Proceedings of the 11th Colloquium for Information Systems Security Education
Boston University
Boston, MA June 4-7, 2007

"Hands-On Crypto": Experiential Learning in Cryptography

N. Paul Schembari, Ph.D., *East Stroudsburg University of Pennsylvania*

Abstract— Experiential learning has been shown to be one of the best methods for learning, especially when combined with other forms of instruction. While much of the literature has illustrated experiential learning techniques for information assurance curriculum in general, the "Cryptography" course has not been studied in great detail with regard to experiential learning. We discuss exercises of multiple forms which demonstrate the intersection of experiential learning and cryptography.

Index terms – **Education, Experiential Learning, Computer Security, Information Assurance, Cryptography**

I. Introduction

In recent years, an increasing number of universities have introduced curricula in information assurance (IA). Logically, many have looked to standards to determine what content should be put into their coursework. One source for information assurance curriculum standards is the US Committee for National Security Systems [1]. The standard entitled *NSTISSI 4011: National Training Standard for Information System Security Professionals* is the most introductory and gives a broad overview of required topics for information assurance education and training.

As part of NSTISSI 4011, the topic of *Cryptography* is considered important, as it is one of the major tools used to protect data confidentiality. In fact, NSTISSI 4011 indicates that the following areas should be emphasized:

- Cryptographic strength (e.g., complexity, secrecy, characteristics of the key)
- Encryption types (e.g., point-to-point, network, link)
- Key management

Furthermore, most universities which offer information assurance coursework will have a course, or part of a course, devoted to cryptography. Also, many universities which do *not* offer information assurance degrees or certificates will offer an elective in cryptography. We also see other "standards" such as the CISSP (Certified Information System Security Professional) Common Body of Knowledge [2], and other standardized tests in information assurance, require knowledge in cryptography. Clearly, the topic of cryptography has been deemed important to the study of information assurance, and so students need to have this knowledge.

Since cryptography is a topic which should be included in information assurance curriculum, we must consider the teaching and learning of cryptography. One decision which is necessary concerns emphasis. Some cryptography courses will emphasize a deep study of cryptographic algorithms, while others will emphasize application of the algorithms. As a case study, consider East Stroudsburg University of Pennsylvania (ESU), where students are required to understand cryptographic algorithms. All information assurance students learn historical and modern encryption algorithms by completing the ESU *Applied Computer Cryptography* course. ESU also offers an elective entitled *Cryptographic Application Development* for those who wish to learn how to build applications using encryption. In these courses, as well as others which involve cryptography, we have successfully used experiential learning techniques since 1999. In fact, as our curriculum evolves, we have increased our use of experiential learning. For more information on how a university can implement cryptography and computer security curriculum at the undergraduate level, see the ESU Center for Computer Security and Information Assurance [3] and Schembari's paper [4] on a Bachelor's program in information assurance.

Another logical step for any university which enters into the information assurance field is the determination of the best teaching methods for this field. Hence, we find in the literature many examples of information assurance pedagogy. Quite a few examples include laboratory experiences in information assurance. Research, as well as anecdotal evidence, has shown that laboratory experiences give students good understanding and retention of the important topics, and hence many instructors use these experiences. However, while cryptography has been deemed an important topic for information assurance students, there has been little

N. Paul Schembari is a Professor of Computer Science and Computer Security at East Stroudsburg University of Pennsylvania. He is also the Director of the University's Computer Security Program.

research in the use of experiential learning techniques in cryptography.

The goal of this paper is to illustrate some uses of experiential learning in cryptography. We begin by giving an overview of experiential learning in information assurance to illustrate its benefits. We then consider some of the experiential learning techniques which have been used for cryptography by others. Finally, we examine some of our own techniques to demonstrate the intersection of experiential learning and cryptography.

II. Experiential Learning and IA Curriculum

Much research has been performed on experiential learning in information assurance and related fields. This research indicates that experiential learning may very well be the best method for learning IA, especially when combined with other methods of instruction.

