PROJECTS MANUAL

CRYPTOGRAPHY AND NETWORK SECURITY

PRINCIPLES AND PRACTICE

SIXTH EDITION

WILLIAM STALLINGS

TABLE OF CONTENTS

All the folders and documents referenced in this manual are at the password-protected online **Instructors Resource Center (IRC)** for this book, or in a few cases, in the **Premium Content** Web site, the latter available to students.

PART ONE SAGE

One of the most important features for this book is the use of Sage for cryptographic examples and homework assignments. Sage is an open-source, multiplatform, freeware package that implements a very powerful, flexible, and easily learned mathematics and computer algebra system. A computer algebra system (CAS) is software that can perform symbolic as well as numerical calculations. CASs have been used for teaching since their inception some decades ago, and there is now a considerable literature on their use. A CAS is a natural tool for extending the learning experience in a cryptography course.

Unlike competing systems such as Mathematica, Maple, and MATLAB, there are no licensing agreements or fees involved with Sage. Thus, Sage can be made available on computers and networks at school, and students can individually download the software to their own personal computers for use at home. Another advantage of using Sage is that students learn a powerful, flexible tool that can be used for virtually any mathematical application, not just cryptography. The Sage Web site (http://www.sagemath.org) provides considerable documentation on how to install, set up, and use Sage on a variety of computers and how to use it online via the Web.

The use of Sage can make a significant difference to the teaching of the mathematics of cryptographic algorithms. Appendix B provides a large number of examples of the use of Sage covering many cryptographic concepts. Appendix C lists exercises in each of these topic areas to enable

the student to gain hands-on experience with cryptographic algorithms. Copies of the code for both appendices are available online so that students do not have to key in lines of code that are printed in the appendices. They are in the folder **SageCode** in the Premium Content section of this book's Web site.

The IRC contains solutions to all of the exercises in Appendix C in the folder **SageExercises**.

Dan Shumow of Microsoft and the University of Washington developed all of the examples and assignments in Appendices B and C.

PART TWO HACKING PROJECTS

The aim of this project is to hack into a corporation's network through a series of steps. The Corporation is named Extreme In Security Corporation. As the name indicates, the corporation has some security holes in it and a clever hacker is able to access critical information by hacking into its network. The instructor's resources include what is needed to set up the Web site. The student should start at the web site and find his or her way into the network. At each step, if the student succeeds, there are indications as to how to proceed on to the next step as well as the grade until that point.

In addition to the project, some exciting exercises on Server-side and Client-side scripting vulnerabilities are also included. Practice with these exercises will help learn the concepts better and make it easier to finish the project.

The instructors supplement includes the files needed for this project in the folder **HackExercises**:

1. Web Security project named extremeinsecure (extremeinsecure.zip)
2. Web Hacking exercises (XSS and Script-attacks) covering client-side and server-side vulnerability exploitations respectively (webhacking.zip)
3. Documentation for installation and use for the above (description.doc)
4. A PowerPoint file describing Web hacking (Web_Security.ppt). This file is crucial to understanding how to use the exercises since it clearly explains the operation using screen shots.

This project was designed and implemented by Professor Sreekanth Malladi of Dakota State University.

PART THREE BLOCK CIPHER PROJECTS

This is a lab that explores the operation of the AES encryption algorithm by tracing its execution, computing one round by hand, and then exploring the various block cipher modes of use. The lab also covers DES. In both cases, an online Java applet is used (or can be downloaded) to execute AES or DES.

For both AES and DES, the lab is divided into three separate parts:

- **Block cipher internals:** This part involves encrypting plaintext and analyzing the intermediate results after each round. There is an online calculator for both AES and DES that provides the intermediate results and the final ciphertext.
- **Block cipher round:** This part involves calculating one round by hand and comparing the results to those produced by the calculator.
- **Block cipher modes of use:** Enables the student to compare the operation of CBC and CFB modes.

