Investigation 34

RSA Encryption

Focus Questions

By the end of this investigation, you should be able to give precise and thorough answers to the questions listed below. You may want to keep these questions in mind to focus your thoughts as you complete the investigation.

- What is public key encryption, and what are some of its applications?
- How does RSA encryption work, and how is the security of RSA encryption related to prime factorization of integers?
- What mathematical results are necessary to establish the validity of RSA encryption, and how do these results follow from previously established properties of the integers?

Congruence and Modular Arithmetic

In this investigation, we will use "mod" notation in two distinct but related ways. As we have done before, we will write

$$a \equiv b \pmod{n}$$

when we want to specify a *relationship* between the integers a and b—namely, that n divides $(a-b)$, or equivalently, a and b have the same remainder when divided by n. This is the standard usage that we are familiar with from Investigation 2, except that we have omitted the parentheses that typically surround the "mod n" portion of the notation. This omission is common, and it will often make our notation easier to read, especially when we are working with expressions that already contain several sets of parentheses.

We will also sometimes write $a = b \pmod{n}$ (note the use of $=$ instead of $\equiv$) for the purposes of *defining* a to be the unique remainder guaranteed by the Division Algorithm when b is divided by n. This means that

$$a \equiv b \pmod{n} \text{ and } 0 \leq a < n.$$

For example, if $x = 56 \pmod{17}$, then since $56 = 17 \cdot 3 + 5$, we see that $x = 56 \pmod{17} = 5$.

Preview Activity 34.1. Determine the value of x for each of the following:

(a) $x = 29 \pmod{17}$.

(b) $x = 138 \pmod{17}$.

(c) $x = 200 \pmod{17}$.

We can also define a function using this mod operator. For a natural number n, we let $R_n = \{0, 1, 2, \ldots, n-1\}$ and define $f : \mathbb{Z} \to R_n$ so that for each integer x,

$$f(x) = x \pmod{n}.$$

This is sometimes referred to as the **mod *n*** function.

(d) For $n = 17$, compute $f(29)$, $f(138)$, and $f(200)$, and $f(546)$.

Introduction

Throughout history, secret codes have been used to send private messages from one person to another since in many situations, there is a desire for security against unauthorized interpretation of coded data. That is, there is a desire for secrecy. **Cryptography** is the science and art of concealing the content of communications between parties where the communications channel between them can be accessed by an unfriendly third party. Following are some standard terms used in modern cryptography,

- Plaintext. This is the original message that is in readable form.
- Encryption. The process of transforming the plaintext message into a disguised message that is then transmitted to another party.
- Encrypted message. This is the disguised message that is transmitted to another party.
- Decryption. The process of converting the encrypted message back to the original plaintext message.
- Decrypted message. The message that is converted from the encrypted message by the decryption process. The decrypted message should be identical with the original plaintext message.
- Cryptosystem. This term refers to the system that contains both the encryption process and the decryption process.

One of the first known cryptosystems is the so-called Caesar cipher, which was used by Julius Caesar to send messages to his troops. To encrypt a message, Caesar simply shifted each letter in the alphabet three places to the right. Using the English alphabet, this means that the letter A is encrypted as D, the letter B is encrypted as E, and so on. When we get to the end of the alphabet, we "wrap around" to the beginning of the alphabet. So X, Y, and Z are encrypted as A, B, and C, respectively. To decrypt a message, we simply shift each letter in the encrypted message 3 letters to the left.

Although it may not be necessary for this cryptosystem, we will illustrate the modern practice of converting text into numerical information. We will encode each letter with the numbers shown in the following table.

A	B	C	D	E	F	G	H	I	J	K	L	M
00	01	02	03	04	05	06	07	08	09	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

So a plaintext message such as SAY would be encoded as 180024, where it is understood that each block of two numbers corresponds to a letter. We can then use the mod 26 function f to encrypt this message. So

$$f(x) = x + 3 \pmod{26},$$

where x is a two-digit representation of a letter.