For example, Perez-Hardy [5] has illustrated the use of a laboratory approach in teaching network administration, a topic closely related to information assurance. Perez-Hardy explains that this approach "provides the best mechanism for making theory a reality for the students." We should also be aware that most practitioners will combine experiential learning with the typical "lecture" model. As Perez-Hardy states, labs should be "designed to reinforce the theory covered in lecture."

In an illustration of experiential learning in information assurance, Hazari [6] discusses some of the downside of the lecture experience: "Students are passive learners in this process, feedback from all students may not be evident, the lecture is delivered assuming average understanding and comprehension for the entire class thereby isolating learners who may be advanced or those needing remediation." Hazari later explains that "when used along with active learning techniques, the lecture becomes even more powerful in achieving instructional goals." It is also important to note the type of information assurance laboratory experiences discussed by Hazari – personal firewall installation, configuration of security zones, program control, alert logs, and email protection.

Dellacca and Justice [7] also promote experiential learning in information assurance. They state: "We continue teaching foundational principles while moving into more advanced content by providing students the benefits of hands-on learning." They also discuss multiple laboratory and outside-of-class learning experiences including the "process through which servers hand out Internet Protocol addresses" which may involve DHCP, security solutions on home wireless networks, and IT Risk Assessments.

Wagner and Wudi [8] also discuss experiential learning in information assurance by looking at the development of cyberwar laboratory exercises for a computer security course. As part of their course, they developed multiple laboratory experiences including

- Ethics
- Policies and Social Engineering
- General Information Gathering (Ping, Traceroute, Finger, Whois, Nslookup/Dig, ARP, Netstat, Etc.)
- Packet Sniffing (TCPDump, Ethereal)
- Password Cracking (John The Ripper, L0phtcrack)
- Cryptography (PGP)
- Port Scanning (Nmap)
- Vulnerability Assessment (Nessus, Chkrootkit)
- Intrusion Detection (Snort)

They also discuss the benefits of experiential learning: "The students better understood the principle we'd passed on the first day of the course – that security is a process, not a product … They reported a better understanding of the use of attacking tools to perform vulnerability assessments of their own network and systems."

One of the best papers covering experiential learning in information assurance is Crowley's "Experiential Learning and Security Lab Design" [9]. Here Crowley discusses some information assurance laboratory exercises, but also gives a good overview of the theory behind experiential learning. He discusses Kolb's learning model [10], where four modes of learning are examined

- *Abstract Conceptualization* (AC), where the focus is on using logic, ideas, and concepts.
- *Reflective Observation* (RO), where the focus is on understanding the meaning of ideas and situations by carefully observing and impartially describing them.
- *Concrete Experience* (CE), where the focus is on being personally involved in experiences.
- *Active Experimentation* (AE), where the focus is on actively influencing people and changing situations.

Crowley also points to a study by Stice [11] which shows the effectiveness of using as many of these modes of learning as possible. Stice's work can be summarized by the following table, where "Retention" indicates the amount of information retained by students after using the shown modes of learning:

Modes of Learning	Retention
AC	20%
AC + RO	50%
AC + RO + CE	70%
AC + RO + CE + AE	90%

The typical lecture and homework model for a course would only use the AC, RO, and perhaps CE modes of

learning, while adding experiential learning would allow all four modes to be included.

Besides discussing educational theory, Crowley also illustrates how to apply experiential learning in IA. He states that "a hands-on lab experience enhances the students understanding of the theoretical concepts." Further, he gives multiple laboratory examples:

- Scanners (Nmap). Use Nmap to:
  - Launch a DOS attack
  - Attempt to avoid intrusion detection
- Sniffers (TCPDump, Ethereal)
- Intrusion Detection (Snort, ZoneLog)
- Capture and analyze packets
  - WINDump
  - Ethereal
- Analyze TCP/IP in real or near real time
- Recognize hostile/attack signatures
- Demonstrate Intrusion Detection by recognizing system scan signatures from:
  - Packet dumps
  - Firewall logs
  - Log analyzers (Zonelog)

With this review of research in IA and experiential learning, two important points emerge. First, each of the above authors has indicated that experiential learning is an excellent goal for information assurance instruction. Second, very few of the illustrated labs include a study of cryptographic algorithms or the applications of these algorithms.

