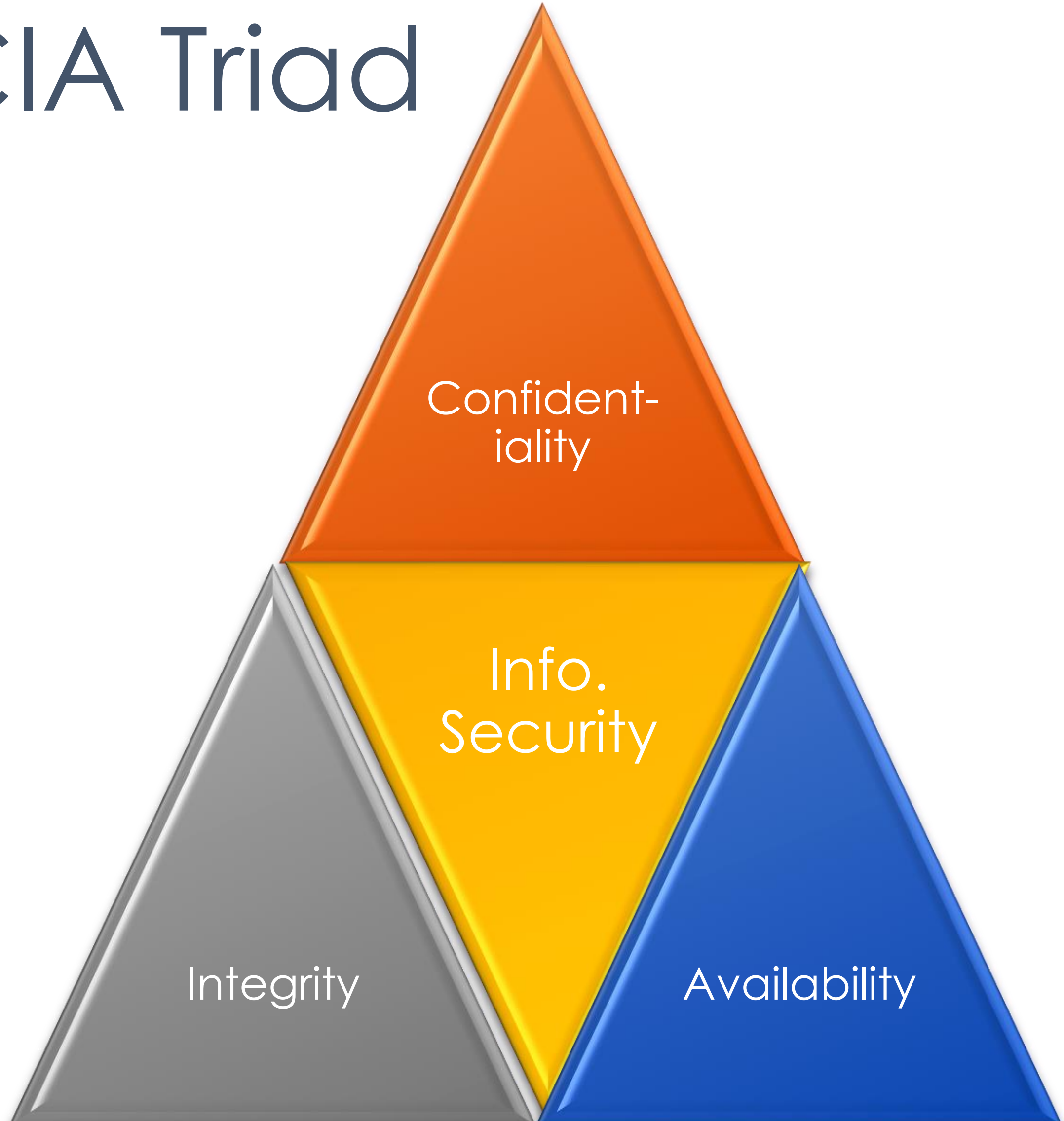
ENGG 5383
Applied Cryptography

Sherman Chow
Chinese University of Hong Kong

Spring 2023

Lecture 1: Introduction

Goals: CIA Triad



Confidentiality

- Prevent the disclosure of info. to unauthorized party
- Encryption: use a “key” to turn a *plaintext* into a *ciphertext*
- Without the “secret key”, the ciphertext is not “useful”
- What constitutes an encryption?
  - Framework / A suite of algorithms

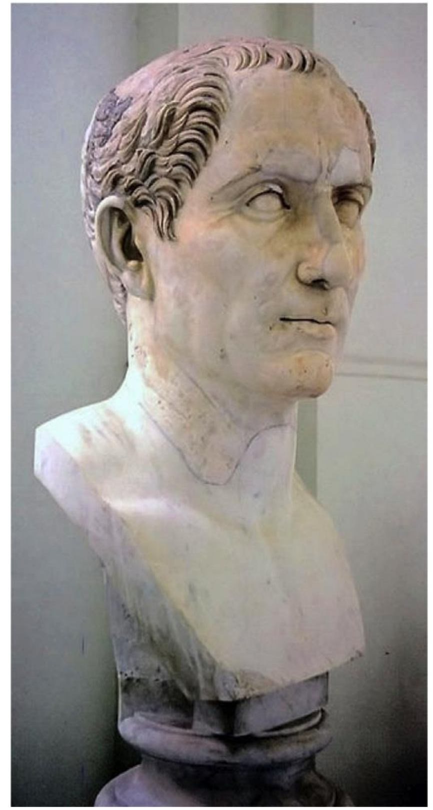
What constitutes an encryption scheme?