Activity 34.2.

(a) Use the encryption function f to encrypt the message 180024.

(b) Suppose you receive an encrypted message 16070022060301 that you know was encrypted using the Caesar cipher. Decrypt this message and obtain the plaintext message. Describe the decryption function that you used to do this.

(c) If somebody knows how to encrypt a message using the Caesar cipher, do they then know how to decrypt a message using the Caesar cipher? Explain how this is a weakness in the cryptosystem.

The Caesar cipher used a shift of 3 letters to the right. This can be generalized to the concept of a **shift cipher**, which will be explored in Exercise (1).

RSA Encryption

Preview Activity 34.3. In Investigation 1, we learned about divisibility, greatest common divisors, and prime factorization in the integers. In this investigation, we will use these ideas together to study an interesting and important application: public-key encryption. The system we will study is one that is commonly used to transmit sensitive data over the internet. Its security rests on an important observation that should become apparent as you attempt to complete the following tasks.

(a) For each of the parts below, find a number whose prime factors are exactly the numbers listed:

(i) 4861 and 2621

(ii) 7907 and 619

(iii) 1753 and 1759

(b) Each of the numbers below has exactly two prime factors. Using any mathematically correct method, find these two factors.

(i) 13494211

(ii) 3902233

(iii) 1776977

(c) Which was easier: part (a) or part (b)? Explain why.

From the primitive Caesar cipher, which simply shifts each letter in the original message by a fixed amount, to the fascinating Enigma machines used by the Germans during World War II, numerous encryption schemes and devices have been invented in an attempt to keep sensitive data out of the hands of unauthorized (and potentially malicious!) third parties. In this investigation, we will study one of the most commonly used modern encryption schemes, named *RSA encryption* for Ron Rivest, Adi Shamir, and Leonard Adleman, who publicly described the system in 1977. An equivalent system was secretly developed in 1973 by Clifford Cocks at the Government Communications Headquarters (GCHQ), an intelligence and security organization responsible for providing signals intelligence and information assurance to the government and armed forces of the United Kingdom. This work was declassified in 1997.

Since the late 1970s, RSA encryption has become one of the most important systems for what is now known as *public-key encryption*. The basic idea behind a public-key scheme is that anyone should be able to send a message, but only the intended recipient should be able to read it. Thus, the key to *encrypt* a message is made public, but the key to *decrypt* the message is kept secret and distributed only to those who are authorized to view the encrypted text. In order for such a system to work, it has to be very difficult or even impossible for a potential attacker to determine the *private* decryption key just by knowing the *public* encryption key. Compare this to the Caesar cipher (or any shift cipher) where knowledge of the encryption key makes it easy to determine the decryption key.

Public-key encryption is particularly useful for tasks such as sending data over the Internet. For instance, a banking web site might want to allow any user to send information (such as a user name and password) over a secure connection. However, for the transaction to be truly secure, only the bank should be able to decode the information sent. RSA encryption achieves this design feature by using the properties of prime factorizations suggested in Preview Activity 34.3. In particular, RSA encryption exploits the fact that it is relatively easy to multiply two prime numbers together (even if they are large primes), but nearly impossible to efficiently factor a large number into its prime factors. As we will see shortly, the RSA scheme translates this theoretical fact into a very practical and secure encryption method.

The Basics of RSA Encryption

The first step in any encryption scheme is to decide what *alphabet* will be used and how the elements of the alphabet will be assigned numerical representations. We will use the same method of assignment used in the Caesar and shift ciphers. So we will use the standard A through Z alphabet, with A represented by 0, B represented by 1, and so on. As we will see, however, there are many security advantages to using a larger alphabet that consists not only of letters, but entire words. So, for instance, we might use a two-digit numerical representation of each letter (00 to 25) and encode our messages using blocks of letters. If we were to do so, the word "SECRET" would be encoded as 180402170419. Keep in mind that at this point, we haven't actually encrypted anything. We have simply developed a way of translating letters or blocks of letters into the numerical representations that will be used by our encrypting function. Note that we could have also included numerical codes for spaces, numbers, punctuation, and the like.

