

1. CRYPTOGRAPHY

The science of cryptography deals with sending and receiving coded messages. Not only governments, but also financial institutions and businesses need frequently to transfer sensitive information from one user or from one computer to another in such a way that even if a message is intercepted by the wrong party, it cannot be read. The general public also needs secure methods of transmitting information, so that, for example, a credit card purchase made over the Internet does not allow one's name and credit card number to fall into the hands of an unscrupulous thief.

We use the term *cipher* to mean a system for encoding and decoding messages. We use the term *encryption* to denote the process of transforming (“encoding”) a plain text message into a coded message and the term *decryption* to denote the process of transforming (“decoding”) the coded message back into the original plain text message. All modern ciphers are based on mathematics and many are based on techniques and results from number theory.

Historically, people have not only used ciphers to keep their messages secret, but they also have devised ways to keep people from knowing that a message was even being sent. An example of a very old and very simple cipher, based on number theory and purportedly used by Julius Caesar, is the so-called *Caesar Cipher*. The idea of the Caesar cipher was to use a simple shift of letters. Replace every letter in the plain text message by the letter three letters to the right to get the coded message. To decode the coded message, one needs only replace each letter in the coded message by the letter three places to the left. The correspondence is shown in the table below.

Cleartext:	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Ciphertext:	D E F G H I J K L M N O P Q R S T U V W X Y X A B C

For example, the word **CAT** would be encrypted as **FDW**. This is obviously not a very sophisticated system and would be relatively easy to crack if the message was longer than a few letters.

When discussing cryptography, one usually translates letters to numbers via the following scheme:

Letters:	A B C D E F G H I J K L M N O P
Numbers:	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
Letters:	Q R S T U V W X Y Z
Numbers:	16 17 18 19 20 21 22 23 24 25

Note that if we translate letters to numbers, the Caesar cipher amounts to adding 3 and working modulo 26. For example, to encrypting the letter **U** we add 3 to **20** and get **23**, which corresponds to **X**. To encrypt **Y**, we add 3 to **24** and get **27**, which is **1** modulo 26, corresponding to the letter **B**. To make this precise, the encryption function ϵ for the Caesar cipher is

$$\epsilon(m) = (m + 3) \% 26$$

and the decryption function δ for the Caesar cipher is given by

$$\delta(m) = (m_1 - 3) \% 26.$$

Notice that δ is the *inverse function* for ϵ . In other words, for any message m , we have

$$\delta(\epsilon(m)) = \delta((m + 3) \% 26) = ((m + 3) - 3) \% 26 = m.$$

In general, a *shift cipher* is described mathematically by $\epsilon(x) = (x + b) \% 26$ for some chosen integer b . The decryption function is then given by $\delta(x) = (x - b) \% 26$. You can check for yourself that $\delta(\epsilon(x)) = x$ for any message x .

To facilitate discussions in cryptography, we usually assume there are two individuals — Alice and Bob — who are wanting to communicate privately, without their opponent — Oscar — knowing what they are saying to each other. We assume that Oscar has full access to the encrypted messages, however. Of course, we also assume that Oscar knows how to translate letters into numbers and conversely. Shift ciphers are far from secure for several reasons. First, there are only 26 possible shift ciphers, so if we assume that Oscar knows that Alice and Bob are using a shift cipher, it is very easy for him to figure out which one it is. Second, Oscar has only to correctly guess one letter correspondence (which would reveal the value of b) to crack the whole code.

We can take a step up in complexity from shift ciphers by considering *affine ciphers*. The idea here is that the encoding function ϵ has two parameters: a and b . We must choose a so that $\gcd(a, 26) = 1$, but b can be any integer. Then $\epsilon(x) = (ax + b) \% 26$. Let's consider an example:

(1.1) **Example.** Consider the affine cipher described by $\epsilon(x) = (5x + 11) \% 26$. We have, for example, $\epsilon(4) = (5(4) + 11) \% 26 = 5$, so the letter **E** is encrypted as **F**.

We then worked in groups on the following question:
Suppose you intercept the message **PDQZPSFA** which was encrypted using the affine cipher $\epsilon(x) = (5x + 11) \% 26$. What does the message say?

There are several approaches one can take here. The “brute force” approach would be to encrypt every letter of the alphabet to get a complete correspondence of all the letters. This requires 25 separate calculations. Rachel did this for the first few letters until she was able to guess the message (GO BIG RED) – which she then checked. Another method, and ultimately more efficient than the brute force approach, is to find the decryption function $\delta(x)$. Note that δ is the inverse function of ϵ , i.e., the function δ such that $\delta(\epsilon(x)) = x$ for any input x . How do we find $\delta(x)$? Actually, it's the same method you use to find inverse functions of real numbers.

Start with $y = \epsilon(x) = 5x + 11 \% 26$. Now interchange the roles of x and y : $x = 5y + 11 \% 26$. We want to solve for y in terms of x . Subtracting 11 from both sides, we have $5y \equiv x - 11 \pmod{26}$. To get y by itself on the left-hand side, we need to “divide” by 5. In the modular world, we do this by multiplying by the inverse of 5 modulo 26. (This is why we require $\gcd(a, 26) = 1$ in the definition of affine ciphers.) We quickly calculated the inverse of 5 modulo 26 to be 21. That is, $21(5) \equiv 1 \pmod{26}$. Multiplying both sides of our equation by 21, we get $y \equiv 21(x - 11) \equiv 21x + 3 \pmod{26}$. Thus, $\delta(x) = 21x + 3 \% 26$. Once we have the decryption function, we can quickly decode each letter. E.g., the first letter **P** gets translated as the number 15. Then $\delta(15) = 21(15) + 3 \% 26 = 6$, which translates to **G**.