III. Experiential Learning in Cryptography

A review of the literature on experiential learning in cryptography yields little results. We have found almost no peer-reviewed research, and some web sites devoted to cryptographic laboratory experiences, and these will be summarized below. It seems that while many view a course in cryptography as an important part of IA, computer science, or mathematical curriculum, most will teach this course using the Lecture - Homework model. The reason behind this choice may be because the Cryptography course is quite mathematical and can be taught in the traditional manner of mathematics classes. After we review some of the existing research in experiential IA learning in this section, we then illustrate some of our own methods in the next section.

We have been able to find some online publications which illustrate experiential learning in cryptography. We believe this catalog is a good starting point for instructors interested in using these techniques in cryptography instruction. For example, a laboratory experience in "Certificates and Cryptography" is available on an MIT webpage [12]. This extensive lab states that it will give the student "an introduction to mathematics and the algorithmic building blocks of modern cryptographic protocols." Part 1 of the lab deals with brute force attacks, and Questions 1.1, 1.2, 1.3, and 1.4 ask the student to estimate the time of brute force attacks and compare these to other estimates (like the age of the universe). Question 1.5 discusses the effects of "Moore's Law" on brute force attacks.

Part 2 of the MIT lab deals with hash algorithms, asking students to perform the hash of a phrase, to estimate the chance of finding another MIT document with the same hash, and to estimate the time of finding a hash collision with the first hash. Part 3 of the MIT lab deals with the encryption product GNU Privacy Guard (GPG), certificates, and signatures. After experimenting with GPG the students are directed to answer questions concerning signatures, GPG, finding altered documents based on signatures, using key servers, and finding legitimate signatures.

A laboratory experience in cryptography is also available on a webpage from Columbus State University [13]. First the students use Microsoft Excel to help in the computation of RSA public and private keys. Microsoft Excel is also used to perform RSA encryption and decryption. Part II of this lab allows the students to view key sharing / splitting techniques and then recover the original key using the Chinese Remainder Theorem. It should be noted that the above exercises are also available from a webpage from Montclair State University [14]. The Columbus State University lab is completed with some discussion questions concerning perfect ciphers, different types of data and the length of time they need to be protected, and US government encryption selection and export controls.

A webpage from the University of Pittsburgh [15] gives laboratory experiences in the Advanced Encryption Standard (AES) using *Matlab* routines. The lab overviews three topics

- Randomness of output and the avalanche effect
- Triple encryption
- Brute force attack with partial knowledge of the key

Randomness and the avalanche effect is studied by allowing the students to perform multiple encryptions using the Matlab code with slight changes to the plaintext or key. Triple encryption allows the students to perform triple-AES using one or two keys. The brute force attack portion of the lab gives students a partial AES key and then allows them to brute force the rest of the key using Matlab code.

Perhaps the best source for experiential learning experiences in cryptography is the collection of web pages created by Brown [16], who has also created Microsoft PowerPoint slides for use with Stallings text

[17] on cryptography. These labs proceed through multiple topics:

- Students create their own classical ciphers (not product ciphers) with limited key sizes, and discuss why their ciphers are secure.
- Students crack the ciphers created by their peers.
- AES encryption where the students view and analyze input and output in each round.
- Students perform the Add-Round-Key stage of AES by hand.
- Students properly encrypt with AES via the Cipher Block Chaining and Cipher Feedback modes of operation.
- Students complete Number Theory exercises including finding discrete logarithms, greatest common divisors, the Euler Totient Function, and modular arithmetic in GF(2^5).
- Students complete RSA encryption
- Students complete El Gamal digital signatures.
- Students use "Secure Email" via certificates. They should also be able to exchange certificates.
- Students sign email and should be able to check valid signatures

