CAS CS 538: Cryptography and Cryptanalysis
Fall 2004

1 Administrative Stuff

Official Description (note that the description in the printed bulletin is incorrect)

Basic algorithms to guarantee confidentiality and authenticity of data. Definitions and proofs of security for practical constructions. Topics include: perfectly secure encryption, pseudorandom generators, RSA and ElGamal encryption, Diffie-Hellman key agreement, RSA signatures, secret sharing, block and stream ciphers.

Prerequisites

CAS CS 332 or permission of instructor. I expect you to be comfortable with the notion of reduction from one problem to another, and with such concepts as "polynomial-time algorithm." The course will also require a good comfort level with mathematical proofs and elementary probability theory. If you haven't taken CS 332 or CS 535, please talk to me in the first week of class.

Instructor

Leonid Reyzin, `reyzin@cs.bu.edu`, (617-35)3-3283, MCS (111 Cummington St) room 287
Office hours (for now, may change): Monday 2:30–4:00 Thursday 11:00–12:30

I encourage you to come to my office hours. If you need to talk to me but can't make the office hours, please send me email. I check it a few times on a weekday and at least once on a weekend.

Lectures and Notes

The lecture is in MCS (111 Cummington St) room B33, Tuesdays and Thursdays 9:30 am–11:00 am

I expect you to come to lecture and I encourage you to participate. This class is small enough that we can keep it interactive. There is no good textbook for the class, and lectures are your primary source of information. Be sure to take good notes. On occasion, I will hand out supplementary reading material.

My own lecture notes will be made available on-line after every few lectures. I do *not* guarantee that they will be made available in time for problem set due dates. They cannot substitute for coming to lecture, as they will be dry, condensed and (necessarily) non-interactive.

Textbooks

There is no good textbook for this class. An *optional* textbook that some students found helpful last year is Modern Cryptography: Theory and Practice by Wenbo Mao, available at BU Bookstore (among others). We will not be following it in order; moreover, it does not contain all the material we care about. There will be no required reading from it, but you may find it helpful when you have questions. For background on number theory and algebra, you may find A Computational Introduction to Number Theory and Algebra by Victor Shoup helpful (the entire book is on-line at `http://shoup.net/ntb/`).

In addition to the textbook, there are many other books on cryptography that may be of interest to you. In particular, for those interested in studying theoretical cryptography in more depth, I recommend Foundations of Cryptography: Basic Tools by Oded Goldreich (see `http://www.wisdom.weizmann.ac.il/~oded/foc.html`). For those interested in learning more about the applied side of things, I recommend the Handbook of Applied Cryptography by Alfred Menezes, Paul van Oorschot, and Scott Vanstone (the entire book is on-line at `http://www.cacr.math.uwaterloo.ca/hac/`). Neither book is required for this course.

Finally, there are many books on the history (rather than the science) of cryptography and related fields, for those of you who are interested. I enjoyed Crypto by Steven Levy. Other well-known books (which I have not had a chance to read) include The Codebreakers: The Story of Secret Writing by David Kahn and Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography by Simon Singh.

Other Communication

The class has a home page: `http://www.cs.bu.edu/~reyzin/teaching/f04cs538/index.html`. On occasion, I will send out email to the class list. Please sign up for the list by typing csmail -a cs538 on csa or csb as soon as possible.

Homework and Academic Conduct

There will be (roughly weekly) problem sets, assigned in class. **I encourage you to discuss course material, including problem sets, with other students in the class, subject to the following rules:**

1. You must write up your solutions completely on your own, without looking at other people's write-ups.
2. In your solution to each problem, you must write the names of those with whom you discussed it.
3. You may not consult solution manuals or other people's solutions from similar courses or prior years of this course.

I expect you to follow these rules as well as the academic conduct code of CAS/GRS. If you have any questions or are not sure what is appropriate, consult me *before* taking steps that you are afraid may violate the rules. If you violate the academic conduct code, you will be reported to the Academic Conduct Committee and fail the course.

Late assignments will not ordinarily be accepted. If an assignment is due at the beginning of class, I expect you to hand it in at the *beginning* of class. If, for some compelling reason, you cannot hand in an assignment on time, please contact me as far in advance as possible.

