Cryptology course packet

Wesley Pegden

Version: September 4, 2009

Chapter 1

Classical Cryptology

1.1 The Caesar cipher and modular arithmetic

More than 2000 years ago, the military secrets of the Roman empire were kept secret with the help of cryptography. The 'Caesar cipher', as it is now called, was used by Julius Caesar to encrypt messages by 'shifting' letters alphabetically.

For example, we could encrypt the message `MEET AT TEN` by replacing each letter in the message with the letter which comes 3 letters later in the alphabet; `M` would get replaced by `P`, the `E`'s would get replaced by `H`'s, and so on. The encrypted message—called the **ciphertext**—would be `PHHW DW WHQ`.

This kind of encryption can be formalized mathematically by assigning a number to each letter:

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

To encrypt a message, we convert its letters to numbers, add 3 to them, and then convert them back into letters:

	M E E T	A T	T E N
	12 4 4 19	0 19	19 4 13
add 3:	15 7 7 22	3 22	22 7 16
	P H H W	D W	W H Q

The person we are sending the message to receives `PHHW DW WZR`, and has been told they can decrypt it by shifting the letters *back* by 3. This corresponds to subtracting three when we convert to numbers:

	P H H W	D W	W H Q
	15 7 7 22	3 22	22 7 16
subtract 3:	12 4 4 19	0 19	19 4 13
	M E E T	A T	T E N

This lets them decrypt the ciphertext and recover the original message (the **plaintext**).

When Caesar used the cipher, he always shifted by 3, but there's no reason for us to stick with this convention. For example, we could have encrypted the message `MEET AT TEN` by shifting the letters by 5 instead of 3:

	M E E T	A T	T E N
	12 4 4 19	0 19	19 4 13
add 5:	17 9 9 24	5 24	24 9 18
	R J J Y	F Y	Y J S

Now the plaintext is still `MEET AT TEN`, but the ciphertext is now `RJJY FY YJS`. We need to tell the person we are sending the message to how much we added in the encryption step (5 in this case) so that they know how much to subtract to recover the original message. This number is called the **key**. Just like before, they would decrypt `RJJY FY YJS` by subtracting:

	R J J Y	F Y	Y J S
	17 9 9 24	5 24	24 9 21
subtract 5:	12 4 4 19	0 19	19 4 16
	M E E T	A T	T E N

Ex. 1.1.1. Encrypt the message `MATH` with the Caesar cipher with 4 as the key.

Ex. 1.1.2. Encrypt the message `CRYPTO` with the Caesar cipher with 6 as the key.

Ex. 1.1.3. The message `QIIX PEXIV` was encrypted using the Caesar cipher with 4 as the key. Decrypt the message.

Ex. 1.1.4. The message `SKKZ NKXK` was encrypted using the Caesar cipher with 6 as the key.

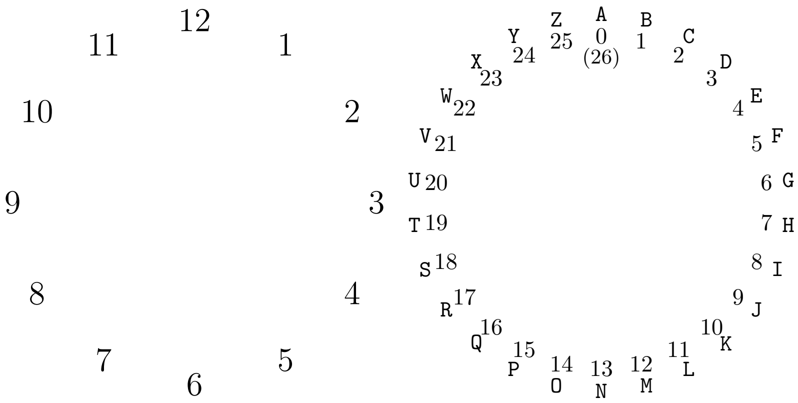
There's a subtlety to the Caesar cipher that hasn't come up yet. Let's return to our original example, and but change it just a little bit. We'll try to encode the message `MEET AT TWO` (note the change) with 5 as a key.

	M E E T	A T	T W O
	12 4 4 19	0 19	19 22 14
add 5:	17 9 9 24	5 24	24 27 19
	R J J Y	F Y	Y (?) T

What should go in the place of the question mark? It doesn't seem like there is a letter corresponding to the number 27. Or is there? Such a letter would be two places 'past' the letter `Z`. Whenever we are looking for a letter past the letter `Z`, we simply wrap around, and start back at the beginning of the alphabet again. In this way, the letter two 'past' `Z` is `B`; so the encrypted message will be `RJJY FY YBT`.

1.1.1. `QEXL`
1.1.3. `MEET LATER`

This is the same way we add when we're talking about time: what time will it be 5 hours after 10 o'clock? The answer isn't 15 o'clock (unless you're using 24 hour time): it's 3 o'clock.



The rings above can be used to add for time and for the Caesar cipher, respectively. What time is it 10 hours after 10 o'clock? Count 10 places past 10 on the left wheel, and you get 8. What letter would S get encrypted to using the Caesar cipher with key 10? Count 10 places past S on the wheel to the right, and get C.

Counting places on the wheel can get a bit tedious however; fortunately, we don't really have to do that. In the case of the clock, for example, observe that $10+10=20$, which is 8 more than 12 (which is one complete run of the clock). We write this fact as $20 \equiv 8 \pmod{12}$, which is read as "20 is congruent to 8 modulo 12". Similarly, we have that the letter S corresponds to the number 18, and $18+10=28$, which is 2 more than 26 (which is one complete turn of the letter wheel, since there are 26 letters). We write this $28 \equiv 2 \pmod{26}$. Note that we got the same answer as by counting on the wheel, since 2 corresponds to the letter C.

If we add big enough numbers, we can go around the wheels multiple times: For example, what time is it 21 hours after 9 o'clock? $9+21=30$, which is 6 hours past two complete runs of the clock (24 hours), thus it will be 6 o'clock. We can write $9+21 \equiv 6 \pmod{12}$. In general, the notation $a \equiv b \pmod{m}$ means that a is b more than some multiple of m. For example: $5 \equiv 2 \pmod{3}$ since $5 = 2+3$, and 3 is a multiple of 3; $8 \equiv 2 \pmod{3}$ since $8 = 2+6$, and 6 is a multiple of 3; and $7 \equiv 3 \pmod{2}$, since $7 = 3+4$ and 4 is a multiple of 2.

Ex. 1.1.5. Which of the following are true?

(a) $11 \equiv 5 \pmod{3}$
(b) $13 \equiv 4 \pmod{5}$
(c) $9 \equiv 6 \pmod{5}$
(d) $9 \equiv -6 \pmod{5}$

Ex. 1.1.6. Which of the following are true?

(a) $6 \equiv 3 \pmod{2}$
(b) $6 \equiv 2 \pmod{3}$
(c) $15 \equiv 3 \pmod{6}$
(d) $6 \equiv -3 \pmod{5}$

Returning to the example of the letter S (corresponding to the number 18) being encrypted by the Caesar cipher using the key 10, we already pointed out that $18+10 \equiv 2 \pmod{26}$, which means that the encryption results in the letter C. If you think about it, though, $18+10 \equiv 54 \pmod{26}$ is also true, since $28 = 54 + (-52)$, and -52 is a multiple of 26. In fact, its even true that $18+10 \equiv 28 \pmod{26}$, since $28 = 28+0$, and 0 is a multiple of 26! In fact, there are infinitely many numbers that 28 is congruent to modulo 26. For the purposes of encrypting the letter S, however, we don't use any of these other congruences, since they don't give numbers between 0 and 25. In general, given any problem of of the form $a \equiv \underline{\quad} \pmod{m}$ there is exactly *one* number which can fill in the blank which lies between 0 and $(m-1)$. How can we find this number? This is just the distance between a and the closest multiple of m smaller than a. If we divide a by m, then the remainder of the division problem corresponds to this distance. We say that a *reduces* to the remainder modulo m. For example, 28 reduces to 2 modulo 26 because $26\overline{)28}$ gives a remainder of 2. (Note that 28 is 2 more than 26, which is the closest multiple of 26 smaller than 28.) We use the notation MOD to indicate this reduction modulo m, so $28 \text{ MOD } 26 = 2$. Notice the difference between the problems $28 \equiv \underline{\quad} \pmod{26}$ and $28 \text{ MOD } 26 = \underline{\quad}$. The first question has infinitely many correct answers (2, 28, 54, -24, *etc.*), while the second question has only one correct answer (2).

Ex. 1.1.7. Reduce each integer to the given modulus.

(a) $34 \text{ MOD } 26 = \underline{\quad}$
(b) $55 \text{ MOD } 26 = \underline{\quad}$
(c) $26 \text{ MOD } 26 = \underline{\quad}$
(d) $5 \text{ MOD } 26 = \underline{\quad}$

Ex. 1.1.8. Reduce each integer to the given modulus.

(a) $11 \text{ MOD } 26 = \underline{\quad}$
(b) $59 \text{ MOD } 26 = \underline{\quad}$
(c) $63 \text{ MOD } 26 = \underline{\quad}$
(d) $28 \text{ MOD } 26 = \underline{\quad}$

1.1.5. (a)True, since $11 = 5+6$ and 6 is a multiple of 3; (b) False, since $13 = 4+9$, but 9 is not a multiple of 5; (c) False, since $9 = 6+3$, but 3 is not a multiple of 5; (d)True! since $9 = (-6)+15$, and 15 is a multiple of 5.

