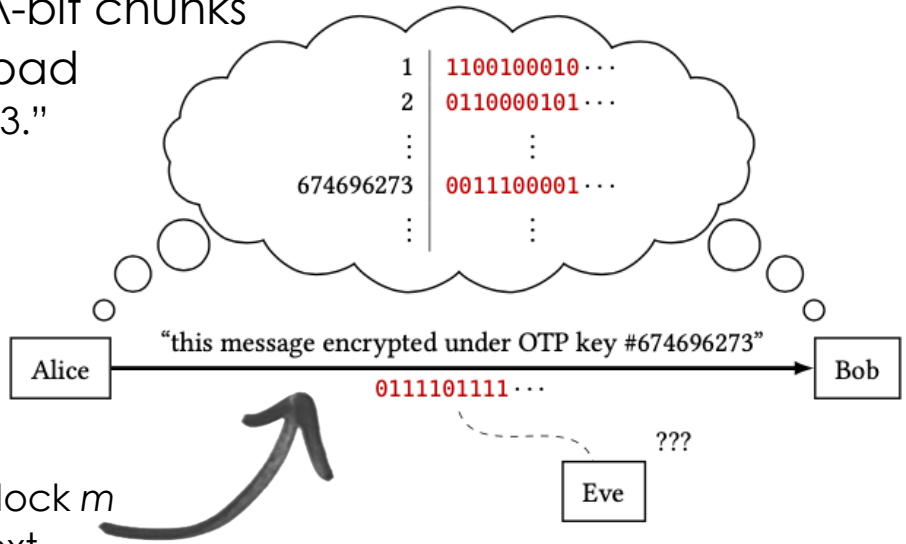
- Luckily, or unluckily, we do not cover how to build a PRG
- But we will cover, when given one, how do you "stretch" it
  - with a PRG having output length longer than the input length by 1
  - we show how to build one with output length much longer than 1

- Applications in secure messaging applications
  - *e.g.*, Signal, Telegram

iii. Block cipher or "Fast-forward" PRG

- If Alice & Bob shared an ∞ amount of randomness
- they split up the randomness into λ-bit chunks
- and use each one as a one-time pad
  - "This OTP uses key = chunk #674696273."

- Such a "look-up table" is called:
- Pseudorandom function (PRF)
  - taking a secret key *k* and an input x
  - outputting a random-looking bitstring
- or Block Cipher (*e.g.*, AES)
  - taking a secret key and a message block *m*
  - outputting a random-looking ciphertext



Questions to ponder (to be answered)

- Can we invert the function?
  - *i.e.*, given the output, recover the input
  - *cf.*, decryption
  - How to build a *permutation* given a pseudorandom function?
- What if we need to encipher more than one block?
  - Just process each block in parallel? Can we do "better"?

iv. Integrity and Birthday "Paradox"

- Imagine you are downloading a movie of many GBs
- and you really care that every byte of the movie file you downloaded is "error-free"
- Will it be nice if there is a *short digital fingerprint* of the file you can read from an authentic homepage?
  - The fingerprint is called **hash value**, produced by a **hash function**
- Can a short bitstring "represent" a giant file?
- How many people need to be in this room for the prob. that two people "collide" on the same birthday is > 0.5?
- Why do we care this question?

Many more questions

- Can we play rock-paper-scissors "fairly" online?
- Can we have sealed-bid auction online?
- How to defend against spam?
- How to fight against "sybil" in Bitcoin?
- How to authenticate a data structure beyond just data?
  - *e.g.*, blockchain = a chain of block (of records)
- Can I store your password without storing your password?!

v. Public-Key Cryptography

- Is 5383 a prime number?
- Find p and q such that $p \cdot q = 5383$?

