Crypto and Infosec – Semester Project

Due on APRIL 7, 2015

This is a group project. No solo efforts. Max group size = 4 persons.

> **"Coming together is a beginning.**
> **Staying together is progress.**
> **Working together is success."**
> **-- Henry Ford, Founder, Ford Motor Company**

Project Option #1 Implementation of a Symmetric Block Cipher

Implement one of the more straightforward block ciphers, such as RC5 or Tiny Encryption Algorithm (http://www.cl.cam.ac.uk/ftp/papers/djw-rmn/djw-rmn-tea.html). Verify that the implementation works, then perform some experiments, such as timing trials, exhaustive key search, tests versus known good answers, randomness checks, etc.

Grading: A grade of "A" will be given to a project report that adequately implements the following:

- Detailed written technical report
- Research that includes literature search
- Complete listings and output (as applicable) of any custom code developed for this project.
  - There must be a design plan for the code.
  - Code must be formatted and well-commented.
- Include all relevant references, figures tables, diagrams, etc.

Your can "delight the customer" (i.e., receive extra credit) through providing additional analysis, handsome reporting, etc.

Project Option #2 Implementation/proof of RSA

Purpose: To demonstrate understanding of RSA and its resistance to cryptanalytic attacks

Requirements: Implement RSA in software. Show that a 200-digit (667-bit) RSA crypto-system can be implemented on a home computer. You will need to find two large (at least 100-digit) prime numbers p and q to generate the modulus for RSA. Also, show that a 50-digit (167-bit) key can be broken within a week. Since n is known to be the product of two prime numbers, factoring this is sufficient for cracking an RSA secret key. What are the average and worst-case times to factor some (e.g., a few thousand) large (e.g., 40-digit) integers? What makes the times better or worse? Determine how long key needs to be to make the crypto-system secure. You will need to implement the following functions:

- CreateKey - Generate RSA public and secret keys, using a 100-digit prime number generator.
- Encrypt - Take a plaintext perform RSA encrypt on it.
- Decrypt - Take a ciphertext and perform RSA decrypt on it.
- Factor - Implements a factoring heuristic to factor large (in excess of 50-digit) numbers.
- Grading: A grade of "A" will be given to a project report that adequately implements the following:
  1. Finding and multiplying two hundred-digit prime numbers to generate a 200-digit RSA public key and a corresponding secret key.
  2. Encrypting, using the public key, and decrypting, using the secret key, a sample message. To show that the algorithm is implemented properly, it is sufficient to show that E(D(m)) = D(E(m)) = m.
  3. Write a program for factoring large numbers using some factoring heuristic. Determine the average and worst-case times for factoring 40-digit keys. Tell what variables affected these times.
  4. Tell how secure the RSA public-key crypto-system is. That is, applying Moore's Law and

extrapolating, determine how long RSA keys must be for crypto-systems of the future. What key size is needed to be safe from a hacker with a home computer? What key size is need to be safe from thousands of crackers working together over the Internet?

5. Supply any other conclusions you can make from this exercise.

- Your can “delight the customer” (i.e., receive extra credit) through providing additional analysis, handsome reporting, etc.

Project Option #3 Implementation of a secure E-Commerce Web site

Requirements: Implement a secure purchase order system that allows a customer to securely enter a purchase request (i.e., with encryption between client and server), and securely routes it to a supervisor for signature and then to the purchasing department. [Or, an alternative application: A request for computer account, routed to academic advisor and then to Academic Computing.]

Your project needs to explain the type(s) of security chosen and why. I should be able to access your Web site URL and enter a purchase order, then receive a confirmation by email for the order. You should have some means for simulating the supervisory approval of the transaction (credit card verifier, encrypted email routed to supervisor, etc.)

Please include some high-level diagrams or data flow descriptions to aid in understanding of your design. If possible, the program should be in compilable form, written in HTML, C, C++, Java, Perl, other scripting languages as needed for the Web-based front end with adequate commenting. The project and/or build files should be included, along with a ReadMe for building the program. If the program requires multiple clients (i.e., a local and remote machine) to run, please indicate this.

Grading: A grade of "A" will be given to a project which contains a **Report** with an executive summary explaining the problem, the program, and the approach used. The report should also include output listings from the program, screen shots, diagrams, and/or any other data applicable to explaining the design approach and methods used.
Note: You need not "re-invent the wheel." This project can be implemented by integrating off-the-shelf components; in which case, you would still need to provide some front end user interface, a secure transaction facility, and secure communications. Perhaps you will use some scripting language to accomplish this. If you borrow someone else's script, make sure you properly attribute it.

Sample project components needed:

1. Internet access
2. Apache, IIS (or other) Web server set up, script files loaded
3. Web form (Index.html)
4. Script to parse form
5. Script to decrypt
6. Script to encrypt
7. Script to email for supervisor approval
8. Script to email confirmation to client?

Project Option #4 Development/Implementation of custom Steganography program

You will determine a way to hide a file of indeterminate type within another file (either audio such as MP3; video, such as MP4, AVI, or other; or image, such as JPG, GIF or other); in such a manner that it does not significantly alter the appearance of the primary file and secretly conceals the contents of the hidden file. The steg system should also implement an authentication system – i.e., a means to retrieve the file should only be available to those who have the proper authentication code (e.g., UserID and password). Finally, you should analyze your steg program to see how well it hides data without being noticeable, while not also significantly altering the quality of the sound or image file.

Grading: A grade of "A" will be given to a project report that adequately implements the technical requirements and all of the following:

- Detailed written technical report
- Research that includes literature search
- Complete listings and output (as applicable) of any custom code developed for this project.
    - There must be a design plan for the code.
    - Code must be formatted and well-commented.
- Include all relevant references, figures tables, diagrams, etc.
- Analysis of the steg embedment for security and file degradation.

Your can “delight the customer” (i.e., receive extra credit) through providing additional analysis, handsome reporting, etc.

Project Option #5 Development/Analysis of a Pseudorandom Number Generator (PRNG)

John von Neumann once cautioned about the misinterpretation of a PRNG as a truly random generator, and joked: "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin."

However, you will develop a deterministic random bit generator – i.e., a program based on an algorithm for generating a sequence of numbers that approximates the properties of random numbers. Of course, the sequence is not truly random in that it is determined by an initial state based on a seed. You can base your PRNG on a popular scheme such as Blum Blum Shub, Linear Feedback Shift Register, Fortuna, or Mersenne Twister -- however, you must put your own "twist" on it and add your own modifications or methods for seed generation.

Then, you must take output from your PRNG program, and run statistical analysis (e.g., chi-square test, kappa test, other tests) to show how "good" your PRNG is.

Grading: A grade of "A" will be given to a project report that adequately implements the technical requirements and all of the following:

- Detailed written technical report
- Research that includes literature search
- Complete listings and output (as applicable) of any custom code developed for this project.
    - There must be a design plan for the code.
    - Code must be formatted and well-commented.
- Include all relevant references, figures tables, diagrams, etc.
- Analysis of the PRNG for statistical soundness in the following areas:
    - http://en.wikipedia.org/wiki/Pearson%27s_chi-squared_test
    - http://en.wikipedia.org/wiki/Friedman_test

Your can “delight the customer” (i.e., receive extra credit) through providing additional analysis (perhaps for cryptographically “strong” tests), handsome reporting, etc.