CS 120/CSCI E-177: Introduction to Cryptography

Salil Vadhan and Alon Rosen September 19, 2006

Lecture Notes 1:

What is Cryptography?

Recommended Reading.

- Goldreich, Section 1.1

Cryptography is an ancient art, dating back as far as 500 BC. However, it is only in the past 25-50 years, with the development of computer science, that it has really become a legitimate *science*. This modern approach to cryptography is the subject of this course.

1 Encryption

- The canonical problem of cryptogaphy. Alice wants to send Bob a message privately over an "insecure channel".
- Insecure channel: an adversary, Eve, can listen in. (Later in the course: Eve can even intercept and modify messages.)
  - courier (e.g. ancient military communications)
  - radio broadcast
  - phone line, cellphone transmission
  - Internet connection
  - ...
- Solution: *encryption*. Alice & Bob meet in advance, agree on a "secret code" (an *encryption scheme*).
- Main Questions:
  1. Can we define "security"?
  2. How do we know we've achieved it?

2 Some History

- 500 BC – WWII
  - design$\rightarrow$ break $\rightarrow$ repair $\rightarrow$ break $\rightarrow$ repair $\rightarrow \cdots$
- Shannon (1940's)
  - first rigorous mathematical treatment of cryptography

  - "Perfect security"
  - pessimistic conclusion — not much can be achieved

- Diffie & Hellman (1970's)
  - "Computational security": design cryptosystems that are *infeasible* to break (rather than impossible to break)
  - Base cryptography on computational complexity theory — "hard problems"

- Goldwasser & Micali (1980's)
  - Satisfactory definitions of security.
  - Constructions that *provably* meet these definitions based on widely believed complexity assumptions (similar to $\mathbf{P} \neq \mathbf{NP}$, but a bit stronger).

3 The Expanding Scope of Cryptography

- Public-key cryptography (Diffie & Hellman, RSA, Rabin)
  - Alice and Bob don't need to meet in advance.

- Authentication & Integrity
  - Alice can verify that the message came from Bob, and that no one has modified it.

- Beyond communication: Secure Computation
  - Protocols for Auctions, Information Retrieval, Electronic Cash, Voting, ...
  - Many parties
  - Different functionalities
  - Different security concerns

4 The Layers of Cryptography

1. Hard computational problems
   - The basic building blocks of crypto.
   - These come from complexity theory and computational number theory.
   - Typically require assumptions (because things like $\mathbf{P} \neq \mathbf{NP}$ are still unproven).
2. Cryptographic primitives
   - Achieve basic "cryptographic tasks" that are useful in a wide variety of settings.
   - Examples: encryption, digital signatures, pseudorandom generators, zero-knowledge proofs
   - By now, almost all of these can be precisely defined and provably built from appropriate hard problems (albeit not always efficiently enough for practice).
3. Protocols

- Examples: Auctions, information retrieval, electronic cash, voting, ...
- These are typically much more complex (built using many primitives), yet still are within reach for precise definitions and proofs.

4. Secure Systems & Implementations

   - Hard to capture all "real-life" constraints in mathematical definitions, but evaluating systems becomes much easier built using protocols with well-defined properties.

This course will focus mostly on Items 1 and 2, with maybe an example from Item 3 if there's time at the end. The idea is to go from build cryptographically useful primitives from clear, widely believed complexity assumptions via a sequence of implications such: "There is no polynomial-time algorithm for factoring integers" $\Rightarrow$ "We can construct one-way functions" $\Rightarrow$ "We can construct pseudorandom generators" $\Rightarrow$ "We can construct private-key encryption schemes" $\Rightarrow$ "We can construct secure election protocols"