# Cryptography and Computer Security for Undergraduates

Paul De Palma (Moderator)
Gonzaga
depalma@gonzaga.edu

Charles Frank
Northern Kentucky
frank@nku.edu

Suzanne Gladfelter
Penn State York
sg3@psu.edu

Joshua Holden
Rose-Hulman
holden@rose-hulman.edu

## ABSTRACT

The panel discusses solutions to the problem of computer security education.

## Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: security **and protection. E.3 [**Data Encryption]**: public key cryptosystem**

## General Terms: Security

## Keywords: cryptography, security

## 1. SUMMARY

**Paul De Palma**

Diffie and Hellman wrote these inspiring words in 1976: "We stand today on the brink of a revolution in cryptography. "[6]. A little over twenty years later, then Deputy Director of the NSA, William Crowell estimated the depth of the revolution: "If all the personal computers in the world - 260 million computers - were put to work on a single PGP-encrypted message, it would still take an estimated 12 million times the age of the universe, on average, to break a single message [5]. Since PGP is free for the downloading, why don't we feel safe? The answer is that cryptography is mathematics, but security requires that cryptographic algorithms be embedded in real, live systems, systems whose staggering complexity is designed by people, with all of our charms and warts. Though I owe this observation to Bruce Schneier [14], it should be obvious to anyone who has designed a large system: principles are great but they come wrapped in a context, and the context is usually messy.

Nevertheless, we in computer science education are being asked to train a generation of practitioners in the mysteries of computer security. In late 2002, the President signed the Cyber Security Research and Development Act [1] that provides over $216 million to support undergraduate and graduate education in computer security. What constitutes this education is the big question. The
cryptographic aspect is now supported by an array of textbooks, many written by mathematicians [3,9,17]. The problem is that cryptography straddles mathematics and computer science. Computer science students often do not have the mathematics to understand the material. A common solution is to present the relevant mathematics as needed. Not a bad idea, though one might argue that a short presentation makes a difficult subject more difficult. Turf is an attendant issue. I've been told that mathematicians feel possessive when computer scientists begin talking about Euler. Let's assume with Schneier that computer security may well begin with cryptography but certainly does not end there. It's easy to imagine teaching a one-semester course on cryptography to suitably prepared students. But what about the rest? What about secure operating systems, secure networks, secure wireless networks, and more. It's a tall order, maybe too tall for the already crowded computer science curriculum. Chomsky [4] once observed that the size of a discipline is inversely proportional to its maturity. On this view, cryptography is mature and tractable, while computer security is a great wooly monster. It requires years of hands-on experience, experience most of us did not acquire in graduate school, nor while teaching undergraduates. And if we really do things right, we ought to consider history too, since crypto, though mature, has had a contentious childhood. An increasing number of computer science programs are confronting these issues.

## 2. POSITION STATEMENTS

**Charles E. Frank**

Computer science students need to understand modern cryptographic algorithms and protocols. They need to be able to apply cryptography to enhance the security of the software they will develop and of the systems that they will manage. I incorporate cryptography into the computer security courses that I teach to advanced undergraduate and graduate computer science students at Northern Kentucky University [7]. I use [12] as the textbook for the cryptography part of my courses. The ciphers I cover include DES, TripleDES, Rijndael, RSA, and Diffie Hellman. I also cover the MD5 and SHA-1 hashing algorithms. Since I want my students to be able to incorporate cryptography into their programs, we look at a variety of Java applications [8] that employ cryptography. I assign my students several small Java programs including a network application that uses cryptography. We encrypt our email with PGP. We also cover SSL/TLS and IPsec.

My students have had a discrete mathematics course with 10 hours of calculus. I cover sufficient number theory so my students can understand RSA and the generation of probable primes. This includes modular arithmetic and Euler's theorem. My students have already studied basic algorithm analysis. I cover algorithms for fast exponentiation and for finding modular inverses. I do not cover classical, yet obsolete, ciphers or the history of cryptography. My course is not a mathematically rigorous presentation. It is applied cryptography.

**Suzanne E. Gladfelter**

Students need to appreciate historical cryptology and understand modern cryptological algorithms and protocols before applying cryptography to computer security issues. I team-teach an introductory upper-division, multi-disciplinary (history, mathematics and computer science) cryptology course at Penn

*SIGCSE'04,* March 3–7, 2004, Norfolk, Virginia, USA.
ACM 1-58113-798-2/04/0003.

