Advanced Tools from Modern Cryptography

Lecture 0

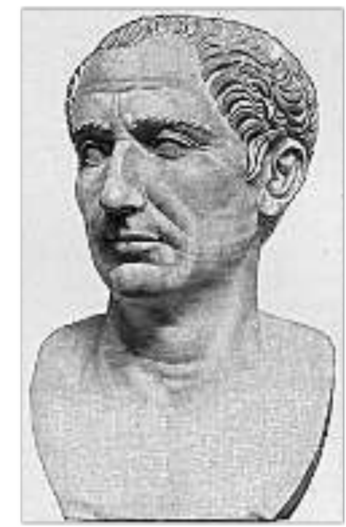
Manoj Prabhakaran

IIT Bombay

“Old” Cryptography

Scytale (ancient Greece)

Caesar Cipher



100 BC - 44 BC

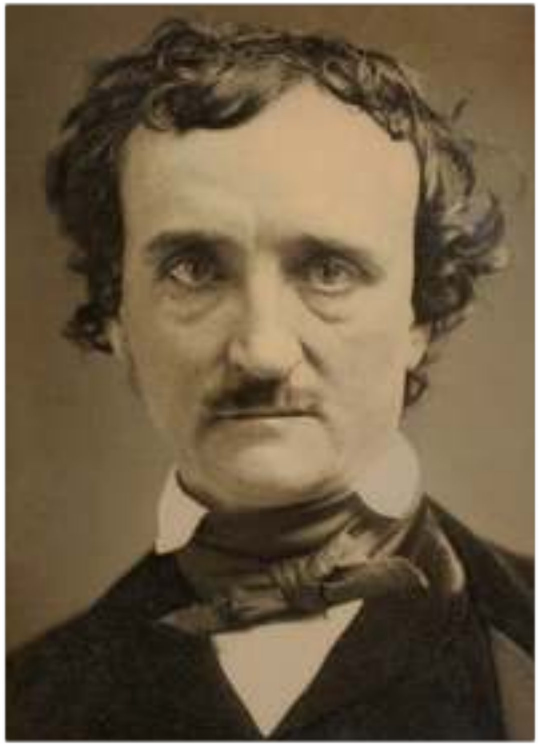
801-873

Cryptanalysis (simple frequency analysis) of Caesar cipher by Al-Kindi

“Old” Cryptography

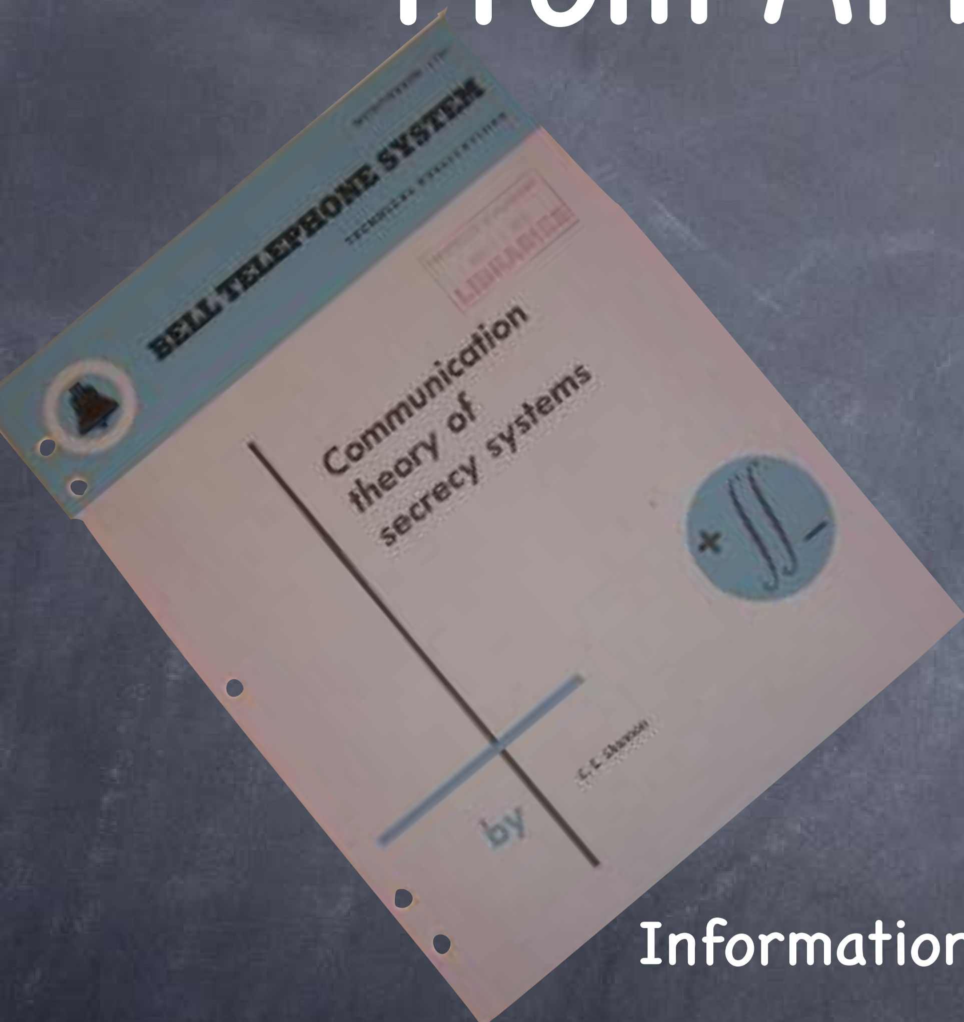
“Human ingenuity cannot concoct a cypher which human ingenuity cannot resolve”