All of the material for the AES lab is in the folder **AESCalc** inside the folder **BlockCipherProjects** in the IRC download. The file AESlab.html contains the actual student assignments. You can load the .html and .jar files needed to set up these labs on your own Web site, or allow students to access it at http://williamstallings.com/Crypto/AESCalc/AESlab.html . For part (a) of the lab, each student is assigned a triple (key, plaintext, ciphertext). You can assign a triple from the file **AES-Triples.doc** or generate random triples using the GenAES program, which is also included in the JAR file, and which generates *n* random triples when run as:

java -cp AEScalc.jar GenAES *n*

All of the material for the AES lab is in the folder **DESCalc** inside the folder **BlockCipherProjects** in the IRC download. The file DESlab.html contains the actual student assignments. You can load the .html and .jar files needed to set up these labs on your own Web site, or allow students to access it at http://williamstallings.com/Crypto/DESCalc/DESlab.html . For part (a) of the lab, each student is assigned a triple (key, plaintext, ciphertext). You can assign a triple from the file **DES-Triples.doc** or generate random triples using the GenAES program, which is also included in the JAR file, and which generates *n* random triples when run as:

java -cp DEScalc.jar GenDES *n*

Lawrie Brown of the Australian Defence Force Academy developed these projects.

PART FOUR LABORATORY EXERCISES

Professor Sanjay Rao and Ruben Torres of Purdue University have prepared a set of laboratory exercises that are part of the instructor's supplement in the folder **LabExercises**. These are implementation projects designed to be programmed on Linux but could be adapted for any Unix environment. These laboratory exercises provide realistic experience in implementing security functions and applications.

PART FIVE RESEARCH PROJECTS

A suggested project assignment form is the document **ProjectForm**, included in the folder **ResearchProjects**. The form includes guidelines for a final report and a set of slides. If the deliverable is slides, class time should be set aside for each team to present its results.

If projects are assigned to teams rather than individuals, problems can arise. A paper on this subject, in the document **Team.pdf** included in the folder **ResearchProjects**, might be useful to the instructor.

Here are some ideas for research project topics:

Session Keys

Examine the establishment of session keys such that the session key is uncomputable, and unspoofable. You may wish to study systems that rely on:

- A shared secret
- Authenticated public keys (including Diffie-Hellman)
- A single public key
- One time passwords (including Lamport's Hash)

To better understand how these methods are used, study their implementation in various protocols/products, such as:

- Kerberos V4

- Kerberos V5
- Secure Socket Layer

Show attacks and defenses to these session key establishment protocols at all levels, including one or both of the communicating machines being compromised.

An Evaluation of SDSI: A Simple Distributed Security Infrastructure

SDSI is a proposal for a public-key security infrastructure designed to provide an efficient and secure means of defining group membership and certification of such groups that is being developed by Ronald Rivest of MIT and Butler Lampson of Microsoft.

Some of the design goals of the SDSI proposal are:

- To design a public-key infrastructure that is simpler than existing proposals (such as X.509-based schemes) by not requiring global certificate hierarchies.
- To borrow from and expand upon similar design efforts (such as that of the IETF SPKI: Simple Public-Key Infrastructure working group)
- To provide ideas and techniques that facilitate the construction of secure systems by providing simple clear data structures and emphasizing clarity and readability at the expense of economical encodings, although efficient representations of its data structures are provided.

For this project

- Provide a functional description of SDSI.
- Identify issues with the way it handles group-membership and certification.

- Identify its strengths and weaknesses with respect to it actually being placed into general use (e.g., complexity and performance issues).

http://theory.lcs.mit.edu/~cis/sdsi.html

Key Escrow System

Investigate key escrow and its implementations by government and commercial groups. Key escrow system (KES) can be defined as an encryption system with a decryption capability that allows authorized persons (e.g. officers of an organization and government officials) to decrypt the ciphertext with the help of trusted parties holding special data recovery keys. The aspects that will be covered about key escrowed system will be security, its operation, and the security measures taken in programming the chips.

In addition to a general discussion, examine the Escrowed Encryption Standard (EES), Yaksha (commercial key escrow) , and Entrust (commercial key recovery). Then describe the philosophical and technical approaches of each one.

The Security of Diffie-Hellman Algorithm.

The security of Diffie-Hellman relies on the difficulty of the discrete logarithm problem. This project describes attempts to determine what size primes are required for security. In particular, evaluate the secure identification option of the Sun Network File System, which uses Diffie-Hellman algorithm with a prime p of 192 bits.

Objectives:

- To describe the different steps necessary to solve the discrete logarithm problem (DLP).
- To discuss the state-of-the-art results obtained for the solution of DLP.
- Present the SUN NFS cryptosystem, and discuss some of its deficiencies.
- Recommendations in order to have a secure size for the prime number p used in the Diffie-Hellman algorithm.

Fault Based Cryptanalysis

Starting in the fall of 1996, a flurry of research announcements and preprints followed a prominent New York Times article extolling a fault-based attack model on the security of computing devices developed by Bellcore researchers. The attack was referred to as Differential Fault Analysis. Search the web for articles and commentary on this attack, and provide a survey of results and algorithms.