Things seem a bit trickier if we are trying to reduce a negative number, but the meaning of the MOD operation is the same. For example, what is -32 MOD 26? The closest multiple of 26 less than -32 is -52, and $-32 = -52+20$, so -32 MOD $26 = 20$. To use division to perform the MOD operation, we would say that $26\overline{)-32}$ is -2, with a remainder of 20, since $26 \cdot (-2) = -52$, and $-52 + 20 = -32$. Long division with negative numbers can seem a bit confusing, but there is an easy way out! Given any number a, you can always find a MOD M just by adding or subtracting multiples of m until you have something between 0 and $(m-1)$. For example, If we want to compute -5 MOD 26, we can add 26 to -5. This gives 21, so -5 MOD $26 = 21$. If we want to find -37 MOD 26, we can add 26 to -37, giving -11. This still lies below 0, so we add 26 again, to get 15. So we got that -37 MOD $26 = 15$. Note that since dding or subtracting m doesn't change the distance between a and the next smallest multiple of m, this will always end up giving the correct reduction.

Ex. 1.1.9. Reduce each integer to the given modulus.

(a) -6 MOD $26 = \_\_$
(b) -12 MOD $26 = \_\_$
(c) -34 MOD $26 = \_\_$
(d) -55 MOD $26 = \_\_$

Ex. 1.1.10. Reduce each integer to the given modulus.

(a) -10 MOD $26 = \_\_$
(b) -15 MOD $26 = \_\_$
(c) -43 MOD $26 = \_\_$
(d) -62 MOD $26 = \_\_$

Armed with this new modular arithmetic, lets return to the Caesar cipher. Let's consider encryption of the phrase `THEY COME BY SEA` using the Caesar cipher with a key of 18. As before, first we translate letters into numbers:

	T	H	E	Y	C	O	M	E	B	Y	S	E	A
	19	7	4	24	2	14	12	4	1	24	18	4	0

Then we add the key (18 in this case) and reduce the results modulo 26:

	T	H	E	Y	C	O	M	E	B	Y	S	E	A
	19	7	4	24	2	14	12	4	1	24	18	4	0
add 18:	37	25	22	42	20	32	30	22	19	42	36	22	18
MOD 26:	11	25	22	16	20	6	4	22	19	16	10	22	18

1.1.7. 8,3,0,5. **1.1.9.** 20,14,18,23.

Finally, we convert back to letters to get the ciphertext:

	T	H	E	Y	C	O	M	E	B	Y	S	E	A
	19	7	4	24	2	14	12	4	1	24	18	4	0
add 18:	37	25	22	42	20	32	30	22	19	42	36	22	18
MOD 26:	11	25	22	16	20	6	4	22	19	16	10	22	18
	L	Z	W	Q	U	G	E	W	T	Q	K	W	S

So we would send the message `LZWQ UGEW TQ KWS`. If the receiving party knows that the key is 18, they can recover the original message by subtracting 18 and reducing modulo 26:

	L	Z	W	Q	U	G	E	W	T	Q	K	W	S
	11	25	22	16	20	6	4	22	19	16	10	22	18
subtract 18:	-7	7	4	-2	2	-12	-14	4	1	-2	-8	4	0
MOD 26	19	7	4	24	2	14	12	4	1	24	18	4	0
	T	H	E	Y	C	O	M	E	B	Y	S	E	A

1.2 Breaking the Caesar cipher

The normal function of an encryption scheme is that one person ('Alice') sends a message to another ('Bob'). As long as Bob knows the key, he can decrypt the message. But what if a third party ('Carla') intercepts the message? Can she figure out what it says, even without knowing the key? Of course, the whole point of encrypting the message is to prevent this!

Consider the intercepted message

`T QZFYO ESP MLR`

which was encrypted with the Caesar cipher. Even without knowing the key, we have a lot of information; for example, we know that the message begins with a one-letter word. Assuming the message is in English, the should mean that `T` was encrypted either from the letter `A` or the letter `I`.

`T` corresponds to the number 19, and `A` to the number 0, which means that for `A` to get encrypted to `T`, the key would have to be 19. Based on this guess, we can try decrypting the message as if it was encrypted with 19 as the key:

	T	Q	Z	F	Y	O	E	S	P	M	L	R
	19	16	25	5	24	14	4	18	15	12	11	17
subtract 19:	0	-3	6	-4	5	-5						
MOD 26	0	23	6	22	5	21						
	A	X	G	W	F	V						

Since the beginning doesn't work out, we don't even have to bother trying the rest of the message: it seems like 19 is definitely not the key. So what `T` in the ciphertext corresponds to `I` in the plaintext (instead of `A`)? Since `T` corresponds to 19 and `I` corresponds to 8, this would mean the encryption key is 11. Let's try that out:

	T	Q	Z	F	Y	O	E	S	P	M	L	R
	19	16	25	5	24	14	4	18	15	12	11	17
subtract 11:	8	5	14	-6	13	3	-7	7	4	1	0	6
MOD 26	8	5	14	20	13	3	19	7	4	1	0	6
	I	F	O	U	N	D	T	H	E	B	A	G

And we've broken the message. The important thing to notice from this example is that **if we can guess just one letter of the plaintext correctly, we can break a whole message encrypted with the Caesar cipher**.

Ex. 1.2.1. Break these Caesar ciphers:

(a) `PAXG LAHNEW B KXMNKG`

(b) `QUCN ZIL U JBIHY WUFF`

(c) `GUR ENOOVG PENJYRQ BHG BS VGF UBYR` (Hint: what three letter words are likely to appear at the beginning of an English sentence?)

It's clear that the spacing of a message already gives lots of information which can be used to break it. For this reason, encoded messages have traditionally been written without their original spacing so that someone trying to break the code can't use this information. For example, if we wanted to send the message `WHEN WILL YOU RETURN` using the Caesar cipher with 10 as a key, we first break the message into groups of 5 letters, ignoring the original spacing:

`WHENW ILLYO URETU RN`

Now if we encrypted this message with 16 as a key, for example, it would become

`LWTCL XAAND JGTIJ GC`

and if someone intercepts the message who doesn't have the key, they would have to try to break it without knowing the lengths of any words. The intended recipient, using the key, can recover the message `WHENW ILLYO URETU RN` and understand it even without the correct spacing.

Even without word spacing intact, it is still possible to break the cipher! Imagine we have intercepted the following message, encrypted using the Caesar cipher with an unknown key:

`THTWW CPEFC YLQEP CESCP POLJD`

The letters which appear most frequently in this message are `C` (4 times) and `P` (4 times). The most common letter in the English language is `E`, so it is likely that `E` was encrypted to either `C` or `P`. `E` corresponds to the number 4, and `C` corresponds to the number 2, so for `E` to be encrypted to `C` the key would have to be 24 (since $4 + 24 = 28$, and 28 MOD 26 = 2). Decrypting with key 24 gives: `VJVYY ERGHE` . . . , which is nonsense. Since this didn't work, we guess instead that `E` was encrypted to `P`; in this case, the key would have been $15 - 4 = 11$. Decrypting with 11 as the key gives

`IWILL RETUR NAFTE RTHRE EDAYS`

and so the message is 'I will return after three days'. This technique to break codes is called **frequency analysis**, since it uses the ordinary frequency of letters in the English language to figure out how a message was encrypted. The table below shows the frequencies of letters in Project Gutenberg's collection of public-domain English-language books.

1	e	12.58%	14	m	2.56%
2	t	9.09%	15	f	2.35%
3	a	8.00%	16	w	2.22%
4	o	7.59%	17	g	1.98%
5	i	6.92%	18	y	1.90%
6	n	6.90%	19	p	1.80%
7	s	6.34%	20	b	1.54%
8	h	6.24%	21	v	0.98%
9	r	5.96%	22	k	0.74%
10	d	4.32%	23	x	0.18%
11	l	4.06%	24	j	0.15%
12	u	2.84%	25	q	0.12%
13	c	2.58%	26	z	0.08%

Table 1.1: Frequencies of letters in English text.

Notice that the letter `C` in the ciphertext above corresponded to the letter `R` in the correctly decoded plaintext; even though `C` was just as common as `P` (which turned out to be `E`) the letter `R` is only the 9th most common letter in English. With messages as short as the one above, this kind of variation means that there can be a lot of trial and error in the application of frequency analysis.

Ex. 1.2.2. Break the following message (which was encrypted with the Caesar cipher) using frequency analysis.

`MAXLX TKXGM MAXWK HBWLR HNKXE HHDBG ZYHK`

It appears that, in Caesar's time, his cipher was never broken, although there is a reference by the writer Aulus Gellius to a "rather ingeniously written treatise by the grammarian Probus" concerning Caesar's cryptographic techniques.

The earliest surviving account of a work describing how to break the cipher is "A Manuscript on Deciphering Cryptographic Messages", written in the 9th century by the Arab philosopher, scientist, and mathematician Al-Kindi, which contains the first known description of the technique of frequency analysis.

1.3 Modular multiplication and the affine cipher.

The Caesar cipher worked by 'adding' a key to a message. What about doing some other operation instead? Subtracting actually wouldn't be any different:

subtracting by a number modulo 26 is always the same as adding some other number modulo 26 (for example, adding 10 (mod 26) is the same as subtracting 16 (mod 26)), so an encryption scheme based on modular subtraction would actually just be the Caesar cipher.

We could try basing an encryption scheme on modular multiplication, however. Let's try encrypting the message `MEETA TTEN` ('meet at ten', broken into blocks of length 5) by multiplying by 2 (mod 26).

