University of London International Programmes

Computing and Information Systems/Creative Computing

CO3326 Computer Security

Coursework assignment 1 2017-18

IMPORTANT: all students have been allocated a unique set of cipher text and keys to use for this coursework assignment. You can obtain your cipher text and keys using your Student Reference Number (SRN) from the following URL: http://foley.gold.ac.uk/cw18/api/cw1/{srn}. For example, if your SRN is 077242419, you would obtain your data from http://foley.gold.ac.uk/cw18/api/cw1/077242419. If you have difficulties obtaining your cipher text and keys, please email us at: intcomp@gold.ac.uk

This coursework assignment is designed to help you enrich your learning experience and to encourage self-study and creativity. Chapter 8 (pages 87-94) of the subject guide and the suggested supplementary reading will help you in completing this assignment. You should read the coursework assignments carefully and pay particular attention to the Submission requirements.

You are expected to submit **two** files: a **report** and a **results sheet**. The *report* counts as **60%** of your coursework assignment mark, in which you're expected to answer the questions below. The *results sheet* counts as **40%** of your mark, in which you're expected to summarise the results of your calculations in a specific format. Please use the cipher text and keys provided when answering the questions and when compiling the results sheet.

To complete the coursework assignment, it will make your life easier if you write a program. You are welcome to use any programming language. Please include key snippets of your code as an annex at the end of your report.

The RSA cryptosystem and factorisation

Suppose you are an avid hacker working for a security agency and you are eavesdropping on a converstion between *Alice* and *Bob*. You intercept a cipher text that you know has been sent from Alice to Bob. You look up both Alice and Bob's public keys in a key directory. You know that they are using an RSA cryptosystem. You assume that Alice wants to be sure that only Bob can decrypt the message. You also assume that Bob wants to be sure that Alice has sent the message. You would like to decrypt the message.

IMPORTANT: To answer the questions below, please use the cipher text and public keys (of Alice and Bob) that you obtained using your SRN.

Question 1

Describe briefly the steps you need to take to decipher the message.

Question 2

Which algorithms would be feasible for factorisation for the particular keys that you are trying to break? Compare the algorithms very briefly in terms of:

- number of digits they can deal with,
- time efficiency,
- memory consumption.

Question 3

Break Alice and Bob's key. State the factorisation algorithm you use and state the running time. If you rely on an online service, state the service you're using and investigate what algorithm is backing the service. *Note*: you will only be able to achieve half marks for this question if you rely on an online tool.

Question 4

What are the implications of the following assumptions on the keys used?

- Alice wants to be sure that only Bob can decrypt the message,
- Bob wants to be sure that Alice has sent the message.

Question 5

Are there any restrictions on the ordering of the keys when encrypting? Using your numbers explain why. Is the ordering the same on decryption?

Question 6

Decipher the message. Show your work.

Annex

Briefly - in one paragraph - describe the design of your code. Attach key snippets to the annex. Don't forget to acknowledge all sources. Make sure you acknowledge any code re-use.

REMINDER: It is important that your submitted coursework assignment is your own individual work and, for the most part, written in your own words. You must provide appropriate in-text citation for both paraphrase and quotation, with a detailed reference section at the end of your coursework. Copying, plagiarism and unaccredited and wholesale reproduction of material from books or from any online source is unacceptable, and will be penalised (see our guide on how to avoid plagiarism on the VLE).

Submission requirements

You should upload **two** single files only. These must not be placed in a folder, zipped, *etc.*

The **report** should be submitted as a PDF document following a *strict naming scheme*: `YourName_{srn}_CO3326_cw1.pdf`. For example, Carl Davis with SRN 077242419 would submit `CarlDavis_077242419_CO3326_cw1.pdf`.

The **results sheet** should be submitted as a JSON file with a *strict format* and *strict naming scheme*. This summarises the results of your calculations and will be automatically checked by an algorithm, so pay particular attention to its format. The name of the file should be `YourName_{srn}_CO3326_cw1.json`; for example, Carl Davis with SRN 077242419 would submit `CarlDavis_077242419_CO3326_cw1.json`.

You have obtained the *cipher text* and the *keys* in the following format (this is an example for illustration):

```
{
   "srn":"077242419",
   "name":"Carl Davis",
   "alice":{
      "modulus":"864153397514862469179935802 47",
      "publicKey":"71748972550420797587053965121"
   },
   "bob":{
      "modulus":"10370370240735783518580370 3747",
      "publicKey":"75540347194440969090680493719"
   },
   "cipherText":"9ctthcohr21pnmf6ip8"
}
```

For this *cipher text* and set of *keys*, Carl Davis would submit the following JSON, which reflects a correct solution:


```
{
   "srn":"077242419",
   "name":"Carl Davis",
   "alice":{
      "p":"111111151111111",
      "q":"777737777777777",
      "r":"864153397514853580690646913 60",
      "modulus":"86415339751486246917993580247",
      "privateKey":"560517578434413827656606742 41",
      "publicKey":"71748972550420797587053965121"
   },
   "bob":{
      "p":"888888877777777",
      "q":"116666666666611",
      "r":"10370370240735682963025925 9360",
      "modulus":"1037037024073578351858037037 47",
      "privateKey":"945134391662953447902846809 99",
      "publicKey":"75540347194440969090680493719"
   },
   "plainText":"university",
   "cipherText":"9ctthcohr21pnmf6ip8"
}
```


You can use this well-formed JSON to adapt for your numbers and your calculation results.

As the JSON is evaluated by an algorithm, every quote, comma, colon, curly brace upper/lower case is crucial. Please pay attention to these. It would be a shame to lose a potential **40%** of the total marks for this coursework assignment because of a misplaced comma or a missing quote. There are online tools you can use for JSON formatting and validation (for example this), so double-check that your JSON is syntactically correct.

Please note that the numbers are enclosed within quotes. There is a hint here: the numbers are too large to be represented as `long` or `int`.

Help

In order to encrypt/decrypt a message, you need to encode the text (string) as a number. If you write your code in `Java`, you can use the following methods:

- for encoding:

```
public static BigInteger encode(final String text) {
    return new BigInteger(text, Character.MAX_RADIX);
}
```

- for decoding:

```
public static String decode(final BigInteger number) {
    return number.toString(Character.MAX_RADIX);
}
```

Alternatively, if you're writing your program in a language other than `Java`, you can rely on the following web service for encoding/decoding:

- http://foley.gold.ac.uk/cw18/api/encode?text=university
- http://foley.gold.ac.uk/cw18/api/decode?number=3113163156336982

obviously replacing `university` or `3113163156336982` in the URL with the text or number you want to encode/decode. The service works for one-word texts (so no space or special characters). There is another hint here: if your decryption is correct, you will get a recognisable English word.