Once we have established our alphabet, it takes only a few simple steps to encrypt a message:

(1) First, we generate two different prime numbers, p and q, and calculate the quantity $m = pq$ (called the *modulus*). For the resulting system to be secure, p and q need to be extremely large, perhaps having hundreds or even thousands of digits. Because of the size of the primes required, the two prime numbers are generated by a computer. * The value of m becomes public—it can be shared with anyone—but p and q must be kept secret. In practice, modern computers can easily generate prime numbers with several hundred digits. However, in that case, the value of m must be factored to determine p and q and this can take several hundred years to do so using current technology.

(2) Next, we obtain a positive integer e such that

$$\gcd(e, (p-1)(q-1)) = 1.$$

This number e is called the *encryption key*, and the value of e is made public. The quantity $(p-1)(q-1)$ is often called the *totient*, denoted t. The totient must be kept secret, as it plays an essential role in the decrypting process.

(3) Finally, to encrypt a message, we form blocks of letters of a specified size and input the numerical representation of each block of letters into the encoding function f defined by

$$f(x) = x^e \pmod{m}.$$

Once we have encrypted a message, the decryption process is similar and can be described as follows:

(1) First, we find a positive integer d (the *decryption key*) such that

$$ed \equiv 1 \pmod{(p-1)(q-1)}.$$

(2) To decrypt an encoded message, we input each block of data into the decoding function g defined by

$$g(x) = x^d \pmod{m}.$$

It should be noted that not all of the steps described above are straightforward or easy to complete. For instance, the decryption key is defined to be an integer that satisfies a particular congruence relation. It is not immediately obvious that such an integer will always exist. We will have to use what we have learned about the integers to prove not only that a suitable decryption key exists, but also that it can be found in a relatively straightforward manner, and that the corresponding decoding function actually returns encrypted messages to their original, unencrypted state.

Two Examples

Before we go any further into investigating the details of RSA encryption and why it works, we will examine two different examples. For the first example, most of the computations can be done using

*As we will see later on, in order for RSA encryption to work, p and q each need to be larger than the number of elements in the given alphabet. This is usually not a problem since the security of RSA encryption relies on choosing primes that are huge, and certainly larger than the size of any reasonable alphabet.

a calculator but it would be easier to use a computer algebra system. A computer algebra system was used to do the computations in the second example. It is only necessary to do one of the examples.

Example 34.4. For the first example, we will use very small prime numbers so that some of the calculations can be done using a calculator. Namely, the calculations for the encryption process can be done with a calculator, but something like a computer algebra system is required to do the computations for the decryption process.

So we will use $p = 29$ and $q = 41$. So the modulus m and totient t are

$$\begin{aligned} m &= pq = 1189 \\ t &= (p-1)(q-1) = 1120 \end{aligned}$$

(This example is only for illustration and would not be used since the values of p and q could be easily determined from $m = 1189$.)

Our encryption key e must be chosen so that $\gcd(e, t) = 1$. We will use $e = 3$. So the modulus $m = 1189$ and encryption key $e = 3$ are made public so that anyone can send a message to this receiver. Suppose someone wants to sent the message "DOG". The numerical code for this is 031406. We now divide this into blocks of digits. Because of the size of m, we will use only two-digit blocks since a 4 digit block could be greater that m. (Note: Three digit blocks could also be used.) To encrypt the message, we use the encryption function f where

$$f(x) = x^e \pmod{m} = x^3 \pmod{1189}.$$

For the message 031406, we have

$$\begin{aligned} f(03) &= 03^3 \pmod{1189} = 27 \\ f(14) &= 14^3 \pmod{1189} = 366 \\ f(06) &= 06^3 \pmod{1189} = 216 \end{aligned}$$

