Teaching Cryptography to Continuing Education Students

Anatoly Temkin

Department of Computer Science, Metropolitan College, Boston University
808 Commonwealth Avenue, Room 250, Boston, MA 02215, USA
temkin@bu.edu

Abstract. Knowledge of mathematical foundations of Cryptography is of paramount importance for students wanting to succeed in graduate degree programs in Computer Science with concentration in security. Cryptography, a relatively new field, has yet to establish a core set of topics and the optimal sequence of their presentation to prepare students for a career in the field of IT security. This paper presents syllabi of two courses on public and private key cryptography offered to continuing education students at Boston University.

1 Introduction

The Computer Science Department at Boston University Metropolitan College offers, among other degrees and certificates, a Master's in Computer Science with a concentration in security and a graduate certificate in computer security. Cryptography, data communications and computer networks, network security, network management and computer security, software security, and digital forensics are among the courses offered by the department [1]. Cryptography, a two semester course, plays an essential role in the curriculum by serving as a foundation for all other security related courses. Public key cryptography is covered in the first semester followed by private key cryptography in the second semester.

2 Students' Background

Students taking elective courses, and cryptography courses as electives, have a significant amount of programming experience. Some are professional programmers working as developers for software companies. Many students did not major in Engineering, Computer Science or Mathematics in their undergraduate programs, so math courses other than introductions to Calculus and Finite Math were not part of their curriculum. They entered the programming field at the end of the last century with majors in history, business, psychology etc and learned programming on the job or by completing certificate programs. By the time they enroll into a Master's degree program in Computer Science, their math background is very weak or almost nonexis-

Please use the following format when citing this chapter:

Temkin, A., 2007, in IFIP International Federation for Information Processing, Volume 237, Fifth World Conference on Information Security Education, eds. Futcher, L., Dodge, R., (Boston: Springer), pp. 121–128.

tent. Students use their working knowledge of programming to grasp new concepts. They try to explain new material in terms of familiar programming concepts.

There are two different types of students in the program: students with a math, computer science, physics or engineering academic background and students without a technical background. Most students from both groups hold full-time programming jobs. In addition to having different academic backgrounds, students vary in their work experience and by the number of years since graduating from college. The average age of students is approximately thirty. Having students with different background in the same classroom presents obvious problems. To make a course successful, we cannot rely on the knowledge students acquired in their undergraduate education, rather we should tailor instructions in a way that will be engaging to all students. Lectures and presentations should be structured to stimulate student interest without overwhelming the student. Overpowering information results in the lack of interest which is necessary to pursue studies. This principal of Zone of Proximal Development was introduced by Vigodsky [3] and is considered to be a major factor in successful education. The problem of keeping a balance between the amount of math in the course on cryptography and the students' ability to understand is a delicate issue. Having taught the course to about 200 students allows me to claim that my careful selection of topics to cover and their presentation order enabled most students' success in a math intensive crypto course.

Of course, there are prerequisite and core courses in the program, like Discrete Mathematics and the Analysis of Algorithms which prepare students to handle mathematical and computational concepts, but there is still a gap between the content of these courses and the abstract content of the group theory. It is not quite clear which crypto topics should be taught to students. Even less clear is the teaching sequence of these topics. For example, should public key or private key cryptography be taught first? We think the answer to this question depends on the objectives set forth before a course is offered. Our objective is to cover mathematical fundamentals of cryptography and that dictates first the coverage of the public key cryptography followed by the private key cryptography. We believe that the content of our courses on cryptography prepares students for successful work in the field.

