An experience teaching a graduate course in cryptography

Aviel D. Rubin *

rubin@bellcore.com
Bellcore
445 South St.
Morristown, NJ 07960

Abstract

This article describes an experience of teaching a graduate-level course in cryptography and computer security at New York University. The course content as well as lessons learned and plans for the future are discussed.

1 Introduction

This paper describes a course titled "Cryptography and Computer Security" that was taught at New York University in the Fall of 1995. The department head at NYU requested a course for practitioners, with an emphasis on applications and real-world problems. Thus, there were four phases to the course, classical cryptography, conventional cipher systems, applications of cryptography, and number theory. Grading was based on five homework sets and a semester project. The course used Bruce Schneier's book, *Applied Cryptography* [38] as the primary text, which was supplemented by a course pack of selected publications. In addition, materials were used from the following books: Doug Stinson's, *Cryptography: Theory and Practice* [41], Dorothy Denning's *Cryptography and Data Security* [13], Garfinkel and Spafford's *Practical Unix Security* [16], Kaufman, Perlman and Speciner's *Network Security* [24], and William Stalling's *Network and Internetwork Security Principles and Practice* [39].

The cryptography and computer security course was offered in the Courant Institute of Mathematical Sciences (`http://cims.nyu.edu/`) on Tuesday evenings from 5-7, and consisted entirely of graduate students. Some of them had full-time jobs during the day, and were taking courses at night. There were 20 registered students, about 10 auditors who showed up regularly, and a teaching assistant.

One idea for the class that came from Stuart Haber, who teaches a cryptography course at Columbia, was somewhat successful. Each lecture was assigned at least one student scribe. The scribe was responsible for taking careful notes and producing a write-up of the lecture, which was then shared with the class. The students received a homework grade on their write-up. The write-ups were of varying qualities, but the feedback from the students was that they were useful. Three samples can be found in `ftp://thumper.bellcore.com/pub/rubin/course/`.

Whenever possible, current events were discussed as they occurred. There were several headlines in the New York Times during the semester, and these were presented and discussed at the start of the following lecture. For example, articles appeared about the Berkeley group who successfully attacked NFS [8]; the Princeton group who found flaws in the random number generator of an early version of

*Author web page: ftp://thumper.bellcore.com/pub/rubin/rubin.html

Netscape [11]; and Paul Kocher's timing attack [25] on RSA (which was published the week after RSA was taught).

2 Classical cryptography

Classical cryptography includes ciphers that are of historical significance. They are of little or no practical use for today's systems. However, there is some insight into how they were designed and attacked that is useful for looking at modern systems.

2.1 Material covered

Students were introduced to some simple ciphers such as substitution, affine, Hill [20], etc. Then, the progression to polyalphabetic ciphers and finally Vernam ciphers was explored. At this time, the notions of known plaintext, chosen plaintext, chosen ciphertext, etc. were introduced. Various techniques, such as index of coincidence and the Kasiski method [23] for cryptanalyzing classical ciphers, were discussed and demonstrated. Several examples from David Kahn's *Codebreaks* [22] were also used.

The next subtopic under classical ciphers was Shannon's information theory, which revolutionized cryptanalysis. Students were taught how to measure key equivocation, unicity distance, and other properties of cryptosystems.

2.2 Homework assigned

Some homework problems involved cryptanalyzing ciphertext using the techniques taught in class. In addition, there were some thought-provoking problems. For example, one problem asked students to discuss the strengths and weaknesses of the one-time pad.

2.3 How students responded

Most students seemed to find this part of the course interesting. Unfortunately, the historic significance of classical cryptography seemed to be lost on several students, who were more interested in creating secure applications than understanding the evolution of cryptography. However, this section is an excellent place to provide students with an insight into cryptography. For example, the notion of chosen ciphertext vs. known plaintext is easier to understand for cryptanalyzing a classical cipher than for breaking DES. The classical cryptography section will be repeated when this course is offered again.

3 Conventional cipher systems

This part of the course focused on the symmetric cipher systems, their use, implementation, and analysis.

3.1 Material covered

The first system covered was the Data Encryption Standard, DES [30]. The algorithm was discussed in depth, including the individual rounds, the various permutations and substitutions, and what little is known about the design of the S/Boxes. The course touched on differential [5] and linear [29] cryptanalysis, exhaustive key search, DES breaking machines [43], and various results from the literature on attacking DES. In addition, hardware and software implementations were compared. Finally, the class learned about key lengths, security and export issues.