- Encryption: E (*m*) → *c*
- Decryption: D (*c*) → *m*
- Need to generate a key *k*
- Key generation algorithm
  - Input: security parameter
  - Output: a key *k*
- E_k (*m*) → *c*, D_k (*c*) → *m*
  - i.e., they are key-ed function
  - All these algorithms are supposed to be public (more on this later)
- *Symmetric*-key encryption

Caesar Cipher

- Consider the 26 alphabets of English
- Encoded them as a number in [0, 25]
- E_k $(m) \rightarrow m + k \bmod 26$
- D_k $(c) \rightarrow c - k \bmod 26$
- salad -> wepeh ($k = 4$)

- Frequency analysis



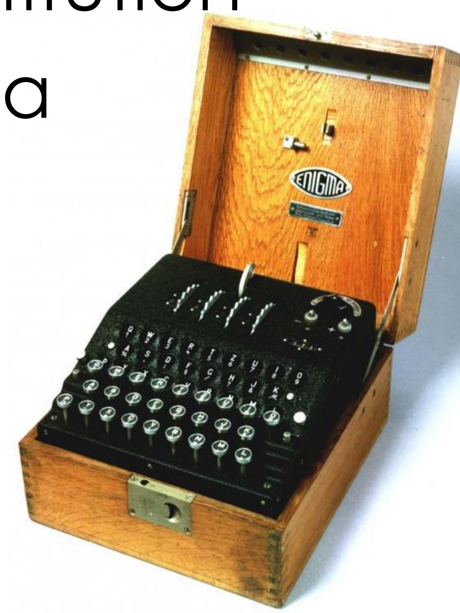
Vigenère Cipher

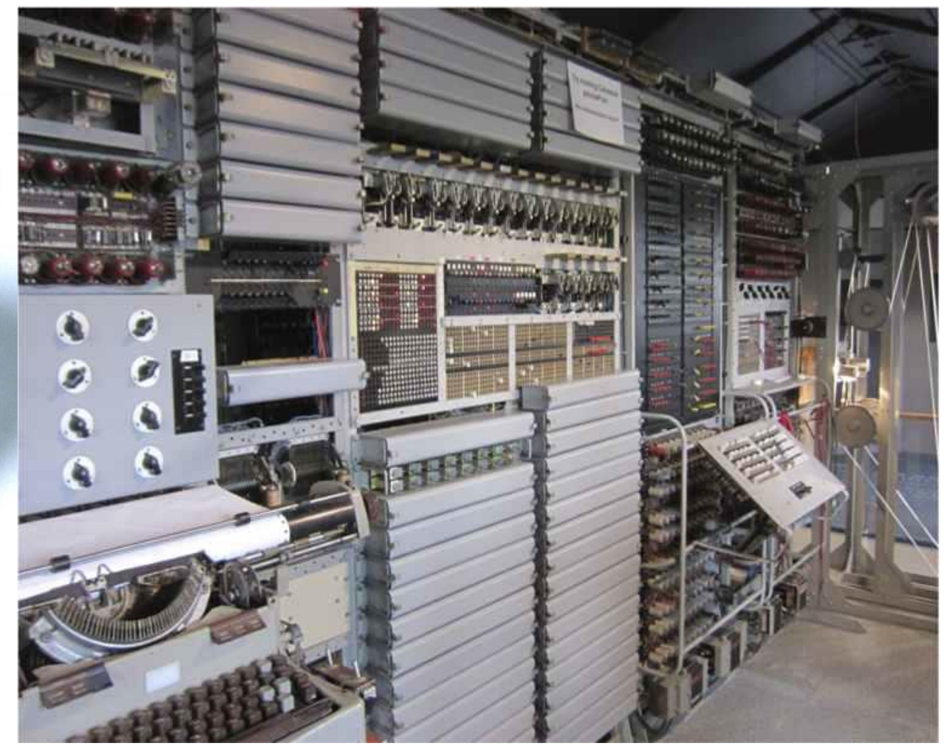
- Variants of Caeser Cipher
- Idea: not always map a plaintext to the same ciphertext
- Plaintext: AttackAtDawn (case insensitive)
- Key: Lemon
- Key “Sequence”: LEMONLEMONLE
- Ciphertext: LXFOPVEFRNHR

- How to attack?

