School of Electronic and Electrical Engineering

ELEC5471M Data Communications and Network Security

UNIVERSITY OF LEEDS

MATLAB Project on Public-Key Encryption

Issued: Week 8 **Due: 2 Dec 2016, at 11:59PM**

This project would help you better understand the RSA encryption technique. This project shall be done in a group of two. The recommended time to finish this project is within the first two lab sessions. You can submit your work nearly a week after the second lab for further flexibility. Please note that your second project will be given by week 10, and it would be best if you finish this one before starting the other. **N.B.:** At the start of the lab session in week 11, you will be asked to demonstrate how your code works and are expected to explain its different parts. This will be used in marking your MATLAB project.

All project related material can be found on the VLE. The submission of your work is also via the VLE. As a group, only one of you needs to submit the work. In order to submit your work, zip all relevant project files into one, and just submit the zipped file. You have to strictly follow the below format for your file's name:

ELEC5471M_FT16_MATLAB_Project_Lastname1_Lastname2.zip

NB: Only one zip file must be submitted. Do not attach your .m files separately. Do not use .rar format. 1 (out of 15) point will be deducted if you do not follow the above guidelines including the name formatting.

Please include the full names and the student IDs of all group members in the "Comment" box on the VLE submission page, as well as in any word/pdf file in the zip file.

Late submission penalty: 5% per calendar day, for up to two weeks. No submissions will be marked after two weeks.

****

General Setting

You have learned in the lectures how the RSA protocol works. Here, you are going to implement RSA at a rather small scale. What is expected from you is to write your code in such a way that it can efficiently work with prime numbers less than 10000. You can find a list of the first 1000 prime numbers at http://primes.utm.edu/lists/small/1000.txt to check your codes. It would be, of course, better if your codes can handle larger numbers as well, but it is not required at this stage.

You are going to write several MATLAB functions (m files) corresponding to different tasks required in the RSA protocol. The first task is to create public and private keys, i.e., to generate parameters N = p x q, d, and e, as will be explained later. The second task is that, given the public key, you write a function that

encrypts a given message. Similarly, given the private key, you should write a function that decrypts a ciphertext. Finally, you should write a code that hacks your RSA implementation, that is, by getting the ciphertext and the public key, it will find the original plaintext. Here, I describe each of these tasks in more detail.

Task 1: Generating Public and Private Keys [4 points[i]]

You are expected to write a function RSA_Gen.m that accepts two prime numbers p and q at its input, and gives us three integer numbers N, d, and e at its output:

function (N, d, e) = RSA_Gen (p,q);

For $N = p \times q$ and $z = (p-1)(q-1)$, d and e must satisfy the following conditions:

1. $e < N$
2. e and z have no common factors, that is, if you factor z into its prime-number constituents, e would not be divisible by any of those prime numbers. Equivalently, the greatest common divisor of e and z must be one.
3. Choose d such that $ed - 1$ is divisible by z, that is, $ed = 1 \mod z$.

Your public key is then given by the pair (N,e), and your private key is given by (N,d).

These MATLAB functions may help you in this project: rem, mod, gcd, and factor. Make yourself familiar with them and their limitations. Do they correctly do their job for very large numbers?

Task 2: RSA Encryption [4 points]

You are expected to write a function RSA_Enc.m that accepts the public key (N,e) along with a message m (a natural number less than N) at its input, and gives us the encrypted version of m at its output:

function c = RSA_Enc (N,e,m);

You basically need to write a code that calculates

$c = m^e \mod N$.

If m or e are large numbers, you may need to calculate c in multiple steps; remember the limitations of the MATLAB functions that you use and that (a x b) mod N = (a mod N) x (b mod N).

Task 3: RSA Decryption [2 points]

You are expected to write a function RSA_Dec.m that accepts the private key (N,d) along with a ciphertext c (a natural number less than N) at its input, and gives us the plaintext m at its output:

function m = RSA_Dec (N,d,c);

You basically need to write a code that calculates

$m = c^d \bmod N$.

Again, If c or d are large numbers, you may need to calculate m in multiple steps; remember the limitations of the MATLAB functions that you use and that (a x b) mod N = (a mod N) x (b mod N).

Task 4: Hacking RSA [3 points]

You are expected to write a function RSA_Hack.m that accepts the public key (N,e) along with a ciphertext c (a natural number less than N) at its input, and gives us the plaintext m at its output:

function m = RSA_Hack (N,e,c);

Make the assumption that $N < 10^8$.

Task 5: Hacking RSA in real world [2 points]

Do some research to find out about successful attacks on the RSA, or possible future attacks on the RSA.

---

[i] Your codes will be assessed against the assessment criteria for software assignments as published on the module's VLE.