Affine Ciphers

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	z	a	b	c

For simplicity in our examples, we shall just run the words together with no encoding of spaces or punctuation, so that “I LOVE MATH” would be viewed as “ILOVEMATH”. Using the Caesar cipher, the encoded message would read “loryhpdwk”. This is obviously not a very sophisticated system and would be relatively easy to crack.

We can identify the letters of the alphabet with the numbers 0 through 25. We have:

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

Once we’ve done this, the Caesar cipher amounts to adding 3 and working modulo 26. For example, the letter “U” corresponds to the number 20. Enciphering it, we add 3 and get 23, which corresponds to “x”. If we start with the letter “Y”, we think of that as 24. Adding 3, we get 27, and modulo 26, this is 1 (or 01). The letter “b” corresponds to the number 01, and so “Y” is enciphered as “b”. To decrypt, we simply shift backward, which amounts to subtracting 3 (still working modulo 26).

To help us describe cryptography mathematically, we can introduce two functions: an *encryption* function which we call $\epsilon(m)$, and a *decryption* function which we call $\delta(s)$. Think of the variable m as representing the unencoded *message*, and the variable s are representing the *secret* encoded message. For the Caesar cipher, we have

$$\epsilon(m) = (m + 3) \pmod{26}$$

and

$$\delta(s) = (s - 3) \pmod{26}.$$

Notice that if we start with the message m, encrypt it to get s, and then decrypt s, we end up with m again. To see this, we have $\delta(s) = (s-3) \pmod{26} = ((m+3)-3) \pmod{26} = m \pmod{26} = m$. It’s important that our encryption function ϵ and our decryption function δ have this relationship; otherwise, we wouldn’t be able to recover our original messages!

The Caesar cipher is an example of a *shift cipher*. In general, a shift cipher shifts the entire plaintext alphabet by some amount, say b, to obtain the ciphertext alphabet. For the Caesar cipher, we have $b = 3$. In general, the encryption function for a shift cipher looks like $\epsilon(m) = (m + b) \pmod{26}$ and the decryption function looks like $\delta(s) = (s - b) \pmod{26}$. Shift ciphers are incredibly easy to crack because there are only 26 of them, including the one in which the ciphertext is the same as the plaintext. This means that if one knows that a shift cipher is being used, then one can simply try all 26 possibilities to find one that actually decrypts the ciphertext into something intelligible.

Shift ciphers are special cases of *substitution ciphers.* With a general substitution cipher, the ciphertext alphabet is a (possibly random) scrambling (rather than an orderly shift, as occurs with shift ciphers) of the plaintext alphabet. There are

$$26! = 26 \cdot 25 \cdot 24 \cdot \ldots \cdot 3 \cdot 2 \cdot 1 = 403,291,461,126,605,635,584,000,000$$

substitution ciphers possible, so simply trying each one is not a feasible option. (To see why there are this many, notice that there are 26 choices for the ciphertext letter corresponding to the plaintext "A", there are 25 choices remaining for the ciphertext letter corresponding to the plaintext "B", and so on.) Nonetheless, substitution ciphers are easy to crack. In particular, one can use common knowledge about the English language (for example, the fact that "e" is by far the most common letter and "th" is the most common pair of letters) to make guesses about what the various letters stand for. The "CRYPTOQUOTES" puzzles you find in the newspaper are examples of substitution ciphers.

Another type of substitution cipher is the *affine cipher* (or *linear cipher*). Even though affine ciphers are examples of substitution ciphers, and are thus far from secure, they can be easily altered to make a system which is, in fact, secure. To set up an affine cipher, you pick two values a and b, and then set $\epsilon(m) = (am + b) \pmod{26}$.

For example, if we take $a = 3$ and $b = 8$, then the encryption function is

$$\epsilon(m) = (3m + 8) \pmod{26}.$$

To encrypt the letter "C", we first note that "C" corresponds to the number 02. Plugging this in for m, we get $\epsilon(02) = 3(02) + 8 = 14$, and so "C" is encrypted as "O". To find our decryption function, we set $s = \epsilon(m)$ and solve for m in terms of s. We have:

$$\begin{aligned} s &\equiv 3m + 8 \pmod{26} \\ s - 8 &\equiv 3m \pmod{26} \text{ so} \\ 3m &\equiv s - 8 \pmod{26} \end{aligned}$$

Since $\gcd(3, 26) = 1$, we know that there will be an x with $3x \equiv 1 \pmod{26}$. We could use the Extended Euclidean Algorithm to find x, or we can simply notice that $(3)(9) = 27 \equiv 1 \pmod{26}$ and so $x = 9$ works. Now we multiply both sides by 9:

$$\begin{aligned} 27m &\equiv 9(s - 8) \pmod{26} \\ &\equiv 9s - 72 \pmod{26} \\ &\equiv 9s + 6 \pmod{26} \end{aligned}$$

which tells us that $m = \delta(s) = 9s + 6 \pmod{26}$.

