**Pre-Calculus Encryption Project: Introduction to Public Key Cryptography**

**Brief Overview:**

Public key cryptography is a common means of securing information on the Internet. It is often used to protect credit card transmissions for on-line shopping. One of the earliest public key algorithms was the RSA algorithm, named for its inventors Ronald Rivest, Adi Shamir and Leonard Adelman. RSA is an important but often transparent part of Internet browsers such as Internet Explorer. In the lessons outlined below, students will learn how to encrypt and decrypt using RSA. In the process of learning RSA, students will become familiar with modular reduction of natural numbers and gain an understanding of some prime number theory.

**Lesson 1: An Introduction to Prime Number Theory**

 The RSA encryption algorithm builds encryption and decryption keys from two extremely large prime numbers (in a manner that will be discussed in Lesson 3). As a result, prime number theory is the foundation for RSA. This lesson will start from the definition of prime numbers and proceed to introduce the basic number theory concepts that are needed for RSA.

**Definitions:** A *prime number* is a positive integer greater than one that is divisible by no positive integers other than one and itself. A positive integer which is not prime and which is not equal to one is called a *composite number.*

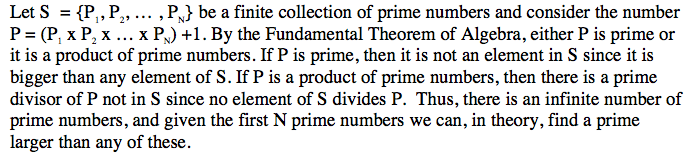
The importance of prime numbers can be seen in the following result.

**The Fundamental Theorem of Arithmetic:** Every positive integer greater than one can be written uniquely as a product of primes, with the prime factors in the product written in order of nondecreasing size.

This result tells us that prime numbers are the basic building blocks of the integers.

Furthermore, it allows us to create an encryption algorithm using prime numbers. For RSA encryption, we start with two primes *p* and *q* and use the number *n* = *pq.* In order for decryption to be possible, there must be a way to get back to *p* and *q* from *n.* The Fundamental Theorem guarantees that *n* has a unique decomposition into primes and thus a factorization of *n* must yield only *p* and q. The strength of the RSA algorithm depends on the size of *p* and *q* and the resulting difficulty an adversary would have in factoring the product not knowing *p* and *q.*

Although the Fundamental Theorem indicates that we can build an encryption algorithm using primes, it is not clear that we can find large enough primes to develop a secure algorithm. Therefore, we next show that given any collection of primes we can always find another prime number not in that collection.



However, in practice, in is not easy to find prime numbers. For the final part of this lesson, we discuss the sieve of Eratosthenes, one of the earliest methods of finding prime numbers and which serves as a foundation for the modern methods of finding prime numbers.

To illustrate this method for finding prime numbers, we illustrate how to find all prime numbers less than 100. The positive integers between 1 and 100 are written in a ten by ten grid. We first cross out all numbers other than two in the grid which are multiples of two. Then, all remaining integers other than three, which are multiples of three, are crossed out. Similarly, all remaining multiples of 5 and 7 are in turn crossed out. The numbers in the grid which have not been crossed out are one and all the prime numbers less than one hundred. Note that a composite integer has no prime factors larger than its square root. So, in our example, it was not necessary to worry about crossing out multiples of numbers larger than ten.

The students now have enough information to answer problems 1 and 2 of the handout.

**Lesson 2: Modular Arithmetic and Greatest Common Denominator**

Modular arithmetic is a useful tool in computer science, mathematics and cryptography. In general terms, the modulus (mod) is the remainder of a division problem. For example:

73 mod 10 = 3 because 73 = 7 \* 10 + 3

Written mathematically,

*a* mod *n* = *b* where *a* = *k* \* *n* + *b*

Since *k* can be any integer, this equation has an infinite number of solutions. In order to limit our answer, *b* is chosen such that 0 *b* *n.*

This is a useful concept in cryptology. The English alphabet only has 26 letters but numbers are infinite. If we represent the letters by the numbers 0 – 25, where a=0, b=1, c=2, ..., y=24, z=25, then we can reduce any number to a value within the range of letters by setting the modulus to 26.

Another useful concept is modular congruency. Two numbers *a* and *b* are congruent modulo a number *n* if their difference (a *- b)* is a multiple of *n.* Formally, we express congruence in the following way.



Here are some examples to illustrate this concept.



The final concept that we will need in order to understand RSA encryption is how to find the greatest common denominator (gcd) of two numbers. The gcd of two numbers, *a* and *b,* is the largest number that divides both *a* and *b.* For example:

gcd (18, 48) = 6

gcd (17, 28) = 1

The easiest way to determine the gcd of two numbers is to perform the prime factorization of both numbers. Then multiply the common factors together to obtain the greatest number that can divide both of the original numbers. In our previous examples

