CS 120/ E-177: Introduction to Cryptography

Salil Vadhan and Alon Rosen Dec. 18, 2006

Lecture Notes 22:

Conclusions

1 What to take away

How to think about cryptographic problems *precisely*.

- Command of basic cryptographic notions — encryption, one-way functions, pseudorandom generators, MACs, etc.
- Defining security
  - Adversary's goal
  - *Probability* of success
  - Adversary's computational resources
  - Adversary's access to system and the communication model
  - Conservative approach
- Constructions
  - Build "complex" cryptographic objects from simpler objects/assumptions.
  - Justify via *reductions*.
  - Always analyze wrt success probability.
  - Stated asymptotically, but can be analyzed concretely
- Some Q's to ask yourself when encountering a new cryptographic protocol:
  - What are we trying to achieve?
  - What are the building blocks? And what are reasonable assumptions about them?
  - Do the assumptions about the building blocks provably imply security of the protocol? If not, are the building blocks at least being used in a way intuitively appropriate to their properties?
- Assumptions we have used
  - complexity assumptions (stronger than $\mathbf{P} \neq \mathbf{NP}$, e.g. one-way functions)
  - adversary's computational resources
  - one protocol running over single communication line, with passive or active adversary in between
  - public keys readily available
  - secret keys truly secret, generating using perfect random bits
  - "party" = "algorithm" = black box mapping inputs to outputs

2 What we didn't cover

- Concurrency and composability
  - Want security when many protocols running concurrently, even under a coordinated attack. ('universal composability')
  - Very active research area
- Key management
  - Key exchange protocols
  - Issues with Public-Key Infrastructure (PKI), Certificate Authorities
  - Human passwords
  - Compromised keys
- Attacks outside the basic models
  - Network security: traffic analysis, denial of service
  - Physical attacks: power analysis, timing analysis, fault analysis
  - Human error
  - Dangerous programs: buggy/insecure code, viruses, worms
- Symbolic analysis of protocols (formal methods)
  - Logic to describe crypto protocols, with idealized model of encryption
  - Can apply automated deduction to analyze these protocols, but does not imply security when implemented with computationally secure primitives
  - Closing this gap is an active research area
- Alternative models
  - Quantum cryptography
  - Bounded-storage model — Rabin's hyperencryption protocol
  - Both allow information-theoretic (statistical) security, no complexity assumptions.
- Social, legal, and policy issues
  - What data *should* be protected? (as opposed to how to protect it)
  - Should digital signatures be legally binding?
  - Should citizens be allowed to use strong cryptography?
  - Tension between privacy/anonymity and security/accountability.

3 What next?

- CS 199r (Computation and Society: Privacy, Technology, and the Law)
  - Mike Smith, Jim Waldo, Alon Rosen, Allan Friedman
  - Spring 2007, TuTh 2:30-4
- More Theory of Cryptography:
  - CS 220r (Cryptography: Trust & Adversity): Graduate-level cryptography course. Starts from first principles, so some amount of overlap with this course, but covers a number of different topics and has a different emphasis.
  - MIT 6.875 (Cryptography and Cryptanalysis): Graduate-level cryptography. Covers almost exactly the same topics as we did, except with a bit more depth and more emphasis on theoretical issues.
  - MIT 6.876J (Advanced Cryptography): Covers recent results and current research directions in cryptography, topics vary from year-to-year. You are probably sufficiently prepared for this course (depending on what they plan to cover), if you are willing to do a little extra reading to fill in any gaps.
  - Further readings
    - Goldreich's books: most comprehensive, most theoretical
    - Lecture notes of Bellare et al.
- Security & Practice of Cryptography
  - Keep a critical eye!
  - MIT 6.857 (Network & Computer Security, Rivest)
  - CS 143 (Computer Networks, Kung)
  - Further Readings:
    - C. Kaufman, R. Perlman, M. Speciner. *Network Security: Private Communication in a Public World.*
    - W. Stallings. *Cryptography and Network Security.*
    - A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography.*
    - B. Schneier. *Applied Cryptography.*
    - D. Stinson. *Cryptography: Theory & Practice.*
- Other areas of theoretical CS highly influenced by cryptography
  - CS 121, 124 (if you haven't taken them yet)
  - Almost all CS 22∗ courses, e.g. CS 224r (Randomness in Computation), CS 225 (Pseudorandomness), CS 221 (Computational Complexity), CS 228 (Computational Learning Theory)
- Number Theory
  - Math 124, and many other courses in the math department.