Next, we introduce the two course sequences on cryptography

3 A Course on Public Key Cryptography.

Despite its name, the course on public key cryptography is not only on encryption theory but also on cryptanalysis. The course begins with a review of the number theory, paying special attention to the unique factorization of integers into primes, the Euclidean Algorithm for finding the greatest common divisor of two integers and its extension to finding multiplicative inverses modulo n. Basic group theory is reviewed and extended with thorough discussion of the multiplicative groups Z_p for p prime and Z_n for a composite n, which are used in the ElGamal and the RSA ciphers. Subgroups, Lagrange Theorem, Euler and Fermat Theorems, index of a subgroup and cyclic subgroups are among the basic facts about groups covered. Having familiarized students with the basic ideas, the next step is a deeper coverage of groups, including,

but not limited to, roots and powers in groups. This allows introducing the definition of a discrete logarithm followed by the ElGamal cipher along with some algorithms to compute discrete logs in cyclic groups. The algorithms presented are the Baby-step Giant-step algorithm and the Index Calculus method. The second part of the course begins with a thorough coverage of the RSA cipher. The AlGamal cipher is followed by the RSA cipher along with a discussion of various attacks, such as forward search, common modulus and small decryption exponent. Right after the RSA cipher several important protocols are covered, more specifically secret sharing, oblivious transfer and zero-knowledge proofs. Quadratic Reciprocity is introduced to present Euler probabilistic primality test. Along with Fermat, Euler and Rabin-Miller pseudo-primes the corresponding algorithms are covered. Random number generators, including Blum-Blum-Shub and Naor-Reingold generators are discussed in detail. In the third part of the course much attention is devoted to factorization attacks. Although this material requires a certain amount of mathematical preparation, not all of which is required of our students, with a certain diligence it is quite possible not to just explain the concepts but to carry them through to the practical algorithmic implementation. Pollard's Rho Method along with Pollard's $p-1$ method is presented. Among modern factorization attacks most attention is devoted to Dixon and the Quadratic Sieve algorithms. The primary texts for this course are textbooks [4] and [5] supplemented with lecture notes. The course is in the lecture format with weekly homework assignments and final projects.

Students are required not only to understand in depth theoretical foundations of the subject matter, but also to utilize their programming skills to write codes for algorithms used in breaking cryptographic codes. Our belief is that once students learn the ideas behind algorithms used in breaking codes they will be much better prepared to understand how to securely implement these algorithms.

The assignments for each of the lectures include problem solving and code writing. All algorithms covered in the course have to be coded by students and the final project is based on the previously coded algorithms. Among algorithms covered in the course are Euclidean and the extended Euclidean algorithms to find multiplicative inverses modulo n, the fast exponentiation algorithm, a primitive root search algorithm, a Baby-step Giant-step and the Index calculus algorithms to find Discrete Logarithms in Cyclic groups, a Miller-Rabin test, Naor-Reingold and Blum-Blum-Shub random number generator algorithms, Pollard's Rho, Pollard's $p-1$ and the Quadratic Sieve factoring algorithms.

For two final projects the class is broken into groups of three students [2]. For the first project, if A, B and C are students in a group, then each of them encrypts two messages using the Diffie-Hellman Key Exchange Protocol and sends them to other members of the same group. That is, A encrypts a message and sends it to B, while C, the adversary, tries to break this message by using Baby-step algorithm and the Index Calculus algorithms to find discrete logs in cyclic groups. Then A encrypts another message and sends it to C, while B, the adversary, tries to break into the message by using the same algorithms. Then B and C encrypt their messages and the whole process is repeated. So every member of a group is involved in encoding two messages and recovering two messages as an adversary.

For the second project, the same members of a group are involved in the exchange of messages within the same scheme using the RSA algorithm. They use the Pollard's Rho algorithm, Pollard's $p \square 1$ method and Quadratic Sieve algorithm to break into the messages.

Here is a syllabus for the course on public key cryptography with some comments

Week 1: Integers, prime numbers, relatively prime numbers, greatest common divisors, factorization into prime numbers, computation of $\square(n)$ based on Principle of Inclusions and Exclusions, the Euclidean and the Expanded Euclidean Algorithms, multiplicative inverses, equivalence relations, classes of integers modulo n, defining binary operations of addition and multiplication on classes of integers modulo n.

Comments: Operations of addition and multiplication are defined on Z_n before a formal definition of a group is introduced. The definition comes in the second lecture and Z_n and $Z_n^{\square}$ serve as examples of groups.

Week 2: Definition of a group, examples of groups (finite and infinite), groups Z_n and $Z_n^{\square}$, subgroups, cosets, Lagrange Theorem, cyclic groups, the exponent of a group, Euler and Fermat theorems.