	M	E	E	T	A	T	T	E	N
	12	4	4	19	0	19	19	4	13
times 2:	24	8	8	38	0	38	38	8	26
MOD 26:	24	8	8	12	0	12	12	8	0
	Y	I	I	M	A	M	M	I	A

There's a problem here.... Both `A` and `N` got encrypted to the same letter (`A`). And in fact, other letters also have this problem: had it been part of the original message, `G` would have been encrypted to `M`, just like `T` was. Here's how multiplying by 2 (mod 26) affects the all the letters in the alphabet:

	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
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
×2	0	2	4	6	8	10	12	14	16	18	20	22	24	26	2	4	6	8	10	12	14	16	18	20	22	24
	A	C	E	G	I	K	M	O	Q	S	U	W	Y	A	C	E	G	I	K	M	O	Q	S	U	W	Y

You can see that, for every possible ciphertext letter, there are two different plaintext letters that would get encrypted to it. And, on the other hand, some letters—`B`, `D`, `F`, *etc.*—never appear as ciphertext letters.

All this means there can't possibly be some reliable way to decrypt messages that were encrypted like this. Even if we know the key (in this case 2), we can't necessary figure out what the message was. For example, if we receive the message `AAM`, encrypted by multiplying by 2, the original message could have been `ANT`, or `NAG`, or `NAT`, *etc.*

What if we tried multiplying by a different number? Here's how the alphabet is transformed under multiplication by 3:

	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
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
×3	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2	5	8	11	14	17	20	23
	A	D	G	J	M	P	S	V	Y	B	E	H	K	N	Q	T	W	Z	C	F	I	L	O	R	U	X

Notice that, here, no two plaintext letters got sent to the same ciphertext letter, meaning that it *should* be possible to recover a message which was encrypted with multiplication by 3. Why do 2 and 3 behave so differently?

Multiplicative inverses and modular multiplication

The important difference between 2 and 3 from the standpoint of the previous example is that 3 has a *multiplicative inverse* to the modulus 26. A multiplicative inverse is something you can multiply a number by to get 1. So, $\frac{1}{3}$ is a multiplicative inverse for the number 3, in the usual sense. But for our purposes, we want an *integer* that when multiplied by 3, gives something which is congruent to 1 (mod 26). 9 is such a number, since $3 \times 9 = 27 \equiv 1 \pmod{26}$. Just like we decrypted Caesar cipher messages by subtracting the encryption key, we can decrypt a message encrypted under multiplication by multiplying by the multiplicative inverse of the key, since this 'reverses' the multiplication operation.

For example, if we encrypt the message `MEETA TTEN` with the multiplication cipher with the key 3, we get

	M	E	E	T	A	T	T	E	N
	12	4	4	19	0	19	19	4	13
times 3	10	12	12	5	0	5	5	12	13
	K	M	M	F	A	F	F	M	N

And now we can decrypt:

	K	M	M	F	A	F	F	M	N
	10	12	12	5	0	5	5	12	13
times 9	12	4	4	19	0	19	19	4	13
	M	E	E	T	A	T	T	E	N

Here the multiplication and MOD steps are shown as a single step; so, for example, `K` decrypts to `M` because $9 \cdot 10 = 90 \equiv \underline{12} \pmod{26}$. Reducing 90 (mod 26) can be done quickest with division: $26\overline{)90}$ gives a remainder of 12.

We could decrypt the message because we could find a multiplicative inverse for 3 (mod 26). You can check, on the other hand, that there is *no* such multiplicative inverse for 2: 2 times any number is never congruent to 1 (mod 26), and decryption is not possible for the message `YIIMA MMIA` given at the beginning of the section.

Carrying out modular multiplication can get a bit tedious, so it's worthwhile to have a Modulo 26 multiplication table (Table 1.2). With the table, its easy to check which numbers have multiplicative inverses modulo 26: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, and 25, whose inverses are 1, 9, 21, 15, 3, 19, 7, 23, 11, 5, 17, and 25, respectively. The numbers which have no inverse modulo 26 are 0, 2, 4, 6, 8, 10, 12, 13, 14, 16, 18, 20, 22, and 24. What distinguishes these two sets of numbers? The numbers with inverses are those which are *relatively prime* to 26 (they have no common factors other than 1 with 26). The numbers without inverses are those which share some divisor other than 1 with 26. Note that this is all of the even numbers (which share the divisor 2 with 26), and 13 (which shares the divisor 13 with 26).

	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
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
2	0	2	4	6	8	10	12	14	16	18	20	22	24	0	2	4	6	8	10	12	14	16	18	20	22	24
3	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2	5	8	11	14	17	20	23
4	0	4	8	12	16	20	24	2	6	10	14	18	22	0	4	8	12	16	20	24	2	6	10	14	18	22
5	0	5	10	15	20	25	4	9	14	19	24	3	8	13	18	23	2	7	12	17	22	1	6	11	16	21
6	0	6	12	18	24	4	10	16	22	2	8	14	20	0	6	12	18	24	4	10	16	22	2	8	14	20
7	0	7	14	21	2	9	16	23	4	11	18	25	6	13	20	1	8	15	22	3	10	17	24	5	12	19
8	0	8	16	24	6	14	22	4	12	20	2	10	18	0	8	16	24	6	14	22	4	12	20	2	10	18
9	0	9	18	1	10	19	2	11	20	3	12	21	4	13	22	5	14	23	6	15	24	7	16	25	8	17
10	0	10	20	4	14	24	8	18	2	12	22	6	16	0	10	20	4	14	24	8	18	2	12	22	6	16
11	0	11	22	7	18	3	14	25	10	21	6	17	2	13	24	9	20	5	16	1	12	23	8	19	4	15
12	0	12	24	10	22	8	20	6	18	4	16	2	14	0	12	24	10	22	8	20	6	18	4	16	2	14
13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13
14	0	14	2	16	4	18	6	20	8	22	10	24	12	0	14	2	16	4	18	6	20	8	22	10	24	12
15	0	15	4	19	8	23	12	1	16	5	20	9	24	13	2	17	6	21	10	25	14	3	18	7	22	11
16	0	16	6	22	12	2	18	8	24	14	4	20	10	0	16	6	22	12	2	18	8	24	14	4	20	10
17	0	17	8	25	16	7	24	15	6	23	14	5	22	13	4	21	12	3	20	11	2	19	10	1	18	9
18	0	18	10	2	20	12	4	22	14	6	24	16	8	0	18	10	2	20	12	4	22	14	6	24	16	8
19	0	19	12	5	24	17	10	3	22	15	8	1	20	13	6	25	18	11	4	23	16	9	2	21	14	7
20	0	20	14	8	2	22	16	10	4	24	18	12	6	0	20	14	8	2	22	16	10	4	24	18	12	6
21	0	21	16	11	6	1	22	17	12	7	2	23	18	13	8	3	24	19	14	9	4	25	20	15	10	5
22	0	22	18	14	10	6	2	24	20	16	12	8	4	0	22	18	14	10	6	2	24	20	16	12	8	4
23	0	23	20	17	14	11	8	5	2	25	22	19	16	13	10	7	4	1	24	21	18	15	12	9	6	3
24	0	24	22	20	18	16	14	12	10	8	6	4	2	0	24	22	20	18	16	14	12	10	8	6	4	2
25	0	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Table 1.2: Multiplication modulo 26

Theorem 1.3.1. *A number a has a multiplicative inverse modulo some number n if and only if they are relatively prime. In other words, the congruence* $a \cdot x \equiv 1 \pmod{n}$ *has a solution x if and only if* $\gcd(a, n) = 1$.

In the above theorem, $\gcd(a, n)$ stands for the 'greatest common divisor' of a and n, which is the number divisible by all common divisors of a and n. For example, $\gcd(30, 75) = 15$, since the common divisors of 30 and 75 are $1, 3, 5, 15$, 15 is divisible by all of these. By convention, $\gcd(0, x) = x$ (e.g., $\gcd(0, 3) = 3$), since, in a certain sense, 0 is divisible by any number: for any number x, $x \cdot 0 = 0$. Notice that $\gcd(a, n) = 1$ just means that a and n have no common divisors, and so are relatively prime.

Ex. 1.3.1. The ciphertext `IASSC GW` was encrypted using the multiplication cipher with 4 as the key, while `KADDI U` was encrypted by multiplication with 5 as the key. It is possible to decrypt one of these messages. Indicate which can be decrypted, briefly explain why, and give the decryption.

What we've learned in this section is that we can encrypt a message using modular multiplication so long as the key used is relatively prime to 26, in which case the encrypted message can be decrypted by multiplying by the inverse. However, this points out a serious weakness of the multiplication cipher: there

are only 12 possible keys, (1,3,5,7,9,11,15,17,19,21,23,25), and only 11 keys if we discount 1 as a key, since it doesn't change the message at all. Compared with 26 possible keys for the Caesar cipher (25 which change the message), and the multiplication cipher is actually *less* secure than the Caesar cipher in terms of how many possible keys there are.

It is possible, however, to combine the operations of the Caesar and Multiplication ciphers into a single cipher which is more secure.

1.4 The Affine Cipher

The affine cipher works through a combination of modular multiplication and modular addition. To encrypt a plaintext letter with a key given by a pair of numbers (a, b), we convert the letter to a number, then multiply it by a modulo 26, and then add b to the result modulo 26, and convert the result to a letter. In other words, we take a plaintext letter corresponding to a number x and turn it into a ciphertext letter corresponding to the number y with the congruence $y \equiv a \cdot x + b \pmod{26}$. Let's see how this works when encrypting the message `MEETA TTEN` with the affine cipher, using the key $(3, 10)$:

	M	E	E	T	A	T	T	E	N
x	12	4	4	19	0	19	19	4	13
$y \equiv 3x + 10 \pmod{26}$	20	22	22	15	10	15	15	22	23
	U	W	W	P	K	P	P	W	X

Table 1.2 is a big help when carrying out the multiplications.