After the design and analysis of DES was taught, students learned how to use this function. Different modes of operation (e.g. ECB, CBC, etc) were discussed, as well as performance issues and block ciphers versus stream ciphers.

In addition to the in-depth exploration of DES, the class learned about other symmetric ciphers, such as IDEA [27] and RC4 [34]. During this part of the course, the notion of key escrow (government and commercial) was discussed. The class covered clipper [12], skipjack [9], and TIS's Key Recovery Center [42], along with the potential effect of escrow on allowable key lengths for export. Then, lecture time was devoted to the effects of combining different algorithms.

Another important topic that was covered was hash functions. MD5 [32] and SHA [1] were given as good examples of collision free one-way functions. Various uses of hash functions were discussed. The time-stamping system of Haber and Stornetta [18] is one example of an interesting application of hash functions. Finally, various message authentication codes (MAC's) were discussed.

3.2 Homework assigned

Homework for this section included numerous readings on the material and a problem set. One of the homework problems required students to describe an application for each mode of operation of block ciphers covered in class, and to explain why this mode of operation is best suited for the particular application. Another problem involved producing the verification algorithm of a MAC, given the algorithm for generating it.

3.3 How students responded

The conventional cipher system generated a lot of interest from the students. One Ph.D. student sent e-mail saying that he had decided that this is going to be his area of research for his thesis. An interesting conversation took place one day about the issue of government key escrow. The students were especially impressed with the time stamping system - the idea that an undisputable time stamp can be produced, given only a cryptographic one-way hash function, without trusting anyone.

4 Applications of cryptography

4.1 Material covered

This section of the course dealt with security issues in actual systems. Unfortunately, there was not enough time in one course to cover everything, so only several systems were analyzed in detail, and the rest were briefly studied. Security at different network layers, especially the IETF work on transport and network layer security, received considerable attention. In particular, the class studied IP-layer encapsulation and key management [21]. One of the groups chose to do their semester project on this topic. The students were introduced to PGP [44], which they were required to understand and use. Tradeoffs between PGP and PEM [2] were discussed, along with issues about the web of trust versus a strict hierarchy of certificates. Message formats for both systems (PGP and X.509) were covered. The Betsi system [35], which uses PGP, and provides integrity assurances to users who download software from the Internet, was also mentioned in class.

Considerable time was spent on authentication systems. Among the systems covered in class were one-time password schemes [19, 37], Kerberos [40], Khat [36], and Kryptoknight [6]. In addition, secure file systems, such as CFS [7] and SFS [17], were covered. Next, the class was introduced to the concept of evaluation criteria, in particular, the orange book [14] definitions were given.

Key agreement protocols, such as Diffie-Hellman [15], were also covered. Man in the middle attacks, and an authenticated version of the protocol were discussed. Several performance and implementation issues wrapped up the conversation on this topic.

As part of the computer security section of the course, Anish Bhimani of Bellcore gave a guest lecture on firewalls [4]. The class was taught the purpose of firewalls, how they are built and maintained, and their limitations.

Two topics of practical significance were password choosing and Unix®security. Several studies that demonstrate how poorly users choose passwords were discussed. Mechanisms for checking a system for bad passwords, as well as preventing them were also covered. There is a definite trade-off between convenience and security when it comes to user passwords. Next, Unix security was studied. Some classical attacks were presented, as well as common sense system administration. This section concluded with a lecture on computer viruses on three different platforms, Macintosh, PC, and Unix.

Finally, Jack Lacy of Bell Labs came to talk about his Cryptolib package [26]. It is a library of extremely fast cryptographic routines for almost any platform. The software was released to the students, most of whom used it for their projects. The library proved to be extremely useful.

4.2 Homework assigned

There were several homework assignments in this section. One assignment involved using PGP. The students were given an assignment that was encrypted with the private key of the instructor. The only way for them to legitimately understand the assignment was to use PGP, obtain the instructor's public key, and decrypt the assignment. This particular assignment dealt with public key infrastructure. Among other things, students were required to have their key signed by some other student in the class. One student posted his key to the class mailing list and asked people to sign it. It was quickly pointed out that he had misunderstood the concept of the web of trust.

Another homework assignment required students to compare and contrast security at the following levels: application, transport, network and link. Tradeoffs involved performance, security, user-friendliness and ease of use.

4.3 How students responded

This section was very popular. Many students' project ideas came from the material that was covered here. There were several who pursued further reading on their own and seemed highly motivated to learn about topics that were not covered because of time restrictions. This was especially true of electronic commerce, which was only briefly mentioned in the lectures. However, a handful of students were more eagerly awaiting the number theory section which followed.

