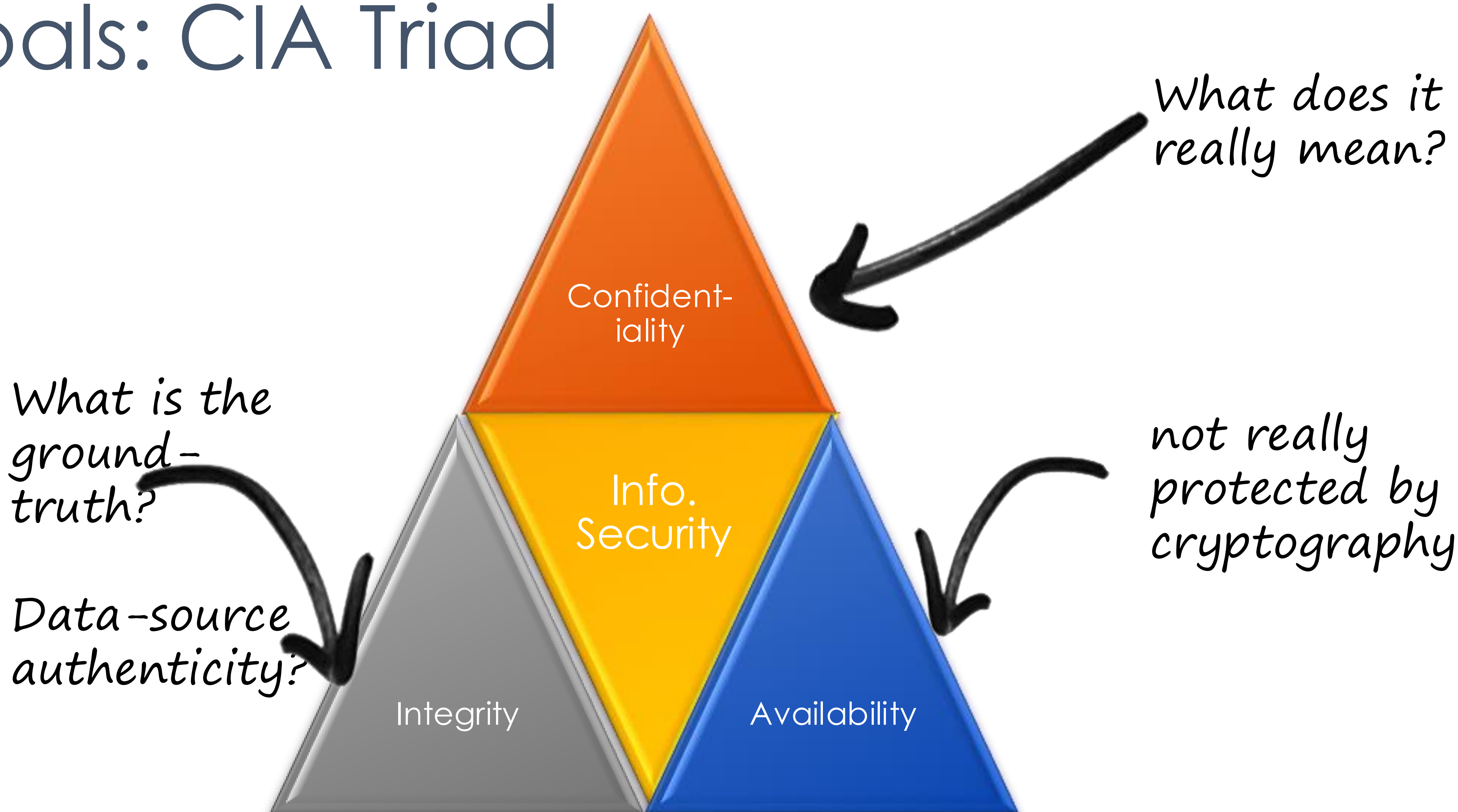
ENGG 5383 Applied Cryptography

Sherman Chow
Chinese University of Hong Kong
Spring 2025
Lecture 1: Introduction

Goals: CIA Triad

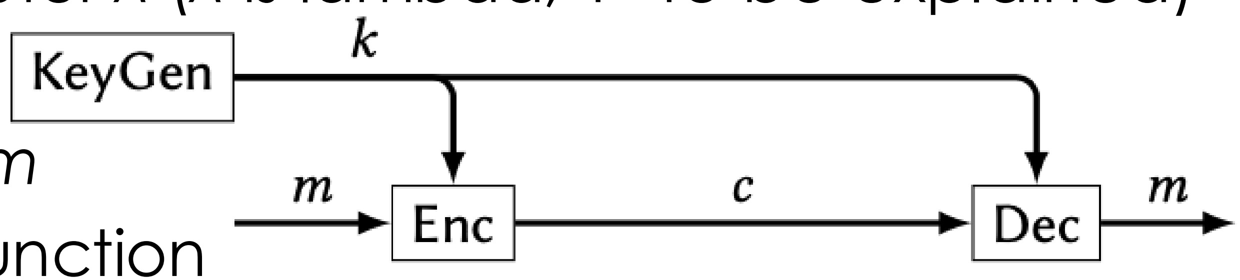


Confidentiality

- 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?

- A crypto scheme/construction is a collection of algorithms
  - we may refer to the entire scheme by a single variable, *e.g.*, Σ
- *Symmetric*-key encryption Σ = (KeyGen, Enc, Dec)
- Key generation algorithm (KeyGen(1^λ) → k)
  - Input: security parameter λ (λ is lambda, 1^λ to be explained)
  - Output: a key k



- $\text{Enc}_k(m)$ → c, $\text{Dec}_k(c)$ → m
  - *i.e.*, they are key-ed function
  - All these algorithms are supposed to be public

Caesar Cipher

Review concepts:
Encoding (is not encryption)
Modular arithmetic
(mod operation: finding remainder)

Letter	Frequency
e	12.7
t	9.1
a	8.2
o	7.5
i	7.0
n	6.7
s	6.3
h	6.1
r	6.0
d	4.3
l	4.0
c	2.8
u	2.8
m	2.4
w	2.4
f	2.2
g	2.0
y	2.0
p	1.9
b	1.5
v	1.0
k	0.8
j	0.15
x	0.15
q	0.10
z	0.07