State York [10]. We discuss monoalphabetic ciphers through public key cryptography; we use [2] and [15] as the textbooks along with an extensive bibliography of resources. The on-line course web site [10] provides links to the syllabus and reference materials.

The course is an elective, open to all majors and labeled as STS (science, technology and society). In addition to fifth semester standing, we require that students have a background in algebra and minimal computing experience. As part of the course, our mathematician teaches modular arithmetic, Euler's theorem, and sufficient number theory for students to be able to understand both classical and modern cryptological algorithms. Some of the mathematical presentations are rigorous, but are "for information only." Students are not expected to master the theory, but they are expected to practically apply the mathematics to course topics. The course offers a variety of assignments: thoughtful historical questions, mathematical applications and computer programming. Students can go on to take other courses at Penn State York to extend their knowledge of cryptology to computer security.

**Joshua Holden**

I have two goals in teaching cryptography to computer science students: to use cryptography as a "cool way" of introducing important areas of mathematics and computer science theory and to educate students in something that may be necessary for them to know in the future. For the past two years I have co-taught a course in cryptography at the Rose-Hulman Institute of Technology with David Mutchler, a colleague from the Computer Science department [11]. The course is cross-listed in both the Computer Science and Mathematics departments, but most of the students are CS majors. The published prerequisites are one quarter of discrete mathematics and two quarters of computer science.

The mathematical part of the course introduces basic number theory; the goal is to get all of the mathematics necessary to understand RSA and AES. We find that most students require quite a bit of extra background, despite the mathematics prerequisite. The mathematical and technical aspects of the course use [16] as a textbook, along with some handouts and web sites. We cover both modern and historical systems. These are chosen based on how they illustrate mathematical and algorithmic principles, such as modular arithmetic, algebraic structures, iteration, recursion, diffusion and confusion, etc. Since cryptography is a fast-moving subject, we feel that it is more important to give students the mathematical and theoretical background behind the ciphers rather than to worry about the details of implementations. Discussions of cryptography and society revolve around readings from [13].

# 3. REFERENCES

[1] American Association for the Advancement of Science. "Cyber Security Bill Becomes Law." Retrieved 9/9/03 from : aaas.org/spp/cstc/news/articles2002/021220_cyber.shtml

[2] Beutelspacher, A. *Cryptology*. MAA. DC, 1994.

[3] Bishop, D. *Introduction to Cryptography with Java Applets*. Jones and Bartlett, Sudbury, MA, 2003.

[4] Chomsky, N. *New Horizons in the Study of Language and Mind*. Cambridge U. Press, NY, 2000.

[5] Crowell, W. Statement to the House on H.695 "Safe Act." Retrieved 8/6/03: www.columbia.edu/~ariel/hr695/crowell-mar20.html.

[6] Diffie, W. and Hellman, M. New Directions in Cryptography. *IEEE Transactions on Information Theory* 22,6 (Nov. 1976), 644-654.

[7] Frank, Charles: http://www.nku.edu/~frank/682.html

[8] Garms, J. and Somerfield, D., *Professional Java Security*, Wrox Press, 2001.

[9] Garrett, P. *Making Codes, Breaking Codes: An Introduction to Cryptology*. Prentice Hall, Upper Saddle river, NJ, 2001.

[10] Gladfelter, Suzanne: http://www.yk.psu.edu/~sg3/sts497a

[11] Holden, Joshua: rose hulman.edu/class/ma/holden/Math479

[12] Kaufman, C., Perlman, R., and Speciner, M. *Network Security.*Prentice Hall 2002.

[13] Levy, Steven. Crypto: Penguin Books, 2002.

[14] Schneier, B. *Secrets and Lies. Wiley. NY 2000.*

[15] Singh, Simon. *The Code Book.*Anchor Books, NY 2000.

[16] Stallings, W. *Crytpography and Network Security: Theory and Practice*. Prentice Hall, Upper Saddle River, NJ 2002.

[17] Trappe, W. and Washington, L. *Introduction to Crytpography with Coding Theory*. Prentice Hall, 2002.