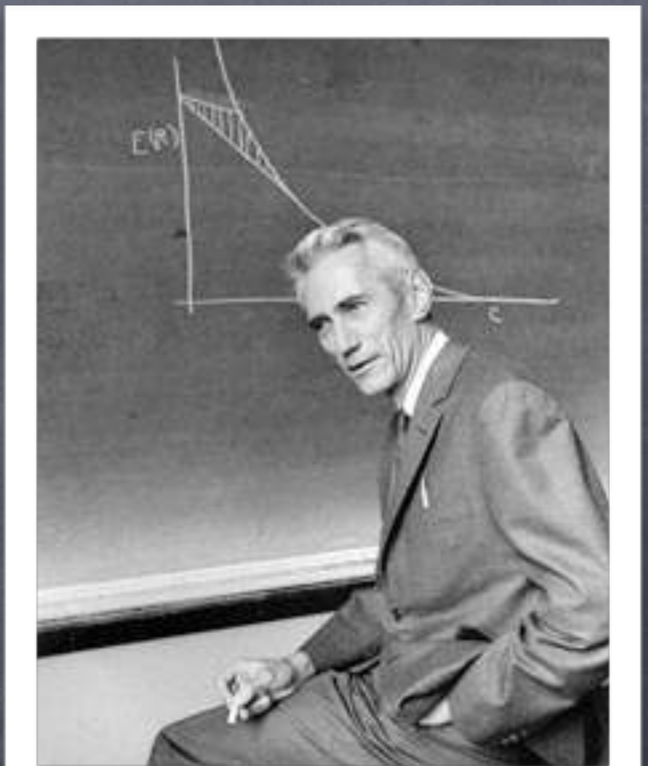
–Edgar Allan Poe



1809-1849

From Art to Science