How can we decrypt the message? The message was encrypted according to the congruence

$$y \equiv 3x + 10 \pmod{26}.$$

When decrypting the message, we know y and are trying to figure out x; so let's solve this congruence for x. First we can subtract 10 from both sides of the congruence:

$$y - 10 \equiv 3x \pmod{26}.$$

Note that -10 is congruent to 16 modulo 26, so, if we want, we can make this change:

$$y + 16 \equiv 3x \pmod{26}.$$

Finally, to deal with the 3, we can multiply by 9, since that is the multiplicative inverse of 3:

$$9(y + 16) \equiv 9 \cdot 3x \pmod{26}$$

which simplifies to

$$9y + 14 \equiv x \pmod{26}.$$

($9 \cdot 16 = 14$ can be found by looking at the table.) And we have found the decryption congruence for the key (3,10), and can use it to decrypt the message:

	U	W	W	P	K	P	P	W	X
y	20	22	22	15	10	15	15	22	23
$x \equiv 9y + 14 \pmod{26}$	12	4	4	19	0	19	19	4	13
	M	E	E	T	A	T	T	E	N

Note that to find the decryption congruence, it was necessary to multiply the by inverse of 3. This brings up an important point: for the same reason that the multiplication cipher requires a key which is relatively prime to 26, **the number a in a key (a, b) used for the affine cipher must be relatively prime to 26**, otherwise it will not have an inverse and there will be no suitable decryption congruence.

Ex. 1.4.1. Indicate all the key-pairs in the following list which can be used for the affine cipher: $(5, 6)$, $(13, 17)$, $(5, 5)$, and $(6, 6)$.

Ex. 1.4.2. Indicate all the key-pairs in the following list which can be used for the affine cipher: $(6, 5)$, $(18, 19)$, $(17, 13)$, and $(17, 15)$.

Ex. 1.4.3. Encrypt the message MATHI SFUN using the affine cipher with key $(7, 11)$.

Ex. 1.4.4. Encrypt the message CRYPT OISFU N with the affine cipher with $(11, 15)$ as a key.

Ex. 1.4.5. Decrypt the message OAAXG XLCSX YD, which was encrypted with the affine cipher using $(5, 6)$ as a key.

1.4.1 Breaking the affine cipher

If an eavesdropper's only approach to breaking an encryption system is to try all possible keys, the affine cipher is already doing much better than the Multiplication or Caesar ciphers (which took 12 and 26 keys, respectively).

Ex. 1.4.6. How many possible keys (a, b) are there for the affine cipher? (Remember, a must be relatively prime to 26!)

However, just like the Caesar cipher, it is possible to break the affine cipher *without* having to try all the keys.

Assume we have intercepted the following message, encrypted with the affine cipher:

```
MCCLL IMIPP ISKLN UHCGI MCKBI XCUMT IPLKX
LRIGW MCXLA MWALV CCDGJ KXYCR
```

We can use frequency analysis to try to break the message. Counting shows that the most common letters in the message are C, I, and L, which occur 9, 7, and 7 times, respectively. Since e is the most common letter in English text, it is natural for us to make the guess that the ciphertext letter C was encrypted from the plaintext letter E.

1.4.3. RLOIP HUVY

1.4.1. $(5, 6)$ and $(5, 5)$

Can we work backwards to break the message now? We know that the message was encrypted using the formula

$$y \equiv ax + b \pmod{26}, \tag{1.1}$$

where the pair (a, b) is the affine cipher key. We guessed that E got encrypted to C; this would mean that for the plaintext $x = 4$, we get the ciphertext $y = 2$. Plugging these values into line (1.1), we get that

$$2 \equiv 4a + b \pmod{26}. \tag{1.2}$$

Can we solve this congruence to figure out the key (a and b) so that we will be able to decrypt the message? No we can't! We have only one congruence, but two unknowns! Just like when solving equations, it is necessary to have at least as many congruences as unknowns to find a solution. How can we get another congruence?

We can make another guess based on frequency analysis. For example, referring to Table 1.1, we see that t is the second most common letter in the English language, so it is natural to guess that T in the plaintext was encrypted to either I or L (the most common letters in the ciphertext after C). If we make the guess that T was encrypted to I, this implies that $y = 8$ for $x = 19$. Plugging this into line (1.1) gives that

$$8 \equiv 19a + b \pmod{26}. \tag{1.3}$$

Now, we can solve the system of congruences

$$\begin{cases} 2 \equiv 4a + b \pmod{26} \\ 8 \equiv 19a + b \pmod{26} \end{cases} \tag{1.4}$$

for a and b. One way to solve a system of congruences or equations is by subtracting multiples of one equation from the other one. In this case, subtracting the second congruence from the first one gives

$$-6 \equiv -15a \pmod{26},$$

which is equivalent to

$$20 \equiv 11a \pmod{26}.$$

Now we can solve for a by multiplying both sides by the multiplicative inverse of 11 (mod 26), which we can see is 19 by looking at Table 1.2. So we get:

$$19 \cdot 20 \equiv 19 \cdot 11a \pmod{26}$$

and so

$$16 \equiv a \pmod{26}. \tag{1.5}$$

However, we see we have a problem. Recall that a must always be relatively prime to 26 for the affine cipher to work; thus one of our guesses must have been wrong. Let's still guess that E is encrypted to C, but now let's guess that T is encrypted to L. Now our system of congruences is

$$\begin{cases} 2 \equiv 4a + b \pmod{26} \\ 11 \equiv 19a + b \pmod{26} \end{cases} \tag{1.6}$$

Subtracting these equations gives

$$-9 \equiv -15a \pmod{26}$$

which is equivalent to

$$17 \equiv 11a \pmod{26}$$

Multiplying both sides by 19 (the inverse of 11 (mod 26)) gives

$$a \equiv 11 \pmod{26}. \tag{1.7}$$

We can find be now by plugging this into either of the equations from line (1.6). For example, plugging into the first gives

$$2 \equiv 11 \cdot 4 + b \pmod{26}$$

which simplifies to

$$2 \equiv 18 + b \pmod{26},$$

giving us

$$b \equiv 10 \pmod{26}, \tag{1.8}$$

We have found the key $(11, 10)$. It is still possible (especially since the message was rather short) that we got unlucky with frequency analysis, so we don't know that this key is actually correct until we've actually tried decrypting the message.

To decrypt the message, we need to find the decryption congruence. The encryption congruence is

$$y \equiv 11x + 10 \pmod{26}.$$

Solving this congruence for x gives the decryption congruence:

$$x \equiv 19y + 18 \pmod{26}.$$

And now we can try decryption the beginning of the message:

	M	C	C	L	L	I	M	I	P	P	...
y	12	2	2	11	11	8	12	8	15	15	...
$x \equiv 19y + 18 \pmod{26}$	12	4	4	19	19	14	12	14	17	17	...
	M	E	E	T	T	O	M	O	R	R	...

And the decryption works out, verifying our frequency analysis guesses. The whole message will decrypt to

```
MEETT OMORR OWATF IVECO MEALO NEIMP ORTAN
TDOCU MENTS MUSTB EEXCH ANGED
```

When solving systems of congruences, the number of solutions can sometimes be greater than 1 (although still often small). Consider, for example, the situation where we have intercepted the message

```
B FNPKK D CDI
```

encrypted with the affine cipher. The original word spacing is still intact, thus it seems natural to guess, for example, that B corresponds to the plaintext letter I and D corresponds to the plaintext letter A. These guesses lead to the system

$$\begin{cases} 1 \equiv 8a + b \pmod{26} \\ 3 \equiv 0a + b \pmod{26} \end{cases}, \tag{1.9}$$

which, upon subtracting, give the congruence

$$24 \equiv 8a \pmod{26}. \tag{1.10}$$

Unlike in the previous example, however, the coefficient of a here does not have an inverse modulo 26. And in fact, examining Table 1.2 shows that $8 \cdot 3 \equiv 24 \pmod{26}$ and $8 \cdot 16 \equiv 24 \pmod{26}$ are both true congruences, thus we need to consider both $a \equiv 3$ and $a \equiv 16$ as possible solutions. Fortunately, in this case, we can immediately rule out the solution $a \equiv 16$, since a must be relatively prime to 26 for the affine cipher to work. Plugging $a \equiv 3$ back into one of the original congruences to solve for b gives $b \equiv 3$, and at this point, the decryption formula can be found and used as in the previous example.

Ex. 1.4.7. Decrypt the message B FNPKK D CDI, encrypted with the affine cipher using the key $(3, 3)$.

Ex. 1.4.8. Solve the following systems of congruences, or state that there is no solution. Be sure to state if there are multiple solutions.

(a) $\begin{cases} 6 \equiv 13a + b \pmod{26} \\ 13 \equiv 4a + b \pmod{26} \end{cases}$

(b) $\begin{cases} 14 \equiv 17a + b \pmod{26} \\ 8 \equiv 7a + b \pmod{26} \end{cases}$

(c) $\begin{cases} 1 \equiv 15a + b \pmod{26} \\ 10 \equiv 9a + b \pmod{26} \end{cases}$

Ex. 1.4.9. Decrypt the message

```
ZVUKE OGDGI HQZIL EUQQV GIFLT UZGLE HUCZZ VUOEX LAEZV KREUA ZGDGH
OEXMZ HIUKX LQGIX LNILM UOUXZ QKTGI ZAVKZ URUHC GOUQT UDGHU EZ
```

encrypted using the affine cipher with an unknown key. A letter count shows U, Z, and G are the most common letters in this ciphertext, occurring 14, 12, and 10 times, respectively.