To decrypt this message, the receiver must now determine d so that $ed \equiv 1 \pmod{(p-1)(q-1)}$ or $3d \equiv 1 \pmod{1120}$. To do this, we use the Euclidean Algorithm to determine the coefficients for $1 = 3r + 1120s$ according to Bezout's Identity (Theorem 1.17 on page 13.) We will omit the details, but it can be verified that

$$3(-373) + 1120 = 1$$

and so $3(-373) \equiv 1 \pmod{1120}$. By adding $3 \cdot 1120$ to both sides of this congruence, we obtain

$$\begin{aligned} 3(-373) + 3(1120) &\equiv 1 + 3(1120) \pmod{1120} \\ 3(747) &\equiv 1 \pmod{1120} \end{aligned}$$

and so $d = 747$. So the user must now decrypt each of these blocks using the decryption function g where

$$g(x) = x^d \pmod{m} = x^{747} \pmod{1189}.$$

For our three blocks, we use a computer algebra system to obtain

$$\begin{aligned} g(27) &= 27^{747} \pmod{1189} = 3 \\ g(366) &= 366^{747} \pmod{1189} = 14 \\ g(216) &= 216^{747} \pmod{1189} = 6 \end{aligned}$$

So the numerical decrypted message is 031406, which is the plaintext message DOG.

Example 34.5. All computations in this example were done using a computer algebra system such as Maple, Mathematica, or Sage. Suppose we want to use RSA encryption to encode the highly sensitive, top-secret message, "JOHNNY LOVES SALLY". We will begin by choosing p and q. In practice, p and q are often hundreds of digits long, but we will choose two smaller primes,

$$p = 400043344212007458013 \text{ and } q = 500030066366269001203.$$

With these choices of p and q, our modulus m and totient t are

$$m = pq = 200033699955714283345172521584008468989639$$

and

$$t = (p-1)(q-1) = 200033699955714283344272448173430192530424.$$

Recall that m will be made public, but p, q, and t will be kept secret. This is significant, since in order to calculate t from m, we would have to first factor m, a task that a computer could fairly easily complete for this example, but not for examples involving larger primes. In fact, one website on RSA cryptography notes that "if p and q are each 1024 bits long, the sun will burn out before the most powerful computers presently in existence can factor the modulus into p and q." †

The next step in encoding our message is to choose an encryption key. We need to choose a number e such that $\gcd(e, (p-1)(q-1)) = 1$. Note that any prime number that does not divide $t = (p-1)(q-1)$ will suffice here; a common choice is $e = 2^{16} + 1 = 65537$, and this is what we will use.

To perform the actual encryption, we will first break our message into three 6-character blocks. We will use the standard 00 – 25 encoding for the letters A – Z, and we will also use the code 99 to denote a space. Thus, the numerical representation of our message is:

$$091407131324 \;\; 991114210418 \;\; 991800111124$$

We then apply our encoding function to each of these 12-digit numbers:

$$\begin{aligned} f(091407131324) &= (091407131324)^e \pmod{m} \\ &= 009505729493564929202343371764084584555016 \\ f(991114210418) &= (991114210418)^e \pmod{m} \\ &= 012196119237767316793050190360104919489384 \\ f(991800111124) &= (991800111124)^e \pmod{m} \\ &= 124080637343749317837866219863773135637684 \end{aligned}$$

Note that we have appended zeros to the beginning of each encoded block so that each has exactly 42 digits. This would allow the entire message to be sent as a single string and then unambiguously decomposed into its three distinct parts prior to decryption.