IV. Our Methods for Experiential Learning in Cryptography

While we have used multiple experiential learning techniques in our Cryptography course, some of the examples given in this section are from other courses as well. As a core component of IA, cryptography is typically discussed in multiple places in the curriculum. We have used three types of experiences in cryptography including Pencil and Paper Exercises, Electronic Exercises, and Required Program Exercises.

A. Paper and Pencil Experiences

One of our paper and pencil exercises involves students solving cryptograms – illustrating the weakness of substitution ciphers, even when attacked by hand. An important illustration here is the increase in difficulty of attacking a substitution cipher when the words are *not* spaced, as is typical for a cryptogram.

A second paper and pencil exercise involves key reuse. Students are given two ciphertexts which have been encrypted with the same key via a stream cipher (character based). They are also given some information about the source of the ciphertext (an artillery unit), and are asked to then try and cryptanalyze the texts.

In a third paper and pencil exercise, we allow students to experience the difficulty of cracking rotor ciphers. We have developed our own simple rotor cipher, using one or two rotors, which is illustrated after a discussion of the German *Enigma* machine, and the students are asked to try and partially crack the cipher. In fact, the students use a known-plaintext attack to partially recreate the cipher wheels.

B. Electronic Experiences

For electronic experiences, students are given computer based tools, sometimes tools they have developed themselves, and they use these to experience cryptography, crack ciphertext, etc.

Using a sniffer such as Ethereal / Wireshark, students are able to watch encrypted traffic, key exchange, or certificate exchange. As two participants use SSH, for example with *PuTTY* [18], other participants sniff the communications to see what is available to the eavesdropper. It is also possible for students to view key or certificate exchange in their own created applications.

The students can also experience cryptanalysis with the help of various tools, which they may have created on their own. We have had students crack substitution ciphers using bigram and trigram analysis. We have had students crack Vigenere [17, pp. 40-43] ciphertext by using the Index of Coincidence [19] to find the period of the key and then using a frequency tool to determine the actual key.

Students can also use tools to view brute force attacks in action. We have had students crack Simplified-DES [17, pp. 56-63] via brute force. Another lab which we have created, but not yet tested in class, involves determining the limits of software like *Mathematica* in cracking RSA – what key size can Mathematica crack practically?

C. Programming Experiences

As mentioned in the previous section, we often ask students to build tools which they can use to encrypt, decrypt, or cryptanalyze. We have created many labs and programming projects in this regard. The students performing these exercises are typically junior Computer Security majors who have already completed an introductory programming course and two courses in data structures. They also have completed a mathematics course in Discrete Mathematics. Hence, we can recommend these exercises for students who have programming experience.

These application-building projects are included below with commentary on student learning or pedagogy.

Caesar Cipher

- Students perform encryption / decryption
- Brute Force Attack: Given ciphertext without key, students convert to all possible plaintexts so that they can decide on the correct plaintext.

- This is typically an easy exercise for most students. We have seen some students who had difficulty in understanding how to give output on the Brute Force attack to see it work.

Simple Ciphers
- Students create an application which allows encryption / decryption via Substitution, Permutation, or XOR
- The idea of this exercise is to get students to build a working application which performs simple encryption. Once the student implementations are tested, they can be used for future exercises.

Crack Simple Ciphers
- Students create a cryptanalysis helper which allows for character, bigram, and trigram analysis.
- Students are given individual ciphertexts, randomly chosen (substitution, permutation, XOR), and asked to build applications which help crack the text - a ciphertext only attack.
- Interestingly enough, some students will have a hard time in counting bigrams and trigrams. We usually include a question on an exam on this topic.
- Approximately 10% of students have difficulty in cracking a random simple cipher - that is, if they are not told the type of cipher.

