Cryptography and Network Security

For this semester, you will do an assignment.

1. Talent students have to do a presentation assignment. Each group is formed of at most 2 students.
2. Other students have to do one from programming assignments and presentation assignments as a group. Each group is formed of at most 3 students.
3. Deadline of deciding your assignment (send email to us: dat@cse.hcmut.edu.vn):
   Fall semester: Nov 20th; Spring semester: Mar 5th.
4. Presentation should be sent email to us 4 weeks before the end of semester.
5. The time slot of presentation will be given to you 3 weeks before the end of semester.
6. Source code and final report should be submitted 1 weeks before the end of semester.

I. Presentation assignments

1. List of Possible Presentation Topics

A list of possible topics (you have to choose one from the list, unless you really convince me that you have an excellent topic that is not included here).

a. Some Interesting topics

- Cryptographic security of reactive systems -- linking cryptography and formal methods
- Direct anonymous attestation (DAA)
- Distributing trust on the Internet / SINTRA
- Enterprise privacy technologies
- Federated identity management
- idemix - pseudonymity for e-transactions
- Privacy-enhancing crypto
- Regulations Expressed As Logical Models (REALM)
- Secure networked storage
- A Practical Dynamic Buffer Overflow Detector
- A Comparison of Publicly Available Tools for Dynamic Buffer Overflow Prevention
- Practical Network Support for IP Traceback
- An Effective Defense Against Spoofed Traffic
- Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense
- Monitoring and Early Warning for Internet Worms
- An Effective Architecture and Algorithm for Detecting Worms with Various Scan Techniques
- SOS: An Architecture for Mitigating DDoS Attacks
- Alliance Formation for DDoS Defense
- Defense and Detection Strategies against Internet Worm
- Security issues in cloud computing
- Security issues in wireless sensor networks and RFID systems

b. The topics are papers from ACM CCS conference 2010, 2009 and 2008:

- You can present papers from these conferences also.
2. **Reports format**

   a. **Presentation format**

   Your presentation should be done in electronic format (either in PPT or PDF or any other proper format). Make sure that your slides are clean, concise, and easy to follow. During your presentation, please do **not** just read your slides directly. Do NOT use the whiteboard a lot (although you can use it occasionally). The number of slides should be proper such that you or your group can finish the talk in about 15-20 minutes. Practice your talk before your formal presentation. Do not expect to give good presentation without any practice.

   b. **Final report format**

   • You choose a topic from a list of possible topics given by me, or you come with some new topics with my approval. Then you start to do some research on the topic: surfing the web, going to the library and so on. Then you write a report about the topic. The report will be a 10 to 15 pages research paper. The paper should be in postscript, PDF, or WORD format.

   • Typically, your report might be organized as follows:
     1. abstract --- summarize what your research project did
     2. introduction --- background information, literature search, scope and limitations of assignment
     3. body --- methods used
     4. analysis and conclusions
     5. recommendations --- future work, unsolved problems
     6. references --- journals, WWW references

   • Your final report should look like a technical report in ACM format or in IEEE format.


   • Here are some links on writing that may help you in preparing your proposal and final report write-up. For those of you pursing research careers I strongly suggest that you take a look at these links if you have not come across them already:

   1. Advice on research and writing: [http://asarefin.wordpress.com/2009/07/05/advice-on-research-and-writing/](http://asarefin.wordpress.com/2009/07/05/advice-on-research-and-writing/)

II. **Programming assignments**

1. **List of programming assignments**

   a. **Programming 1**

   In this Programming assignment you will build a simple cryptography program in a programming language of your choice. Then you will generate a few ciphertexts. Finally you will try to crypt-analyze the ciphertexts. For simplicity, we assume that the input alphabet is {a-z, A-Z, 0-9} plus a special empty space character. Your program shall perform the following functions:

   • From a plaintext produce an monoalphabetic substitution cipher. In other words, your key is a substitution rule for each possible input character.

   • From a plaintext produce a transposition cipher. For uniform encryption by all students, we assume that the cipher will work on a block of 8 characters. In other words, you always permute within a block of 8 characters.