To decrypt the message, we would first need to find the decryption key, d. It is at this step that some of the theory we have been studying in previous investigations is particularly useful (and in fact necessary). For now, however, we will skip over the "how" and simply assume that we have been able to find an integer, say

$$d = 921894177863251936178098635065731653140 81,$$

such that

$$ed \equiv 1 \pmod{(p-1)(q-1)}.$$

† `http://fringe.davesource.com/Fringe/Crypt/RSA/Algorithm.html`

A computer algebra system can then readily verify that

$$\begin{aligned} &g(009505729493564929202343371764084584555016) \\ &\qquad = (009505729493564929202343371764084584555016)^d \pmod m \\ &\qquad = 091407131324, \end{aligned}$$

$$\begin{aligned} &g(012196119237767316793050190360104919489384) \\ &\qquad = (012196119237767316793050190360104919489384)^d \pmod m \\ &\qquad = 991114210418, \end{aligned}$$

and

$$\begin{aligned} &g(124080637343749317837866219863773135637684) \\ &\qquad = (124080637343749317837866219863773135637684)^d \pmod m \\ &\qquad = 991800111124. \end{aligned}$$

In other words, the decoding function g returns each block of the encrypted message to its original, unencrypted state, as desired. It is interesting to note that, even in this example, the computations in the decryption process involve raising one 40-digit number to another 40-digit exponent. Fortunately, there are efficient algorithms for performing such exponentiations, even when the numbers involved are much larger (as would be the case if larger, and hence more realistic, values of p and q were chosen.)

Why RSA Decryption Works

Now that we've seen an example, we are ready to get to work and show that RSA encryption actually works the way it is intended. In particular, we must show three important facts:

- First, we must show that no matter what primes are used for p and q, it will always be possible to find an encryption key, e, that satisfies
$$\gcd(e, (p-1)(q-1)) = 1.$$
- Next, we must show that it is always possible to find a decryption key, d, such that
$$ed \equiv 1 \pmod{(p-1)(q-1)}.$$
- Finally, we must show that the decoding function $g(x) = x^d \pmod m$ is the inverse of the encoding function $f(x) = x^e \pmod m$. In other words, we must show that for all x in our alphabet,
$$(x^e)^d \pmod m = x^{ed} \pmod m = x.$$

The goal for the rest of this investigation is to establish these three facts, and the activities below suggest a series of steps that will accomplish exactly that goal.

Task 1: Finding the Encryption Key

Activity 34.6. Let p and q be any prime numbers.

(a) Explain why it is always possible to find a prime number e that does not divide $(p-1)(q-1)$.

(b) Explain why the number e found in part (a) would always satisfy

$$\gcd(e, (p-1)(q-1)) = 1.$$

Task 2: Finding the Decryption Key

Activity 34.7. Consider the fact that the encryption key e is chosen specifically so that

$$\gcd(e, (p-1)(q-1)) = 1.$$

(a) Use Bezout's Identity (Theorem 1.17 on page 13) to write down a linear combination corresponding to $\gcd(e, (p-1)(q-1))$.

(b) Use your answer to part (a) to explain why there must exist an integer d such that

$$ed \equiv 1 \pmod{(p-1)(q-1)}.$$

(c) What process or algorithm would allow you to actually determine the value of d that is guaranteed to exist by part (b)? (Hint: We have studied this algorithm in a previous investigation.)

(d) Suppose that we were able to find an integer d' such that

$$ed' \equiv 1 \pmod{(p-1)(q-1)},$$

but $d' < 0$. Explain how we could use d' to find a positive integer d that also satisfies

$$ed \equiv 1 \pmod{(p-1)(q-1)}.$$

(Hint: Add a convenient quantity to d'.)

Task 3: Proving an Inverse Relationship Between f and g

In order to prove that the decoding function g actually undoes the work of the encoding function f, we must show that for all x in our alphabet,

$$x^{ed} \pmod{m} = x.$$

Doing so will require the following three intermediate results:

Theorem 34.8 (Binomial Theorem)**.** *Let n be a positive integer, and let a and b be any real numbers. Then*

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k,$$

where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Theorem 34.9 (Freshman's Dream)**.** *Let p be a prime number. Then for all integers a and b,*

$$(a+b)^p \equiv (a^p + b^p) \bmod p.$$

Theorem 34.10 (Fermat's Little Theorem)**.** *Let p be a prime number. Then for every positive integer a,*

$$a^p \equiv a \bmod p.$$

Proofs of the Binomial Theorem and the Freshman's Dream are outlined in Exercises 10 and 11 of Investigation 5. (See page 81.) The proof of Fermat's Little Theorem follows from the Freshman's Dream. (If you have completed Investigation 23, an alternative proof using group theory is suggested in Exercise (20) on page 330.)