Caesar Bigram Cipher
- Students put all 676 possible bigrams in "lexicographic" order
- Students create an application which will accept a number from 1 to 675 as a key, and add / subtract this amount (mod 676) to encrypt / decrypt a bigram
- The idea of this exercise is to get students to see the difficulty of cryptanalysis when we move from encrypting one character at a time to more.

Crack Caesar Bigram
- Each student is assigned an individual ciphertext file created with the Caesar Bigram cipher - a ciphertext only attack. Students must write code to find their key.
- Again, students may have trouble in determining how to output their results to find the appropriate key or plaintext.

Playfair Cipher [17, pp. 35-37]
- Encryption or decryption
- The decryption implementation is difficult for some students.

Vigenere Cipher
- Simple Encryption and Decryption, also using Vigenere Auto-Key [17, p. 42]
- Student must create a Vigenere Cryptanalysis Helper which calculates the Index of Coincidence of partitioned ciphertext.
- Students are given individual Vigenere ciphertexts and asked to build applications which help crack the text.
- The encryption and decryption implementations are simple for students, but we have seen some students try to place an entire Vigenere Tableau [17, p. 41] into code!
- The Index of Coincidence implementation is much more difficult, and some students will not be able to crack Vigenere ciphertext without assistance.

Linear Feedback Shift Register (LFSR) as key generator
- Create key stream and then encrypt / decrypt
- This is a simple exercise for students.

Non Linear Combination Generator for Stream Ciphers
- A number of LFSRs are input and combined in a nonlinear fashion to create key stream. Then encrypt / decrypt.
- This builds on the last exercise and can be given in combination with it.

RC-4
- Input key and "shuffle" as defined in RC-4. Then encrypt / decrypt.
- Since RC-4 is well defined in the literature, with pseudocode available from multiple sources, students can usually complete this exercise easily.

Simplified-DES
- Encryption / decryption
- Also, brute force attack - known plaintext.
- Students will not typically have trouble implementing encryption and decryption of S-DES, but it is a great algorithm for the students to help them learn DES.
- The brute force attack might be difficult for some students.

"Partial-DES"
- Students write an implementation which performs one round of DES. Implementation also uses only one DES S-box
- The idea here is to get students to write their own code for DES. Using only one round and one S-box makes it less likely that the students are copying code from online sources. Some students may find this difficult.

AES
- Students implement AES round functions including ByteSub using a modified S-Box
- Students implement one full round of AES

- The idea here is similar to "Partial DES"

RSA Support

- Student implement various mathematical functions which will allow for easier programming of RSA: Finding Primes, Greatest Common Divisor, Extended Euclid Algorithm, Exponentiation Using Square and Multiply Technique
- We have found that asking students for a full RSA implementation without background was problematic - some students did not know even where to start. Hence, we now start students with building functions which can be used in RSA implementation.

RSA Implementation

- User enters plaintext or ciphertext and the students' implementations should calculate appropriate keys and then encrypt or decrypt.
- With the functions given in RSA Support, the students can now complete RSA, even though some will still have trouble.
- We have found that asking students to first input the plaintext or ciphertext, and then compute appropriate keys allows for easier checking of the students' implementations.

Semester Projects involving group work

- Diffie Hellman Key Exchange
- AES (full algorithm)
- DES (full algorithm)
- 3-DES (full algorithm)
- DES Brute Force (unsuccessful)
- Certificate Exchange – Students implement a Public Key Infrastructure
- Each of these projects allowed students to complete full implementations of the algorithms. In the Certificate Exchange project, students were able to define their own certificates and create a small PKI.

Clearly, a large number of experiential learning opportunities are possible in a cryptography course.