Enigma

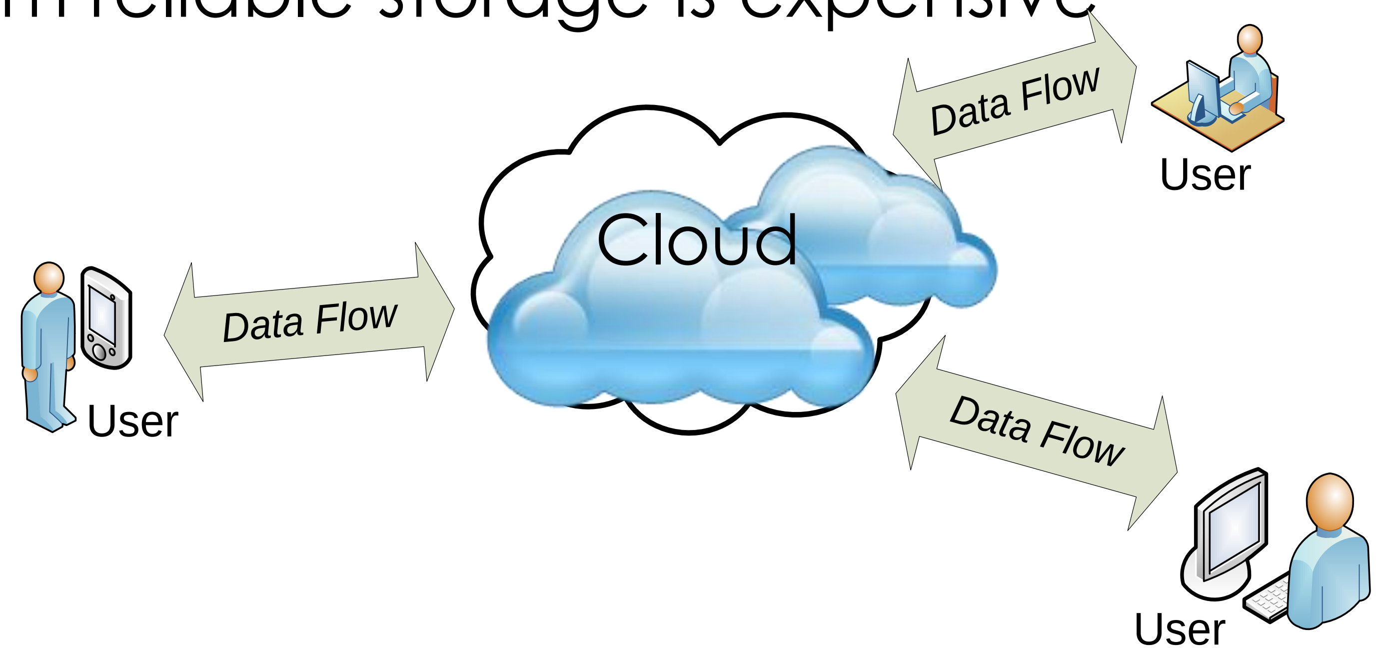
- Caeser and Vigenère Ciphers are both polyalphabetic
- Based on Substitution
- So does Enigma





Basic Settings of Cloud Storage

- Client stores (large) files with the server
  - Online backup, Software as a Service (SaaS), etc.
- Long-term reliable storage is expensive



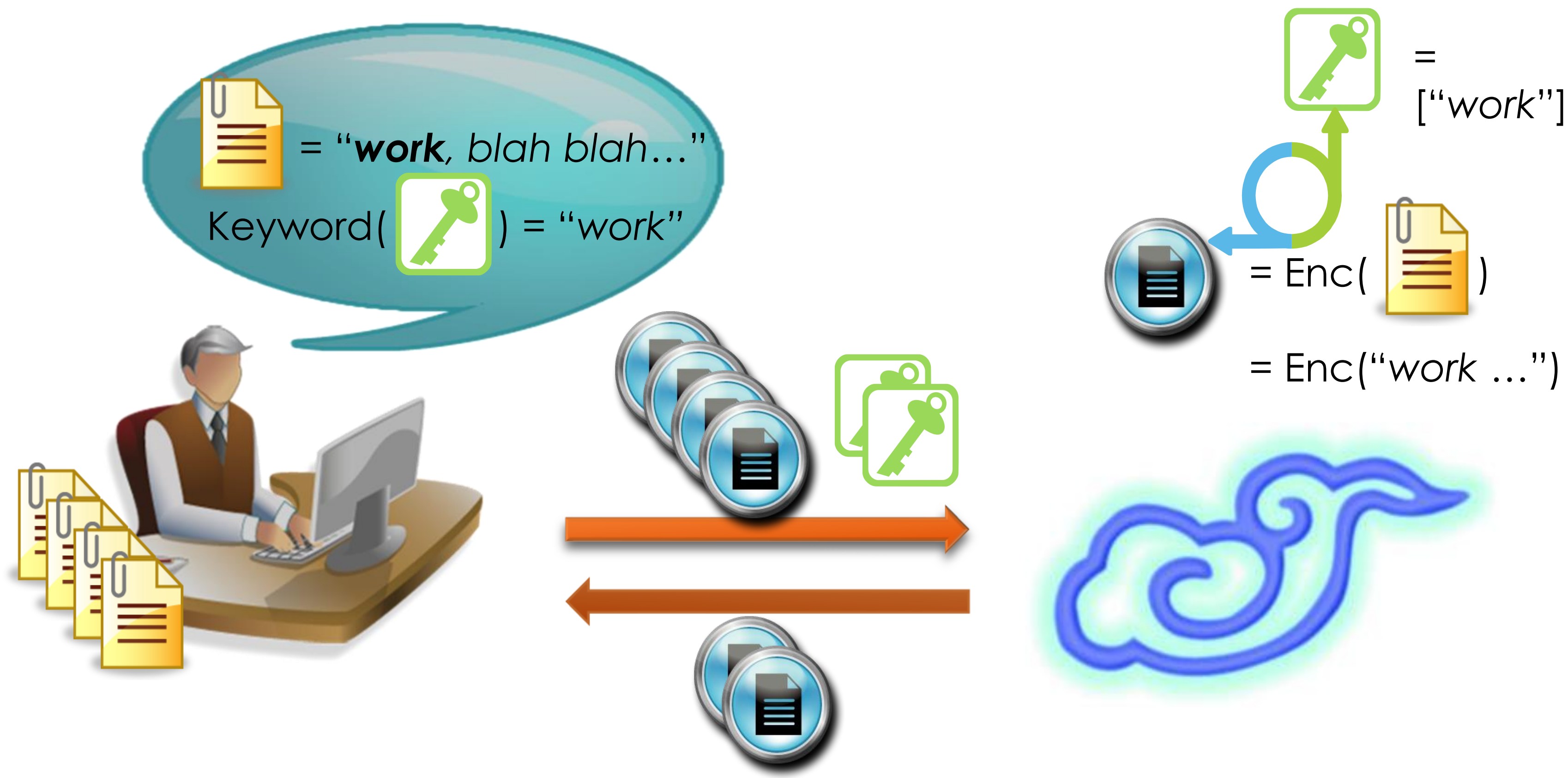
Is "full" confidentiality always desirable?