GSM Security and Encryption

The motivation for security in cellular telecommunication systems is to secure conversations and signaling data from interception as well as to prevent cellular telephone fraud. Investigate the security system embedded in GSM (Group Special Mobile) system, which is a European standard, is currently in use on almost every continent. Topics to cover: overview, authentication, signaling and data confidentiality, subscriber identity confidentiality, encryption algorithms, and conclusions.

http://www.gsmworld.com

Miscellaneous

Some of these are also suitable for programming projects. Investigate and then demonstrate and/or report on one of the following:

- A basic, intermediate, advanced or esoteric cryptographic protocol, such as authentication, key distribution, secret splitting, secret sharing (threshold schemes), time stamping, bit commitment, fair cryptosystems, key escrow, zero-knowledge, secure elections, or digital cash (See [SCHN96], Chapters 3-6 and 23 for additional information.)
- The generation of (pseudo)random numbers used for keys, IVs, and other cryptographic parameters.
- A cryptographic-based network security protocol or application, or extension thereof, such as SHTTP, SSL, WinPCT (compare with SSL), SSH, IPSEC, TLSP, NLSP, MSP, Netware, KryptoKnight, SNMP Security,

Lotus Notes, DCE, NFS, NIS, RPC, Java, JavaSoft, ActiveX, iKP, SET, or Secure Courier.

- The development of public-key infrastructures, including efforts such as X.509, PKIX, SPKI, SDSI, or MISSI.
- The unique issues involved in the application of cryptographic techniques in a special telecommunications environment, such as video, speech, mobile communications, or teleconferencing.
- Cryptographic applications programming interfaces, such as the Internet GSS-API, Open Group GCS- API, Microsoft CryptoAPI, and/or RSA Labs Cryptoki.
- The use of formal techniques (e.g., BAN logic) to analyze cryptographic protocols.
- The relative advantages and disadvantages of implementing cryptographic protocols at the application layer (layer 7), network layer (layer 3) and transport layer (layer 4) of the OSI basic reference model. Perform as a group project, culminating in a panel-style presentation, with each member arguing for a different layer.

PART SIX PROGRAMMING PROJECTS

The guidelines discussed in Part Five concerning research projects can apply for programming projects as well. You may also wish to specify how an implementation is to be presented or demonstrated.

Anonymous Message Broadcast

Anonymous message broadcast is a scheme based on the famous Dining Cryptographers Problem. The purpose of anonymous message broadcast is to enable clients to be able to communicate among one another without divulging the identity of who said what. All the chat participants are, however, aware of who the other participants are. A useful application of this type of system can be having discussion between, for example, a manager and her employees, in which none of the employees wanted the manager to know who said what. Although the manager maybe able to guess, she will not have a way of conclusively proving it (i.e. by performing some kind of IP trace).

Implement a system composed of two components: the Server software and the Client software. The Server software is where the communication is hosted and clients must connect and authenticate with the server in order to participate in a discussion. The client software implements the mechanism required for the client to anonymously communicate with the server. Discussions are formed in groups that are pre-configured on the server. In order for the discussion to take place all the group members must be present. The reason this is done will become clear later on in the discussion.

The group is assigned a single password that all members must know in order to be able to authenticate with the server. A client has to enter their name as well in order to initially identify himself with other group members so that everyone is aware of who is currently present in the discussion.

Anonymous message broadcast is described in [SCHN96], p. 137.

Firewall Implementation

The purpose of this project is to provide a prototype implementation of a firewall, based on RFC 1928 (SOCKS Protocol Version 5). The firewall toolkit from Trusted Information Systems may be used as a starting point. The firewall should handle forwarding of Telnet, HTTP, FTP, and SMTP traffic.

http://www.fwtk.org/

Digital Cash

This project is based on the protocol number 4, described in [SCHN96], p. 142. Basically this protocol implements an electronic cash system, in which the digital cash cannot be copied or reused more than once and the privacy of the customer's identity is guaranteed.