Comment: I consider it very important to cover the introduction into the group theory as soon as possible in the course. My experience from teaching this course to about 200 students in the last four years supports this practice.

Week 3: Exponentiation Algorithm, Primitive Roots, Discrete Logs, ElGamal Cipher

Comments: A discrete log problem is introduced in an abstract group as well as the ElGa mal cipher. In the following week lecture this cipher is run in $Z_p^{\square}$

Week 4: The Diffie-Hellman Key Exchange Protocol, Primitive Root Search Algorithm, Baby-Step Giant-Step Algorithm, The Index Calculus Algorithm Public-Key Ciphers.

Comments: It seems very important not just to explain ciphers but to introduce probabilistic algorithms used to break ciphers.

Week 5: Introduction to Public Key ciphers, The RSA Cipher and attacks on RSA. Key distribution, mutual authentication, certificates.

Week 6: Chinese Remainder Theorem, Euler Criterion, Roots Mod Composites.

Week 7: Oblivious Transfer Protocol (factorization and discrete log based). Zero-knowledge proofs, authentication.

Week 8: Quadratic Reciprocity.

Week 9: Pseudorandom numbers, Fermat, Euler, and strong pseudoprimes, Solovay-Strassen test, Miller-Rabin test.

Week 10: Random Number generators, Linear Congruential generator, Feedback Shift generator, Noar-Reingold Generator, Blum-Blum-Shub Generator

Week 11: Modern Factorization Attacks. Pollard's Rho Method, Pollard's p-1 Method

Week 12: Dixon's algorithm, Non-Sieving Quadratic Sieve, The Quadratic Sieve Factoring Algorithm.

4 A Course on Private Key Cryptography.

The second semester course on private key cryptography begins with a thorough introduction into the theory of finite fields with coverage of commutative rings, irreducible elements of rings, group of units, the Euclidean algorithm in a polynomial ring, fields Z_p and $GF(2^m)$. The coverage continuous with the ring of polynomials mod P, operations of addition, multiplication and finding inverses in that ring. Discussing finite fields is important for the in-depth coverage of DES and Whirlpool hash function. The course goes on with the coverage of encryption systems and attacks against them. Fiestel ciphers and DES are discussed in detail. A lot of attention in lectures is given to block cipher modes of operation and hash functions based on block ciphers.

Security of hash functions, iterated hash functions, message authenticated codes are covered along with digital signatures and authentication protocols. Public Key Infrastructure, certificates, trust models, IP security protocols, Transport Layer Security/ Secure Sockets Layer protocols are discussed by the end of the course and, finally, an introduction into elliptic curves and ElGamal public key encryption is covered.

The primary texts for this course are textbooks [4], [5], [6] and [7] supplemented with lecture notes. The course is in the lecture format with weekly homework assignments and a final project. Homework assignments include writing codes for the algorithms covered and the final project. For the final project, students are broken in groups of two. If A and B are in the same group, A encrypts a message using AES, hashes it, "signs" it and sends the encrypted message and its signature to B. B verifies both the authenticity of the message and its integrity and decrypts it. Then A and B switch roles.

Here is a syllabus for the course on private key cryptography with some comments.

Week 1: Rings, commutative rings, zero divisors and integral domain, cancellation property in a commutative ring, irreducible elements of a ring, the additive group of a ring, the group of units, fields, polynomial rings, the Euclidean algorithm in a polynomial ring over a field.

Week 2: Finite fields, fields Z_p, congruence classes of a polynomials modulo P, irreducible polynomials of degree n, the ring of polynomials mod P as a finite field, field extensions; addition, multiplication and multiplicative inverses in the ring of polynomials mod P.

Week 3: Advanced Encryption Standard block cipher.

Comments: For some reasons students think that whatever the mathematical apparatus used for public key cryptography is totally different from the one used for private key cryptography. It is very important to emphasize the mathematical foundations of block ciphers like AES, to go over a group $GF(2^8)$, irreducible polynomials, inverse elements etc.