- Consider you want to upload your files to the cloud.
- What do you want your cloud service providers do?
- They cannot do much more than storage.
- How about encrypted e-mail?
- You may want your mobile devices only download e-mails marked w/ the keyword "urgent" from the server.
- You don't want the server to know what are the keywords associated with each email.

Retrieval of Encrypted Data

- Download all data, then decrypt
  - O(N) communication
  - N: number of documents
- Build a local index, then download
  - O(N) local storage
- Ideally, O(n) complexity (at least at client)
  - n: number of matching documents ($n \ll N$)

Searchable (Symm.) Encryption



What we talked about so far…

- Primitive / Building block: Encryption
- Some constructions of encryption / encryption schemes
- Some attacks
- We identified some higher application of encryption
- Some “attacks”/”weakness” can be a useful feature
- Some discussion of desired performance parameters
- Three initial tasks of “crypto study”:
  - Identification of the problem / application scenario
  - Identification of the primitive which may be useful
  - Definition of Functional Requirements and Security requirements

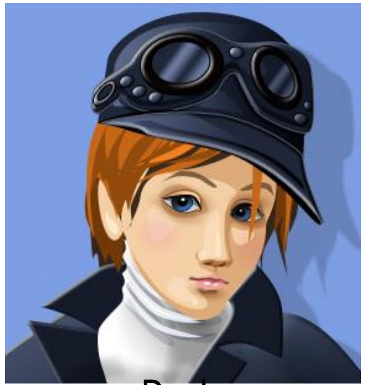
Integrity

- Prevent undetectable modification of data
- Non-repudiation: cannot deny having sent a message
- Message Authentication / Digital Signature
- Is non-repudiation / public-verifiability always desirable?

Motivating Scenario

- Alice is making an offer to Bob
- Bob acquires a signed offer from Alice
- But Alice doesn't want Bob to show it to anybody else
- Bob can not use Alice's offer as leverage to negotiate better terms with, say, Carol


Alice


Bob


Carol

Applications

- Love letters
- Job offers
- Contracts
- Receipt-free elections
- Selling of verified (*e.g.*, malware-free) software

Vehicle Safety Communications

- Safer and more efficient driving
  - electronic brake light
  - road condition warning
  - curve speed assistance
  - collision warning
  - emergency vehicle signal preemption
  - ...
- Cannot be misused to create accidents
- But we want to avoid invading privacy of the drivers


Accident ahead!
Let's go to
another tunnel

Possible Solutions

- Requires the driver to sign on every messages
- This compromises (location) privacy.