- Romans employed such an “encryption” scheme
- Consider the 26 alphabets of English
- Encoded them as a number in [0, 25]
- $E(m) \rightarrow m + k \bmod 26$
- $D(c) \rightarrow c - k \bmod 26$
- my salad -> qc wepeh ($k = 4$)

- Vulnerable to Frequency Analysis
  - with knowledge of plaintext distribution
  - cryptii.com/pipes/caesar-cipher
  - crypto.interactive-maths.com/frequency-analysis-breaking-the-code

Vigenère Cipher: a variant of Caesar Cipher

- Idea: not always map a plaintext to the same ciphertext
- Plaintext (*m*): AttackAtDawn (case insensitive)
- Key (*k*): Lemon
- Key "Sequence" (*s*): LEMONLEMONLE
- Ciphertext (*c*): LXFOPVEFRNHR

Concept to be revisited later:
Generating a longer pseudorandom sequence

s	l	e	m	o	n	l	e	m	o	n	l	e
m	a	t	t	a	c	k	a	t	d	a	w	n
c	l	x	f	o	p	v	e	f	r	n	h	r

- How to attack?
  - index of coincidence to figure out the key length (if not known) [**]

Enigma

- Caesar and Vigenère Ciphers are both "polyalphabetic"
- Based on *Substitution*
- So does Enigma
- employed by
  - Nazi Germany
  - during World War II