V. Conclusion

The typical university Cryptography course is ripe for experiential learning techniques. Such a course should give students an understanding of the theory, algorithms, and applications of cryptology, allowing students to learn via multiple methods. At East Stroudsburg University of Pennsylvania, we have been successful in incorporating experiential learning in this course, as well as other courses where cryptography is studied, and suggest this method for others teaching cryptography. The students enjoy the experiences, and theory suggests that they will better retain this vital element of information assurance curriculum.

VI. References

[1] Committee for National Security Systems; "CNSS Instructions", Retrieved on March 9, 2007 from http://www.cnss.gov/instructions.html.

[2] International Information Systems Security Certification Consortium, "CISSP Common Body of Knowledge", Retrieved on March 22, 2007 from https://www.isc2.org/cgi-bin/content.cgi?category=8

[3] East Stroudsburg University, "East Stroudsburg University Center for Computer Security and Information Assurance", Retrieved on March 9, 2007 from http://www.esu.edu/compusec/.

[4] N. Paul Schembari, "A Bachelor of Science Degree in Computer Security: The Experiences of a National Center of Academic Excellence in Information Assurance Education", *Proceedings from the Ninth Colloquium for Information Systems Security Education*, pp. 6 – 11 (2005).

[5] Sylvia Perez-Hardy, "A Unique Experiential Model for Teaching Network Administration", *Proceedings of the Fourth Conference on Information Technology Curriculum*, pp. 119 – 121 (2003).

[6] Sunil Hazari, "Instructional Strategies for a Graduate Level Information Security Management Course", *Proceedings of the 1st Annual Conference on Information Security Curriculum Development*, pp. 71 – 75 (2004).

[7] David Dellacca and Connie Justice; "Building Tomorrow's Information Assurance Workforce Through Experiential Learning", *Proceedings of the 40th Hawaii International Conference on System Sciences*, 2007. Retrieved on March 10, 2007 from http://csdl2.computer.org/comp/proceedings/hicss/2007/2755/00/27550271c.pdf

[8] Paul J. Wagner and Jason M. Wudi, "Designing and Implementing a Cyberwar Laboratory Exercise for a Computer Security Course", *ACM SIGCSE Bulletin*, Vol. 36, Issue 1, pp. 402 – 406 (2004).

[9] Ed Crowley, "Experiential Learning and Security Lab Design", *Proceedings of the Fifth Conference on Information Technology Education*, pp. 169 – 176 (2004).

[10] D.A. Kolb, *Experiential Learning*, Prentice Hall Publishing (1984).

[11] J. E. Stice, "Using Kolb's Learning Cycle to Improve Student Learning", *Engineering Education*, 77, pp. 291-296 (1987).

[12] "Hands-On 7: Cryptography and Certificates", Retrieved on March 10, 2007 from http://web.mit.edu/6.033/www/assignments/handson-crypto.html

[13] "Assignment 2 – Elementary Cryptography", Retrieved on March 10, 2007 from http://csc.colstate.edu/summers/NOTES/6126/lab2.html

[14] James W. Benham, "Laboratory Exercises in Encryption", Retrieved on March 10, 2007 from http://www.csam.montclair.edu/~benham/enclabs/

[15] "AES Lab", Retrieved on March 10, 2007 from http://www.tele.pitt.edu/~telelab/labs/TELE%202820/pdf/TELE2820~AES%20Lab~07.08.05.pdf

[16] Lawrie Brown, "Cryptography Labs", Retrieved on March 10, 2007 from http://www.itee.adfa.edu.au/coursework/ZITE3102/labs/labs.html.

[17] William Stallings, *Cryptography and Network Security*, Third Edition, Prentice Hall Publishing (2003).

[18] "PuTTY: A Free Telnet/SSH Client", Retrieved on March 22, 2007 from http://www.chiark.greenend.org.uk/~sgtatham/putty/.

[19] W. F. Friedman, *The Index of Coincidence and Its Applications in Cryptography*, Riverbank Publication No. 22, Riverbank Labs (1922). Reprinted by Aegean Park Press (1987).