

MCS 320, Introduction to Symbolic Computation, Project 1
RSA Cryptosystem
Due date: Friday, March 13, 2009, 9 AM

Project description

The goal of the project is to implement a public key crypto system (RSA) in Maple. The main references are:

1. Rivest, R.; A. Shamir; L. Adleman (1978). “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, Communications of the ACM 21 (2): pp. 120–126, available for download from blackboard.
2. <http://en.wikipedia.org/wiki/RSA>

You need to develop two Maple procedures and a test example all in one worksheet, preferably, classical. If for some reason you must use a worksheet that is not classical, you must use Maple input, 1D mode (red colour, bold) when you enter commands — other versions will not be graded. You must also document all steps in your worksheet to show your understanding and help me recognising it.

1. Your test example must choose two distinct large prime numbers, your encryption exponent, satisfying the conditions described in the RSA paper, the block size, and a test message (medium size meaningful sentence in English: take a book by Agatha Christie, Arthur Conan Doyle, Ernest Hemingway, Jerome Jerome, Adam Smith, Karl Marx, or Vladimir Nabokov and pick one sentence from the middle of the book).
2. Encrypting a message which is a string containing a text. The input for the procedure must be of the form:
 - (a) a string *message* in the alphabet $A \dots Z$ (that contains a message to encrypt),
 - (b) a public key (modulus n which is an integer),
 - (c) an encryption exponent a (integer)
 - (d) the block size *blocksize*. It means the number of letters in each block to encrypt separately. The procedure must determine whether the length of the message is divisible by the block size. If it is not, it must append “Z” to the end. The reason for splitting a message into blocks is because your procedure will convert the message into an integer to further process it with the algorithm. To each letter you will associate a number, for example:

$$A \mapsto 00, \dots, Z \mapsto 25.$$

But to do computation more efficiently, it could help to do this in blocks. Let

$$\textit{blocksize} = 4.$$

Then *HELLO* will be first split into blocks of size 4 as

$$HELL \quad OZZZ$$

and each block will be replaced with an integer separately:

$$HELL \mapsto 07041111,$$
$$OZZZ \mapsto 14252525.$$

If you encrypt each letter separately (*blocksize* = 1) or do not split into blocks it might be inefficient. Choose a reasonable block size when you are testing your procedure.

The output must be list of integers, each integer is the encryption of each block in the original text.

3. Decrypting an encrypted message. The input must be of the form:

- (a) list of integers *encrypted*, each integer is the encryption of each block in the original text,
- (b) two integers p and q that form the private key,
- (c) encryption exponent a ,
- (d) *blocksize*.

The output must be a string in the alphabet $A \dots Z$ that is the RSA decryption of *encrypted*.

Note that if you encrypt a test message and the decrypt it you must get your original message back plus, possibly, several letters Z at the end depending on your *blocksize*.

Note

Bring *your* solution to the project to class. *Your* is emphasised to stress that your solution is the result of an individual effort. Collaborations are not permitted. The solution to this project consists of two parts:

1. A print out of the Maple worksheet that you bring to class. Deliver a well written document, with grammatically correct and complete sentences, without spelling mistakes, appropriately structured into sections and subsections.
2. The Maple worksheet that you email to alexey_ov@yahoo.com as an attachment. **Do not** send your projects to **other** e-mail addresses: they will not be reviewed if not sent to the e-mail address mentioned above. The worksheet should run as a computer program, from top to bottom with consistent output and without errors.