5 Number Theory

5.1 Material covered

The number theory section of the course began with elementary number theory and modular arithmetic. Then, several algorithms, such as square and multiply for fast exponentiation, were given. The Diffie-Hellman problem was visited again in this section in more detail. Next, multiplicative inverses were covered, along with many theorems related to their properties. Later, Euler's totient function, $\phi(n)$,

---

Unix is a registered trademark of Unix Systems Laboratories.

was introduced. This led to Fermat's theorem that given a prime p and an a such that $gcd(a,p)=1$, then

$$a^{p-1}\ mod\ p = 1$$

and Euler's generalization that for every a and n such that $gcd(a,n)=1$,

$$a^{\phi(n)}\ mod\ n = 1$$

Several results were proven in class, and others were given as homework assignments.

Other algorithms, such as Euclid's extended algorithm for finding inverses and the Chinese remainder theorem for solving simultaneous systems of equations were covered in class and in the homework assignments. The next section in the number theory lectures dealt with Galois fields. GF(2^n) and GF(p) were discussed and compared.

After Galois fields were taught, the students were presented exponentiation ciphers, such as the Pohlig-Hellman scheme [31], which can be used as a conventional cipher. The security of this scheme rests on the difficulty of the discrete log problem, which was the next topic in the class.

After Pohlig-Hellman, the RSA [33] scheme was introduced. Some lecture time was devoted to giving examples of using RSA, and discussing various attacks on it, including the current status of factoring. Several topics related to implementing RSA were covered next. Various techniques for obtaining large primes with high probability were discussed. The class was taught about Legendre and Jacobi symbols, and several number theory results that use them. Primality tests such as the Solovay-Strassen and the Miller-Rabin tests were covered, as well as methods for combining the techniques.

Once the students understood how RSA worked, an interesting application (developed by the same authors), mental poker, was presented. In this scheme, users who share a common modulus can play poker using the commutative property of RSA encryption. However, by observing which cards were quadratic residues and which were not, a cheating party can obtain a bit of information about each card and cheat [28]. The lesson here is that cryptographic algorithms cannot be used as black boxes. A developer must consider the environment in which the algorithm will run, as well as subtleties of the algorithms themselves.

The last part of the number theory section was given by a guest lecturer, Stuart Haber. He taught the class about zero-knowledge proofs using number theory.

5.2 Homework assigned

The homework in this section was the most challenging for the students. The tasks they were given included proving various theorems and applying the algorithms taught in class to several problems. For example, the students had to compute inverses by hand using Euclid's extended algorithm. Another problem consisted of solving a set of equations using the Chinese remainder theorem.

5.3 How students responded

There were two general responses to the number theory section. A small group of students was somewhat overwhelmed by this section and did not have the mathematical background (nor interest) to follow the material very well. A more substantial group of students felt that this was the one of the most interesting parts of the course and wished that more material had been offered. Therefore, there will be a greater focus on number theory and its application to cryptography in the second course.

6 Student projects

One important element of the course was the student projects. Students formed groups of one to four people, where each member of the group received the same grade. Projects accounted for 40% of the semester grade. The groups were given complete freedom to choose their topics, but the format of the project was defined. However, several exceptions to the format were approved.

The first stage in the project was problem definition. There was an early deadline for students to submit a problem statement. At least two groups submitted problem statements that ended up completely unrelated to their final projects. The next deadline was a survey of previous work on the problem. Most projects included some implementation work, and a final report was due on the last day of class. The report consisted of a problem statement, survey report, implementation section, results, and conclusions. On the last day of classes, each group presented their project to the class. Final project grades were based on the report and the presentation.

The quality of the projects ranged from poor to excellent. At least two of the groups will submit papers for publication, and three of the groups are continuing work on their projects on their own. Two of the best students in the class will probably work at Bellcore this summer on projects related to their course projects, and one of these students is now pursuing a problem, under the direction of the instructor, that might lead to his Ph.D. thesis topic. It is likely that this work will be funded by Bellcore in the future.

One of the groups' projects was to improve the security of the Courant Institute's computer center. The group developed a site security policy. In addition, they obtained many public domain security packages, such as Tripwire, COPS, S/KEY, etc, and explored their installation at Courant. The group also installed version V of Kerberos on some test machines, to test the feasibility of deploying it on the actual network. This group worked together with the system administrators and a faculty member in charge of a committee to help secure their network.