Week 4: Encryption schemes, unconditionally, computationally and provably secure encryption systems, attacks against the encryption scheme (ciphertext only, known- plaintext and chosen-plaintext, chosen ciphertext), a simple substitution ci-

pher, polyalphabetic ciphers, block and stream ciphers, the Vernam cipher, a one-time pad.

Fiestel ciphers and DES. Linear and differential cryptanalysis.

Week 5: The New Data Seal (NDS) cipher and a chosen-plaintext attack on NDS. Tweakable block ciphers and modes of operation: Tweak Block Chaining, Tweak Chain Hash and Tweakable Authenticated Encryption.

Comments: As NDS uses the same key in all rounds, the chosen-plaintext attack is successful and although NDS is never used in "real" life situations, it is beneficial for students to learn how the key is recovered as the result of the attack.

Week 6: Double DES, its vulnerability to meet-in-the-middle attack, triple DES, block cipher modes of operations. Electronic codebook (ECB), Cipher-block chaining(CBC), cipher feedback (CFB), Output feedback (OFB), Counter (CTR). Error propagation, integrity protection.

Week 7: One way functions, trapdoor functions, confidentiality and non-repudiation, hash functions, properties of cryptographic hash functions (preimage, second preimage and collision resistance), the random oracle model. Iterated hash functions, the Matyas-Meuer-Oseas and Miyaguchi-Preneel hash function constructions. The Mercle-Damgard generic construction of cryptographic hash functions. The Digital Signature Algorithm

Week 8: A Whirlpool cryptographic hash function. Coverage of MD-4, MD-5, SHA1, SHA-256, 384, 512, a commitment scheme and verification of message integrity.

Comments: The coverage of a Whirlpool hash function seems to be important as the hash function's construction is based on a block cipher similar to AES.

Week 9: Message authentication codes, message authentication codes built from block ciphers, HMACs, Swcurity of MACs, vulnerability of MACs to birthday, collision and other attacks.

Week 10: Public Key Infrastructure, certificates, trust models, IP security protocols, Transport Layer Security/ Secure Sockets Layer protocols

Week 11: Introduction into elliptic curves over finite fields Z_p and F_q, a group operation on an elliptic curve, points at infinity.

Comments: This material seems to be the most difficult for students since it is math intensive. However, most students grab the concept of elliptic curves and are prepared to further their knowledge through independent reading.

Week 12: Elliptic curve cryptography: ElGamal public key encryption, Massey-Omura encryption, ElGamal Digital Signatures, the Digital Signature Algorithm.

5 Conclusion

Students' feedback is highly encouraged and plays a significant role in the design and updates of the crypto course. Being academic in nature, this course has the potential to make an immediate impact on students' ability to utilize the knowledge gained in the classroom in their professions. Since many students taking this class work full-time for leading companies in the security field, like RSA Security and Cisco, or are employed as computer security professionals at other companies, very often holding

leading positions, their opinion of course content and its modes of offering is very valuable. Students' satisfaction with the course on security as well as their satisfaction with the whole program is a litmus test for the faculty. A large number of students consider the course on cryptography to be highly challenging but manageable. There should be a significant effort on the part of the student to learn. The efficacy of the course is conveyed in almost every student's evaluation. Working professionals always evaluate courses based on how much they learn and on the amount of applicability to their immediate line of work. I am listing some of the student's feedback comments.

"The usage of this class will help me evaluate the security we can build into our Web services, our password encryption in particular. The understanding of cracking the codes for the RSA and the Diffie-Hellman key exchange will help me choose good keys for these secure communication protocols."

"I work as a systems integration engineer in the International Air Traffic Control division. My responsibilities are to help engineers integrate radar software and hardware components. Two years ago, I made the decision to pursue my Master's degree in Computer Science with specialization in security. At work with the proliferation of networks and the internet, increasingly security is playing a prominent role in every aspect of the projects I have worked on. The cryptography course helped me a great deal to get a better understanding and appreciation for the role and benefits of application, system and network security.

I feel compelled if not obligated to learn more about security in order to perform my job adequately. I find myself increasingly involved in the role of a security engineer which requires a strong understanding of cryptography. I have enjoyed and learned a lot from the Cryptography class and I believe it is the most important course in computer science degree program with security concentration."