Photo taken at Bletchley Park

"Rail-Fence" Cipher via Transposition

DISGRUNTLED EMPLOYEE

↓

```
D   R   L   E   O
 I G U T E   M L Y E
  S   N   D   P   E
```

↓

DRLEOIGUTE MLYESNDPE

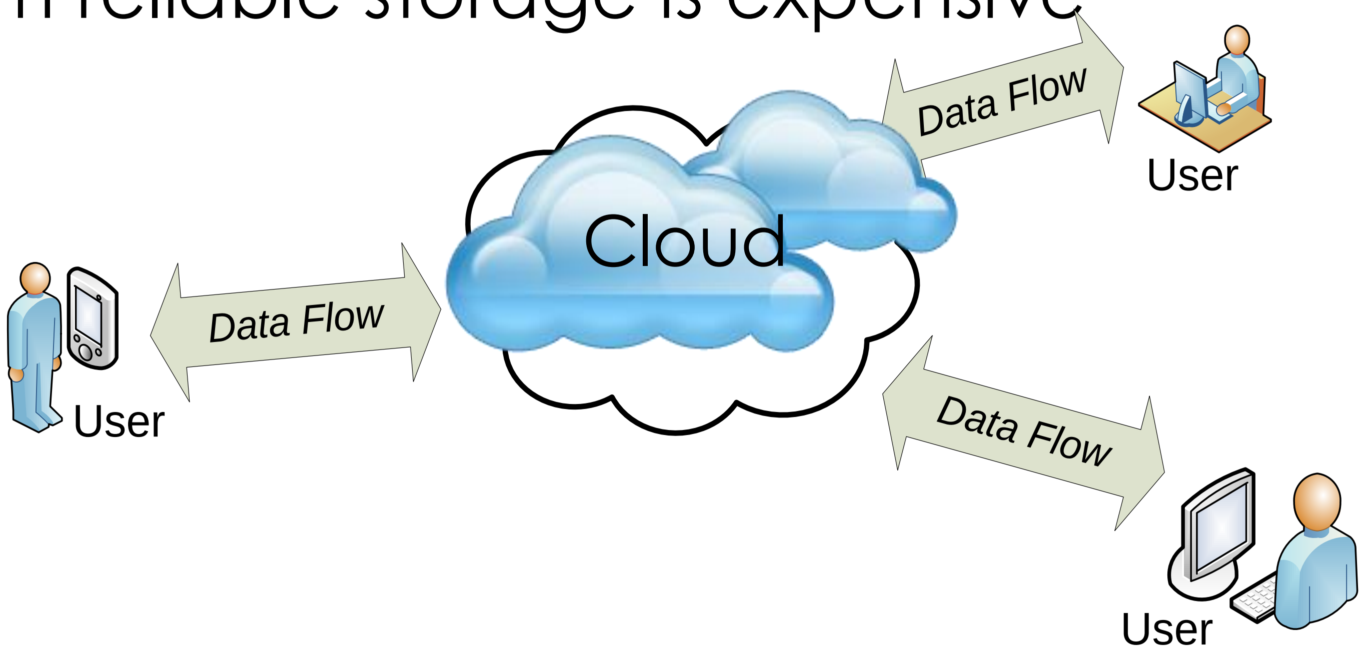
Defining Security

- Making the nebulous concept of "security" concrete
- Breaking the vicious circle of "cat-and-mouse" games
- We will try to model the attacker as "powerful" as possible
- Keep this in mind: we define (*i.e.*, limit) our problems
- We first define the problem and the system

"To define is to limit."
—Oscar Wilde
(Irish poet and playwright)

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



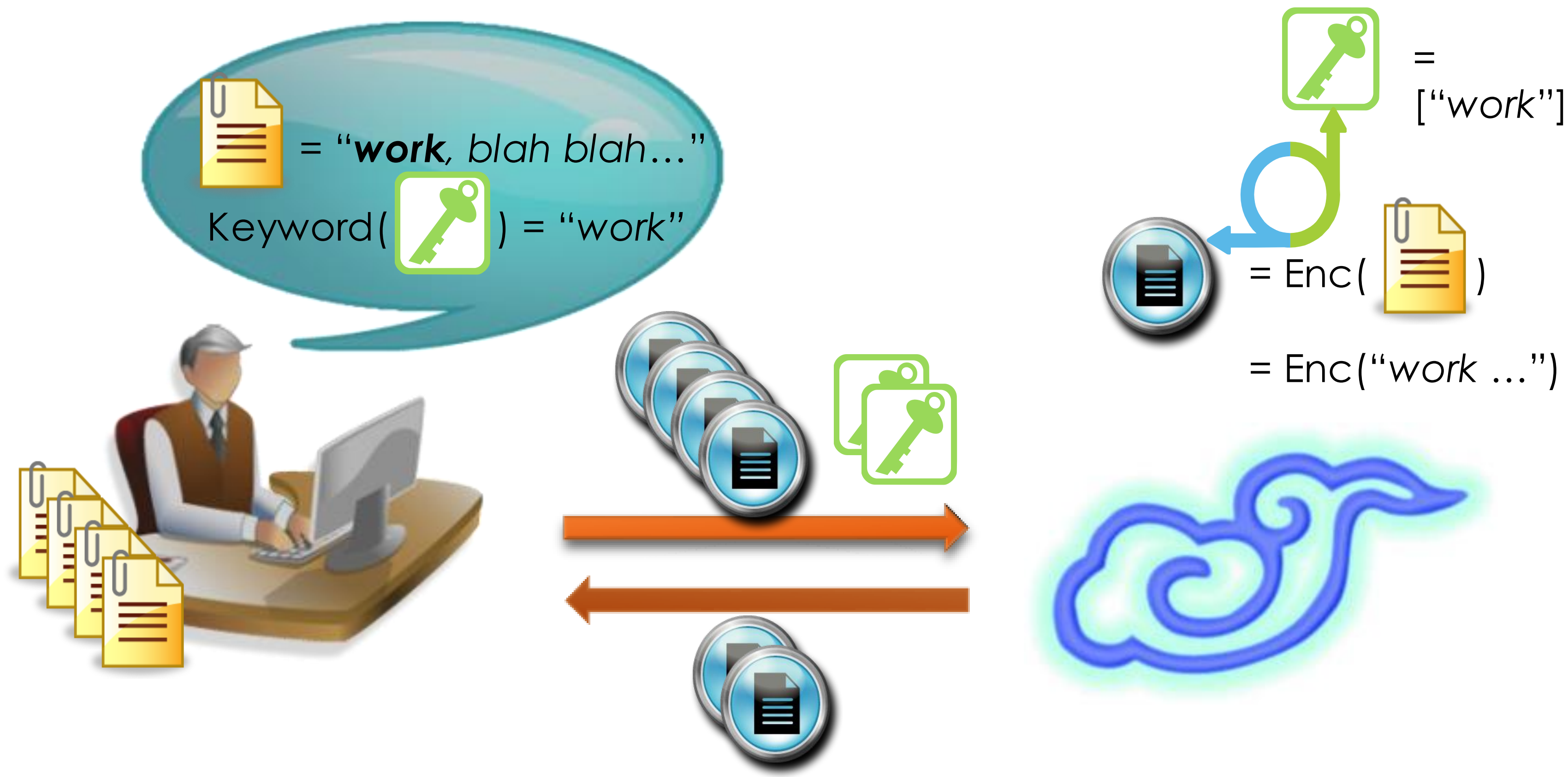
Is “full” confidentiality always desirable?