Implementation: The system allows money transaction between three parties: Customer, Merchant and Bank. The electronic cash (ecash) used during these transactions is a file that contains:

- the amount of the transaction involved;
- an uniqueness string number;

- identity strings which contain the identity of the customer (this information remains secret unless the customer tries to use the ecash illicitly – more than once);
- bank's signature (before the customer can use the ecash) The services provided for each party is described as follows:
- **Customer**
- generates N orders for each money order the customer wants to make and assigns a different random uniqueness string number for each of the N ecash money orders
- implements the secret splitting and bit commitment protocols used to generate the identity strings that describe the customer's name, address and any other piece of identifying information that the bank wants to see.
- implements a blind signature protocol for all N money orders
- automatically complies to reveal the half of the identity string chosen by the merchant
- **Merchant**
- verification of the legitimacy of the bank's signature
- random generator of the selector string, which determines the half of the identity string the customer is required to reveal
- **Bank**
- random choice of 1 out of N money orders sent by the customer to remain unopened
- an algorithm that certifies that all the N-1 money orders have been filled with valid information
- a procedure to certify that the orders received from merchants have not been used previously and storage of the uniqueness string and identity strings of the orders in a database file
- Appropriate measures against reuse of the ecash

Virtual Election Booth

This project implements the secure election protocol described in [SCHN96], p. 127 (Voting with Two Central Facilities). A more theoretical discussion is found in [SALO96]. The implementation will provide a secure way for people to vote online, which eliminates the hassle of physically being present at designated election locations.

Since computerized voting will not replace general elections unless there is a protocol that both maintains individual privacy and prevents cheating, the ideal protocol must meet these requirements:

- Only authorized voters can vote.
- No one can vote more than once.
- No one can determine for whom anyone else voted.
- No one can duplicate anyone else's votes.
- Every voter can make sure that his vote has been taken into account in the final tabulation.
- Everyone knows who voted and who didn't

Your design should use two central facilities: Central Tabulating Facility (CTF) and Central Legitimization Agency (CLA). CLA's main function is to certify the voters. Each voter will send a message to the CLA asking for a validation number, and CLA will return a random validation number. The CLA retains a list of validation numbers as well as a list of validation numbers' recipients to prevent a voter from voting twice. Then, the CLA completes its task by sending the list of validation number to the CTF. CTF's main function is to count votes. CTF checks the validation number against the list received from the CLA. If the validation number is there, the CTF crosses it off (to prevent someone from voting twice). The CTF adds the identification number to the list of people who voted for a particular candidate and adds one to the tally. After all the votes have been received, the CTF publishes the outcome.

An effective way to implement this is via the web, using CGI programs to implement CTF and CLA, and a Java applet to do encryption on the client side.

Digital Certified Mail

This project implements the digital certified mail scheme described in [SCHN96], p. 122; a more theoretical discussion is found in "A Randomizing Protocol for Signing Contracts," by Even et. al., *Communications of the ACM*, June 1985. The objectives of digital certified mail are (1) to require that a recipient sign for a message prior to receiving it, and (2) to prevent a sender from forging a receipt.

The solution involves several exchanges of keys to ensure that when the sender has a valid receipt, the recipient will also have a decrypted version of the text . Both sides must generate random DES keys, and encrypt receipts and one bogus message (to be sent by the Sender). Both sides will transfer the keys to each other using the oblivious transfer protocol.

Web Based Secure Purchase Order

Implement a secure purchase order system that allows the user to enter a purchase request and routes it (by secure email) to a supervisor for signature and then to the purchasing department.

- All user interactions will be Web-based.
- All connections between parties will be preceded by public-key mutual authentication.
- The signatures of both the purchaser and the supervisor will be public key based, and will be performed on a hash of the purchase order. The

signature of the purchaser will be sent to both the supervisor and the orders department along with a timestamp. If an order is approved by the supervisor, the orders department can cross-check the digest signed by the supervisor with the digest signed by the purchaser. The signature and time-stamping is obviously important in preventing repudiation. I am purposely ignoring the possibility that a user will "publish" their key to back up a repudiation. Ideally, the user's key will not be easily accessible and, since the whole process takes place in one organization, the possible means of revealing a key are very limited. The biggest threat is a user using another user's machine the forge an order.

- All messages will be encrypted using RSA public-key cryptography. Depending on performance (and time) this might be optimized by using RSA to only send a one-time secret key.

Poker on the Internet

This project implements poker on the Internet. It will accept one to five players. More than five players tend to spread the deck out too much.

Every player will need a public/private key to play. They will also need to have an account with the house. Startup will have to be coordinated amongst the players. Session keys will be established between the house and the players. The session key is used to encrypt players. cards and for players to sign their bets. The house is responsible for totaling the pot and announcing bets to all players and tracking players' banks. Session keys will be destroyed after a players leaves a current session.

