CS558 Assignment 4
Due: 11:59pm July 3 (Sunday)

This assignment is done individually.

Students need to complete both PART I and PART II

PART I (5% of the total score)

1. **[20 points]** Consider the following Mandatory Access Control Policy **(TS > S > C > U)**, where u1 is a user, and o1, o2, o3, and o4 are objects.
   Security clearance: (u1, S)
   Security level: (o1, TS), (o2, S), (o3,C), (o4,U)
   Answer the following questions: (1) What objects can u1 read? (2) What objects can u1 write?
2. **[30 points]** In secure electronic transaction, the merchant receives the dual signature **DS**, the order information **OI**, the **pub-key pubc** of the customer, and the **message digest for payment information (PIMD)**. The bank receives the dual signature **DS**, **the payment information PI**, the **public-key pubc** of the customer, and the **message digest for order information (OIMD)**. Note that PIMD = h(PI) and OIMD = h(OI). Answer the following questions: (1) How to compute the dual signature DS from OI, PI, h, and the private key of the customer? (2) How does the merchant verify DS? (3) How does the bank verify DS?
3. **[25 points]** Consider the following SQL query:
   SELECT * FROM u WHERE login = '" + userName + "' and password= '" + password + "';
   Show that the query is vulnerable to the SQL injection attack by providing an input which allows an unauthorized user to access table u.
4. **[25 points]** Consider the following program **(f.c)**:
   ```
   void main(int argc, char **argv) {
           int x = 123;   char buffer[4];
     strcpy(buffer,argv[1]);    printf("%d\n",x);}
   ```
   The output of "**f abcde"** is 169348921. Give an outline explanation of why this happens.

PART II (20% of the total score)

In this assignment, you need to write two programs: genpassword and verifypassword (you can give different names).

genpassword is used to generate a file ***password*** which has the following format:

<user ID> <encrypted password> <time>

where <user ID> is the ID of the user, <encrypted password> is computed by encrypting the password using a key and a symmetric encryption algorithm (e.g. AES, 3-DES), and <time> is the time when the password is created. **You can either hardcode the key in your program, or randomly generate a key and save the key in a file. You can use the existing implementation of 3-DES, or AES (e.g. the implementation provided in java.security, openssl, etc.) in this assignment**.

When genpassword is invoked, it prompts the user who invokes genpassword to enter each user's

ID and password. Your program then encrypts the password using the key, and saves the ID, the encrypted password , and the time when the password was created in a file password. If the ID entered is the same as that stored in the password file, then print "the ID exists" and prompts the user to enter a different ID.

verifypassword is used to verify the ID and the password of a user. When verifypassword is invoked, it prompts the user to enter his/her ID and password. If the ID does not exist in file password, then print "ID does not exist" and prompts the user to enter another ID. Otherwise, your program will retrieve the encrypted password ep of the user from file password. Your program then decrypt ep and compare the password entered by the user against the decrypted password. If they are the same, then print "correct password. The password creation time was <time>" where <time> is the time stored in the password file; otherwise print "incorrect password".

Submission guideline:

- Create a directory with a unique name (e.g. p4-[userid]), which contains the source code, the key (if the key is saved in a file), and a README file.
- **README** file (text file, please do not submit a .doc file) contains
  - Your name and email address.
  - Whether your code was tested on remote.cs.
  - How to compile and execute your program.
  - (Optional) Briefly describe your algorithm or anything special about your submission that the grader should take note of.

-Tar the contents of this directory using the following command.

tar –cvf p4-[userid].tar p4-[userid]

E.g. tar -cvf p4-pyang.tar p4-pyang/

- Upload the tared file you create above, and part1.pdf containing the solution to PART I questions to brightspace.