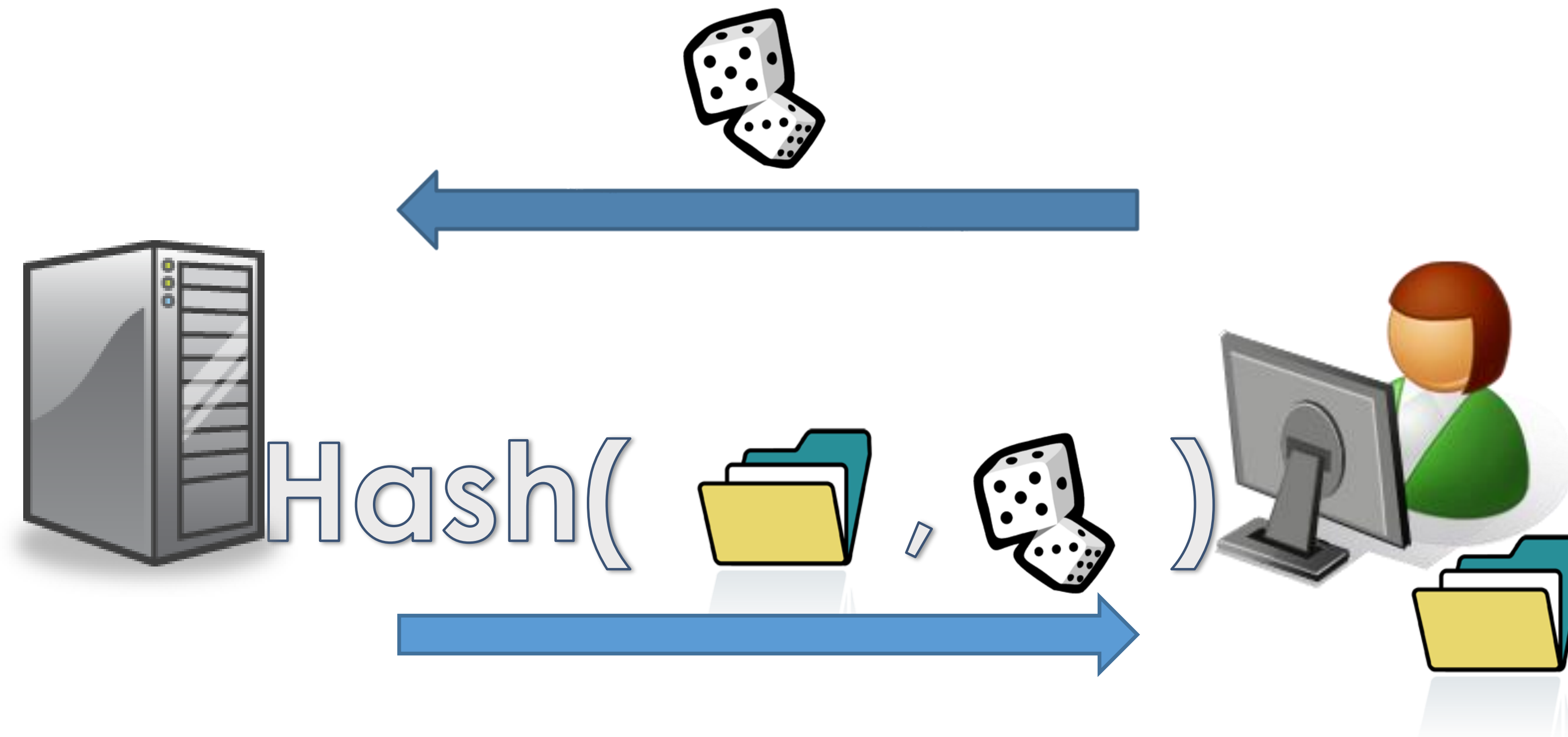
- Signatures are "anonymous" in normal circumstances
  - What does that mean?
- A "trusted" party can "open" a signature if necessary.
  - Opening a signature means revealing its true signer.
- Good enough? Too powerful?
- Any alternative formulation?

Availability

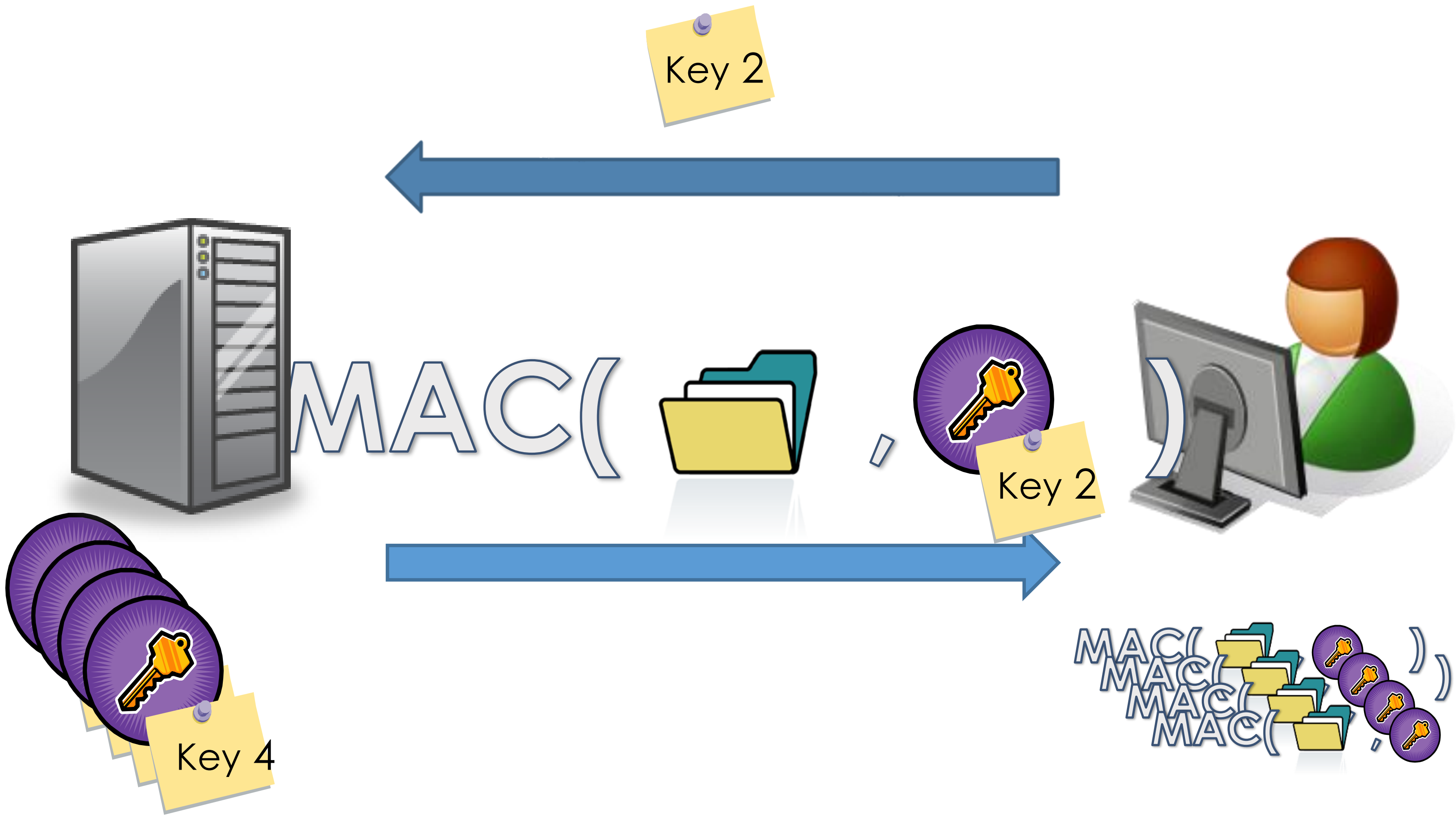
- A system must be serving the info when it is needed.
- How can cryptography help to ensure availability?