Note that breaking the affine cipher was significantly more of a nuisance than breaking Caesar's cipher: apart from having to solve a system of congruences, we had to make *two* correct guesses from frequency analysis to come up with the correct key. Nevertheless, it still seems to be a weakness that discovering the correct decryption of two letters is enough to break the whole cipher. The substitution cipher, covered in the next section, requires substantially more guesswork to break.

1.5 The Substitution Cipher

The Caesar, multiplication, and affine ciphers all have something in common: all three ciphers use the same rules for encoding a letter regardless of its position in the message: for example, if an E in one part of the plaintext gets encrypted to the letter O, than *all* E's in the plaintext will get encrypted to the letter O. For this reason, these three ciphers are all just special cases of the substitution cipher, which works by specifying an arbitrary substitution for letters in the alphabet. For example, under the following specified substitution:

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

the message

```
WHENW ILLYO URETU RN
```

would be encrypted to

```
EJFOE MBBKN PWFYP WO
```

Decryption works by reading the substitution table in reverse.

Ex. 1.5.1. Decrypt the message YNTFN WONYY NTF, which was encrypted using the above substitution table.

A key for the substitution cipher consists of a table like the one given above. There are obviously lots of such tables, and the substitution cipher has *far* more possible keys than the Caesar or affine ciphers.

Ex. 1.5.2. How many possible keys are there for the substitution cipher? Keep in mind that a letter can't appear more than once in the bottom of the table, otherwise the substitution can't be reversed. You don't need to give the answer as a number, you can leave it as an expression involving some numbers and operations.

The number of possible keys is so great, in fact, that it is practically impossible to break the cipher just by guessing keys. This is not the case for the Caesar cipher of Affine cipher; in those cases, there are few enough keys that even by just by hand it would be possible (though possibly very tedious) to break the cipher just by trying decryption with all possible keys. With the substitution cipher, the number of possible keys is so great that, even using a modern desktop computer, this could take on the order of billions of years. This might lead one to conclude that the substitution cipher is very secure.

In fact, it is actually relatively straightforward to break the substitution cipher—even by hand—so long as the ciphertext is long enough, although this involves a fair amount of guesswork. Consider, for example, the following ciphertext:

```
GAYRI NGQKI CYHHY HCBLC IBOIZ VBYZI ELPQY BBYHC KVTIZ QYQBI ZLHBT

IKGHU GHELP TGOYH CHLBT YHCBL ELLHR ILZBN YRIQT ITGEJ IIJIE YHBLB

TIKLL UTIZQ YQBIZ NGQZI GEYHC KSBYB TGEHL JYRBS ZIQLZ RLHOI ZQGBY

LHQYH YBGHE NTGBY QBTIS QILPG KLLUB TLSCT BGAYR INYBT LSBJY RBSZI

QLZRL HOIZQ GBYLH
```

We want to apply frequency analysis. Counting letters indicates that the most common letters are B, I, L, Y, and H, occurring 26, 25, 24, 23, and 20 times, respectively. It is reasonable to assume that the plaintext letters T and E correspond to some of these most common letters.

If we assume that E was encrypted to B and T was encrypted to I, we can make the following substitutions:

```
GAYRI NGQKI CYHHY HCBLC IBOIZ VBYZI ELPQY BBYHC KVTIZ QYQBI ZLHBT
    t     t         e   te t   e  t       ee       t     et    e
IKGHU GHELP TGOYH CHLBT YHCBL ELLHR ILZBN YRIQT ITGEJ IIJIE YHBLB
t                    e     e        t  e    t   t     tt t    e e
TIKLL UTIZQ YQBIZ NGQZI GEYHC KSBYB TGEHL JYRBS ZIQLZ RLHOI ZQGBY
 t      t     et      t         e e          e   t        t    e
LHQYH YBGHE NTGBY QBTIS QILPG KLLUB TLSCT BGAYR INYBT LSBJY RBSZI
       e       e   e t   t        e       e     e  e    e    e  e
QLZRL HOIZQ GBYLH
        t    e
```

There is something strange about this substitution, however: nowhere does the pattern T_E appear, which would mean that the word "the" never appears in the passage. While this is possible, it seems perhaps more likely that the substitution should be the other way around. Switching T and E gives the following:

```
GAYRI NGQKI CYHHY HCBLC IBOIZ VBYZI ELPQY BBYHC KVTIZ QYQBI ZLHBT
```

```
    e     e         t   et e   t  e       tt       e     te    t
IKGHU GHELP TGOYH CHLBT YHCBL ELLHR ILZBN YRIQT ITGEJ IIJIE YHBLB
e                    t     t        e  t    e   e     ee e    t t
TIKLL UTIZQ YQBIZ NGQZI GEYHC KSBYB TGEHL JYRBS ZIQLZ RLHOI ZQGBY
 e      e     te      e         t t          t   e        e     t
LHQYH YBGHE NTGBY QBTIS QILPG KLLUB TLSCT BGAYR INYBT LSBJY RBSZI
       t       t   t e   e        t       t     e  t    t    t  e
QLZRL HOIZQ GBYLH
        e    t
```

There are now four instances of the pattern T_E: in the fifth block on the first line (ciphertext BOI), straddling the last block of the first line and the first block of the second line (ciphertext BTI, straddling the last block of the second line and the first block of the third line (ciphertext BTI, and in the fourth block of the fourth line (ciphertext BTI). Based on these occurrences, it seems reasonable to assume that T in the ciphertext corresponds to H in the plaintext and that the first instance BOI was just a coincidence. Filling in this substitution, we get:

```
GAYRI NGQKI CYHHY HCBLC IBOIZ VBYZI ELPQY BBYHC KVTIZ QYQBI ZLHBT
    e     e         t   et e   t  e       tt      he     te    th
IKGHU GHELP TGOYH CHLBT YHCBL ELLHR ILZBN YRIQT ITGEJ IIJIE YHBLB
e           h        th    t        e  t    e h eh    ee e    t t
TIKLL UTIZQ YQBIZ NGQZI GEYHC KSBYB TGEHL JYRBS ZIQLZ RLHOI ZQGBY
he     he     te      e         t t h        t   e        e     t
LHQYH YBGHE NTGBY QBTIS QILPG KLLUB TLSCT BGAYR INYBT LSBJY RBSZI
       t     h t   the   e        t h   h t     e  th   t    t  e
QLZRL HOIZQ GBYLH
        e    t
```

As we would expect, our substitutions have produced several instances of TH in the plaintext, suggesting that we are on the right track. Continuing now with frequency analysis, the most common ciphertext letters we have not yet assigned a substitution for are L, Y, and H. Referring to Table 1.1, the most common English letters after *e* and *t* are *a*, *o*, and *i*. Notice however, that the pattern LL occurs three times in the ciphertext: of the letters *a*, *o*, and *i*, only *o* appears commonly as a double letter in English, so it is natural to assume that L was substituted for O:

```
GAYRI NGQKI CYHHY HCBLC IBOIZ VBYZI ELPQY BBYHC KVTIZ QYQBI ZLHBT
    e     e         to  et e   t  e  o    tt      he     te  o th
IKGHU GHELP TGOYH CHLBT YHCBL ELLHR ILZBN YRIQT ITGEJ IIJIE YHBLB
e        o  h       oth    to  oo   eo t    e h eh    ee e    tot
TIKLL UTIZQ YQBIZ NGQZI GEYHC KSBYB TGEHL JYRBS ZIQLZ RLHOI ZQGBY
he oo  he     te      e         t t h   o    t   e o   o  e     t
LHQYH YBGHE NTGBY QBTIS QILPG KLLUB TLSCT BGAYR INYBT LSBJY RBSZI
o      t     h t   the   eo    oo t ho  h t     e  th o t    t  e
QLZRL HOIZQ GBYLH
 o  o   e    t o
```

We can also try frequency analysis on blocks of letters. For example, the three letter block YHC occurs a 5 times in the ciphertext, more than any other triple. The most common English "trigrams" are *the*, *and*, and *ing*. Since our guesses so far rule out the *the*, it is natural to make the substitutions Y→A, H→N, and C→D:

```
GAYRI NGQKI CYHHY HCBLC IBOIZ VBYZI ELPQY BBYHC KVTIZ QYQBI ZLHBT
  a e     e danna ndtod et e   ta e  o  a ttand   he   a te  onth
IKGHU GHELP TGOYH CHLBT YHCBL ELLHR ILZBN YRIQT ITGEJ IIJIE YHBLB
e  n   n o  h  an dnoth andto  oon  eo t  a e h eh    ee e  antot
TIKLL UTIZQ YQBIZ NGQZI GEYHC KSBYB TGEHL JYRBS ZIQLZ RLHOI ZQGBY
he oo  he   a te      e   and   tat h  no    t   e o   o  e     t
LHQYH YBGHE NTGBY QBTIS QILPG KLLUB TLSCT BGAYR INYBT LSBJY RBSZI
on an at n   h ta  the   eo    oo t ho  h t     e  th o t    t  e
QLZRL HOIZQ GBYLH
 o  o n e    tyon
```