- They form the basis of RSA, by Rivest-Shamir-Adleman.
- There are also other methods to do public-key crypto.
- So, we need to learn some mathematics later (modular arithmetic)

- Public-key encryption
  - anyone can encrypt to you using your public key
- Digital signature
  - anyone can verify the signature
  - that only you can create with your private key

III. What defines information security?

- What is information security?
  - Didn't we just describe three security goals? What's more?
- What is cyber security?
- What is a "cyber space"?

- What makes security difficult to get right?
- What makes security different from, *e.g.*, engineering?

What is Cyber Security

- Science and Engineering of guarding computer-related systems and assets against unintentional or malicious behaviors of intelligent adversaries

- Security vs. Reliability (*e.g.*, airplane/building/car safety)
  - Intentional vs. Accidental fault/failure
  - Bad guys in security can be very smart and creative

- "Cyber World" vs. "Physical World"
  - Street-smart users may become silly in the cyberspace
  - Information, unlike physical property, is abstract
  - Users often do not know the consequence of "their cyber actions"

What is Cyber Security (cont.)

- Security is a relative concept
- vs. functional requirements (*e.g.*, usability) of honest users
  - *e.g.*, how much time you wait at the gate for security check ✈
  - *e.g.*, how do you unlock your smartphone?
- vs. goal and capabilities of the adversary
  - *e.g.*, fingerprint unlock
  - What if an attacker has "access" of your finger (while you Zzz…)?

Cyber Landscape

- Desktop workstation
- Laptops
- Smartphones, Tablets
- Smart Grid
- Vehicular Network
- Cyber-Physical Systems
- Internet of Things, *etc*.

- Take cloud as an example:
- Multi-tenant cloud environment
- Tenants may not be trustworthy
  - We need some isolation of them
  - as if each is accessing an independent virtual machine
- Cloud operator (enforcing access control) may not be trustworthy
- Outsourced (computation over) sensitive data
  - *e.g.*, Big Data Analytics

Some Terminologies

- *Adversary (threat agent)*: individual, group, organization, or government that (has the intent to) conduct(s) detrimental activities
- *Attack*: any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy info. system resources or the info. itself
- *Threat*: any circumstance/event w/ the potential to adversely impact
  - organizational operations
    - (including mission, functions, image/reputation)
  - organizational assets, individuals, other organizations, or the Nation
  - via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.

Security Challenges

1. Security is not as simple as it might first appear to the novice
2. In developing a security mechanism or algorithm, one must always consider potential attacks on its security features
3. Procedures used to provide particular services are often counterintuitive
4. Physical and logical placement needs to be determined
5. Security mechanisms typically involve
   more than a particular algorithm/protocol and also
   require that participants be in possession of some secret info.
   raising questions about its creation, distribution, & protection

More Security Challenges

6. Attackers only need to find a single weakness, while the designer must find and eliminate all weaknesses
7. Security is still too often an afterthought to be incorporated into a system after the design is complete (rather than being an integral part of the design process)
8. Security requires regular and constant monitoring
9. There is a natural tendency on the part of users and system managers to perceive little benefit from security investment until a security failure occurs
10. Many users and even security administrators view strong security as an impediment to efficient and user-friendly operation of an information system or use of information

IV. Unique Aspects of Security

- Adversarial (and Long-Term) Thinking:
  - Engineering a product best for its users
  - The design should remain "reliable" against adversarial users
- Precision and Formality:
  - Engineering may want a product works 99.9% of all scenarios
  - The adversarial user makes that 0.1% happen
- Threat Model

Adversarial Thinking

- Security professionals anticipating threats to a physical facility
- Think about designing security sys. for a high-value jewelry store
- Security experts don't assume all visitors are honest.
- They plan for potential burglars with sophisticated tools.

- Security/crypto anticipates "cyber/digital" adversaries who are
  - skilled (*e.g.*, with advanced techniques/tools)
  - motivated
  - creative

Professional Paranoid

- A mindset shift is required to study cryptography/security
- We approach every problem with a healthy dose of skepticism.
- Assuming something is secure because it appears to be?
- No! we rigorously challenge it.
- This mindset leads to in-depth analysis and proof of security.
- We don't just aim for things to "work"; we aim for mathematical proofs that something is secure under well-defined conditions.
- Is the security model comprehensive enough?
- What does it mean in the real-world?