- 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 for client)
  - n: number of matching documents ($n \ll N$)

Searchable (Symm.) Encryption



Deterministic Encryption

- Same inputs (secret key and plaintext) always lead to the same output (ciphertext)
- The first solution idea in most people's mind for search?
- To search for w, secret-key owner encrypts w for the server.
- It lets equality test on ciphertexts carry over to plaintexts.

- However, even before searching, the server knows what ciphertexts are related to the same (unknown) keyword
- Can we do better?

What we talked about so far…

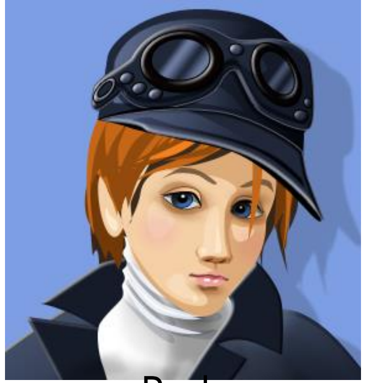
- Primitive / Building block: Encryption
- Some constructions of encryption / encryption schemes
- Some attacks
- We identified some higher application of encryption
- Some "attacks"/"weakness" can be a useful feature
- Some discussion of desired performance parameters
- Three initial tasks of "crypto 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 Story


Alice


Bob


Carol

- 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

- Applications
  - Job offers
  - Contracts
  - Love letters
  - 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.