Unfortunately, there are some things to indicate that this last set of substitutions may have been incorrect. For example, in the first line we now have have the blocks EDANNANDTODET and NOTHANDTO in the plaintext, on the first and second lines respectively. Both of these blocks would seem more reasonable if A and D were replaced with I and G, respectively, suggesting that perhaps the ciphertext triple YHC corresponded with the trigram ING after all. Making these changes gives us:

```
GAYRI NGQKI CYHHY HCBLC IBOIZ VBYZI ELPQY BBYHC KVTIZ QYQBI ZLHBT
  i e     e  inni ngtog et e   ti e  o  i tting   he   i te  onth
IKGHU GHELP TGOYH CHLBT YHCBL ELLHR ILZBN YRIQT ITGEJ IIJIE YHBLB
e  n   n o  h  in gnoth ingto  oon  eo t  i e h eh    ee e  intot
TIKLL UTIZQ YQBIZ NGQZI GEYHC KSBYB TGEHL JYRBS ZIQLZ RLHOI ZQGBY
he oo  he   i te      e   ing   tit h  no    t   e o   o  e     t
LHQYH YBGHE NTGBY QBTIS QILPG KLLUB TLSCT BGAYR INYBT LSBJY RBSZI
on in it n   h ti  the   eo    oo t ho  h t     e  th o t    t  e
QLZRL HOIZQ GBYLH
 o  o n e    tyon
```

and now those troublesome blocks have become EGINNINGTOGET and NOTHINGTO. At this point, we basically playing hangman. For example, _EGINNINGTOGET seems like it could be BEGINNINGTOGET, suggesting the substitution K→B, while NOTHINGTO_O could be NOTHINGTODO, suggesting the substitution E→D. Making these substitutions gives us:

```
GAYRI NGQKI CYHHY HCBLC IBOIZ VBYZI ELPQY BBYHC KVTIZ QYQBI ZLHBT
  i e    be ginni ngtog et e   ti e do  i tting b he   i te  onth
IKGHU GHELP TGOYH CHLBT YHCBL ELLHR ILZBN YRIQT ITGEJ IIJIE YHBLB
eb n   ndo  h  in gnoth ingto doon  eo t  i e h eh d  ee ed intot
TIKLL UTIZQ YQBIZ NGQZI GEYHC KSBYB TGEHL JYRBS ZIQLZ RLHOI ZQGBY
heboo  he   i te      e  ding b tit h  no    t i e o   o  e     t
```

```
LHQYH YBGHE NTGBY QBTIS QILPG KLLUB TLSCT BGAYR INYBT LSBJY RBSZI
on in it nd  h ti  the   eo   boo t ho  h t     e  th o t    t  e
QLZRL HOIZQ GBYLH
 o  o n e    tyon
```

Spanning the end of the second line and beginning of the third, the plaintext block `INTOTHEBOO_` suggests the substitution `U`→`K`. In the third line, we have the plaintext `INGB_TIT`. The `ING` almost certainly represents the end of a word. It seems clear that the blank must be a vowel, and `U` seems the most likely candidate. The substitutions `U`→`K` and `S`→`U` give us:

```
GAYRI NGQKI CYHHY HCBLC IBOIZ VBYZI ELPQY BBYHC KVTIZ QYQBI ZLHBT
  i e    be ginni ngtog et e   ti e do  i tting b he   i te  onth
IKGHU GHELP TGOYH CHLBT YHCBL ELLHR ILZBN YRIQT ITGEJ IIJIE YHBLB
eb nk  ndo  h  in gnoth ingto doon  eo t  i e h eh d  ee ed intot
TIKLL UTIZQ YQBIZ NGQZI GEYHC KSBYB TGEHL JYRBS ZIQLZ RLHOI ZQGBY
heboo khe   i te      e  ding butit h  no    tu  e o   o  e     t
LHQYH YBGHE NTGBY QBTIS QILPG KLLUB TLSCT BGAYR INYBT LSBJY RBSZI
on in it nd  h ti  theu  eo   bookt ho  h t     e  th o t    t  e
QLZRL HOIZQ GBYLH
 o  o n e    tyon
```

On the first line, `TI_EDO__ITTING` becomes `TIREDOFSITTING` under the substitutions `Z`→`R`, `P`→`F`, and `Q`→`S`. On the second line, `ON_EO_T_I_E` becomes `ONCEORTWICE` under the substitutions `R`→`C`, `Z`→`R`, and `N`→`W`. These five substitutions bring us to

```
GAYRI NGQKI CYHHY HCBLC IBOIZ VBYZI ELPQY BBYHC KVTIZ QYQBI ZLHBT
  ice   sbe ginni ngtog et er  tire dofsi tting b her siste ronth
IKGHU GHELP TGOYH CHLBT YHCBL ELLHR ILZBN YRIQT ITGEJ IIJIE YHBLB
eb nk  ndof h  in gnoth ingto doonc eortw icesh eh d  ee ed intot
TIKLL UTIZQ YQBIZ NGQZI GEYHC KSBYB TGEHL JYRBS ZIQLZ RLHOI ZQGBY
heboo khers ister   sre  ding butit h  no   rtu resor co  e  rs t
LHQYH YBGHE NTGBY QBTIS QILPG KLLUB TLSCT BGAYR INYBT LSBJY RBSZI
onsin it nd  h ti stheu seof  bookt ho  h t   c e  th o t   ct re
QLZRL HOIZQ GBYLH
sorco n ers  tyon
```

At this point, it's not too hard to figure out the rest. The plaintext is:

```
ALICE WASBE GINNI NGTOG ETVER YTIRE DOFSI TTING BYHER SISTE RONTH
EBANK ANDOF HAVIN GNOTH INGTO DOONC EORTW ICESH EHADP EEPED INTOT
HEBOO KHERS ISTER WASRE ADING BUTIT HADNO PICTU RESOR CONVE RSATI
ONSIN ITAND WHATI STHEU SEOFA BOOKT HOUGH TALIC EWITH OUTPI CTURE
SORCO NVERS ATION
```

There is no doubt that applying frequency analysis to the substitution cipher in this way can be tedious. Unlike the Caesar and affine ciphers, it is not enough

to figure out just one or two of the substitutions; each one must be determined separately. But the fact that it is possible at all, with a cipher that has such a large number of possible keys, indicates just how powerful frequency analysis is. Messages are not random jumbles of letters, and frequency analysis allows the cryptographer to take advantage of that fact to break codes.

Ex. 1.5.3. Break the following substitution cipher. This is made substantially easier by the fact that the original word spacing is intact.

```
LKZB RMLK X JFAKFDEQ AOBXOV TEFIB F MLKABOBA TBXH XKA TBXOV LSBO

JXKV X NRXFKQ XKA ZROFLRP SLIRJB LC CLODLQQBK ILOB TEFIB F KLAABA

KBXOIV KXMMFKD PRAABKIV QEBOB ZXJB X QXMMFKD XP LC PLJB LKB

DBKQIV OXMMFKD OXMMFKD XQ JV ZEXJYBO ALLO Q FP PLJB SFPFQBO F

JRQQBOBA QXMMFKD XQ JV ZEXJYBO ALLO LKIV QEFP XKA KLQEFKD JLOB
```

Letter count: A: 15, B: 28, C: 3, D: 9, E: 8, F: 19, G: 0, H: 1, I: 8, J: 12, K: 24, L: 22, M: 12, N: 1, O: 19, P: 8, Q: 16, R: 7, S: 3, T: 4, U: 0, V: 9, W: 0, X: 23, Y: 2, Z: 5

1.6 The Permutation Cipher

Substitution ciphers (including the Caesar and Affine ciphers) essentially work by relabeling the letters of the alphabet to disguise the original message. Frequency analysis can be used to figure out the plaintext by discovering how the alphabet was 'relabeled'. The permutation cipher, on the other hand, does not change the letters *per se*, but just moves them to different positions. For example, consider the message

```
MEETA TTENT HIRTY
```

We can break the message into blocks of three letters each:

```
MEE TAT TEN THI RTY
```

and then 'rotate' the letters in each block to the right (moving the right-most letter in each block to the first position):

```
EME TTA NTE ITH YRT
```

and then regroup the letters into blocks of 5 to get a ciphertext to be transmitted:

```
EMETT ANTEI THYRT
```

To decipher the message, the recipient would break the message back into blocks of three, and reverse the permutation of the letters by rotating the letters in each block to the left (moving the left-most letter in each block to the last position).

In this example, encryption was done by rotation in blocks of 3, but the permutation cipher can work on blocks of arbitrary size. In general, the key to the permutation cipher is a *permutation*. For example,

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

is the permutation which rotates three objects cyclically 'to the right': objects in the order (1 2 3) are permuted so that they are in the order (3 1 2): each element has been moved to the right, and the last element has "wrapped around" to the first position. In the case of the plaintext block `MEE`, applying this permutation resulted in the ciphertext block `EME`; `TAT` was transformed into `TTA`, while `ELE` became `EEL` and `VEN` became `NVE`. The key to the permutation cipher is *any* permutation. For example, the permutation

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 4 & 2 & 1 \end{pmatrix} \tag{1.11}$$

acts on blocks of 5 objects. To use it to encipher the message

`MEETA TTENT HIRTY`

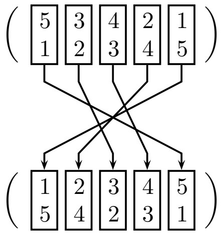
we simply 'apply' the permutation to each of the blocks (since it is already grouped into blocks of the right size). The permutation in line (1.11) specifies that the 5th letter will be moved to the first position, the 3rd letter will be in the second position, and 4th letter will be in the 3rd position, the second letter will be in the fourth position, and the first letter will be in the 5th position. Applying the permutation to the block `MEETA`, then, would give `AETEM`. The entire ciphertext would be:

`AETEM TENTT YRTIH`

To decipher the message, we need to find the permutation which "reverses" the permutation from line (1.11). This is called the *inverse* of the permutation. This would be a permutation that takes objects in the order (5 3 4 2 1) and puts them in the order (1 2 3 4 5). To find the permutation, we first write this first ordering of the objects over the second one:

$$\begin{pmatrix} 5 & 3 & 4 & 2 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix} \tag{1.12}$$

Now we rearrange columns so that the first row is in the standard increasing order:



Thus the message

`AETEM TENTT YRTIH`

can be decrypted by applying the inverse permutation

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 2 & 3 & 1 \end{pmatrix}$$

to each of the blocks of five.