Simple Analysis: Online Banking

- You log in to your bank account & expect it to be secure.
- In security/cryptography, we don't take this for granted.

- We question how the bank ensures your data's security, prompting us to examine encryption, authentication, and access control methods to prevent unauthorized access.

Interdisciplinary (whatever it takes)

- Security/Cryptography combines elements from mathematics, computer science, and engineering.

- Embracing this interdisciplinary approach helps us create robust solutions without requiring in-depth expertise in any single field.

- Consider securing a wi-fi network. It's a collaborative effort:
- Cryptographers work on the math/algo. that encrypt the data
- Network engineers configure the routers
- Computer scientists ensure the encryption software runs.

Security vs. Usability

- Think about designing a secure login system for a smartphone.
- Traditional engineering might prioritize convenience, allowing users to easily access their device.
- For practical systems, we must balance usability with security.
- Cryptographers need to ensure that even if a smartphone is lost or stolen, an attacker can't easily access sensitive information.
- This requires a different mindset, where security often trumps convenience.

Long-Term Thinking

- Consider encrypting health records in healthcare systems
- Security/Crypto professionals must think long-term, ensuring that patient data remains confidential for decades.
- aims to provide security that withstands the test of time
- This differs from some engineering disciplines where components can be easily upgraded.
- requiring a unique mindset that anticipates future advancements in technology and potential attacks.

Precision and Formality

- Precision is crucial. We use mathematical proofs to prove security
- just as architects and engineers use precise blueprints to design buildings

- Example 1: Think of designing a safe, it depends on physical strength
- Cryptographers rely on mathematical algorithms to ensure security.

- Example 2: Traditional engineers ensure the stability of a building.
- Cryptographers secure network communication or data at rest.

- A cryptosystem uses MAC as a tool, MAC is secure means the whole system is secure?
- A cryptosystem is “secure”? We need precise language or precise model

Threat Model (overview)

- Brute-force attack
  - How many number of trials for a 3-digit pin lock?
  - What are the possible patterns to unlock a phone by drawing a figure touching each of the 9 dots in a square rid at most once?

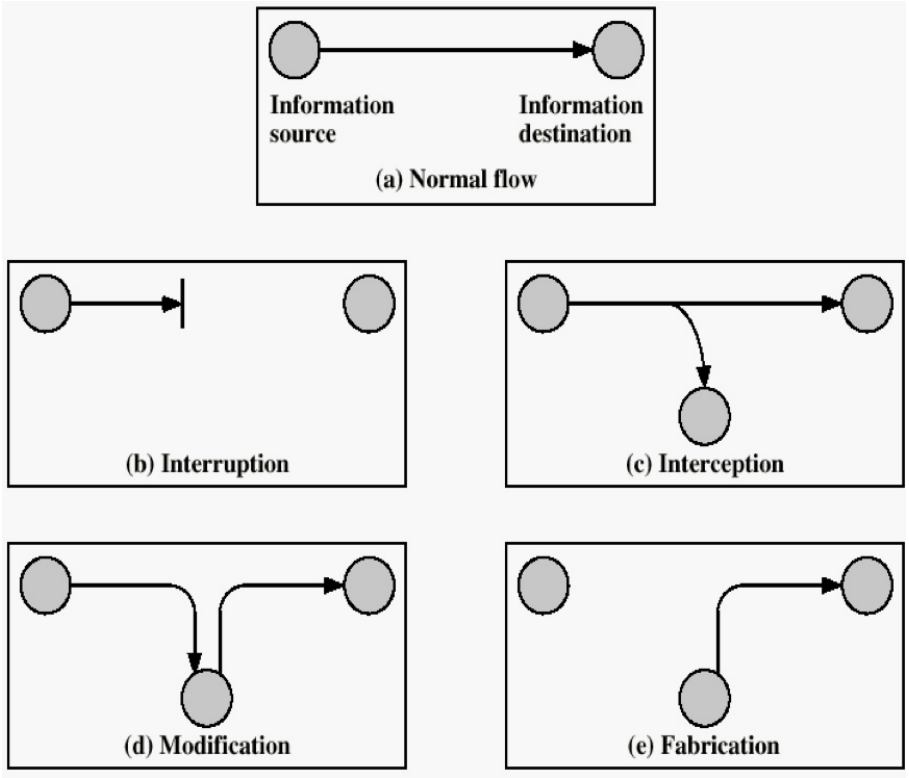
- Adaptive attack (another dimension, can still be brute-force)
  - Civil engineers assume earthquake, typhoon, *etc*.
  - but do not assume the typhoon is under control (of mythical being?)
  - An adversary sends different encrypted http packets to a webserver
  - The webserver might behave differently
    - valid decryption, then perform action
    - invalid decryption, then quickly return error

