*“It is possible to build a cabin with no foundations, but not a lasting building.”*

Eng. Isidor Goldreich (1906-1995)

Cryptography is concerned with the conceptualization, definition, and construction of computing systems that address security concerns. The design of cryptographic systems must be based on firm foundations. This course presents a rigorous and systematic treatment of foundational issues: defining cryptographic tasks and solving new cryptographic problems using existing tools.

The first part of the course focuses on the basic mathematical tools: computational difficulty (one-way functions), pseudo randomness, and zero-knowledge proofs. The emphasis is on the clarification of the fundamental concepts and on demonstrating the feasibility of solving cryptographic problems. This part is based on the book titled "Foundations of Cryptography: Basic Tools" by Oded Goldreich.

In the second part of the course we will look at the basic applications (of the basic tools that we have developed so far) like encryption schemes, signature schemes and general cryptographic protocols. This part of the course will be based on the second volume, titled "Foundations of Cryptography: Basic Applications", of the same series on foundations of cryptography by Oded Goldreich.

## Structure and Prerequisites:

- Clarification of the fundamental concepts is done in a way that is independent of the particulars of some popular number-theoretic examples.
- Concepts are best clarified when presented at an abstract level, decoupled from specific examples.
- Basic knowledge of algorithms (including randomized ones), computability and elementary probability theory.
- Background on computational number is theory is not required. (short appendix appears in the book).

## Details:

- What: CS6285, Topics in Computer Science V1; title “Foundations of Cryptography”.
- When: Lectures every Tuesday, 2-4pm.
- Where: COM1-212.
- No Tutorials, no Labs.

## Books:

1. “Foundations of Cryptography: Basic Tools”. (This is the Volume 1)
   Author: Oded Goldreich. Publication: Cambridge University Press.Year 2001.
2. "Foundations of Cryptography: Basic Applications". (This is the Volume 2)
   Author: Oded Goldreich. Publication: Cambridge University Press.Year 2004.

Useful weblinks:

1. http://www.wisdom.weizmann.ac.il/~oded/foc.html
2. http://www.wisdom.weizmann.ac.il/~oded/foc-book.html
3. Course website: https://www.comp.nus.edu.sg/~rahul/CS6285.html

## Course Outline:

Following is an outline, as suggested by Oded Goldreich,  we will try to follow this schedule. Each lecture below is meant to be of one hour and we will try to club 2 lectures in each two hour class. Lectures 1-15 are covered by Volume 1. Lectures 16-28 are covered by Volume 2.

- Lecture 1: Introduction, Background, etc.

- Lecture 2-5: Computational Difficulty (One-Way Functions)
  Main: Definition (Sec. 2.2), Hard-Core Predicates (Sec. 2.5)
  Optional: Weak implies Strong (Sec. 2.3), and Sec. 2.4.2-2.4.4

- Lecture 6-10: Pseudorandom Generators
  Main: Definitional issues and a construction (Sec. 3.2-3.4)
  Optional: Pseudorandom Functions (Sec. 3.6)

- Lecture 11-15: Zero-Knowledge Proofs
  Main: Some definitions and a construction (Sec. 4.2.1, 4.3.1, 4.4.1-4.4.3)
  Optional: Sec. 4.2.2, 4.3.2, 4.3.3-4.3.4, 4.4.4

- Lecture 16-20: Encryption Schemes
  Definitions and a construction.

- Lecture 21-24: Signature Schemes
  Definitions and a construction.

- Lecture 25-28: General Cryptographic Protocols
  The definitional approach and a general construction (sketches).

## Evaluation:

Continuous assessment 60% (assignments), Final Exam 40%.

Lecture 1

## The Need for a Rigorous Treatment in Cryptography

> *If the truth of a proposition does not follow*
> *from the fact that it is self-evident to us,*
> *then its self-evidence in no way justifies our belief in its truth*.
>
> Ludwig Wittgenstein, *Tractatus logico-phisophicus* (1921)

- Cryptography concerned with constructing schemes robust against malicious attempts.

- Must take into account infinite set of adversarial strategies.

- Adversary can create cheating strategy after knowing cryptographic scheme.

- Cannot assume anything except that Adversary is "computationally bounded".

- Cannot use heuristics and cannot trust our intuitions.

- At this stage of history we do not understand "the nature of efficient computation" very well.

- Track record :
  1. Abandon of papers that derive or jump to wrong conclusions about security.
  2. (1979) Ron Rivest : No signature scheme that was "proven secure assuming the intractability of factoring" could resist "chosen message attack". They made implicit unjustified assumption regarding the nature of a "proof of security".

     (1984) Goldwasser, Micali, Rivest pointed out the fallacy.

- Practical consequences of the Rigorous treatment:
  1. Plausibility results: Establish a connection between two notions. Specific construction may be impractical.
  2. Introduction of paradigms and techniques that may be applicable in practice.
  3. Presentation of schemes that are suitable for practical application.

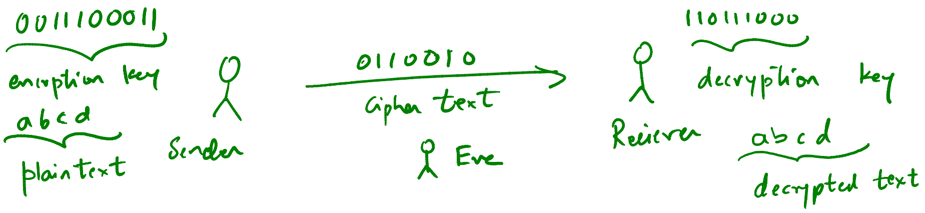
- Overall the approach followed in the book is very "conservative".