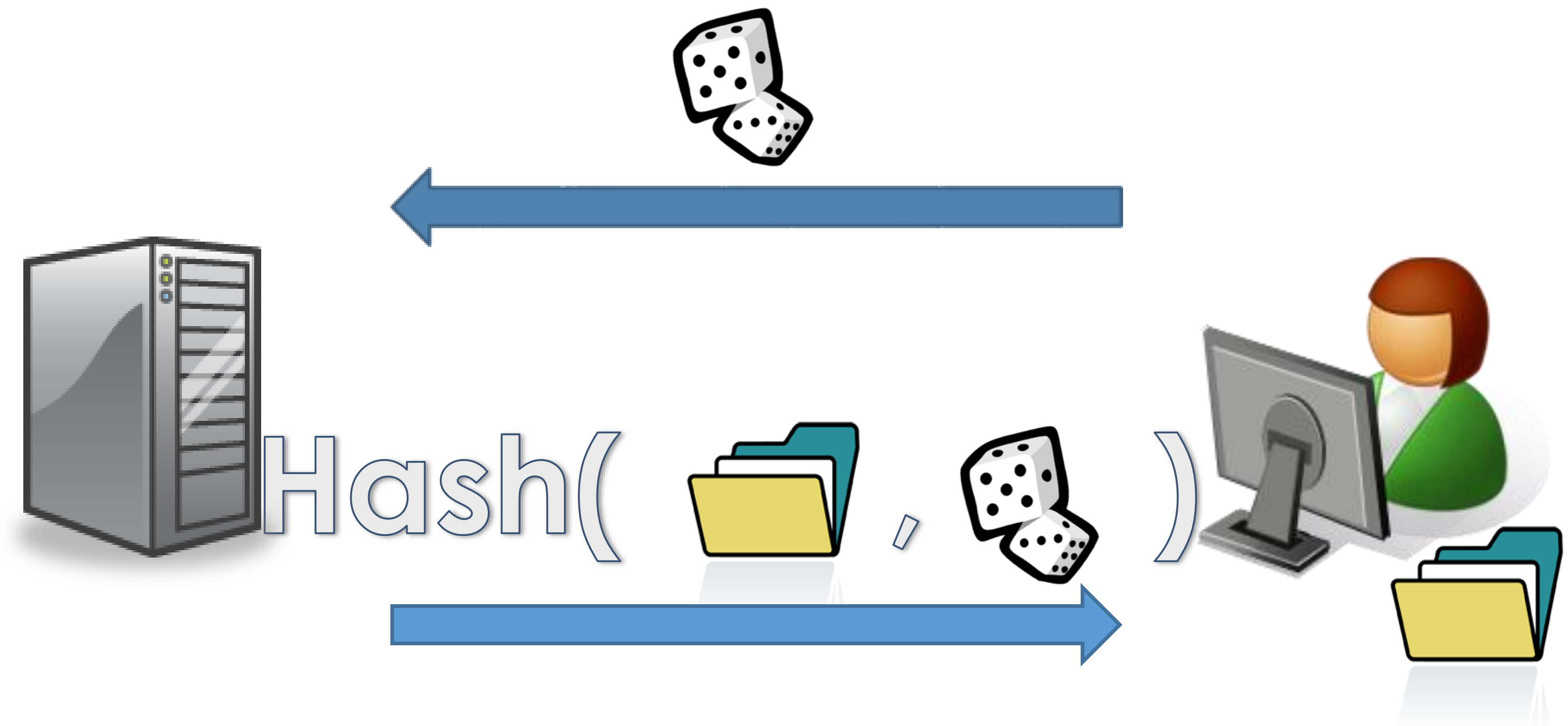
- Signatures are "anonymous" in normal circumstances
  - What does that mean?
- A "trusted" party can "open" a signature if necessary.
  - Opening a signature means revealing its true signer.
- Good enough? Too powerful?
- Any alternative formulation?

