IS53012C Security and Encryption Coursework[1] 2022–23

This coursework carries 30% of the module grade. The two-minute video presentation carries 10% of the module grade. *You are required to attempt and submit ONE of the assignments only, but welcome to attempt more.*

Assignment I

Develop a software prototype in Java[2] to demonstrate how the RSA algorithms work using the simplified algorithms and examples studied in the lectures/workshops. 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.

$$
\begin{array}{ccccc}
\text{Alice} & & \text{Charlie} & & \text{Bob} \\
\hline
\text{m} & & & & \\
\downarrow & & & & \\
?? & \rightarrow & ?? & \rightarrow & ?? \\
 & & & & \downarrow \\
?? & \leftarrow & ?? & \leftarrow & ?? \\
\downarrow & & \downarrow & & \\
?? & & ?? & & \\
\downarrow & & & & \\
?? & \rightarrow & ?? & \rightarrow & ?? \\
 & & \downarrow & & \downarrow \\
 & & ?? & & ??
\end{array}
$$

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.
2. Implement the RSA encryption algorithm.
3. Implement the RSA decryption algorithm.

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] You are encouraged to work as a team of up to three members each playing a role as Alice, Bob or Charlie. This is also for resit of IS53012BS Computer Security

[2] or any programming language freely available

Assignment II

Based on the software prototype that you have developed in the previous part, analyse and implement the protocol below about authentication using a trusted server S.

Suppose a trusted server S that distributes public keys on behalf of others. Thus S holds Alice's public key K_A and Bob's public key K_B. S's public key, k_S, is well known. Now Alice (A) and Bob (B) wish to authenticate with each other and they propose to use the following protocol.

1) Dear S, This is A and I would like to get B's public key. Yours sincerely, A.
2) Dear A, Here is B's public key signed by me. Yours sincerely, S.
3) Dear B, This is A and I have sent you a nonce only you can read. Yours sincerely, A.
4) Dear S, This is B and I would like to get A's public key. Yours sincerely, B.
5) Dear B, Here is A's public key signed by me. Yours sincerely, S.
6) Dear A, Here is my nonce and yours, proving I decrypted it. Yours sincerely, B.
7) Dear B, Here is your nonce proving I decrypted it. Yours sincerely, A.

1. Implement this protocol in Java[3] to demonstrate how it works (again in decimal). There is no specific requirement to the user interface of your prototype but you may like to use the same simple user interface in the previous coursework assignment.
2. Identify and in your program demonstrate if there is an error or/and a subtle vulnerability of this protocol. [Hint: Consider if A uses this protocol to authenticate with a third-party Z.]
3. This protocol looks familiar. What is the name of the familiar protocol?

You may add if necessary assumptions for details to ease your implementation, but you must explain them clearly to gain credits. Also, you may decide where to start your implementation but it might be easier for you to first work out the keys and notations involved in each step. For example, let n_A and n_B be the nonce of A and of B respectively, and $(x,y).k$ be (x,y) with a signature k. The following lines denote the protocol with information flows to be transmitted.

1) A → S: A, B
2) S → A: $(K_B,B).k_S$
3) A → B: $(n_A,A).K_B$
4) B → S: B, A
5) S → B: $(K_A,A).k_S$
6) B → A: $(n_A,n_B).K_A$
7) A → B: $(n_B).K_B$

[END OF COURSEWORK ASSIGNMENT]

[3] or any programming language freely available

Assignment III

Consider the Digital Signature Scheme (DSS) which is based on the El Gamal public key cryptosystem. The basic DSS algorithm can be described as follows:

1. Alice generates public (p, g, y) and private (a) El Gamal keys
2. Alice signs message m using her private key (a) by taking the following steps:
   i. Alice randomly selects an integer k between 1 and $p-2$, $(1 \leq k \leq p-2)$ with the condition that k and $p-1$ are co-prime $(gcd(k, p-1) = 1)$.
   ii. She computes r, k^{-1} and s where:

$$r = g^k \ mod \ p$$

$$k^{-1} = inverse \ of \ k \ mod \ (p-1) \qquad (i.e. \ kk^{-1} = 1 \ mod \ (p-1))$$

$$s = k^{-1}(m - ar) \ mod \ (p-1)$$

   iii. Alice sends Bob the message m and the signature pair (r, s).

Develop a demo prototype to show that, upon receipt of the message m and signature pair (r, s) from Alice, Bob takes the following steps to verify the signature, using Alice's public key (p, g, y).

1. Bob computes v_1 and v_2 where:

$$v_1 = y^r r^s \ mod \ p$$

$$v_2 = g^m \ mod \ p$$

2. Bob accepts the message and signature as genuine if and only if $v_1 = v_2$.

Question 1

You are given the following El Gamal keys: public $(p = 83773, g = 5, y = 34970)$ and private $(a = 407)$. Showing all of your working and the results for each step of the algorithm, use this key to create signatures for the two messages $m_1 = 649$ and $m_2 = 538$.
Use the verification protocol to show that your signatures are correct.

Question 2

My public El Gamal key is: public $(p = 23269, g = 53, y = 13162)$
Decrypt the following message, signature pairs and hence deduce which messages are correct. Show your working out.

(1) $m1 = 933, r1 = 3598, s1 = 12102$

(2) $m2 = 10796, r2 = 12073, s2 = 6669$

(3) $m3 = 3899, r3 = 22789, s3 = 468$

Question 3

Write a paragraph discussing the advantages and disadvantages of the El Gamal based digital signature scheme when compared with the signature scheme based on the RSA.
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]

Submission requirements

These requirements apply to both parts.

1. Naming conventions for any `.pdf` or `.zip` file submissions
   When naming your files, please ensure that you include your full name, student ID number, course code and assignment number, e.g. `FAMILY-NAME.first-name_ID_IS53012A_cwPart#.pdf`
   (e.g. `ZUCKERBERG.david_920000000_IS53012A_cwPart2.pdf).`
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.

   If you use a programming language other than Java, you would need to provide the whole running environment for the marker.
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 5);
   - 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.
6. Group work (by up to 3 members) is allowed but ALL member names must be displayed on the coversheet of the ONE group coursework submission, and an equal grade will be awarded to all members unless specified otherwise.

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]