MATH10001 Cryptography Project

The deadline is 18 December, 5 p.m. The solutions must be submitted via Blackboard.

Put your name, student number and group number at the top of first page of your solutions. You should write your report in LaTeX and *submit the PDF file and the LaTeX source file (the file with .tex extension).* Include your MATLAB code in your report, do not submit it in separate files. If you use a word processor such as Microsoft Word or LibreOffice, you can submit your work as a .doc, .docx, .odt or .pdf file (still a single file containing all your MATLAB code) and it will be marked but you will *not* get any marks for LaTeX and presentation. Your report must be suitable for automated plagiarism checking.

The calculations must be done in MATLAB (use MATLAB R2020a, R2020b or MATLAB Online). Include your MATLAB code and the output in your report as text, *not as images*. You can use scripts or you can do the calculations interactively, but your code should be self-contained, it should not require any input from the user. All the question can be solved by using the MATLAB functions taught in this course. Aim to do the calculations efficiently, your code should not produce large amounts of unnecessary output and you should explain what your code does in comments included in the code or in the text of your report. Put MATLAB code and output between `\begin{verbatim}` and `\end{verbatim}` or use the `listings` package. Long lines in verbatim text should be split so that they fit on the page. Short MATLAB expressions can be included between `\verb@` and `@` in the text. The congruence $a \equiv b \bmod n$ can be typeset as `$a\equiv b \bmod n$`.

You should answer the questions in the text of your report, merely including the MATLAB output verbatim is not sufficient, and you should include explanations and interpretations of the results, where appropriate.

There are 30 marks available for this project:

- 20 marks for the mathematical content (correct calculations and explanations),
- 5 marks for group work (1/4 of the average content mark for your group, excluding those who did not attend the labs, failed to submit the project or whose work does not appear to be a serious attempt) and
- 5 marks for the quality of LaTeX and presentation.

Questions 1–3 are the same for everybody in a group, in question 4 you will have to calculate your own primes. You should work together on the methods to carry out the calculations, implementing them in MATLAB and on interpreting the results, but you must write your own MATLAB code and carry out the calculations yourself, you *must not* copy them from someone else or let someone else copy yours, and your report must be your own work. See also `http://documents.manchester.ac.uk/display.aspx?DocID=2870`.

The project, including writing up should not take more than 10 hours.

If you have any questions, e-mail `Gabor.Megyesi@manchester.ac.uk`.

Question 1. [3 marks in total] Find the numbers p, a and v for your group in the file `Table1.pdf`. You are Alice in the Diffie-Hellman key exchange. p is your prime, g will be the smallest positive primitive root modulo p, a is your random number and v is the number sent to you by Bob.

(a) [1 mark] Find g, the smallest positive primitive root modulo p.

(b) [2 marks] Calculate the number u you send to Bob and the shared key k.

Question 2. [8 marks in total] *You are* not *allowed to use* `factor`, `ifactor` *or any other MATLAB function that involves factorisation of integers in this question.*

(a) [5 marks] Find the number n for your group in the table in the file `Table2.pdf`. Carry out the Miller-Rabin test on n with base 7 and store the sequence of remainders in a vector so that you can use them in (b).
Your program should print out r and the sequence of remainders. Your code should be adaptable to testing different numbers by changing the value of n.
What can you conclude about n from the result of the Miller-Rabin test? (You can do this by observation, you do not have to write MATLAB code to make the decision.)

(b) [3 marks] Take the vector of remainders from (a), add 1 to them and calculate the greatest common divisor of each of the resulting numbers with n. Then subtract 1 from the remainders and calculate the greatest common divisor of each of the resulting numbers with n. Use these numbers to find the prime factorisation of n.

Question 3. [3 marks] A digital signature algorithm uses $p = 30167674936870980426367$, $q = 17456345243$, $g = 18008617784390347685963$ and $y = 61726472517312324125 43$.
Find the values of H, r and s for your group in `Table3.pdf` and determine whether (r, s) is a valid signature for H. You should print out the intermediate results u_1 and u_2, not just the final result of the calculation.

Question 4. [6 marks in total] Find your student number. If it is an 8 digit number, then S will be your student number. If it is a 7 digit number, then add 2000000 to it and this number will be S. In either case, S should be an 8 digit number beginning with 1. Make sure that you have the correct number before continuing.

(a) [4 marks] Find the smallest prime p such that $p > 10S^4$ and $(p-1)/2$ is also prime. Find the largest prime q such that $q < 4S^5$ and $(q-1)/2$ is also prime. (Calculating p and q may take anywhere from a few seconds to a couple of minutes in the online version of MATLAB depending on how far you have to search. If your program takes much longer, then check your MATLAB code to make sure it that it does not go into an infinite loop. You should try your program on smaller numbers first to ensure that it works.) p and q will be your personal primes for RSA. Your public key will be $e = 65537$. Calculate the corresponding private key d.

(b) [2 marks] Find the two encrypted messages corresponding to your student number in the file `Table4.pdf` and decrypt them with your private key. Divide the decrypted messages into 2 digit blocks from the right, 00 denotes a space, convert the other 2 digit blocks to letters by using the table below and then join the messages. (You can do this by hand, you do not have to write a MATLAB program to do it.) What text do you get?

01	02	03	04	05	06	07	08	09	10	11	12	13
A	B	C	D	E	F	G	H	I	J	K	L	M
14	15	16	17	18	19	20	21	22	23	24	25	26
N	O	P	Q	R	S	T	U	V	W	X	Y	Z