- Consider cloud storage again, how can I ensure that the cloud service provider is really storing my file?
- If the cloud deleted your file, not much you can do.
- At least, I can provide (cryptographic) evidence when it fails to do so.

Challenge + Message Digest



Message Authentication Code (MAC)



Can we do more “outsourcing”?

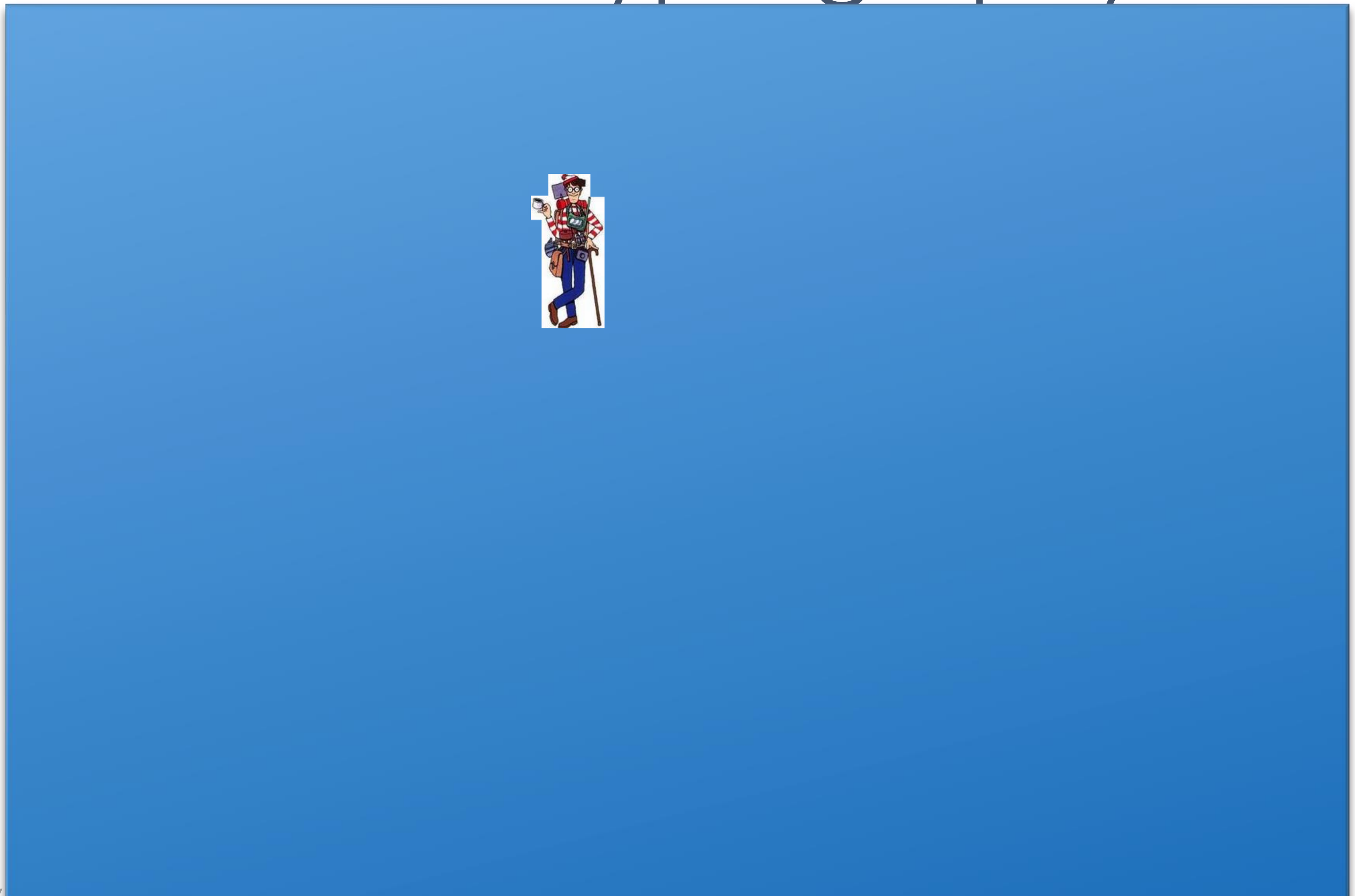
- The storage is outsourced to the cloud.
- Why not outsource the auditing to third-party auditor?