There is one issue we haven't discussed yet: what if the message cannot be broken into blocks of the appropriate size? Imagine, for example, that we want to encrypt the message

`MEETA TTEN`

with the permutation

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}.$$

After grouping into blocks of 4:

`MEET ATTE N`

there is a leftover letter, since the message length was not a multiple of 4. In this case, we can **pad** the message by adding extra nonsense letters to the end:

`MEET ATTE NCTH`

This encrypts to

`ETEM TETA THCN`

or, after regrouping,

`ETEMT ETATH CN`

When the recipient decrypts the message, they will simply discard any nonsense at the end that was added for padding.

Ex. 1.6.1. The message `XIMTI LLAPU` was encrypted with the permutation cipher with key $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 1 & 5 & 4 \end{pmatrix}$. Decrypt it.

Ex. 1.6.2. Encrypt the message PERMU TATIO N with the permutation

$$\begin{pmatrix} 3 & 4 & 2 & 1 \\ 3 & 4 & 2 & 1 \end{pmatrix}$$

How can an eavesdropper break the permutation cipher without knowing the key? Note that, even if we intercepted a very long message, frequency analysis on letters wouldn't be helpful to break the permutation cipher; in fact, *the frequency of letters in the ciphertext is the same as the frequency of letters in the plaintext*. Thus frequency analysis of letters will typically just reveal that that E and T are common letters in the ciphertext, and that Z is uncommon, *etc.*. This might be useful in confirming that the original message is English text, for example, but won't give us any information on the permutation used to encode the message.

On the other hand, knowledge of common pairs and triples of letters in English can be very useful in breaking the permutation cipher. Consider the following message, encrypted with the permutation cipher:

```
RIBNT HGEES MSGEA TTHOE RODPO IPNRL TH
```

The ciphertext is 32 characters long; this already gives us important information, since the length of the permutation must divide the length of the message. In this case, the divisors of 32 are 1,2,4,8,16,32. Let's guess that the permutation has length 4, in which case the cipher works by permuting blocks of this length:

```
RIBN THGE ESMS GEAT THOE RODP OIPN RLTH
```

Now we try to find how to permute the letters in the blocks to give rise to English text. Notice, for example, that two of the blocks contain the pattern TH_E. It seems likely that this pattern arose from occurrences of the word *the* in the plaintext. If this is the case, it tells us that that the decryption permutation maps the 1st, 2nd, and 4th letters into consecutive positions; there are two permutations with this property, namely

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix} \tag{1.13}$$

and

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 2 & 4 \end{pmatrix}. \tag{1.14}$$

Under the first of these permutations, the first few blocks of the message decrypt to

```
RINB THEG ESSM GETA ...
```

which appears to be nonsense. Decrypting with the second permutation, however, gives

```
BRIN GTHE MESS AGET OTHE DROP POIN TRLH
```

so the message was "bring the message to the drop point", padded with RLH to bring the plaintext to a multiple of 4.

Ex. 1.6.3. Decrypt the message HTESE RCHTE SAEEB PNMRE TUTDE, encrypted with an unknown permutation of length 5.

1.7 The Vigenére cipher

Consider a message to be encrypted, for example:

> There is no possibility of thinking of anything at all in the world, or even out of it, which can be regarded as good without qualification, except a good will. Intelligence, wit, judgment, and whatever talents of the mind one might want to name are doubtless in many respects good and desirable, as are such qualities of temperament as courage, resolution, perseverance. But they can also become extremely bad and harmful if the will, which is to make use of these gifts of nature and which in its special constitution is called character, is not good.

To encrypt this message with the Caesar cipher, we would shift each letter by some fixed amount. This encryption could be easily broken, since it is easily subjected to frequency analysis, and correctly figuring out one letter is enough to break the entire message. (On top of all of this, the Caesar cipher admits only 26 distinct keys—25 not counting the key which does nothing—so all the keys could be tried in the worst case.)

Straightforward frequency analysis can even be used to break a general substitution cipher, as we saw in Section 1.5; since letters are encrypted the same way throughout an entire message (*i.e.,* with the same substitution table), the decryption for each letter could be determined one at a time using frequency and other clues.

The Vigenére cipher, first invented in 1553 by Giovan Battista Bellaso and subsequently rediscovered by Blaise de Vigenére in 1586, addresses these issues by shifting letters at different places in the message by different amounts. Suppose, for example, we have agreed on a keyword MORALS. To encrypt the above message, we first write the keyword, repeated, underneath the plaintext, and then 'add' corresponding plaintext and key letters. For example, T corresponds to the number 19, while M corresponds to the number 12. Thus the first ciphertext letter will correspond to $5 \equiv 19 + 12 \pmod{26}$, giving the letter F. Similarly, the second letter will be given by $7 + 14$, which gives the letter V. The rest of the encryption is shown below:

```
THERE ISNOP OSSIB ILITY OFTHI NKING OFANY THING ATALL INTHE WORLD
MORAL SMORA LSMOR ALSMO RALSM ORALS MORAL SMORA LSMOR ALSMO RALSM
FVVRP AEBFP ZKEWS IWAFM FFEZU BBIYY ATRNJ LTWEG LLMZC IYLTS NOCDP

OREVE NOUTO FITWH ICHCA NBERE GARDE DASGO ODWIT HOUTQ UALIF ICATI
```

```
ORALS MORAL SMORA LSMOR ALSMO RALSM ORALS MORAL SMORA LSMOR ALSMO
CIEGW ZCLTZ XUHNH TUTQR NMWDS XACVQ RRSRG ARNIE ZAIKQ FSXWW INSFW

ONEXC EPTAG OODWI LLINT ELLIG ENCEW ITJUD GMENT ANDWH ATEVE RTALE
RALSM ORALS MORAL SMORA LSMOR ALSMO RALSM ORALS MORAL SMORA LSMOR
FNPPO SGTLY ACUWT DXWET PDXWX EYUQK ZTUMP UDEYL MBUWS SFSME CLMZV

NTSOF THEMI NDONE MIGHT WANTT ONAME AREDO UBTLE SSINM ANYRE SPECT
ALSMO RALSM ORALS MORAL SMORA LSMOR ALSMO RALSM ORALS MORAL SMORA
NEKAT KHPEU BUOYW YWXHE OMBKT ZFMAV ACWPC LBEDQ GJIYE MBPRP KBSTT

SGOOD ANDDE SIRAB LEASA RESUC HQUAL ITIES OFTEM PERAM ENTAS COURA
LSMOR ALSMO RALSM ORALS MORAL SMORA LSMOR ALSMO RALSM ORALS MORAL
DYACU AYVPS JICSN ZVADS DSJUN ZCIRL TLUSJ OQLQA GECSY SETLK OCLRL

GERES OLUTI ONPER SEVER ANCEB UTTHE YCANA LSOBE COMEE XTREM ELYBA
SMORA LSMOR ALSMO RALSM ORALS MORAL SMORA LSMOR ALSMO RALSM ORALS
YQFVS ZDGHZ OYHQF JEGWD OECPT GHKHP QOOEA WKAPV CZEQS OTCWY SCYMS

DANDH ARMFU LIFTH EWILL WHICH ISTOM AKEUS EOFTH ESEGI FTSOF NATUR
MORAL SMORA LSMOR ALSMO RALSM ORALS MORAL SMORA LSMOR ALSMO RALSM
POEDS SDAWU WARHY EHAXZ NHTUT WJTZE MYVUD WATKH PKQUZ FEKAT EAEMD

EANDW HICHI NITSS PECIA LCONS TITUT IONIS CALLE DCHAR ACTER ISNOT
ORALS MORAL SMORA LSMOR ALSMO RALSM ORALS MORAL SMORA LSMOR ALSMO
SRNOO TWTHT FUHJS AWOWR LNGZG KIEMF WFNTK OOCLP VOVRR LUFSI IDFAH

GOOD
XOZV
RALS
```

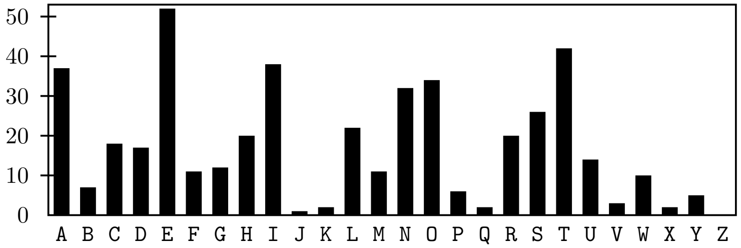
To decrypt the message, the recipient would write the keyword repeated under the ciphertext and subtract.

Ex. 1.7.1. Encrypt the message `FOLLO WTHEY ELLOW BRICK ROAD` with they keyword `OZ`.

Ex. 1.7.2. Decrypt the message `LOSVW AZBSH DHQID ARSLG EL`, encrypted with the Vigenére cipher using `SHOES` as a key.