Implementation of a Symmetric Block Cipher

Implement one of the more straightforward block ciphers, such as RC5 or Tiny Encryption Algorithm (http://143.53.36.235:8080/tea.htm). Verify that the implementation works, then perform some experiments, such as timing trials, exhaustive key search, randomness checks, etc.

Miscellaneous

- Compare several different implementations of DES, RSA, and/or other cryptographic algorithms. Develop and conduct performance analysis of the implementations, either as a whole, or in parts. Measure the key generation and encryption/decryption times. Measure independent internal components of the algorithm. Compare observed performance differences and differences in implementation techniques.
- Using an implementation of DES or some other block cipher, demonstrate the encryption and decryption processes using the ECB, CBC, K-bit CFB, K-bit OFB, and other modes of and/or multiple encryption. Show normal operations, the effects of bit errors on transmitted ciphertext including error extension, and the effects of block boundary errors and re-synchronization.
- Develop computer programs that automatically cryptanalyze simple encrypted files, e.g., a simple polyalphabetic cipher, or an encrypted WordPerfect file.
- Develop and demonstrate a secure telephone by integrating freely available computer programs for voice or audio communication and encryption. Develop and implement a secure communications protocol including encryption and/or authentication, and key management.

PART SEVEN PRACTICAL SECURITY ASSESSMENTS

Examining the current infrastructure and practices of an existing organization is one of the best ways of developing skills in assessing its security posture. Students, working either individually or in small groups, select a suitable small- to medium-sized organization. They then interview some key personnel in that organization in order to conduct a suitable selection of security risk assessment and review tasks as it relates to the organization's IT infrastructure and practices. As a result, they can then recommend suitable changes, which can improve the organization's IT security. These activities help students develop an appreciation of current security practices, and the skills needed to review these and recommend changes.

The document **PracticalAssessments** included at the IRC provides detailed guidance.

PART EIGHT FIREWALL PROJECTS

The implementation of network firewalls can be a difficult concept for students to grasp initially. The IRC includes Network Firewall Visualization tool to convey and teach network security and firewall configuration. This tool is intended to teach and reinforce key concepts including the use and purpose of a perimeter firewall, the use of separated subnets, the purposes behind packet filtering, and the shortcomings of a simple packet filter firewall.

The IRC includes the files needed for the case study assignments in the folder **`Firewall Tool and Exercise`**. The folder also contains a copy of the paper that discusses the case studies: "Network Firewall Visualization in the Classroom," by Warner, et al, *Journal of Computing Sciences in Colleges,* December 2010.

The IRC includes a .jar file that is fully portable, and a series of exercises. The tool and exercises were developed at U.S. Air Force Academy.

PART NINE CASE STUDIES

Teaching with case studies engages students in active learning. The IRC includes case studies in the following areas:

- Disaster recovery
- Firewalls
- Incidence response
- Physical security
- Risk
- Security policy
- Virtualization

Each case study includes learning objectives, case description, and a series of case discussion questions. Each case study is based on real-world situations and includes papers or reports describing the case.

The IRC includes the files needed for the case study assignments in the folder **CaseStudies**. The folder also contains a copy of the paper that discusses the case studies: "Case Studies for Teaching Physical Security and Security Policy," by Yuan, et al, *Proceedings, InfoSecCD '10 2010 Information Security Curriculum Development Conference*.

The case studies were developed at North Carolina A&T State University.

PART TEN WRITING ASSIGNMENTS

Writing assignments can have a powerful multiplier effect in the learning process in a technical discipline such as cryptography and network security. Adherents of the Writing Across the Curriculum (WAC) movement (http://wac.colostate.edu/) report substantial benefits of writing assignments in facilitating learning. Writing assignments lead to more detailed and complete thinking about a particular topic. In addition, writing assignments help to overcome the tendency of students to pursue a subject with a minimum of personal engagement, just learning facts and problem-solving techniques without obtaining a deep understanding of the subject matter.

The suggested assignments are in the document **Crypto6eWriting.doc**, organized by chapter. Instructors may ultimately find that this is the most important part of their approach to teaching the material. I would greatly appreciate any feedback on this area and any suggestions for additional writing assignments.

The following suggestions for using a variety of writing assignments are taken from "Writing Assignments: Pathways to Connections, Clarity, Creativity", R. Brent and R. Felder, *College Teaching*, 40(1), 1992.

