CS 127/CSCI E-127: Introduction to Cryptography

Prof. Salil Vadhan — Fall 2013

Lecture Notes 21:

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

- Secure Multiparty Computation
  - Many parties compute a joint function of their inputs so that no one learns anything other than result.
  - Can be done for arbitrary poly-time functions (fairly easily) using fully homomorphic encryption, but there are constructions (from the 1980's) based on much weaker assumptions
  - Zero-knowledge proofs, electronic voting, secure auctions, etc. are all special cases
  - NB: does not address which *functions* are safe to compute (the result itself may reveal more than you want)
- 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 (high-rate beacon of random bits, adversary can't store all of it)
  - Both allow information-theoretic (statistical) security, no complexity assumptions.

3 What next?

- Cryptography and Legal/Policy/Social Issues
  - Computer Science 105: Privacy and Technology
  - Center for Research on Computation & Society (`http://crcs.seas.harvard.edu/`).
- More Theory of Cryptography:
  - 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 6.892 (Computing on Encrypted Data): secure two-party and multi-party computation protocols, fully homomorphic encryption, functional encryption and program obfuscation. 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.
  - MIT 6.876J, 6.889, 18.426J (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.
  - MIT 6.S898 Evolution of a Proof: Efficient proof systems in complexity theory, cryptography, and quantum computation.
  - Other chapters of Katz & Lindell.
  - Oded Goldreich. *Foundations of Cryptography (Vols I & II).*
  - MIT Cryptography and Information Security seminar (`http://toc.csail.mit.edu/cis`)
  - Harvard Theory of Computation seminar (`http://toc.seas.harvard.edu/`)
- Security & Practice of Cryptography
  - MIT 6.857 (Network & Computer Security, Rivest)
  - CS 252r (Advanced Topics in Programming Languages): focuses on security in some years
  - 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.*
  - Keep a critical eye!
- Other areas of theoretical CS highly influenced by cryptography
  - CS 121, 124 (if you haven't taken them yet)
  - Many CS 22$*$ courses, e.g. CS 221 (Computational Complexity), CS 225 (Pseudorandomness), CS 228 (Computational Learning Theory), CS 229r (depending on topic — was Mathematical Approaches to Data Privacy in Spring '12)

- Number Theory
  - Math 124, and many other courses in the math department.