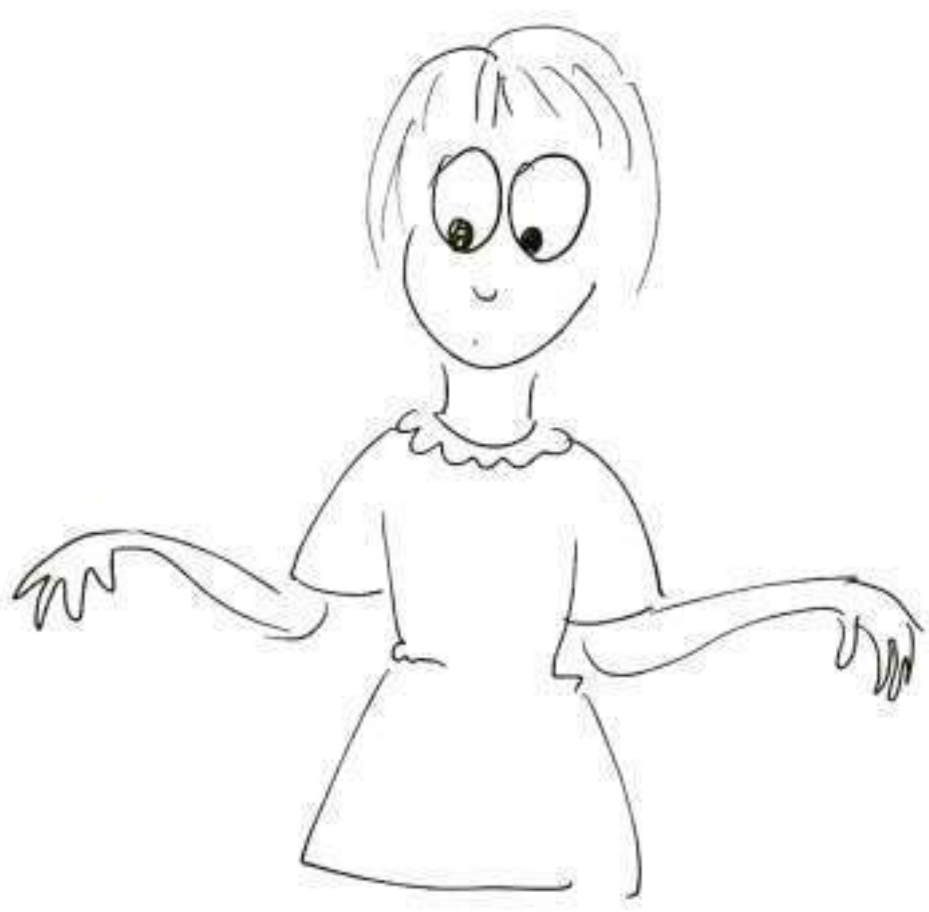
1916 - 2001

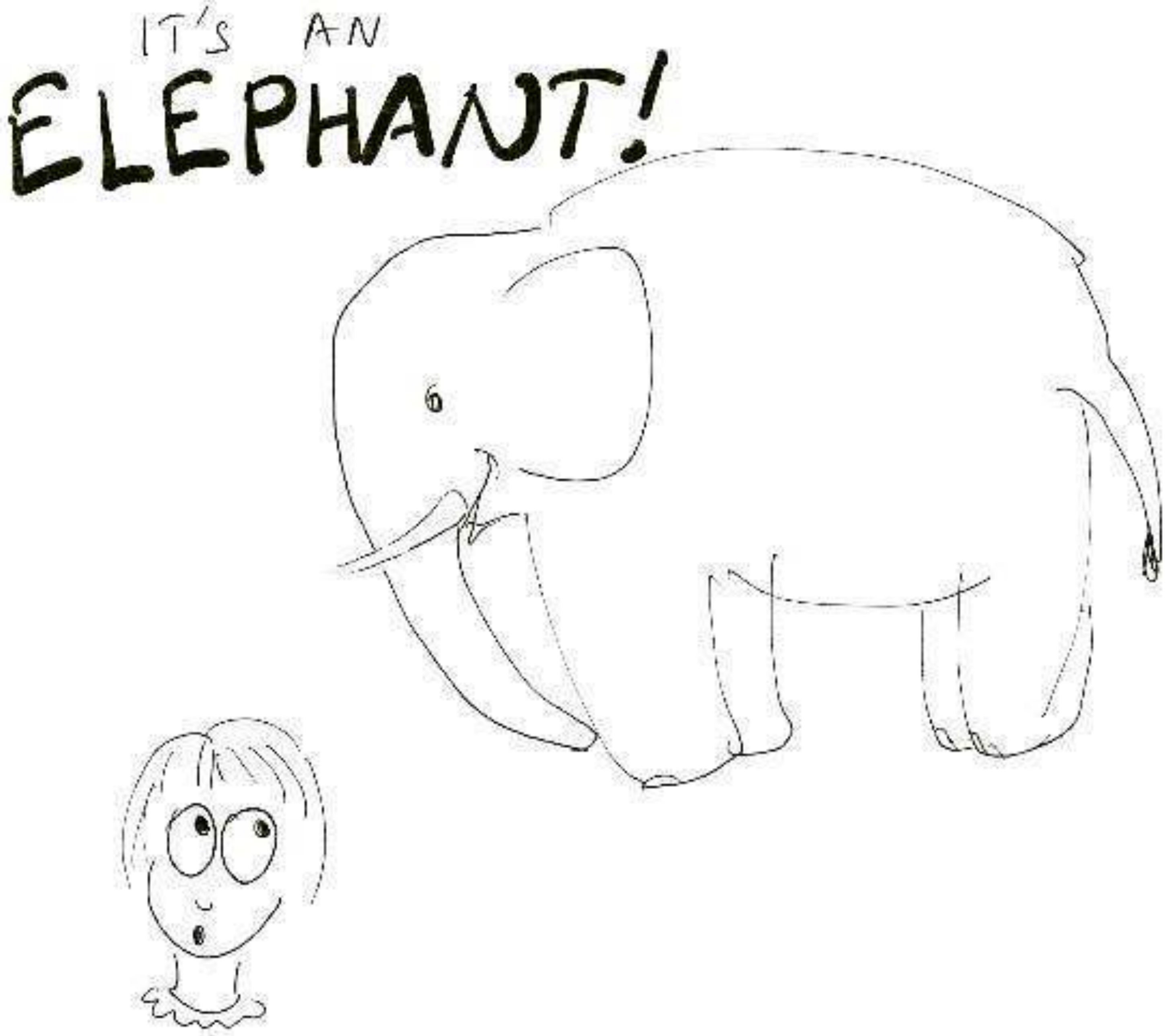
Information can be quantified