Activity 34.11. Use induction on a, along with the Freshman's Dream, to prove Fermat's Little Theorem.

Activity 34.12. Explain why the conclusion of Fermat's Little Theorem (namely, that $a^p \equiv a \pmod{p}$) is equivalent to

$$a^{p-1} \equiv 1 \pmod{p}$$

as long as $p \nmid a$.

We can now use the Freshman's Dream and Fermat's Little Theorem to establish an inverse relationship between the RSA encoding and decoding functions.

Activity 34.13. Let p, q, m, d, and e be as stated previously. Note that, by definition,

$$ed \equiv 1 \pmod{(p-1)(q-1)}.$$

(a) Use the definition of congruence to explain why $ed \equiv 1 \pmod{(p-1)}$ and $ed \equiv 1 \pmod{(q-1)}$.

(b) Use part (a), Activity 34.12, and the fact that p and q were chosen to be very large (and, in particular, much larger than the number of letters in the alphabet, $\mathbb{A}$) in order to prove that

$$x^{ed} \equiv x \pmod{p} \text{ and } x^{ed} \equiv x \pmod{q}$$

for all $x \in \mathbb{A}$.

(c) Use part (b) to explain why for all $x \in \mathbb{A}$, $x^{ed} \equiv x \pmod{m}$.

(d) Explain how your answer to part (c) actually implies that $x^{ed} \pmod{m} = x$. (Hint: It again matters that p and q, and thus m, are larger than the number of elements in $\mathbb{A}$. Remember that if two numbers are both less than m and yet congruent modulo m, then they must be equal.)

(e) Deduce from your answer to part (d) that the decoding function used in RSA encryption always returns an encrypted message to its original, unencrypted state.

Concluding Thoughts and Notes

Before we conclude our investigations of RSA encryption, a few additional observations are worth mentioning.

- Since p and q are chosen to be very large, it is important that e be large enough so that x^e is typically greater than the modulus, $m = pq$. If x^e is not greater than m, then messages can be easily decrypted by simply taking the e^{th} root of the encrypted data, since the encryption function in this case does not involve a reduction modulo m.

- RSA encryption schemes are deterministic, meaning that they have no random component to them. Because of this, potential attackers could use the public encryption key to develop a dictionary of likely words and their encryptions. This dictionary could then be used to try to decipher encrypted messages by comparing the encrypted words to the entries in the dictionary. Encrypting larger blocks of data (instead of individual letters) reduces this security vulnerability.

- One aspect of public key encryption that we have not considered is that of *signing* or *authentication*. Since RSA schemes enable anyone to encrypt a message, it is important to be able to verify that encrypted messages are actually from who they claim to be from. Several methods are available for this purpose, many of which involve introducing an additional private key that identifies the sender. See Exercise (7).

- Most experts agree that with large enough primes, RSA encryption seems to be secure for the near future. That is, even though the values of e and m are made public, the value of the decryption key d cannot really be determined from e and m. However, advances in computer science, and especially in the field of quantum computing, have the potential to solve computational problems such as integer factorization much faster than present-day computers. This would then render RSA encryption obsolete since the decryption key could be determined from e and m.

- Interest in RSA was primarily confined to mathematicians, computer scientists, and cryptography hobbyists until the invention of the World Wide Web. Beginning in the 1990s, there has been an explosion of online commerce. Some form of encryption was needed in order to send credit card numbers and other sensitive information over the Internet. Now, if you log into any secure web server, there is a good chance your computer has the server's public key and used it to secure the information you have sent. However, this information is often not encrypted directly using RSA. This is due to the fact that the RSA system is a so-called asymmetric cryptosystem. This simply means that there are two keys, one for encryption and one for decryption. A symmetric cryptosystem uses the same key for encryption and decryption and is usually much faster and uses fewer resources than an asymmetric system.