Exams and Grading

There will be a final exam, tentatively scheduled for **Thursday, December 16, 12:30-2:30 pm**. There will be no midterm. The problem sets are worth 70% of the grade. The exam is worth 30%. I reserve the right to deviate from this formula in unusual cases.

If you are unsure of your performance in the class, please come and talk to me. Remember that the last day to drop a class without a 'W' is Tuesday, October 12. The last day to drop a class with a 'W' is Friday, October 29. After that, you must receive a real grade for the course. It is your responsibility to talk to me before these dates if you may need to drop the course to avoid receiving a low grade. I am powerless to change these university-wide rules about drop dates.

2 More interesting stuff

What this course is about

The primary focus of this course will be on *definitions* and *constructions* of various cryptographic objects, such as pseudorandom generators, encryption schemes, digital signature schemes, message authentication codes, block ciphers, and others time permitting. We will try to understand what security properties are desirable in such objects, how to properly define these properties, and how to design objects that satisfy them.

Once we establish a good definition for a particular object, the emphasis will be on constructing examples that *provably* satisfy the definition. Thus, a main prerequisite of this course is mathematical maturity and a certain comfort level with proofs. I will be doing proofs in class, and you will be doing them on the problem sets.

Secondary topics that we will cover only briefly will be current cryptographic practice and the history of cryptography and cryptanalysis.

At the end of this course, you should be able to make sense of a good portion of current cryptography research papers and standards.

What this course is *not* about

This course will not teach you how to make your computer secure. Cryptography is only one tool in computer security. The rest of computer security has to deal with such fascinating things as buggy code, poorly managed and ever-too-curious humans, power consumption of smart cards, electromagnetic emissions from monitors, etc. We will mostly abstract all that away. I will, however, point out where the limitations of our models are and what else is needed for actual security.

This course will also not teach you how to implement the techniques we discuss in the most efficient manner. We will stop at cryptographic algorithms. The underlying number-theoretic algorithms will be discussed only briefly; the most advanced and efficient ones require more time to learn than we will have. For example, if you take only this class, you should be able to program RSA, but many existing implementations will probably be much more efficient that yours.

Finally, this course will not teach you how to design the next great block cipher, such as DES or AES, or the next cryptographic hash function, such as SHA-1. (There are very interesting techniques people use for that, but, unfortunately, our current understanding of these techniques does not allow us to prove any security properties of the resulting constructions.) Nor will this course teach you how to "break" such designs.

Just because I will not teach these topics does not mean they are not worth your while. There are plenty of books and research papers to read and people to talk to if you are interested in pursuing any of these topics.

Topics to be Covered

Below is a rather ambitious plan for the topics I intend to cover, week by week. We will probably have to omit some as we go along. If we have to omit anything, we'll try to omit the extra constructions and keep the definitions: constructions tend to evolve all the time, while definitions are here to stay.

Week	Topics
1	Perfect security: Shannon's lowerbound and the Vernam cipher (a.k.a. one-time pad); review of relevant probability theory
2	Pseudorandom generators (a.k.a. stream ciphers): definition, discrete log problem, and Blum-Micali construction
3	Indistinguishability-based definition and composability theorem for pseudorandom generators
4	Integer factorization, Chinese remainder theorem, and Blum-Blum-Shub pseudorandom generator
5	Intuition and first examples of public-key encryption: RSA, Rabin. Definition of security.
6	Encrypting long messages with RSA, Blum-Goldwasser and PKCS #1
7	Brief history. Diffie-Hellman key agreement, decisional Diffie-Hellman assumption, and ElGamal encryption
8	Introduction to one-way and trapdoor functions, hardcore bits, Goldreich-Levin construction; Shamir's secret sharing
9	Signature schemes: intuition, formal definitions, one-time Lamport schemes
10	Signature Schemes: Merkle trees; full-domain hash RSA and Rabin
11	Certificates, public-key infrastructure, brief look at simple protocols
12	Symmetric techniques: examples (RC4, DES, AES). Pseudorandom functions (Goldreich-Goldwasser-Micali) and permutations (Luby-Rackoff).
13	Symmetric encryption and block cipher modes of operation
14	Message authentication codes and conclusions

In the unlikely event we have time left, we may consider more efficient signature schemes, encryption secure against active adversaries, or interactive protocols. I will always try to provide pointers to more advanced topics as we go along.