18 = 1 \* 2 \* 3 \* 3

48=1\*2\*2\* 2\*2\*3

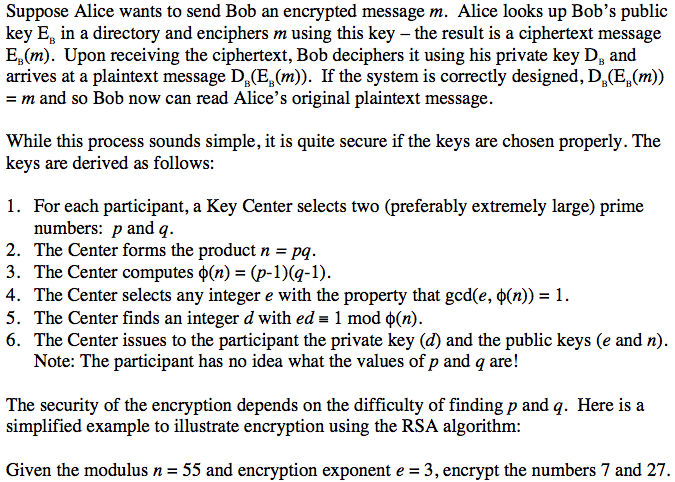
17 = 1 \* 17

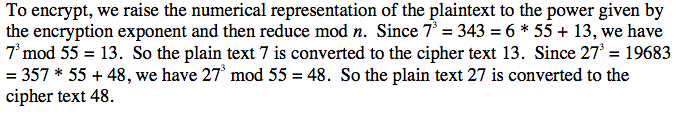
28 = 1 \* 2 \* 2 \* 7

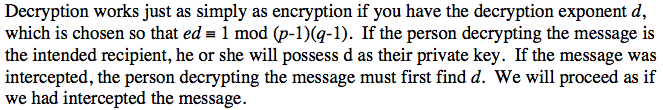
In both of these factorizations, we see that 18 and 48 have one ‘1’, one ‘2’, and one ‘3’ in common. Therefore, the gcd(18,48)=1\*2\*3=6. In the case of 17 and 28, the only common factor they share is 1. Therefore, the gcd (17, 28) = 1. In this case, we say that these two numbers are relatively prime to one another. Many cryptologic techniques, including RSA, require the keys to be relatively prime to one another.

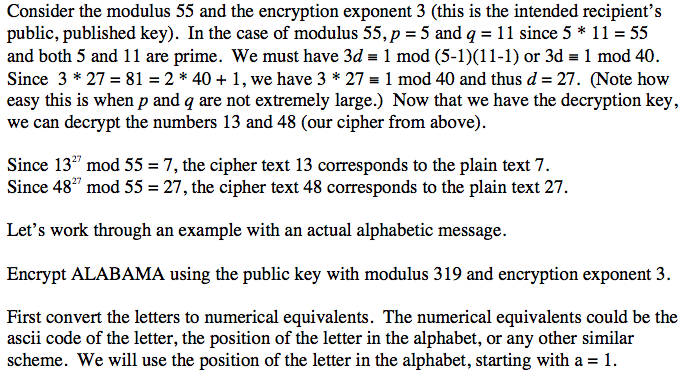
**Lesson 3: Encrypting and Decrypting Messages Using RSA**

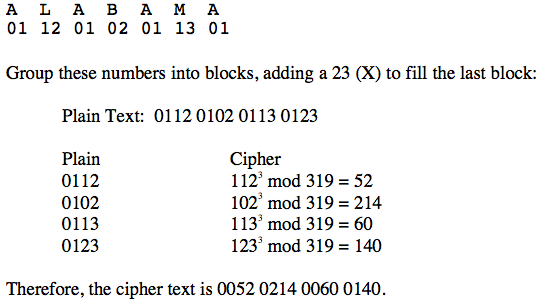
RSA is considered an asymmetric cryptographic system. In such systems, the sender and receiver do not share a common key; rather each participant has a *public key* **E** (for enciphering) and a *private key* **D** (for deciphering). The public and private keys of the sender do not relate to the public and private keys of the receiver of the message. For this reason, the public keys can be published for anyone to see. Here is an example to illustrate this concept.



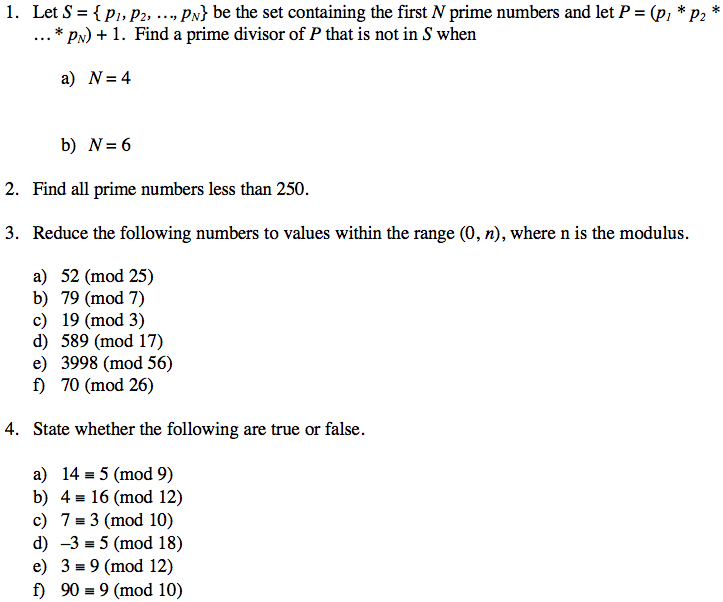








**Apply what you learned:**You will explain how you solved the following problems in your report write-up.



1. Find the gcd.
   1. a)  gcd (15, 5)
   2. b)  gcd (89, 46)
   3. c)  gcd (87, 190)
   4. d)  gcd (54, 7)
   5. e)  gcd (41, 101)
   6. f)  gcd (890, 21)
2. Given the public key with modulus 2911 and encryption exponent 1867, decrypt the following RSA-encrypted message:

*0518 0753 0875 1618 1126 0615 1613 1966 2062 2412 1173 0030 2756 0308 2524 0376 0918 2528 2169 1359 0534 2116 0126 1606 0330*