- Wait, will this auditor need to know the plaintext data?
- Using “proof-of-retrievability” (PoR) protocol, it doesn’t.

- “It doesn’t need” does not imply “It cannot learn”
- “Zero-knowledge” PoR

Where is Waldo/Wally?

Applied "Kid" Cryptography



Yao's Millionaires' Problem

I have \$$x$

I have \$$y$

Is $x > y$?

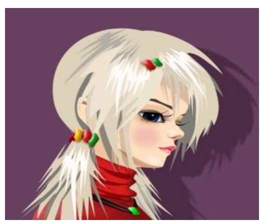
Secure comparison can be applied to, among many,

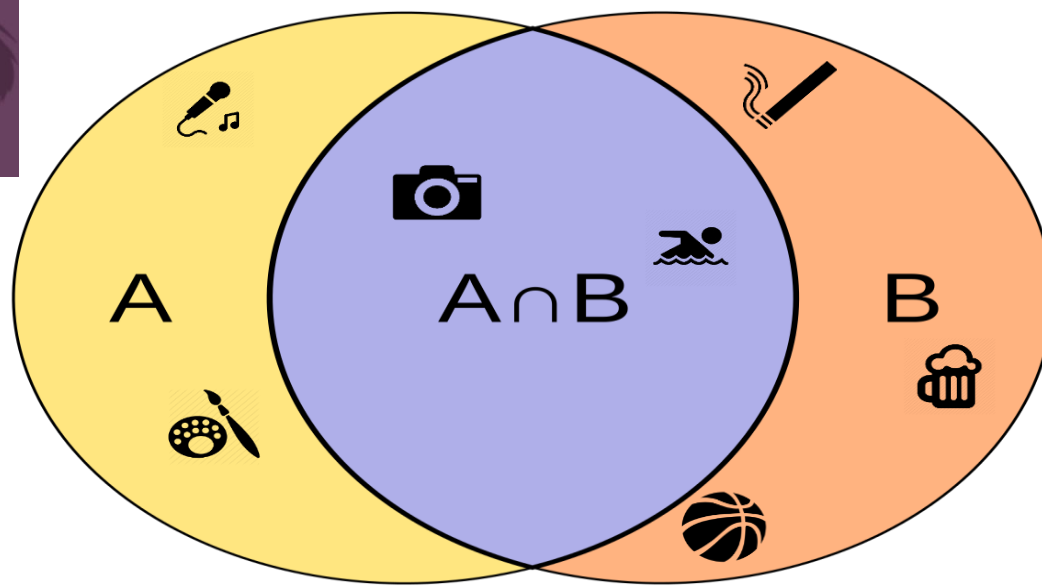
- Training over encrypted data (*e.g.*, ReLU)
- Location-based services (*e.g.*, who are near enough?)

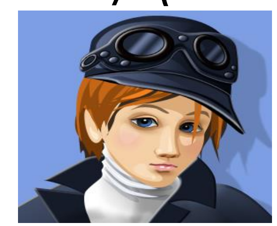
Private Set Intersection (PSI)

PSI can be applied to, among many,

- Privacy-preserving *contact tracing*
- CSAM detection (Apple PSI system)
- Advertisement efficacy (Google PSI sum)