Availability

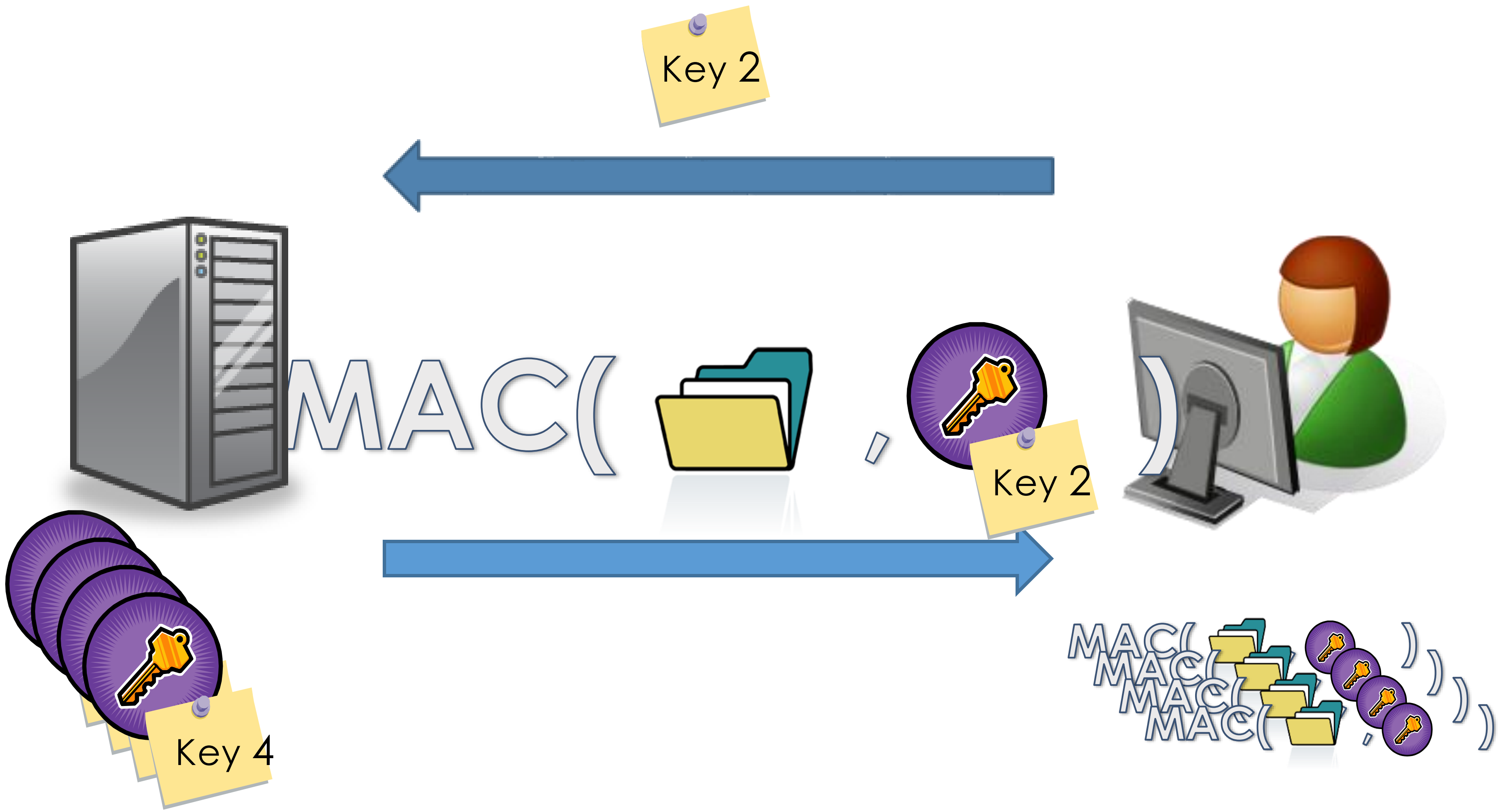
- A system must be serving the info when it is needed.
- How can cryptography help to ensure availability?

- 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”?