- Don't set out to do it all at once. Start by trying one or two of the suggested assignments that appeal to you; then gradually add new ones to an extent that seems appropriate and comfortable.
- Clearly relate the assignments to the course content, and be sure students understand the connections.

- When you decide to use a particular type of assignment (e.g., brainstorming, critical question generation, problem formulation), try it at least three times. The first time that you ask students to do something unfamiliar they are likely to ignore you, as though not believing anyone could want them to do anything that bizarre. The second time they will take the assignment seriously, but many will miss the point. By the third time, you will start to see the sought-after results.
- All of the given writing exercises may be completed in or out of class. In either case, set aside class time for brief sharing and discussion of the responses in groups of three to five. Such discussions expose the students to a wider variety of ideas than most could have thought of individually and reduce or eliminate the need for the instructor to read and critique all the papers.
- Many of the suggested assignments require no feedback from the instructor; the process is more important than the product. But for those assignments that do result in a product (make up a problem, develop critical questions), make sure that the students get constructive feedback on their initial efforts, including suggestions on how they could have done it better.
- The feedback may come from you (initially it will have to), but the students may also provide it to one another after they have gained some experience and understand the object of the exercise.
- Ask students to evaluate particular assignments and the writing experience as a whole. Make changes in assignments that do not seem to be accomplishing their objectives.
- Write articles about assignments that work particularly well so that other teachers can get the benefit of your experience.

Part Eleven Reading/Report Assignments

Many professors like to assign papers for the students to review; this helps to keep them aware that cryptography and network security is a changing field, and that the now-accepted wisdom was once un-formulated. I've suggested some papers organized chapter, but there are many more good ones.

The following is a suggested assignment wording:

> Read and report on the following papers from the research literature. Your report should be one to two pages long; three-quarters of the report should summarize the paper, and one-quarter of the report should be a critique. Introduce your report with a formal citation of the paper, using the format found in the References section of the textbook.

You might also suggest to your students that they follow a format roughly similar to that found at the Treasure Trove of Paper Summaries. There is a link from my Computer Science Student Resource Site at http://www.computersciencestudent.com) under the Other Useful tab. Of course, then you must check that they don't use one of the online summaries!

The suggested papers are in the document **CryptoeeReading.doc** included in the IRC. All of the papers listed in this document are in the folder **Crypto6ePapers** in the Premium Content Web site for this book

PART TWELVE DISCUSSION TOPICS

One way to provide a collaborative experience for students is discussion topics. Each suggested topic relates to material in the book. The instructor can set it up so that students can discuss a topic either in a class setting, an online chat room, a wiki, or a message board. Again, I would greatly appreciate any feedback on this area and any suggestions for additional discussion topics.

The following are suggested discussion topics.

1. Just prior to the announcement of the winner in the NIST SHA-3 competition, well-known security expert Bruce Schneier, who co-authored one of the 5 competition finalists, called for the competition to be cancelled and no award made. Bruce reasoned that SHA-2 was sufficient both in terms of security and performance for the foreseeable future. Discuss the pros and cons of adopting SHA-3 or sticking with SHA-2.
2. At first glance, the key wrapping approach to encrypting symmetric keys for key management applications seems unnecessarily elaborate. The alternative would be a simple encryption using AES or an authenticated encryption using one of the block cipher modes of operation that provides authenticated encryption. Discuss the relative merits of using one of these approaches compared to key wrapping.
3. There are three standardized approaches to digital signatures: RSA-PSS, ECDSA, and the original DSA developed by NIST. One advantage of RSA-PSS is that RSA-based signature schemes are already widely used in commercial applications. One advantage of DSA is that it is the original scheme approved by NIST and is

implemented in numerous packages, especially within the US government and government contractors. One advantage of ECDSA is efficiency. Discuss the relative merits of the three approaches for a new commercial product.

References:

FISH12 Fisher, D. " Forthcoming SHA-3 Hash Function May be Unnecessary." *https://threatpost.com/en_us/blogs/forthcoming-sha-3-hash-function-may-be-unnecessary-092412*

SCHN12 Schneier, B. "SHA-3 to be Announced." http://www.schneier.com/blog/archives/2012/09/sha-3_will_be_a.html

PART THIRTEEN ENCRYPTION LAB EXERCISES

The folder **`EncryptionLabExercises`** contains a set of files that can be uploaded to your Web site and provide and Excel-based set of encryption exercises in the following areas:

- RSA key generation, encryption, and decryption
- Key sharing using the Chinese remainder theorem

This material was prepared by Dr. J. W. Benham of Montclair State University.