"I think that I have re-learned how to think, especially in a course environment! So, this will be of tremendous value as I continue with other courses in the program. I anticipate that I will have a focus in security as part of my responsibilities as an integrated solution architect within the next year. My role is evolving and as the focus integration within the healthcare enterprise increases, the security will become important and the ability to exchange shared secrets for use in secure communications will be essential. I will need to have an understanding of what options are available and the tradeoffs between them. Having the understanding of the mathematical foundations for the protocols will be helpful to truly understand how the protocols should be used and how they might be compromised."

"As far as my professional work is concerned, the course has opened my eyes to the hazards of believing we are "secure". The cryptography course, more than anything, has piqued my interest in math. I kept telling my friends all semester that cryptography is the best math class I had ever taken. As an aside, the project may be one of the most fun projects I've ever had for a course. We have had quite a good time going back and forth trying to break RSA private keys. It's amazing how big these numbers can get yet we are still able to crack them. But it's also incredible to see how well-chosen keys resist Rho, p-1, and other attacks!"

"My role at work is to design and develop our company's IT infrastructure, especially around networks and security. While I've always been considered one of the most security-conscious employees on the team, it wasn't until this course that I

fully came to appreciate the importance of cryptography. For example, in the past when evaluating a vendor's product, I'd have been satisfied to learn "encryption" was available. Given what I now know today, this wouldn't be enough - e.g. what specific encryption algorithms are being used? Who developed the encryption protocols (i.e. in-house vs. standard, well-known protocols like RSA)? What key-length is being used? I'll want to understand the specifics to ensure that choices we make as a company are well-informed and grounded in a strong technological basis and not on marketing hype."

"I found this course very valuable in truly understanding how crypto works while working in a security environment. By making this more a course in cryptology and using real-world examples the student is given a solid education."

"In this class I have learned much of the mathematical basis for algorithms that I use in my professional work. It is helpful to understand the underpinnings for RSA and Diffie-Hellman protocols as that will enable me to make better use of them and avoid pitfalls that I otherwise might encounter.

"I had an interest in finite fields in high school, but had no idea about what applications there might exist. This course has re-stimulated my interest in this field and particularly cryptology. I have a special interest in (t,n)-threshold schemes and new uses for cryptography. I'm also especially interested in provably secure pseudo-random number generators, how pseudorandom number generators can be evaluated, and what it means for something to actually be, or appear to be, random. I spent a lot of time working on my final projects; building the rudiments of a "Cryptologic Workbench" that I hope will help me explore some of these ideas more thoroughly. I also have some ideas on how a scalable web services-enabled cryptologic workbench might enable a larger community to cooperatively explore these ideas.

References

1. Zlateva, T et al. Integrated Curricula for Computer and Network Security Education, Proceedings of the Colloquium for Information Systems Security Education, Society for Advancing Information Assurance and Infrastructure Protection, Washington, D.C., June 2003.
2. Chitkushev, L.T. et al. Laboratory Assignments in Security Education, Proceedings of the 4th World Conference on Information Security Education. Editors: Natalia Miloslavskaya, Helen L. Armstrong. Success Through Information Security Knowledge, IFIP TC11 / WG11.8 Forth World Conference on Information Security Education (WISE 4), June 18-20, 2005, Moscow, Russia. ISBN 5-7262-0565-0
3. Vygotsky, L.S., Mind in Society: The Development of Higher Psychological Processes, Cambridge, Mass: Harvard University Press, 1978
4. Garrett, P. Making, Breaking Codes: An Introduction to Cryptology. Upper Saddle River, NJ: Prentice Hall, 2001
5. Stallings W. Cryptography and Network Security: Principles and Practices, 4th edition.Upper Saddle River, NJ: Prentice Hall, 2006
6. Washington, Lawrence C. Elliptic Curves, Number Theory and Cryptography, Chapman & Hall/CRC 2003
7. Stinson, Douglas R. Cryptography: Theory and Practice, 3d edition, Chapman & Hall/CRC 2006