Another successful project dealt with NFS security. The group working on this project implemented a one-time password scheme to authenticate NFS accesses, and designed an encrypted NFS file system based on this technique. The students are continuing their work and plan to submit a paper to an upcoming security conference. Their preliminary performance numbers indicate that the performance hit is negligible.

There was a group in the class that attempted to implement the emerging IETF standards for IP layer security, under DOS. This project was too ambitious, and the results were marginal at best. A more successful project came from a student who did his project alone. He designed a system to help eliminate fraud in cellular phone systems. The student actually contacted Nynex and Bell Atlantic. His system is very practical, and may actually be implemented some day.

7 Lessons learned

This course is being offered again in the Spring of 1996. The content will remain the same, but there were some valuable lessons learned from the first semester.

Teaching a class that meets only once a week is a challenge. To compensate for the lack of direct contact with students, the class was required to check their e-mail on the course mailing list three times a week. Everyone was accountable for all information in any posting, and several homework assignments were distributed this way. Most students had no problem with this, but a few, who work during the day, complained of difficulties accessing the network from home in the evenings. Overall, however, requiring students to check their mail proved to be invaluable for maintaining contact with the students.

An important lesson learned from the course projects is that most students do not like open-ended problems. A little guidance goes a long way. Next semester, the students will be able to choose from a list of projects, or pursue something on their own. This will eliminate some of the confusion (and fear) that was evident last semester. Another lesson from the number theory section is that there is no such thing as too many examples. Many of the number theory concepts are quite abstract, and until an example demonstrates a particular technique, there is little or no intuition gained the first time something is explained.

The last lesson learned from this class is that computer security and cryptography are very hot topics right now. Several students brought friends of theirs to sit in on lectures, there was one faculty member who attended regularly, and many students felt there was a direct relationship between what they were learning in class and their current jobs.

8 Future course plan

A sequel to the current course is being planned for the fall of 1996 semester at NYU. This new course will assume the current class as a prerequisite. It will delve deeper into the current topics and introduce some new ones.

The second course in cryptography and computer security will cover more topics in number theory. Other cryptographic algorithms and protocols will be taught, including concepts such as oblivious transfer, bit commitment, blind signatures, simultaneous contract signing, anonymous broadcast, byzantine agreement, etc.

Another part of the course will focus on electronic commerce including protocols such as VISA's STT, Mastercard's SEPP, IBM's iKP, and Netscape's SSL.

If if there is time, other topics under consideration are secure operating systems, including multilevel systems and covert channel analysis, the Bell-Lapadula model [3], the use of smart cards for security, Windows NT security, formal methods, and logics of authentication, such as the BAN logic [10].

References

[1] Accredited Standards Committee X9. *Working Draft: American National Standard X9.30-1993: Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry: Part 2: The Secure Hash Algorithm (SHA)*, 1993.

[2] D. Balenson. Privacy enhancement for Internet electronic mail: part iii—algorithms, modes, and identifiers. *RFC 1423*, February 1993.

[3] D. E. Bell and L. J. LaPadula. Secure computer systems: A mathematical model. *Repoert MTR-2547, MITRE Corp.*, 1973.

[4] Steve Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker.* Addison-Wesley Publishing Company, 1994.

[5] E. Biham and A. Shamir. *A Differential Cryptanalysis of the Data Encryption Standard.* Springer-Verlag, 1993.

[6] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung. The kryptoknight family of light-weight protocols for authentication and key distribution. *IEEE Transactions on Networking*, 3(1):31–42, February 1995.

[7] Matt Blaze. A cryptographic file system for unix. *Proc. 1st ACM Conference on Computer and Communications Security*, pages 9–16, November 1993. ftp://research.att.com/dist/mab/cfs.ps.

[8] Eric Brewer, Paul Gauthier, Ian Goldberg, and David Wagner. http://http.cs.berkeley.edu/ gauthier/endpoint-security.html, 1995.

[9] E.F. Brickell, D.E. Denning, S.T. Kent, D.P. Maher, and W. Tuchman. *Skipjack Review, Interim Report: The Skipjack Algorithm*, July 28, 1993.

[10] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8, February 1990.

[11] Drew Dean and Dan Wallach. Security flaws in the hotjava web browser. *Princeton CS Tech Report 501-95*, 1995. http://www.cs.princeton.edu/ ddean/java/.

[12] D.E. Denning. The Clipper encryption system. *American Scientist*, 81(4):319–323, July–August 1993.