In general, to construct an affine cipher, we begin by choosing a and b with $\gcd(a, 26) = 1$. Then $\epsilon(m) = (am + b) \pmod{26}$ and $\delta(s) = x(s - b) \pmod{26}$, where x satisfies $ax \equiv 1 \pmod{26}$.

Exercise: This exercise has two parts. Working in teams of 2,

1. you and your partner will create an affine cipher by choosing a and b with $\gcd(a, 26) = 1$. Using your cipher, $\epsilon(m) = (am + b) \pmod{26}$, encode a message that is between 8 and 12 letters long. Give this encoded message, along with your values of a and b, to another team to decipher.

2. you and your partner will decipher the message that the other team gave you. Using their values of a and b, decode their message using $\delta(s) = x(s - b) \pmod{26}$, where x satisfies $ax \equiv 1 \pmod{26}$

As we mentioned earlier, affine ciphers are not secure because they're really just special examples of substitution ciphers and so one may use "frequency analysis" to crack them. However, we can tweak the idea a bit and consider *affine block ciphers* instead. The mathematics is the same, only now instead of encrypting one letter at a time, we encrypt a block of letters together. As an example, suppose we want to take our block-length to be 4. This means that we divide our message into blocks of 4 letters and encrypt each block separately. The largest number we could end up with is 456,975 (corresponding to the highly unlikely 4-letter block "ZZZZ"), and so we need to be sure that our modulus is greater than 456,975. We could use 456,976 but it's just as easy (if not easier) to use 1,000,000. Now we proceed just as before. We choose a and b and set $\epsilon(m) = (am + b) \pmod{1,000,000}$. As long as we've chosen a so that $\gcd(a, 1,000,000) = 1$, we can find an integer x such that $ax \equiv 1 \pmod{1,000,000}$. In this case, our decryption function is $\delta(s) = x(s - b) \pmod{1,000,000}$.

Because we're now encrypting blocks of letters rather than single letters, frequency analysis will not work here. In other words, affine block ciphers are reasonably secure as long as the block size is large enough (blocks of size four will most likely be big enough).

Even though affine block ciphers are secure, there's still a problem with them. The problem is that they're *symmetric*. This means that anyone who knows the encryption function $\epsilon(m)$ also knows (or can easily figure out) the decryption function $\delta(s)$. For example, all one needs to do to figure out the formula for $\delta(s)$ given that $\epsilon(m) = (am + b) \pmod{1,000,000}$ is use the Extended Euclidean Algorithm to find x such that $ax \equiv 1 \pmod{1,000,000}$. This is easy to do either by hand or with the help of a computer.

To see why having a symmetric cipher is less than desirable, consider an imaginary scenario involving best friends, Jack and Katie. Jack moves to Texas to teach, and, of course, he and Katie plan on keeping in touch. Suppose Jack has a co-worked named Earl who turns out to be a real slime-bucket. Earl has found a way to listen in on all of Jack's phone conversations and he also reads all of his mail. How can Jack and Katie communicate without Earl knowing what they're saying? Certainly they want to use some sort of cryptographic system, and they know that affine block ciphers are reasonably secure. So Katie picks values of a and b and sends them to Jack, and they use an affine 4-letter block cipher with encryption function $\epsilon(m) = (am + b) \pmod{1,000,000}$. They can each use the Extended Euclidean Algorithm to compute the formula for $\delta(s)$, and so they can read the encrypted letters they send to each other. Unfortunately, Earl read the original letter in which Katie sent Jack the choices of a and b. Even more unfortunate is the fact that even though Earl is a slime-bucket, he's an *intelligent* slime-bucket, and so he too can use the Extended Euclidean Algorithm to find the formula for $\delta(s)$. This means that even though they're sending encrypted messages, Earl can still read what they're writing!

What is really needed is an *asymmetric* system — a system in which knowing the formula for $\epsilon(m)$ doesn't mean that you can find the formula for $\delta(s)$.

This handout was modified from Dr. Judy Walker's notes from her Math 398 course taught in the Spring of 2002 at UNL.