  Quite often, a file will be encrypted with a symmetric-key algorithm, and the symmetric key will be encrypted with RSA encryption. Under this process, only an entity that has access to the RSA private key will be able to decrypt the symmetric key. Without being able to access the symmetric key, the original file cannot be decrypted. This method can be used to keep messages and files secure, without taking too long or consuming too many computational resources.

  One widely used symmetric system is the Advanced Encryption Standard or AES. This was developed in the late 1990s by the National Institute of Standards and Technology (NIST) as the new government cipher standard. It is expected that at some time, AES will need to be replaced. The original expectation was that the standard should last 30 years, and NIST is supposed to review it every five years. So far, no problems have arisen with AES, but cryptographers are working on what might succeed AES to be ready just in case problems arise.

Exercises

(1) Caesar's's cipher is an example of a **shift cipher**, in that it encrypts messages by simply shifting each letter in the message by a fixed amount. (For example A $\rightarrow$ E, B $\rightarrow$ F, etc.) The encoding functions associated with shift ciphers always have the form

$$f(x) = x + a \pmod{n},$$

where n is the number of letters in the alphabet (typically 26).

(a) Assuming that the message below was encrypted using a shift cipher, decrypt the message. Make sure you explain how you determined which shift cipher was used.

```
XLMW QIWWEKI AEW IRGVCTXIH YWMRK E WLMJX GMTLIV.
```

(b) Was the following message encrypted using a shift cipher? Why or why not?

```
AJMZ YBZZGLB EGZ QCA BQPSOUABD
KZMQL G ZJMHA PMUJBS.
```

(2) In contrast to a shift cipher (see Exercise (1)), a **stretch cipher** uses multiplication instead of addition to encode messages. That is, instead of using an encoding function of the form

$$f(x) = x + a \pmod{26},$$

it uses one of the form

$$f(x) = ax \pmod{26},$$

where a is some integer.

(a) Use a stretch cipher of your choosing to encode this message: ABSTRACT ALGEBRA MAKES ME SMILE.

(b) Does the stretch cipher you used in part (a) have a corresponding decoding function? In other words, is there a rule that can be used to decode any message encoded by the cipher?no

(c) Is it always possible to decode a message that has been encoded using a stretch cipher? If so, explain how. Otherwise, determine the values of a (assuming a 26 letter alphabet) for which the corresponding stretch cipher is decode-able.

(3) The frequency with which each letter in the alphabet occurs in ordinary English is given in Table 34.1. [‡] Explain in a precise way how this table could be used to break shift and stretch ciphers.

(4) In Exercise (3) , we saw how analyzing the frequency of letters in a message encrypted with a shift or stretch cipher could help someone break such a cipher. So people began devising encryption methods in which a given letter that appears more than once in a message would be encrypted differently depending on its location in the message. For example, there are four A's in the plaintext message is ALABAMA, and we know that the Caesar cipher would encrypt each of these A's with the letter D.

[‡] This table originally appeared in *Applications of Abstract Algebra with Maple and MATLAB* (2nd ed.) by Klima, Stitzinger, and Sigmon, CRC Press, 2006.

Letter	Frequency (%)	Letter	Frequency (%)
A	8.167	N	6.749
B	1.492	O	7.507
C	2.782	P	1.929
D	4.253	Q	0.095
E	12.702	R	5.987
F	2.228	S	6.327
G	2.015	T	9.056
H	6.094	U	2.758
I	6.966	V	0.978
J	0.153	W	2.360
K	0.772	X	0.150
L	4.025	Y	1.974
M	2.406	Z	0.074

Table 34.1
Frequency of each letter in the English language

The Vigenére cipher is a method of encryption that uses a series of interwoven shift ciphers based on a keyword. For example, if the keyword is GOLF, then we determine the keyword number sequence, which is 6 – 14 – 11 – 5. This means that the first letter in the plaintext message will be shifted by 6, then second letter will be shifted by 14, the third letter will be shifted by 11, and the fourth letter will be shifted by 5. When we have used all of the letters in the keyword, we start over at the beginning of the keyword. So the fifth letter in the plaintext message will be shifted by 6, the sixth letter will be shifted by 14, the seventh letter will be shifted by 11, and the eighth letter will be shifted by 5.