Capabilities/Agenda of Adversaries

- Passive attacks: will not alter the data or disrupt the system
  - to eavesdrop on or monitor information, but not modify or disrupt it
  - *e.g.*, Traffic Analysis: Analyzing patterns in communication to extract *meta information* without directly accessing the content
    - such as the frequency and timing of messages,
- Active attacks: make changes to the data/system itself, *e.g.*,:
  - Stop a message flow
  - Modify a message
  - Create a new message flow (forging), replay an old message flow
  - Malware: compromise the data integrity / disrupt system operations
  - (Distributed) Denial of Service (DDoS) Attacks: overloading a system, network, or service to make it unavailable to legitimate users.
  - Impersonation (Spoofing): pretending to be someone/entity else

Classification of Threats

- Leakage
- Tampering
- Vandalism



(Detour :) Various Types of Digital Pest

- Logic Bomb: logic embedded in a program that triggers when specific conditions are met
  - *e.g.*: A programmer inserts code in a financial application that deletes critical data if the program is executed after a specific date
- Backdoor/Trapdoor: secret undocumented entry point in a program, allowing access without passing authentication
  - *e.g.*: A developer includes a hidden code pathway in a network tool, providing unauthorized access to anyone who knows about it
- Trojan Horse: covert undocumented routine *R* embedded within a useful program *P*, execution of *P* results in execution *R*
  - *e.g.*: A seemingly harmless game contains hidden code that, when executed, allows a remote attacker to control the running device

Various Types of Digital Pest (cont')

- Virus: code embedded within a program that *replicates* itself by inserting copies into other programs, performing some functions
  - *e.g.*: A malicious email attachment containing a virus, which spreads to others in the address book or files on infected device.
- Worm: a self-replicating program that spreads across a network
  - *e.g.*: exploiting a vulnerability in a network protocol to spread itself
- Zombie/Bot: a program that covertly takes over an Internet-connected device, using it to launch an untraceable attack
  - *e.g.*: DDoS attacks against a specific service by flooding request

How Crypto “helps others” (sub-areas)

- Security, Privacy
  - ACM Conf. on Computer and Communications Security (CCS)
  - IEEE Security & Privacy (S&P/"Oakland")
  - Usenix Security
  - ISOC Network and Distributed System Security (NDSS)
  - ACSAC, CODASPY, CSF, ESORICS, EuroS&P, PETS, RAID, SACMAT, WiSec, …
- Network/Distributed Computing/WWW
  - IEEE Infocom
  - IEEE Intl. Conf. on Distributed Computing Systems (ICDCS)
  - ACM Principles of Distributed Computing (PODC)
  - ACM The Web Conference
- Theory
  - IEEE Foundations of Computer Science (FOCS)
  - ACM Symposium on Theory of Computing (STOC)
  - ACM Conf. on Innovations in Theoretical Computer Science (ITCS)
  - IEEE International Symposium on Information Theory (ISIT)

