University of London International Programmes
Computing coursework assignments 2015–16

CO3326 Computer security
Coursework assignment 1

This coursework assignment is designed to help you enrich your learning experience and to encourage self-study and creativity. Assumptions may be added if necessary in your coursework answers to simplify implementation tasks or help with understanding issues. You should, however, attempt the exercises at the end of each chapter in the textbook or subject guide before doing the coursework. Otherwise, you may find the tasks in the coursework assignment difficult and the experience less rewarding. You should read the coursework assignments or questions carefully and pay particular attention to the Submission Requirements on page 3. Note: programme source code may be checked using plagiarism-detecting programs.

Develop a software prototype in Java to demonstrate how the RSA algorithms work using the simplified algorithms and examples in decimal on pages 91–93 CO3326 Computer security subject guide. In particular, your prototype should demonstrate how two primes p and q are generated, how the random number e is generated, where $0 < e < r$ and e has no factor in common with r, and how the private key d and public key (e, n) are generated. As part of testing, a good coursework may also demonstrate a special case when your RSA program would *not* work securely. Your program should prompt the user to input certain parameters that would lead to the problematic state.

There is no specific requirement to the user interface of your prototype but you should design at least a simple user interface to allow the user to simulate a communication scenario, where Alice sends an encrypted message to Bob, and Bob decrypts the ciphertext to read the message. Also, Charlie may intercept the data flow and obtain unauthorised information.

For example, the following format may be adopted to demonstrate what happens with the plaintext m that from Alice to Bob, where "??" parts are for you to design.

Alice		Charlie		Bob
m				
↓				
??	→	??	→	??
				↓
??	←	??	←	??
↓		↓		
??		??		
↓				
??	→	??	→	??
		↓		↓
		??		??

You may decide where to start your design but it would often be easier to first divide the task into a number of subtasks. For example,

1. Implement a cryptorandom key generator and the algorithm for modular exponentiation (algorithm on page 77 of the CO3326 subject guide). This part of the program source code is worth up to [20%] credits.
2. Implement the RSA encryption algorithm. This part of the program source code is worth up to [20%].
3. Implement the RSA decryption algorithm. This part of the program source code is worth up to [20%].

You may add if necessary assumptions for details to ease your implementation, but you must explain them clearly to gain credits.

[END OF COURSEWORK ASSIGNMENT 1]

Submission requirements

These requirements apply to both coursework assignments. The available marks are given in square brackets.

1. Naming conventions for any `.pdf` or `.zip` file submissions
   When naming your files, please ensure that you include your full name, student number, course code and assignment number, e.g. `FamilyName_SRN_COxxxxcw#.pdf`
   (e.g. `Zuckerberg_920000000_CO3326cw2.pdf`), where
   - `FamilyName` is your family name (also known as last name or surname) as it appears in your student record (check your student portal)
   - `SRN` is your Student Reference Number, for example 920000000
   - `COXXXX` is the course number, for example CO3326, and cw# is either cw1 (coursework 1) or cw2 (coursework 2).
2. Your coursework submission must include a report Document [40%] and the program Code [60%].
   The Document (preferable in .pdf format) should include the following sections:
   (a) Algorithms (in flow-chart)
   (b) Design (in block diagram or class-diagram in UML)
   (c) Demonstration (in 5 best screen-shots)
   (d) Discussion (including answers to any questions/problems in the Coursework assignment, your experience in attempt of the coursework, and full bibliography)

   The program code should include the
   (a) Java source codes .java
   (b) executable version .class.
3. Execution of your programs:
   [Penalty] A ZERO mark may be awarded if
   - your program(s) cannot be run from the coursework directory by a simple command '*java menu*' (this means that you should name your main class 'menu', or adopt the `menu.java` that can be found in the Appendix on page 4);
   - your source code(s) does not compile and you give no information on your program execution environment;
   - your program(s) does not do what you claim it should do;
   - your program(s) crashes within the first *three* interactive execution steps;
   - your program(s) works for the first time of execution only;
   - there is no comment in your source code.
4. You should monitor and report the time you have spent for each part of the coursework answers, and leave a note to the examiner if you need to raise any issue at the beginning of your coursework answers as follows:

Total Number of Hours Spent	
Hours Spent for Algorithm Design	
Hours Spent for Programming	
Hours Spent for Writing Report	
Hours Spent for Testing	
Note for the examiner (if any):	

5. Show *all* your work. Any use of others' work should be declared at the point of use and referred to in the *Bibliography* section at the end of your coursework answers.

Appendix

This is an example. Please modify accordingly to suit your own purposes.

```
import java.lang.*;
import java.io.*;
// Modify the display content to suit your purposes...
class menu {
private static final String TITLE =
"\n2910326 Computer Security coursework\n"+
"   by firstname-FAMILYNAME_SRN\n\n"+
"\t********************\n"+
"\t1. Declaration: Sorry but part of the program was copied
from the Internet! \n" +
"\t2. Question 2 \n"+
"\t3. Question 3 \n"+
"\t4. no attempt \n"+
"\t0. Exit \n"+
"\t********************\n"+
"Please input a single digit (0-4):\n";
menu() {
int selected=-1;
while (selected!=0) {
System.out.println(TITLE);
BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
// selected = Integer.parseInt(in.readLine());
try {
                       selected = Integer.parseInt(in.readLine());

                           switch(selected) {
                                case 1:  q1();
                                  break;
                                case 2:  q2();
                                  break;
                                case 3:  q3();
                                  break;
                                case 4:  q4();
                                  break;}      }
   catch(Exception ex) {}  } // end while
               System.out.println("Bye!");
}
// Modify the types of the methods to suit your purposes...
private void q1() {
        System.out.println("in q1");
}
private void q2() {
System.out.println("in q2");
}
private int q3() {
System.out.println("in q3");
return 1;
}
private boolean q4() {
System.out.println("in q4");
return true;
}
   public static void main(String[] args) {
new menu();
   }
}
```

[END OF SUBMISSION REQUIREMENTS]