"However practice needs much more than a sound theoretical framework, whereas this book makes no attempt to provide anything beyond the latter. For helpful suggestions concerning practice (i.e. applied cryptography), the reader may *critically* consult other texts." - Oded Goldreich.

Lecture 1

## Cryptography: Main topics

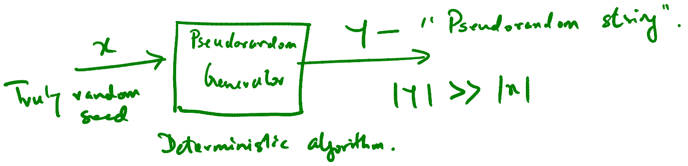
### Encryption Schemes:



1. Information Theoretic Security: Encryption and decryption keys need to be the same and of length equal to plaintext's length.

2. Security based on computational limitations of Eve: Encryption and decryption keys could be much shorter. Encryption key could be known to Eve as well !! (Public-key cryptosystems).

### Pseudorandom Generators:



A main result in the field:

Thm: Pseudorandom generators exist if and only if "one-way functions" exits.

### Digital Signatures:

A *scheme for unforgeable signatures* requires:

i. That each user be able to *efficiently generate his or her own signatures* on documents of his or her own choice.
ii. That each user be able to *efficiently verify* whether or not a given string is a signature of another (specific) user on a specific document, and
iii. That *no one be able to efficiently produce the signatures of other users* to documents that those users did not sign.

### Message Authentication:

A *scheme for message authentication* requires:

i. That each of the communicating parties be able to *efficiently generate an authentication tag* for any message of his or her own choice.
ii. That each of the communicating parties be able to *efficiently verify* whether or not a given string is an authentication tag for a given message, and
iii. That *no external adversary* (i.e. a party other than the communicating parties) *be able to efficiently produce authentication tags* to messages not sent by the communicating parties.

**Note:** In message authentication, **it is not required,** that a third party should be able to verify the validity of the authentication tags produced by designated parties, whereas in digital signatures, it is required that the third party be able to verify the validity of the signatures produced by other users. Hence digital signatures provide a solution to the message authentication problem but not necessarily vice versa.

Fault-Tolerant Protocols:

*"Cryptography is concerned with any problem in which one wishes to limit the effects of dishonest users."*

Simultaneity Protocols e.g Secret Sharing:

i. There are two parties, holding a secret each.
ii. If both are honest, then at end of protocol, each has other's secret.
iii. In any case, the first party will hold second party's secret, if and only if, the second party holds first's party secret.

Secure implementation of Functionalities: A protocol for securely evaluating a specific function must satisfy the following:

i. *Privacy:* No party can "gain information" on the input of the other parties, beyond what is deduced from the value of the function.
*ii.* *Robustness:* No party can "influence" the value of the function, beyond the influence exerted by selecting its own input.

It is sometimes required that these conditions hold with respect to "small" (e.g minority) coalition of parties (instead of single party).

One example is *voting*, in which the function computed in "majority".

Following is a major result in the field.

Thm: Secure evaluation of any function can be done if there is an honest majority (even if the identity of honest parties is not known).

Zero-knowledge Proofs:
Loosely speaking, a *zero-knowledge proof* yields nothing but the validity of the assertion. Zero-knowledge proofs provide a tool for "forcing" parties to

follow a given protocol correctly.

A major tool in cryptography is the following fact:

Thm: Zero-knowledge proof systems exists for all languages in NP (provided that one-way functions exist).

## Some Background from Probability Theory

Random variables, How to read probabilistic statement, notation U_n, X_n, Y_n etc.

Markov's inequality, Chebyshev's Inequality, corollary for Pairwise independent sampling, Chernoff Bound, (epsilon, delta)-approximation, Hoefding Inequality.

## The Computational Model

Turing machines, P, NP, NP-Completeness, belief that "P $\neq$ NP", Probabilistic Polynomial Time, Randomized algorithms, s-t connectivity example.

Randomized Algorithms: Two Points of View.

*Probabilistic Polynomial time Turing machine:* A probabilistic machine that always (i.e. independent of the outcome of its internal coin tosses) halts after a polynomial (in the length of the input) number of steps.

**Remark:** Note that without loss of generality we can assume that on input x, the machine always makes T(|x|) coin tosses, where T(.) is a polynomial.

Comparison of *non-deterministic* v/s probabilistic machines.

Associating efficient computations with BPP:

Thesis: *Efficient computations correspond to computations that can be carried out by probabilistic polynomial time Turing machines.*

Complexity class BPP, discussion of bounded probability*, negligible functions*, negligible functions stay that way when multiplied by a polynomial. Phrase "for all sufficiently large n"

Non-Uniform Polynomial Time

Two equivalent ways of defining *non-uniform polynomial time machine.*

Meta Theorem: *Whatever can be achieved by probabilistic polynomial-time machines can be achieved by non-uniform polynomial time machines.*

It is obviously wrong for example if we consider the task of tossing coins :) but holds for "real theorems".

Class P/poly, alternate definition of P via uniformity,

Thm: BPP contained in P/poly.
Proof: Refer to book.

Non-uniform circuit families, equivalence with non-uniform polynomial time machines.

Interactability Assumptions:

- *Interactable*: Tasks that cannot be performed by probabilistic polynomial time machines.
- The computational approach to cryptography is interesting only if NP is not contained in BPP.
  NP not known to contain BPP.
- Discuss "if <**interactability assumption**> then <**useful consequence**>" .
- "Sufficiently strong" one-way functions imply that BPP=P, but this is not known to hold unconditionally.
- Assuming P not equal NP will **not suffice.** A cryptographic scheme must be unbreakable in "most cases".
- Cannot prove as of now "worst case interactibility" implies "average case interactibility". This will be a breakthrough result.

Oracle Machines : Discuss definition from book.