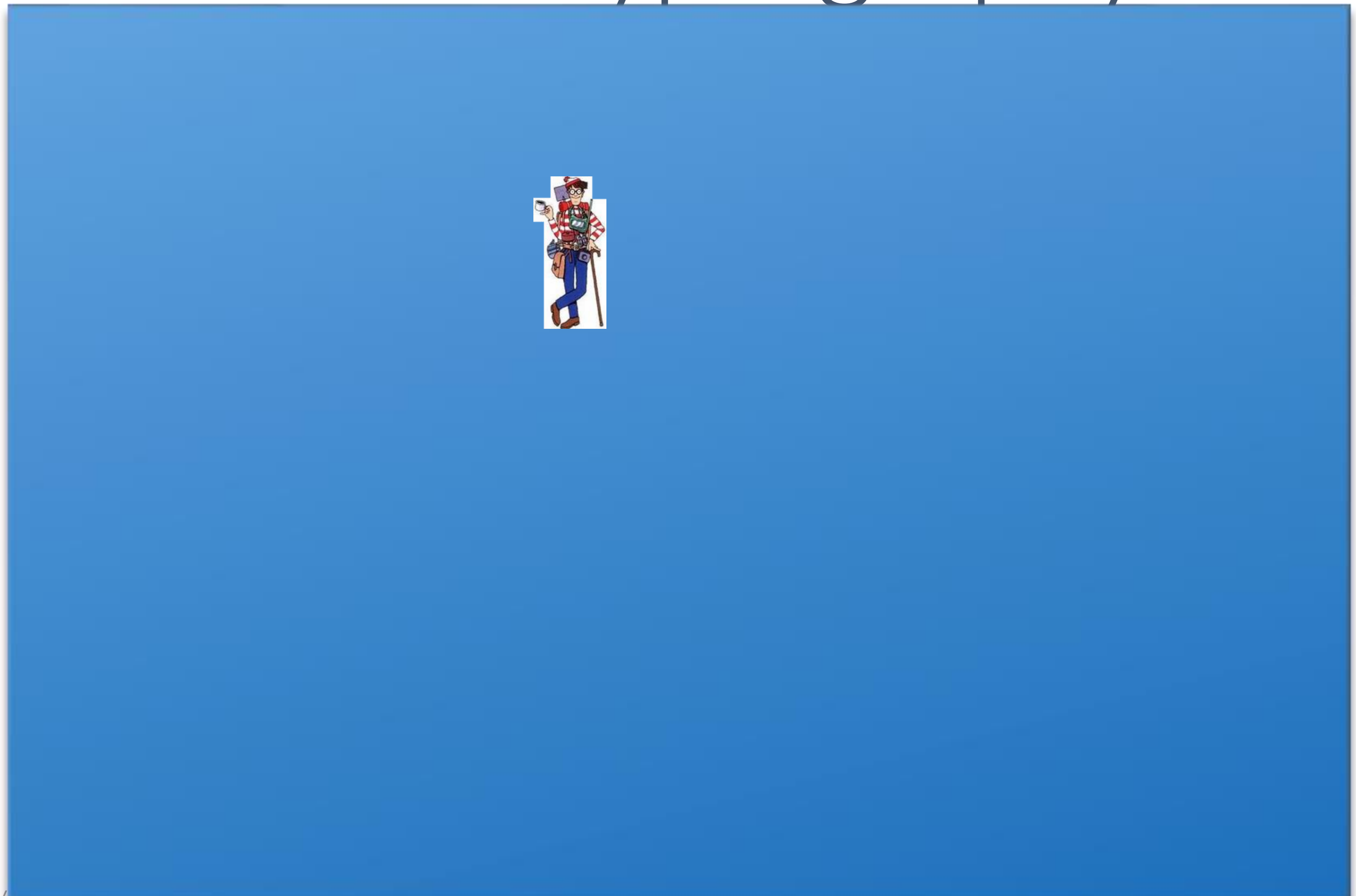
- The storage is outsourced to the cloud.
- Why not outsource the auditing to third-party auditor?

- 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’m richer!

I’m rich!

I have \$x

I have \$y

Is $x > y$?

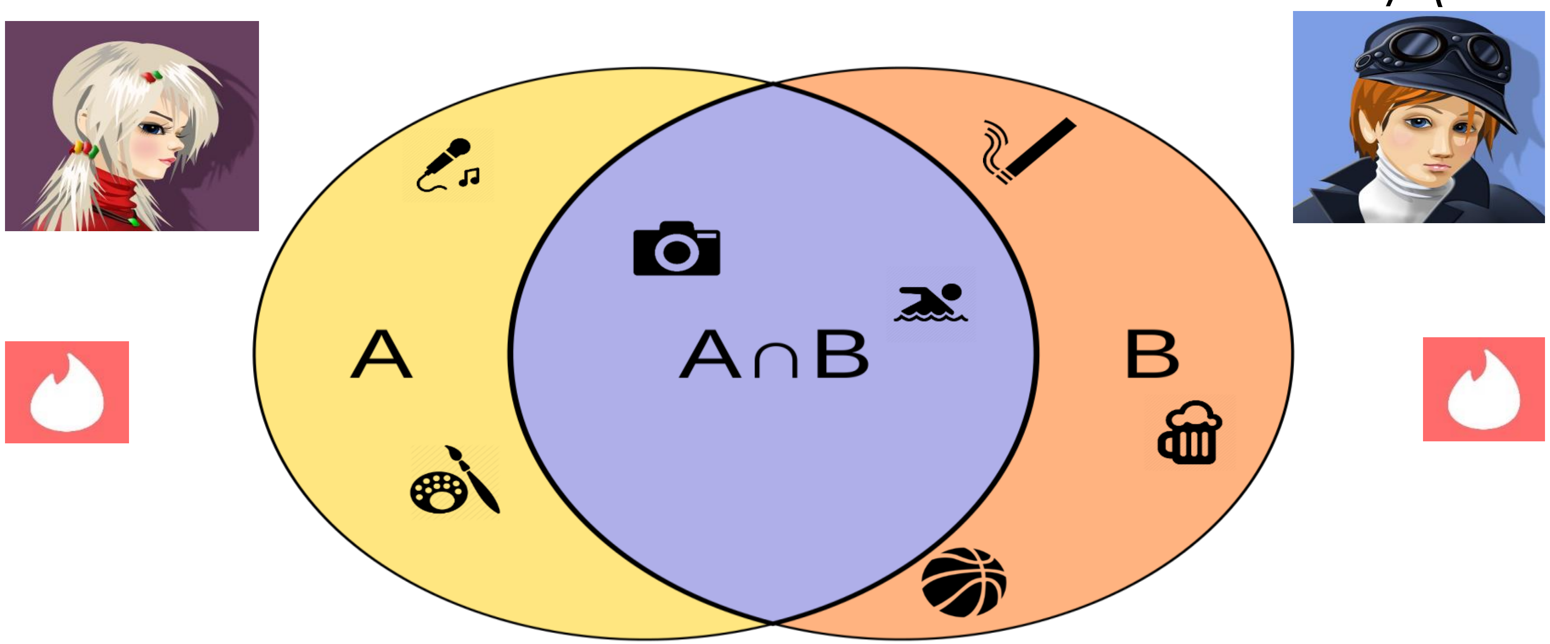
Secure comparison can be applied to, among many,

