# RSA Encryption

## 1 Overview

How are messages transferred in a secure way? How is your browser sure that it is talking to the website it should be instead of an imposter? How can we carry key-chains that tell us a password that changes every five seconds? The answer to all of these questions is asymmetrical encryption. Asymmetric encryption allows us to take a message and turn it into gibberish using a digital key, and turn it back into the original message using a different key. We can give out a **public** key to the world, and let them encrypt messages that can only be decrypted using our **private** key. The public and private keys are mathematically related, but it is computationally intractable to compute a private key from only the public key.

Today we will be implementing one such asymmetrical encryption algorithm known as RSA, named for its inventors Ron Rivest, Adi Shamir and Leonard Adleman. RSA uses randomly generated prime numbers to generate a key pair, and then uses exponentiation and modulation to encrypt and decrypt messages using this key. RSA, while being surplanted by the more sophisticated SHA family of algorithms, is still commonly used for digital security. The mathematics proving RSA to be secure are beyond the scope of this course. The implementation of RSA, however, is of a reasonable level of complexity to be implemented in the our lab, given the tools provided by Java's class library.

## 2 Learning Outcomes

**By the end of this project students should be able to:**

- write algorithms based on a technical problem description;
- utilzie Java types.
- utlize Java's BigInteger to perform high level computation;
- utilize asymmetric encryption to encrypt and decrypt messages;
- work effectively with a partner using pair-programming;
- write an effective report that describes the students' problem solving process.

## 3 Pre-Lab Instructions

**Do this part and turn in your answer on BBLearn before you come to lab:**

This project will require you to use binary input and output streams, and understand how to perform RSA encryption

- Read the Wikipedia article on the RSA algorithm: https://en.wikipedia.org/wiki/RSA_(cryptosystem)
- Walk through the process of applying key generation, encryption, and decryption, where p = 20 and q = 17. Let 60 be the value you encrypt and decrypt. List all of the intermediate variables you get along the way. Include this process in your prelab BBLearn submission
  - The following address can be used for computing the modular multiplicative inverse: http://planetcalc.com/3311/
  - The following address can be used for computing the modular exponentiation: https://www.mtholyoke.edu/courses/quenell/s2003/ma139/js/powermod.html
- Read the documentation for Java's BigInteger class. List the methods that would be necessary to perform the calculations you did while walking through the RSA algorithm.
  - Also, how might your randomly generate prime numbers?

## 4 Lab Instructions

**Do this part in lab:**
Create a class with a main method for our RSA implemenation. Our program will have three parts

1. Generating the public and private keys
2. Encrypting a number using the public key
3. Decrypting the number using the private key

### Step 1

Use Java's BigInteger data type to perform the same process shown in the RSA example in steps 1 to 5. For each of the values computed, print those labeled values to the screen. First uses the example values for p and q to ensure your compuation is running correctly. Once you are sure your code is working properly, comment out the section where you set p and q and add lines to use randomly generated prime numbers instead.You should be able to easily switch back and forth between using the test numbers and randomly generated the numbers by switching which line is commented. Each of these random prime numbres should have a bitlength of 256.

### Step 2

Print a labeled starting value to the screen. Now, go through the encryption process. Remember that in order to do this, we will need the public key (the n and e values.) When using the example values, the number 65 should encrypt to 2790 durring this step. Once you have completed the encryption, print the labeled value to the screen.

### Step 3

Now perform the decryption process on your encrypted value. Remember that in order to do this, we will need the private key (the n and d values.) In the example values, this would decrypt 2790 back to 65. When using the randomly generated keys, this should give you back the original value from step 2.

When this is complete you will have a simple but legitimate RSA implementation. Congratulations!

(n: 25824288551267950344381342303097806894 7, d: 22786136957001132653968909292797344335 3)
https://sites.google.com/a/nau.edu/249751615939781461903757255894139183723/

## 5 Lab Report

**Each pair of students will write a single lab report together and each student will turn in that same lab report on BBLearn. Submissions from each student on a pair should be identical. In order to recieve points, you MUST make a BBLearn submission.**

Your lab report should begin with a preamble that contains:

- The lab assignment number and all partner names
- Your name(s)
- The date
- The lab section

It should then be followed by four numbered sections:

### 1. Problem Statement

In this section you should describe the problem in **your** own words. The problem statement should answer questions like:

- What are the important features of the problem?
- What are the problem requirements?

This section should also include a reasonably complete list of requirements in the assignment. Following your description of the problem, include a bulleted list of specific features to implement. If there are any specific funtions, classes or numeric requirements given to you, they should be represented in this bulleted list. It is recomended that you complete this section before you begin coding the problem, as gathering these requirements may help you organize your thoughts before you begin your soulation.

### 2. Planning

In the second section you should describe what planning you did in order to solve the problem. You should include planning artifacts like sketches, diagrams, or pseudocode you may have used. You should also describe your planning process. List the specific data structures or techniques you plan on using, and why. How do you plan on breaking up the problems into functions, classes and methods?

### 3. Implementation and Testing

In the third section you should demonstrate your implementation. As directed by the lab instructor you should (as appropriate) include:

- a copy of your source code (Submitted in BBLearn as .java files, should not be in your report document)
- a screen shot of your running application / solution. This should include a screenshot of the output, be it a terminal window or graphical interface
- results from testing. This is specific to the lab, and not every lab will include this element.

### 4. Reflection

In the last section you should reflect on the project. What part of the projec was the most difficult? Where there other solutions to this problem that you considered? Where there any new solutions you can think of now that you couldn't then? Do you think the way you chose was the best option, or would you go about the problem differently if you had to start over? How might the problem had been broken up into easier problems? If you had one more day to work this problem, what improvements might you make? Every problem has alternative solutions and solution has tradeoffs or improvements, you are required to identify some of these elements.

### 5. Partner Rating

Every assignment you are required to rate your partner with a rating between 1 and 10. For each patener a name and rating should be submitted in the comment section of the BBLearn submission, and not in the report document. You do not have to tell your partner the rating you assign them. A rating 8 or more

indicates that your partner was particularly helpful or contributed exceptional effort. A rating of 5 indcates that your partner met the class expectations of them. Rating your partner 3 or less means that they refused contribute to the project, failed to put in a resonable effort or actively blocked you from participating. In the case where you give a very low rating, please explain why in the comment section of the submission. Those who recieve low scores may be asked to explain their actions to the lab staff, and additional low ratings after a warning could lead to losing a letter grade, or even failing the course. Consistant high ratings from partners are noted durring the grading process, and may be taken into account when rounding your final grade.

## Colophon

This project was developed by Richard Lester and Dr. James Dean Palmer of Northern Arizona University. 