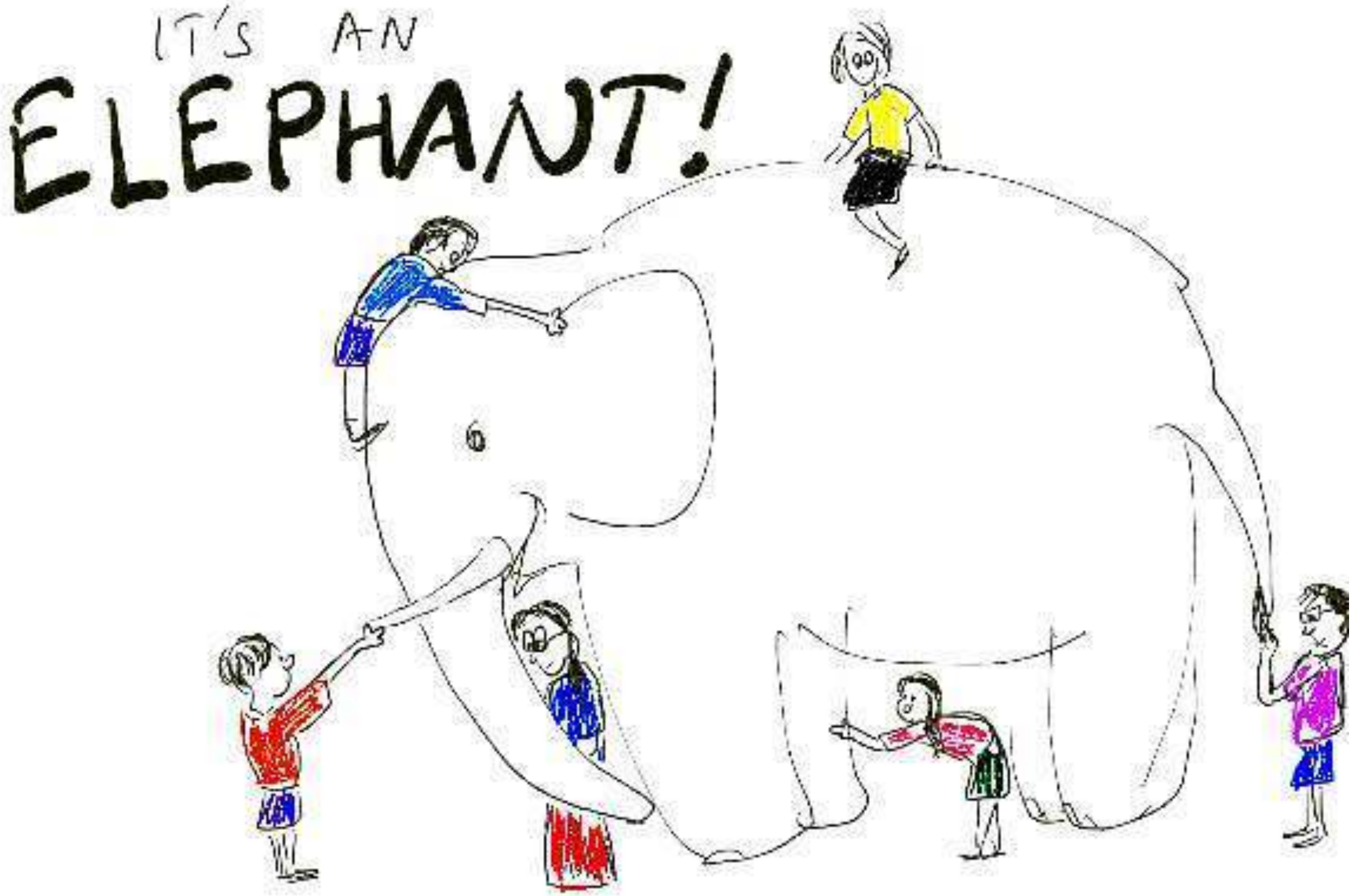
Perfect secrecy: ciphertext has zero information about the message