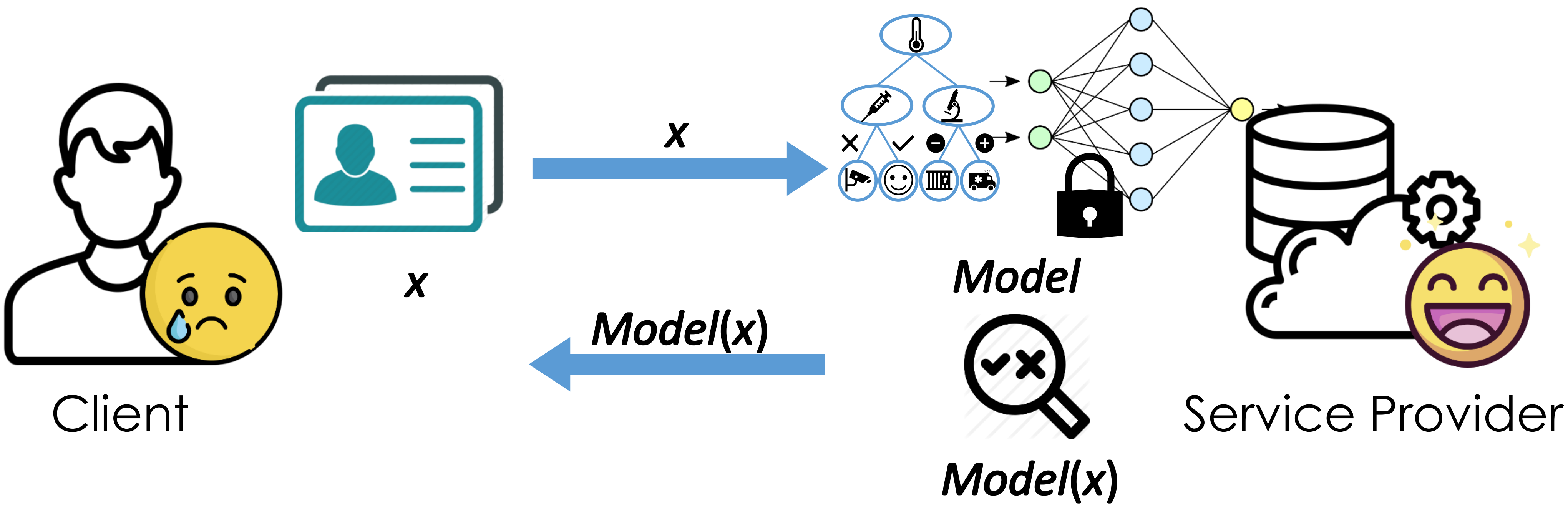






Query Privacy in ML Inference

- Queries in machine-learning (ML) inference can be sensitive
  - Social applications, Medical image analysis, Computer vision, …
- The "natural" way will leak them to the server



Summary of Tools/Primitives Covered

- Searchable Encryption
- "Non-transferable" Signature
  - Undeniable signatures, Confirmer signatures
- Signature with "Fair-Privacy"
  - Group signature, Traceable signature
- Proof of Retrievability
- Zero-Knowledge Proof
- Secure Multiparty (Two-party) Computation
  - Secure Comparison, Private Set Intersection

Possible Topics for Project

- Access Control Encryption
- Outsourcing (Verifiable) Computation
- “Secure” Data Analytics / Machine Learning
- Password Hardening
- Blacklistable Anonymous Credentials
- Cryptocurrency and its “Privacy-Preserving” version
- Specific Zero-Knowledge Proof
- Auto Synthesis/Analysis of Cryptographic Schemes
- Lattice-Based Cryptography

Back to (Basic) Encryption

- $G(1^\lambda) \rightarrow k$, $E_k(m) \rightarrow c$, $D_k(c) \rightarrow m$
- Have we specified the algorithms clear enough?
- D() must always be correct
  - How to relax this requirement? Why do we want to relax it?
- Have we specified the security requirement?
- Have we specified the adversary's power/knowledge?
- G(), E(), D() are all public info. known by the adversary
  - Kerckhoffs' principle (cf. security by obscurity)

How to define security?

- Let the adversary have unbounded computational power
- Exercise: argue that both sender and receiver must share a secret not known to the adversary
- Without the “secret key”, the ciphertext is not “useful”.
  - The ciphertext leaks no information about the plaintext.
- How to define information? (Or rather the lack of it?)
- We use entropy to quantify information
  - How probable is it?
  - e.g., a fair coin toss vs. a dice with all faces being identical
  - Exercise: construct its definition (or check “Information Theory”)

Shannon's Information-Theoretic Security

- We want to say "a priori probability of a plaintext message *m* is the same as the a posteriori probability of *m* given the corresponding ciphertext *c*."
- $H(m) = H(m \mid c)$
  - R.H.S.: conditional entropy of the plaintext given the ciphertext
- This is a definition of confidentiality

(The Almighty) One-Time Pad

- Now I suggest to use the following encryption scheme:
  - pick a random key as long as the plaintext
  - to encrypt: XOR the key with the plaintext bitwise
    - Or bitwise modulo addition (mod 2)
- Exercise 1: prove it is IT-secure
- Exercise 2: prove it is secure for any message distribution
- Exercise 3: prove it is optimal (i.e., minimum key-length)
- Problems?

Tasks of Crypto. Study

- Identification of the problem / application scenario
- Identification of the primitive which may be useful
  - Do not re-invent the wheel
  - Extending existing primitives
  - Relation between primitives (one implies another?)
- Definition of Functional Requirements
  - A suite of algorithms / protocols
  - Input & Output behavior / interfaces
  - Entities involved
  - System model: which entity executes which algorithm/protocols?
- Definition of Security requirements
  - Relation of security notions (one implies another?)
- Construction of the schemes
- Analysis of the proposed construction
  - Security Proof: Provable Security!
  - Efficiency (Order Analysis and/or Experiment on Prototype Implementation)

"Compressed" Secret-Keys

- Pseudo-random number generator (PRNG)
  - outputs a long string of "random-looking" bits
  - from a short random seed
  - a.k.a. stream cipher
- Computationally secure against Next-bit test
  - given the first k bits of a random sequence
  - no polynomial-time algorithm can predict the $(k+1)^{th}$ bit
  - with probability of success better than 50%
  - a generator passing the next-bit test will pass all other polynomial-time statistical tests for randomness [Yao82]

Next Lecture

- Security against computationally-bounded adversary?
- Public-key encryption
- One-way function (OWF)
- One-way permutation (OWP)
- Trapdoor permutation (TDP)
- Crash course on number-theory
- Number-theoretic candidates of OWF, OWP, TDP
- Modeling security of public-key encryption