(1.2) **Example.** Suppose $\epsilon(x) = 73x + 57 \% 101$. Find the decryption function δ .
Let $y = \epsilon(x) = 73x + 57 \% 101$. We need to find x in terms of y . We have $73x + 57 \equiv y \pmod{101}$, so $73x \equiv y - 57 \pmod{101}$. Since $\gcd(73, 101) = 1$, 73 has an inverse

modulo 101. Eric found the inverse to be 18. If we multiply both sides of the equation by 18, we will have solved for x :

$$\begin{aligned}\delta(y) = x &\equiv 18(y - 57) \pmod{101} \\ &\equiv 18y - 16 \pmod{101} \\ &\equiv 18y + 85 \pmod{101}.\end{aligned}$$

Reversing the roles of x and y , we get $\delta(x) = 18y + 85 \%101$.

We then returned to our discussion of ciphers. Although there are many more affine ciphers than shift ciphers (how many more?), they are really not that much more secure. Eve has only to make two educated guesses (instead of one) and he can recover the value of a and b in the encryption function $\epsilon(m)$. Once he has the encryption function, he can calculate the decryption function just as we did in the examples above. Both shift ciphers and affine ciphers are examples of *substitution* ciphers, where the ciphertext alphabet is just some (possibly random) permutation of the cleartext alphabet. A serious drawback to substitution ciphers is that one can use common knowledge about the English language (for example, the fact that “e” is by far the most common letter) to make guesses about what the various letters stand for. The “CRYPTOQUOTES” puzzles you find in the newspaper are examples of substitution ciphers.

One might consider encoding more than one letter at a time. For example, suppose we wish to encode 4 letters as one “block”. We can still translate each letter into a two-digit number as above, but then we consider the four letters together as an eight-digit number m . Such a number is certainly less than 10^8 , so we can use 10^8 as our modulus and proceed as before with our affine cipher. Choose parameters a and b with $\gcd(a, 10^8) = 1$, and set $\epsilon(m) = (am + b) \%10^8$. The frequencies for blocks of four letters are not nearly so well known as they are for individual letters, and so this type of affine cipher is much more secure.

There are still problems, however. The main problem is that before Alice and Bob can communicate using this affine cipher, they must decide on the values of a and b . (These values are called the “key” for the cipher.) They can’t just send these values to each other unencrypted, because then Oscar could read them and he would know the formula for ϵ . So how do Alice and Bob decide on their key?

One solution is to use *public key cryptography*. The basic idea of a public key system is that even if Oscar knows ϵ , he can’t figure out δ . In the RSA cyptosystem, which is the public key system we will focus on, the encryption function has the form

$$\epsilon(x) = x^e \%N$$

where e and N are carefully chosen positive integers. It turns out that the decrypting function has the same form: $\delta(x) = x^d \%N$. In general, given such an ϵ it is very difficult to find δ in any reasonable amount of time, even with the world’s fastest computers. However, Alice chooses N in such a way that she (and only she) can quickly compute δ . The secret lies in the prime factorization of N . Here is how it works:

Pick primes:: The first step for Alice is to pick two prime numbers p and q with $p \neq q$. In practice, these primes need to be very large — about 150 digits each— for the cipher to be secure.

- We’ll take $p = 7919$ and $q = 7937$.

Calculate n and k :: Next Alice simply sets $N := pq$ and $k = (p - 1)(q - 1)$. Since Alice knows the factorization of n , she can compute k easily. Notice that in practice, n will have at least 300 digits. This means that computing k would be very difficult without knowing the factorization of N , and factoring N would also be very difficult.

- In our example, we have $N = (7919)(7937) = 62853103$ and $k = (7918)(7936) = 62837248$.

Choose e — the encoding exponent:: The next step is to pick a value of e at random, making sure that $\gcd(e, k) = 1$. Alice does this by first selecting a value for e and then performing the Euclidean Algorithm to calculate $\gcd(e, k)$. If this gcd is 1, great. Otherwise, Alice simply chooses a new value of e .

- For our example, we'll just take $e = q = 7937$. (We wouldn't want to do this in practice, of course, as it may give away the factorization of N !)

Find d — the decoding exponent:: Now, Alice needs to find a value of d so that $de \equiv 1 \pmod{k}$. She can do this through using the Euclidean Algorithm and back substitution. This could be done by hand for small primes such as mine, but in practice we would do this on a computer.

- Using Maple, I found $d \equiv e^{-1} \equiv 49607937 \pmod{k}$.

When all is said and done, Alice's public keys are N and e , and her private key is d . The encoding function, which anyone in the world can use to send a message to Alice, is $\epsilon(x) = x^e \% N$. The decoding function, which only Alice knows, is $\delta(x) = x^d \% N$. Since in our example N is eight digits long and we want our messages to be less than N (so they will be remainders modulo N), we should break our message into four letter blocks and encode each of these blocks separately. For example, if we wanted to encode the word **MATH**, we plug $m = 12001907$ into our encryption function to get

$$\epsilon(12001907) = 12001907^{7937} \% 62853103 = 53218748.$$

One can also check (again, using a computer) that

$$\delta(53218748) = (53218748)^{49607937} \% 62853103 = 12001907.$$

So we see that the decryption function works (for this message, anyway). At this point, we need to see why $\delta(x) = x^d \% N$ is actually the decryption function. That is, we need to check that $\delta(\epsilon(x)) = x$ for all x . Chris and Erik will prove this in their project. For the proof, they will use the following homework problem:

Let p and q be distinct primes. Prove that $a \equiv b \pmod{pq}$ if and only if $a \equiv b \pmod{p}$ and $a \equiv b \pmod{q}$.