We still follow the basic rule that if a plaintext letter corresponds to the number c, and we shift s places, then the encrypted letter will correspond to the number

$$f(c) = (c + s) \pmod{26}.$$

The numerical code for the plaintext message ALABAMA is 00110001001200.

(a) Use the keyword GOLF with a Vigenére cipher to encrypt the message ALABAMA, first as a numerical sequence and then using letters.

(b) The plaintext message had 4 A's. Were these four A's encrypted with same letter in the encrypted message?

(c) Explain how the receiver would decrypt this message.

(5) **Hill ciphers** use matrices to encode and decode messages. For instance, using the 2×2 matrix

$$A = \begin{bmatrix} 2 & 5 \\ 1 & 4 \end{bmatrix},$$

the message "ATTACK" would be encrypted by multiplying A by the vector representation of each pair of consecutive letters. Doing so, we obtain

$$A\begin{bmatrix}0\\19\end{bmatrix}=\begin{bmatrix}95\\76\end{bmatrix},\quad A\begin{bmatrix}19\\0\end{bmatrix}=\begin{bmatrix}38\\19\end{bmatrix},\text{ and }A\begin{bmatrix}2\\10\end{bmatrix}=\begin{bmatrix}54\\42\end{bmatrix}.$$

(a) Reduce these vectors modulo 26 in order to finish encrypting the message.

(b) Are the two A's in the original message encrypted to the same letter? What about the two T's?

(c) Are Hill ciphers more or less susceptible to frequency analysis (that is, the analysis suggested in Exercise (3)) than shift and stretch ciphers? Clearly explain your answer.

(d) What conditions must be placed on the encrypting matrix in order to guarantee that the resulting Hill cipher will be decode-able? Give a convincing argument to justify your answer.

(e) Use a Hill cipher with a matrix different than A to encrypt the message, "CRYPTOGRAPHY IS FUN". Then find the corresponding decrypting matrix, and verify that it does in fact return the encrypted message to its original form.

(6) Are any of the systems mentioned in Exercises 1 – 5 public-key schemes? That is, is it ever possible to encode messages using one of these schemes without also knowing how to decode messages?

(7) Suppose that person B wants to send a message to person A in such a way that A knows it is from B. To do this, person B needs a digital signature. So for this, we will assume that A has made public their encryption key consisting of n_A and e_A. Person B also has a public encryption key consisting of n_B and e_B. Of course, both of them have their own decryption keys: d_A for person A and d_B for person B.

Explain how B can use their decryption key d_B to "sign" the message to A so that A knows that the message is actually from person B.

Connections

Pure mathematics can be described as the study of mathematical concepts independently of any application outside of mathematics. Mathematicians have had differing opinions as to what constitutes pure versus applied mathematics. One of the more famous (at least among mathematicians) examples of this debate can be found in the essay from 1940 called "A Mathematician's Apology"' by the well-known British mathematician G.H. Hardy. Hardy preferred pure mathematics, which he often compared to painting and poetry, but he argued that the distinction was that applied mathematics sought to express physical truth in a mathematical framework, whereas pure mathematics expressed truths that were independent of the physical world.

In this investigation, we considered one of the applications of number theory and abstract algebra that is very important in our digital world. We saw how congruence and modular arithmetic from Investigation 2, along with Bezout's Identity from Investigation 1 and Fermat's Little Theorem (see Exercises (20) and (24) in Investigation 23) can be used to encrypt data in such a way that unauthorized people cannot decrypt the data. This allows for secure electronic transmission of information.