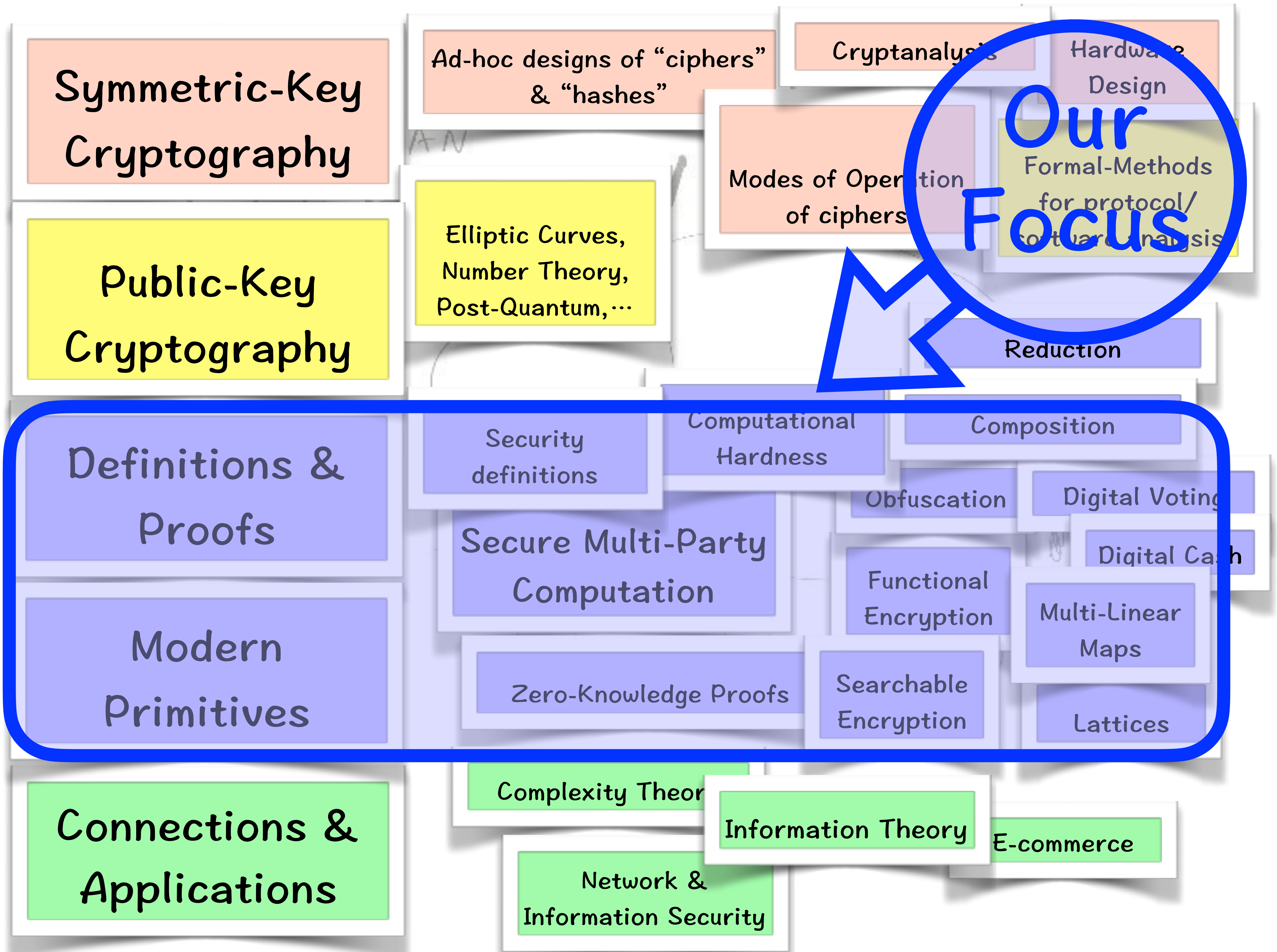
Key to perfect secrecy: Randomness

What is Modern Cryptography?




IT'S AN
ELEPHANT!


IT'S AN
ELEPHANT!