[13] Dorothy Denning. *Cryptography and Data Security.* Addison-Wesley Publishing Company, 1983.

[14] Department of Defense. *DoD 5200.28-STD: Department of Defense (DoD) Trusted Computer System Evaluation Criteria (TCSEC)*, 1985.

[15] W. Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 1976.

[16] Simson Garfinkel and Gene Spafford. *Practical Unix Security.* O'Reilly & Associates, Inc., 1991.

[17] Peter Gutmann. http://www.cs.auckland.ac.nz/ pgut01/sfs.html.

[18] S. Haber and W.S. Stornetta. How to time-stamp a digital document. In A.J. Menezes and S. A. Vanstone, editors, *CRYPTO90*, pages 437–455. Springer-Verlag, 1991. Lecture Notes in Computer Science No. 537.

[19] Neil Haller. The s/key(tm) one-time password system. *Symposium on Network and Distributed System Security*, pages 151–157, February 1994.

[20] L. S. Hill. Cryptography in an algebraic alphabet. *American Mathematical Monthly*, 36:306–312, 1929.

[21] John Ioannidis and Matt Blaze. The architecture and implementation of network-layer security under Unix. *USENIX Security Conference IV*, pages 29–39, October 1993.

[22] David Kahn. *The Codebreakers.* Macmillan Publishing Co., 1967.

[23] F. W. Kasiski. *Die Geheimschriften und die Dechiffrir-kunst.* E.S. Miller und Sohn, 1963. (In German).

[24] Charlie Kaufman, Radia Perlman, and Mike Speciner. *Network Security: Private communication in a public world.* Prentice Hall, 1995.

[25] Paul C. Kocher. Cryptanalysis of diffie-hellman, rsa, dss, and other systems using timing attacks. *Manuscript*, 1995.

[26] John B. Lacy. CryptoLib: Cryptography in software. *USENIX Security Conference IV*, pages 1–18, 1993.

[27] X. Lai and J. Massey. A proposal for a new block encryption standard. In I.B. Damgård, editor, *EUROCRYPT90*, pages 389–404. Springer-Verlag, 1990. Lecture Notes in Computer Science No. 473.

[28] R.J. Lipton. How to cheat at mental poker. *Computer Science Dept., Berkeley, CA*, August 1979.

[29] M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology — Eurocrypt '93*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397, Berlin, 1994. Springer-Verlag.

[30] National Bureau of Standards. Data encryption standard. *Federal Information Processing Standards Publication*, 1(46), 1977.

[31] S.C. Pohlig and M.E. Hellman. An improved algorithm for computing logarithms in GF(p) and its cryptographic significance. *IEEE Transactions on Information Theory*, 24(1):106–111, January 1978.

[32] R. Rivest. The md5 message digest algorithm. *RFC 1321*, April 1992.

[33] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key crypto systems. *Communications of the ACM*, 21(2):120–126, 1978.

[34] R.L. Rivest. *The RC4 Encryption Algorithm.* RSA Data Security, Inc., March 12, 1992. (Proprietary).

[35] A. Rubin. Trusted distribution of software over the Internet. In *Proc. Internet Society Symposium on Network and Distributed System Security*, pages 47–53, 1995.

[36] A. D. Rubin and P. Honeyman. Long running jobs in an authenticated environment. *USENIX Security Conference IV*, pages 19–28, October 1993.

[37] Aviel D. Rubin. Independent one-time passwords. *USENIX Journal of Computing Systems*, 9(1), 1996. to appear.

[38] Bruce Schneier. *Applied Cryptography - Protocols, Algorithms, and Source Code in C.* John Wiley & Sons, Inc., 1994.

[39] William Stallings. *Network and Internetwork security.* Prentice Hall, 1995.

[40] J.G. Steiner, B.C. Neuman, and J.I. Schiller. Kerberos: An authentication service for open network systems. In *Usenix Conference Proceedings*, pages 191–202, Dallas, Texas, February 1988.

[41] Douglas Stinson. *Cryptography: Theory and Practice.* CRC Press, Inc, 1995.

[42] Stephen T. Walker, Steven B. Lipner, Carl M. Ellison, Dennis K. Branstad, and David M. Balenson. Commercial key escrow: Something for everyone now and for the future. *TIS report #541*, January 1995.

[43] Michael J. Wiener. Efficient DES key search. *TR-244, School of Computer Science, Carleton University*, May 1994.

[44] P. Zimmerman. Pgp user's guide. December 4, 1992.