How Crypto Helps Info. Sec. (Overview)

- Confidentiality (for your eyes only) vs. eavesdropping, tracing
- Integrity (has not be altered) vs. tampering
- Authentication (you are who you say you are)
  vs. impersonation, masquerading, spoofing
- Access control (only the intended can "use" the resources)
  vs. unauthorized use / abuse of resources
- Non-repudiation (the order is final)
  vs. denying one's act, backing away from a deal
- Availability (serves whom should be served) vs. (D)DoS attacks

What is Hacking?

- Info. Tech. is getting more powerful and widely deployed
  - FinTech, RegTech, HealthTech, you name it…
  - Opportunity (features) comes with risks (bugs)
- Still, it is mostly a program, and program has bugs
  - Or people make mistakes
- Bugs/Mistakes leads to loophole, which invites "hackers"
- Vulnerability: weakness in an information system procedures, internal controls, or implementation that could be exploited/triggered by a threat source

Why we still make the same mistakes?

- Learn from the History, or can we?
- Back in the old days, computers are not connected…
  - Internet -> Ubiquitous Computing (or IoT)
  - Cloud Computing -> Edge Computing
  - There are paradigm shift, but something "core" still remains
- Indeed, many people/computers/programs today are still making similar kinds of mistakes as decades ago

Authentication

- Who are authorized/legitimate user to do something?
- People use password for authentication 4 decades ago
  - How about now? Well, we do have 2-factor authentication
- Password, or secret-key, requires the verifier has the secret
- Using public-key cryptography, public key = a user

- (Public-key) signature leaves non-repudiable evidence
- Public-key encryption: enforce access control

Authentication and Digital Certificate

- Using public-key cryptography, public key = a user
- In RSA (more details later), public key is just a number
- How to associate a public key to a real-world identity?
- Actually, what is authentication?
- You are what you are, physically?
  - Biometric authentication?
- You are what you have?
  - Passport?
- Who certifies your passport?
- We need a digital *certificate* certified by a certificate authority.
- Alternative approach, a single trusted party to distribute key?

Authenticating a Server

- How do we know cuhk.edu.hk is really cuhk.edu.hk?
- You(r computer) just knows an IP address of the *domain*.
- We need to translate a domain name / URL to an IP.
- We need a certificate to bind the above translation.
- This is done by many domain name servers (DNS).
- Many users are visiting many websites, asking for many IP's
- Which response is for you?
- Attacker exploits this opportunity to give you a false reply.
- An "easy" to see and launch attack if you study hash function!

Phishing (real-world examples)

- Phishing: A fraudulent attempt to trick you to provide personal information, *e.g.*, HKID #, password, credit card #, etc.

From: <hradmin@cuhk.com>
Date: Sunday, November 25, 2018 11:35 PM
To: Sherman Chow (IEG)
Subject: Salary adjustment from Jan 2019
Reply-To: hradmin@cuhk.pw

Dear Colleague,

We assessed the 2018 payment structure as provided for under the terms of employment and discovered that you are due for a salary raise starting Jan 2019.

Your salary raise documents are enclosed below:

Statement.pdf

Human Resources

The Chinese University of Hong Kong

- `From: Prof. xxx <tfeng6997@gmail.com>`
- `Sent: 2 Mar 2021 (Tue) 4:48PM`
- `Subject:` URGENT: Quick Assistance Needed
- Are you free at the moment?
- If yes, Send me your whatsapp number for direct communication ..

Why is it risky to click a random web link?

- Cross-site Scripting, Cross-site Request Forgery
- Attacker exploits "cookie"
  - server-implanted data to ease your life, *e.g.*, no repeated authentication
- Both are technical, but explainable by the same concept: trust
- Other attacks
  - *e.g.*, SQL injection (below shows connecting to an SQL/database server)