Symmetric-Key Cryptography
Ad-hoc designs of "ciphers" & "hashes"
Cryptanalysis
Hardware Design
Modes of Operation of ciphers
Formal-Methods for protocol/ software analysis
Our Focus
Public-Key Cryptography
Elliptic Curves, Number Theory, Post-Quantum,…
Reduction
Definitions & Proofs
Security definitions
Computational Hardness
Composition
Obfuscation
Digital Voting
Digital Cash
Secure Multi-Party Computation
Functional Encryption
Multi-Linear Maps
Modern Primitives
Zero-Knowledge Proofs
Searchable Encryption
Lattices
Connections & Applications
Complexity Theory
Information Theory
E-commerce
Network & Information Security

Symmetric-Key Cryptography

Public-Key Cryptography

Definitions & Proofs

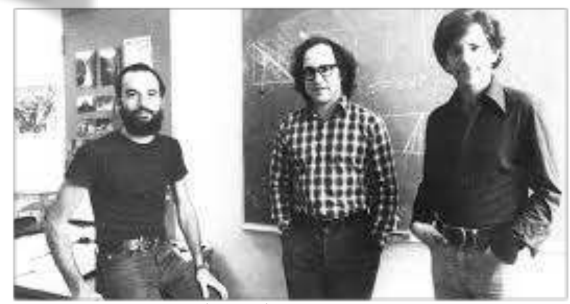
Modern Primitives

Connections & Applications


Claude Shannon


Alan Turing


Merkle, Hellman & Diffie
Turing Award '15


Shamir , Rivest & Adleman
Turing Award '02


Manuel Blum
Turing Award '95


Andrew Yao
Turing Award '00


Goldwasser & Micali
Turing Award '12

Modern Cryptography

- Some tools
  - Secure Multi-Party Computation (MPC)
    - In particular, Zero-Knowledge Proofs
  - Private Information Retrieval (PIR)
  - Fully Homomorphic Encryption (FHE)
  - Functional Encryption (FE)
  - Obfuscation
  - Searchable Encryption
  - Oblivious RAM (ORAM)
  - Leakage-Resilient tools
- Tools for what?

Collaboration

- ... Among mutually distrusting entities
- Secure Multi-Party Computation
  - Example: Company A is shopping for parts for its new product from a supplier, Company B.
  - Example: Auctions, where only the winners' payments need to be revealed
  - Example: Govt. agencies collaborating to enforce laws while respecting the privacy of citizens

Securing Cloud Storage

- **Private Information Retrieval**
  - Don't want the server to see my access pattern
- **Searchable Encryption**
  - Allow search operations on data stored encrypted on the server (OK to reveal the access pattern)
- **Oblivious RAM**
  - Allow read and write operations on data stored on the server, and do not reveal access pattern

Computing on Encrypted Data

- Similar goals as achieved by MPC, but with very restricted interaction among parties (and weaker security guarantees)
- **Fully Homomorphic Encryption**: computing server does not see the data; client need not do the computation, but only encryption/decryption
- **Functional Encryption**: keys can be issued to allow computation of specific functions, with the outcome becoming available to the computing party
- **Obfuscation**: "Encrypted" function that can be run on any input (without needing a key)

Connections

- These are also often tools for building other cryptographic tools
  - e.g., ORAM can be used for MPC
  - e.g., MPC can be used for FE
  - e.g., MPC for leakage resilience
- They share some common underlying primitives
  - e.g., Secret-sharing, Randomized Encoding

Definitions

- Important to be precise about what these (complicated) tools actually guarantee
- Even for a simple tool like encryption, easy to misunderstand its guarantees
  - e.g., malleability, circular (in)security, ...
- Strong security definitions are often provably impossible to achieve for many of these tools
  - e.g., (standard) "universally composable" security for MPC, "virtual black box" security for obfuscation, etc.

Course Plan

- Quick run-through of basic concepts like indistinguishability and basic tools like pseudorandom functions
- Will start with MPC
- As many other topics as possible, as time permits
- Background needed: Mathematical maturity (reading definitions, writing proofs, ...), familiarity with probability, linear algebra, computational complexity

Course Logistics

- Grading:
  - Two Quizzes (60%)
  - ≈3 HW assignments (20%)
  - Course project (20%)
  - > 80% live attendance required!
- "Theory" course: no programming requirement, but your course project could be a programming project
  - We have an MPC programming language now!
- Office hours TBA. Announcements via Moodle or Piazza
- Course webpage: see cse.iitb.ac.in/~mp/teach/