YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CPSC 467b: Cryptography and Computer Security *Handout #1*
Professor M. J. Fischer *January 11, 2005*

Syllabus (Spring 2005)

1 Official Yale catalog listing

CPSC 467 01 (20794) / CPSC 567 01 (23313)

Cryptography & ComputerSecurity

Michael Fischer

TTh 4.00–5.15 AKW 200

A survey of such private and public key cryptographic techniques as DES, RSA, and zero-knowledge proofs, and their application to problems of maintaining privacy and security in computer networks. Focus on technology, with consideration of such societal issues as balancing individual privacy concerns against the needs of law enforcement, vulnerability of societal institutions to electronic attack, export regulations and international competitiveness, and development of secure information systems. *Some programming may be required. After CPSC 202a and 223b.*

Final exam scheduled (Group 27) 05/07/2005 S 2.00

Distributional Group IV

Not CR/D/F

2 Course Description

As the title suggests, this course is about cryptography and computer security. Perhaps a more apt title would be "Cryptography and Information Security." Cryptography, because it is a fascinating field of study that is also a key technology for achieving security objectives, and security because of its central importance to our emerging "information society." Security policies will shape the kind of society we live in, and cryptography will have a major impact on the kinds of security policies that are achievable.

Information security, broadly defined, involves controlling the dissemination of information. It includes issues of privacy, data integrity, authenticity, and authority. Privacy refers to preventing information flow to unintended recipients. Data integrity properties insure that information is correct and undamaged. Authenticity identifies information with a source. Authority describes what actions are permitted by whom. Because of the ease with which information can be copied and transmitted, traditional physical means of control are of limited efficacy. Cryptography gives a way to build logical controls on the flow of information that are largely independent of the physical properties of the devices used to transmit and store information.

Cryptography lies at the center of this course, but we will be approaching the subject broadly. On the one end, we'll look at problems of information security and see how cryptographic tools can be used to solve them. We'll also touch on some social issues surrounding the use of cryptography. At the other end, we'll explore the mathematical structures from which cryptographic primitives are built.

Security properties cannot be verified through testing since there is no way to test all possible attacks. Instead, they must be verified analytically through security modeling. This means establishing mathematical models in which security properties can be formally stated and proved.

This course will spend roughly equal time on security protocols, cryptographic primitives, and cryptographic modeling. Rather than attempt to study each of these topics separately, we will look at a number of natural and well-motivated security problems and discuss each in depth. In the process we will introduce the cryptographic primitives and develop the mathematical tools needed to analyze them.

In greater detail, we will be looking at the following topics, as time permits:

- *Security problems and systems.* Privacy and security in the information age. Multiparty problems: Secret message transmission, authentication, key distribution, key escrow, digital signatures, certified mail, contract-signing, coin flipping, and so forth. Practical systems: Kerberos, SSH, PGP, smart cards, SSL.

- *Cryptographic primitives and specific realizations.* Primitives: Transposition and substitution ciphers, private and public key encryption systems, digital signatures, message digests, key exchange, secret sharing, pseudorandom number generation. Realizations: AES, DES, RSA, RC6, DSA, SHA, MD5.

- *Mathematics for cryptography.* Information theory. Number theory: Prime numbers and factoring, modular arithmetic, computations in finite fields, discrete logarithms. Complexity theory. Theory of distributed systems.

3 Course materials

Required textbook: Wenbo Mao, *Modern Cryptography: Theory & Practice*, Prentice-Hall, 2004, ISBN 0–13–066943–1.

Website: I maintain a course website at http://zoo.cs.yale.edu/classes/cs467/2005s/index.html. You should bookmark it in your browser and visit it often. It will grow as the term progresses and will contain announcements, handouts, lecture notes, revisions to homework assignments, programming hints, and links to documents in the course directory and elsewhere on the web. *Access is restricted to machines on the Yale network.* For off-campus use, you will need to follow the instructions for use of the Yale remote authentication proxy server.

4 Course Mechanics

Prerequisites: This course will be taught at an advanced undergraduate/graduate level and assumes a basic computer science background. Some C programming will be required. CPSC 202a and 223b are prerequisites. Graduate students should have an equivalent background.

Requirements: Course requirements include written problem sets and programming assignments, two hour exams, and a final exam. Graduate students taking the course will be expected to perform at a higher level than undergraduates and may be required to do additional work.

Grading: The final grade in the course will be based on your performance on the programming assignments and other homework, the two hour exams, and the final exam. Assignments, hour exams, and final exam will each count for between 30% and 40

Assignments and other announcements: Written problem sets and programming assignments will posted on the handouts page of the course website from time to time during the course. Other course announcements will be posted on the course home page. It is your responsibility to check these pages frequently.

Email: I am always available for email consultation at fischer-michael@cs.yale.edu. I can't always promise to respond right away, but I can often be reached by email when I am away from the office. Email is also the preferred way to arrange an appointment with me.

5 Policies

Late Policy: Late work will be accepted at the discretion of the instructor and/or TA and will generally be subject to a penalty unless accompanied by a Dean's excuse. Work will not be accepted after graded papers have been returned or solutions released. However, alternative means for making up missed work may be arranged on an individual basis with a Dean's excuse.

Please contact the instructor or TA as soon as you find out that you are unable to submit work on time or to attend a scheduled exam so that suitable makeup arrangements can be made.

Policy on Working Together: Work turned in under your name must be your own work. Plagiarism is unethical and will not be tolerated. You may neither copy from others nor permit your own work to be copied. Therefore, it is important that you keep your files protected so that others cannot read them and that you carefully guard your password. If you think your password may have been compromised, you should change it immediately.

You may of course discuss the lectures and readings with your classmates in order to improve your understanding of the subject. However, all written work must be your own. You are also always free (and encouraged) to come in and ask the TA or instructor for help about anything concerning the course. Please talk to me if you have any questions about this policy.

Policy on Computer Problems: The Yale College policy on "Use of Computers and Postponement of Work" in the Yale College Programs of Study applies to this course. It is reproduced below.

> "Problems that may arise from the use of computers, software, and printers normally are not considered legitimate reasons for the postponement of work. A student who uses computers is responsible for operating them properly and completing work on time. (It is expected that a student will exercise reasonable prudence to safeguard materials, including saving data on removable disks at frequent intervals and making duplicate copies of work files.) Any computer work should be completed well in advance of the deadline in order to avoid last-minute technical problems as well as delays caused by heavy demand on shared computer resources in Yale College."

Particularly relevant for this course are the cautions against leaving a programming assignment to the last minute when machines might be busy, printers broken, and so forth, and about safeguarding your data.

6 Computing Facilities

The Zoo: This course will be using the Computer Science Department's educational computing facility, otherwise known as the Zoo. This facility contains Pentium-based PC's running Linux. You will need to learn how to use these machines if you don't already know how in order to read email, browse course-related web pages, edit and compile C programs, and access the course directory. Look at

`http://zoo.cs.yale.edu/help/`

for information on getting started if you are new to the Zoo.

Your course account: You should request a course account for this course *even if you already have a Zoo account*, for otherwise you will be unable to submit work electronically. To obtain your account, go to

`http://zoo.cs.yale.edu/help/accessing-zoo.shtml`

and follow the instructions there.

Course directory: The shared course directory, `/c/cs467`, is located on the Zoo server. You can access it from your Zoo course account. It will contain software that you will be using for this course and miscellaneous documentation and files. It will also contain the software that you will use when submitting assignments electronically. Public files there can be accessed via the web as well as from a Zoo node. Your class account home directories will also be located there.