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

Private Set Intersection (PSI)

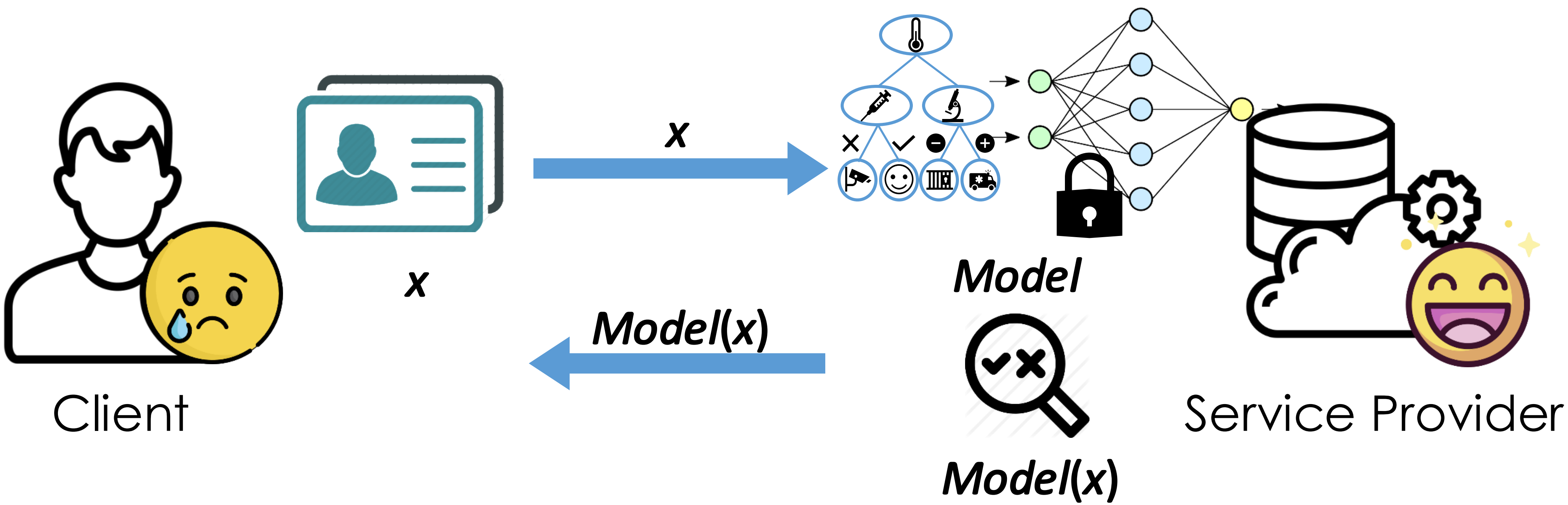
PSI can be applied to, among many,

- 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

- Outsourcing (Verifiable) Computation
- “Secure” Data Analytics / Machine Learning
- Decentralized Anonymous Credentials with Reputation
- Cryptocurrency and its “Privacy-Preserving” version
- Specific Zero-Knowledge Proof (e.g., for matrix circuit)
- Auto Synthesis/Analysis of Cryptographic Schemes
- Lattice-Based Cryptography

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, their input & output behavior / interfaces
  - System model: what entities are involved, 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)

Notation in the Slides
[*]: slightly complicated, slides did not give full details, but it should make sense to you.
[**]: advanced materials, not much details provided, "out-of-syllabus"