   • From a plaintext produce a product cipher based on the previous functions. For simplicity, assume
that the substitution cipher is used first and then the transposition cipher is used to encrypt the
result to get the final ciphertext.

- Generate one cipher-method with each of the three functions (three ciphers in total). For each
cipher method (with a fixed key) you encrypt some arbitrary plain-text that contains the words
"computer" and "security" inside the plaintext. Each text should contain at least 1000 characters and
be of normal type.
- After you produced the ciphertexts using each of the 3 encryption methods (with different keys),
you now start to design methods to find the original plaintext using the given ciphertext. Begin with
the simplesst (1) and continue with (2) and (3).
  1. Build your own tools or use ready-made tools or scripts to cryptanalyze the ciphers.
  2. Try to get the plaintext from the ciphertexts.
  3. Try to get the key used.

b. Programming 2

In this Programming assignment you will have to implement RSA encryption in Java/C++. You
cannot use existing RSA implementations found from web or in Java. What you can use are:
- **Java** has a built-in BigInteger class to store big integers you needed for RSA (such as finding
large prime numbers).
- **C++** has a library as **NTL** (Library for doing Number Theory) or **GMP** (the GNU Multiple
Precision Arithmetic Library).

In other words, you can use these big-integer implementation to manage your data and do module
operation, but **not** use the existing implemented methods (gcd, power, finding prime numbers, and
so on). You have to implement these functions yourself. You can use existing secure function to
produce large random numbers (some functions provided for random numbers cannot be used due
to its weak security). Notice that Java provide tools to get random numbers in java.util.random or
java.security.SecureRandom. Similarly, C++ have **rand()** and **srand()** to generate random numbers.
You can use the random number function provided by Java if implementing random numbers is
really difficult for you. However, these methods there are not secure since Linear congruent method
is the default method set for Java's two built in random number generators. So to enhance security,
you are strongly recommended to implement your own good random number generator.

In your own RSA implementation, assume that the large prime numbers are at least 500 bits (but
could be much larger than this) and you should write several functions yourself:
- A function to find large prime numbers, when given number of bits as an input
- A function to compute **gcd** when given two large integers
- A function to produce a **random** encryption key when given the two large prime
  numbers used for RSA
- A function to compute the **decryption** key when given the encryption key e and the two
  large prime numbers p and q
- A function for encryption when given the message and encryption key e and the modulo
  n
- A function for decryption when given the ciphertext and encryption key e and the
  modulo n

c. Programming 3

In this Programming assignment you will build a simple IAM (Identity and Access Management)
system with open sources. You can use existing libraries or open sources found from web or in the
programming languages such as: JCA/JCE, Open LDAP, Webmin, ...
Your system shall perform the following functions:

- **User management functions**: Create, update, delete, search identity information of users with directory service such as Open LDAP, Active Directory.

- **Authentication user functions** with the following methods: username and password, X.509 digital certificates, personal identification number (PIN).

- **Logging functions** provide the necessary trail to explain authentication events, directory objects modification.

- **Front end**: webUI (HTTP/HTTPS) for user management, logging.

**d. Programming 4**

In this Programming assignment you will have to build a global firewall system with open sources. You can use open sources found from web such as Shorewall, Squid, SquidGuard, DansGuardian, ClamAV, Webmin, …

Your system shall perform the following functions:

- **Firewall**: stateful packet filtering, no limit on the number of network interfaces, multiple interfaces per zone and multiple zones per interface permitted, flexible address management (NAT, PAT, Transparent).

- **Web filtering**: URL/Keyword/Phrase Block, URL Exempt List, blocks Java Applet, Cookies, Active X.

- **Antivirus**: support HTTP/SMTP/POP3/IMAP/FTP/IM and automatic virus database update.

- **AntiSpam**: Real-Time Blacklist/Open Relay Database Server, keyword/Phrase Filtering, IP Address Blacklist/Exempt List, Automatic Update.

- **Management**: WebUI (HTTP/HTTPS) for system management.

**2. What you have to submit**

- Source code and user guide.

- A final report about the assignment. The report will have 20 to 30 pages. The paper should be in postscript, PDF, or WORD format.