University of Maryland
CMSC414 — Computer and Network Security
Professor Jonathan Katz

Problem Set 1

Due at *beginning* of class on Sept. 23

1. (**10 points.**) (From Bishop.) Classify each of the following as a violation of confidentiality, integrity, availability, or some combination thereof.

   - John copies Mary's homework.
   - Carol changes the amount of Angelo's check from $100 to $1000.
   - Rhonda registers the domain name "Cocacola.com" and refuses to the the soft drink company buy or use that domain name.
   - Jonah obtains Peter's credit card number and had the credit card company cancel the card and replace it with another card bearing a different account number.

2. (**10 points.**) (From Bishop.) Give an example of a situation in which the following is true:

   - Prevention is more important than detection or recovery.
   - Detection is more important than prevention or recovery.
   - Recovery is more important than detection or prevention.

3. (**Programming exercise — 30 points.**) You will create a program `decrypt.java` that will decrypt any ciphertext which was encrypted using a shift or a Vigenere cipher. You must write the entire program yourself, and may not get any part of it from other sources.

   For each step below, answer the question and make sure your program implements your answer. You should use the letter frequency data from Bishop, Figure 9.1.

   (a) Your program should take one command-line input, which specifies a file containing the ciphertext. Your program should count the instances of each letter in the input file (you may ignore non-alphabetic characters, and should not distinguish between upper- and lower-case). Based on this data, how can your program determine whether a substitution cipher or a Vigenere cipher was used?

   (b) If the ciphertext was encrypted using a substitution cipher, how can you check whether a shift cipher was used (recall that a shift cipher is a special type of substitution cipher)? Assuming a shift cipher was used, how can your program then decrypt the ciphertext?

   (c) If the ciphertext was encrypted using a Vigenere cipher, how can your program determine the length of the key that was used? Once you have determined the key length, how can your program decrypt the entire ciphertext?

(d) (**Summary of requirements.**) Your program should read a ciphertext from the file specified on the command line. The ciphertext will be encrypted using either a shift cipher, a substitution cipher, a Vigenere cipher, or none of the above. (1) Your program should output (to stdout) which encryption method was most likely used; (2) if the encryption method is a shift cipher or a Vigenere cipher, your program should output the key and the plaintext (also to stdout).

(e) (**Extra credit.**) Modify your program so it can also decrypt ciphertexts formed using a substitution cipher. This is not necessarily a hard problem, but it is hard to do *efficiently* and will likely take a fair amount of time to implement.

4. (**Programming exercise — 50 points.**) You will create a program to encrypt data using the private-key algorithm DES. If you have any questions about implementing this program, do not hesitate to speak to the TAs.

   - You should have a program called `keygen.java`. This program should generate a random DES key and write it to a file called `key.txt`. The key should be represented in hexadecimal format. (Recall that a DES key is 64 bits long, yet only 56 of these are random; the remainder are check bits. Make sure that you generate a *random* but valid DES key.)
   - You should have a second program `DESencrypt.java`. This program should read from the files `key.txt` and `plaintext.txt`. The plaintext file will consist of ASCII text. The key file will contain a 64-bit DES key in hexadecimal format. Your program should also support one command-line flag: "-mode XXX", where XXX is one of ECB, CBC, OFB, or CFB.

     The program should: (1) read the plaintext file and interpret the ASCII characters as bits (where each character maps onto its 8-bit ASCII value). You may assume that the total number of bits is a multiple of 64 (i.e., the total number of characters is a multiple of 8). For this problem, you should be able to work with any ASCII characters as is, so do not ignore whitespace and do not treat upper- and lower-case as the same; (2) encrypt the plaintext, using DES, the key stored in `key.txt`, and the mode of encryption designated on the command line. For the case of CBC, CFB, and OFB, make sure to generate a *random* IV; (3) output the complete ciphertext to a file `ciphertext.txt`, in hexadecimal format.
   - You should have a third program `DESdecrypt.java`. This program should read from the files `key.txt` and `ciphertext.txt`, and should also support a command-line flag, exactly as for the case of encryption. This program should reverse the above procedure (when using the same key file, of course).