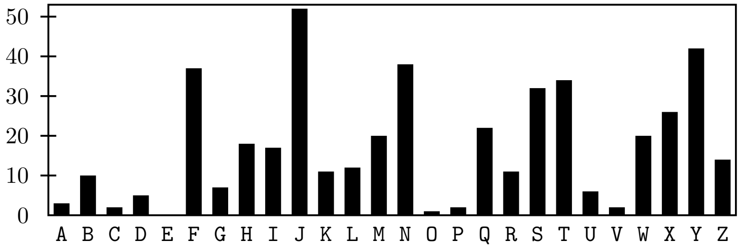
1.7.1 Analysis of the Vigenére cipher

The Vigenére cipher is designed to be resistent to frequency analysis. Consider the following graph of the frequencies of letters in the original plaintext from above:

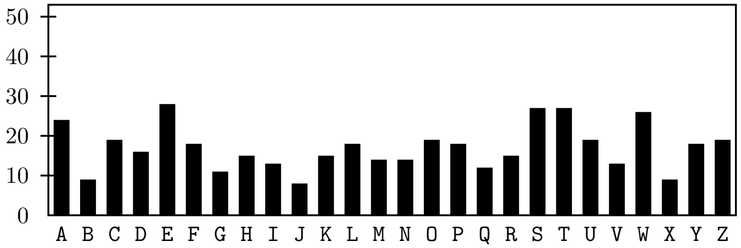


We can see that the plaintext has the typical signatures of English text: common *e*'s and *t*'s, uncommon *z*'s, *etc.*

If we had encrypted the message with the Caesar cipher—or even with any substitution cipher—then the same frequencies would occur in the distribution, but for different letters. For example, here is the distribution of the message after encryption by the Caesar cipher with a key of 5:



And frequency analysis would be possible because we can guess now that that common letters in the ciphertext correspond to common English letters (*e*, *t*, *etc.*). In contrast, the following graph shows the distribution of letters in the ciphertext found above by encrypting the plaintext with the Vigenére cipher (with 'MORALS' as the key).



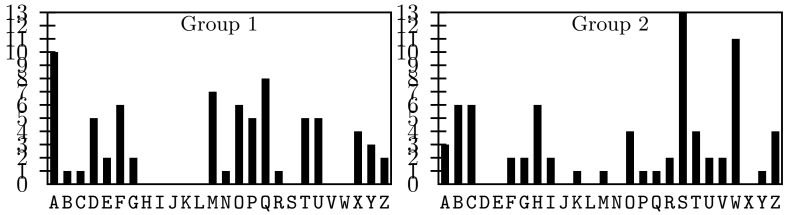
This distribution is quite different than the frequency distribution of the plaintext—it has been 'smoothed out' by the process of encryption. And the resulting distribution is much less useful to someone trying to break the code. For example, given a common letter like `W` in the ciphertext, there does not appear to be a simple way to decide whether it is common because one of its corresponding plaintext letters was 'very common', or because several of them were 'fairly common'. In short, there is no clear way for an eavesdropper to use this frequency distribution to make guesses about the key used for encryption. This feature of the Vigenére cipher makes it seem like it may be impossible to break (and the cipher was actually known for a time as 'the unbreakable cipher'), and no practical attack on the cipher was known until 300 years after its introduction.

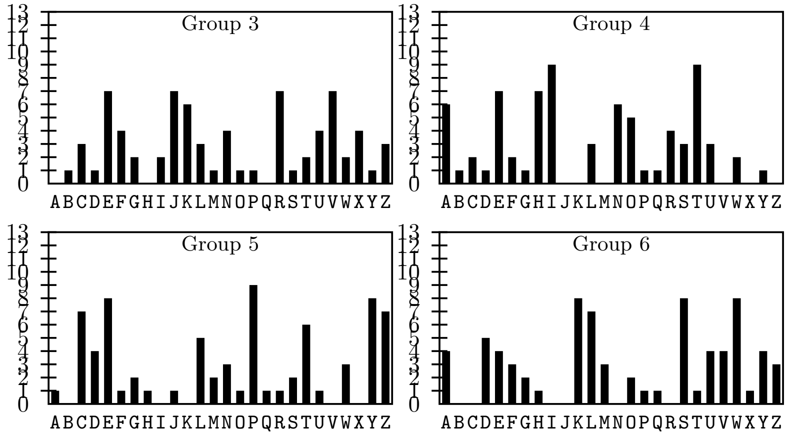
Imagine, however, that an eavesdropper somehow knows the length of the keyword that was used—in the case of the current example, that length is 6. The eavesdropper can then break the message into 6 groups which each consist of letters which were all shifted by the same amount in the encryption process. For example, for the current ciphertext

```
FVVRP AEBFP ZKEWS IWAFM FFEZU BBIYY ATRNJ LTWEG LLMZC IYLTS NOCDP
CIEGW ZCLTZ XUHNH TUTQR NMWDS XACVQ RRSRG ARNIE ZAIKQ FSXWW INSFW
FNPPO SGTLY ACUWT DXWET PDXWX EYUQK ZTUMP UDEYL MBUWS SFSME CLMZV
NEKAT KHPEU BUOYW YWXHE OMBKT ZFMAV ACWPC LBEDQ GJIYE MBPRP KBSTT
DYACU AYVPS JICSN ZVADS DSJUN ZCIRL TLUSJ OQLQA GECSY SETLK OCLRL
YQFVS ZDGHZ OYHQF JEGWD OECPT GHKHP QOOEA WKAPV CZEQS OTCWY SCYMS
POEDS SDAWU WARHY EHAXZ NHTUT WJTZE MYVUD WATKH PKQUZ FEKAT EAEMD
SRNOO TWTHT FUHJS AWOWR LNGZG KIEMF WFNTK OOCLP VOVRR LUFSI IDFAH
XOZV
```

the first group of letters would be F, E, E, F, *etc.* (every 6th letter starting with the first one). The second group would be V, B, W, M, *etc.* (every 6th letter starting with the second), and so on. The letters in each of these groups were encrypted the same way, since the keyword lines up the same way for each of them (all letters in the first group were encrypted by adding the letter M, *etc.*). The important thing to notice is that this means that frequency analysis should work on each group of letters when the groups are considered *separately*! Even though the all the letters of the message were not encrypted in a consistent way, resulting in a frequency distribution that it is not useful for breaking the code, each of the groups of letters was encrypted by a simple shift, and each has a frequency distribution revealing information about that shift.

Shown below are the frequency distributions for each of these 6 groups:





Each of these distributions confers information on the shift used for the respective groups of letters. For example, the fact that Q is common in the distribution for Group 1 corresponds to the fact that E is common in the plaintext and corresponds to the letter Q under a shift by 12 (which corresponds to M, the first letter of the keyword). Similarly, S is common in the distribution for group 2 because it is the shift of E by 14 (corresponding to O, the second letter of the keyword); V is common in group 3 because it is the shift of E by 17 (R); E and T are common in group 4 because it was not shifted at all (a shift of 0 corresponds to A in the keyword); the common letter P in group 5 corresponds E under a shift of 11 (L); and the common letter W in group 6 corresponds to E under a shift of 18 (S).

Ex. 1.7.3. Explain in your own words how the Vigenére cipher can be broken if they keyword length is known.

All of this analysis, however, is predicated on the assumption that the eavesdropper can somehow figure out the length of the keyword. Of course, an eavesdropper with enough time could simply try lots of possible keyword lengths, until one worked out. There is, however, a much better way of efficiently determining the keyword length, called the Kasiski test, named after Friedrich Kasiski who published the first attack on the Vigenére cipher in 1863[1].

1.7.2 The Kasiski test

The Kasiski test works to determine the length of the keyword used for encryption with the Vigenére cipher by taking advantage of repetitions in the ciphertext. For example, an observant cryptanalysist might notice that the strings ATKHP and NHTUT both appear twice in the ciphertext:

[1] Charles Babbage had independently developed the same technique 10 or 15 years earlier, although he never published it.

```
FVVRP AEBFP ZKEWS IWAFM FFEZU BBIYY ATRNJ LTWEG LLMZC IYLTS NOCDP
CIEGW ZCLTZ XUHNH TUTQR NMWDS XACVQ RRSRG ARNIE ZAIKQ FSXWW INSFW
FNPPO SGTLY ACUWT DXWET PDXWX EYUQK ZTUMP UDEYL MBUWS SFSME CLMZV
NEKAT KHPEU BUOYW YWXHE OMBKT ZFMAV ACWPC LBEDQ GJIYE MBPRP KBSTT
DYACU AYVPS JICSN ZVADS DSJUN ZCIRL TLUSJ OQLQA GECSY SETLK OCLRL
YQFVS ZDGHZ OYHQF JEGWD OECPT GHKHP QOOEA WKAPV CZEQS OTCWY SCYMS
POEDS SDAWU WARHY EHAXZ NHTUT WJTZE MYVUD WATKH PKQUZ FEKAT EAEMD
SRNOO TWTHT FUHJS AWOWR LNGZG KIEMF WFNTK OOCLP VOVRR LUFSI IDFAH
XOZV
```

How can it happen that long words are repeated in the ciphertext like this? If we examine the encryption calculation on page 28, we see that these instances of ATKHP and NHTUT correspond to the words OFTHE and WHICH, respectively, which were repeated in the plaintext. These are not the only long strings repeated in the plaintext, however; for example, the string GOODWI occurs both near the end of the second line of the plaintext (as the beginning of "good without qualification") near the beginning of the third line (as the beginning of "good will"). However, the repetition of the string GOODWI in the plaintext does not give rise to a repetition in the ciphertext in this case; the first instance gets encrypted as RGARNI and the second as YACUWT.

The difference between instances of repetition in the plaintext that do give rise to repetition in the ciphertext and those that don't is the distance between the repetitions. Notice, for example, for the two instances of WHICH in the plaintext which were encrypted to NHTUT, the instances lined up with the keyword in the same way (in both cases, WHICH is added to the segment RALSM out of the running keystream ...MORALSMORALSMORALS...). Similarly, the two instances of OFTHE which encrypt to ATKHP both line up with the segment MORAL from the keystream. The important thing to notice is that two instances of a repeated string will line up with the key stream in the same way exactly when *the distance between the first letters of the instances of the repeated string is a multiple of the keyword length.* Thus, for example, we can count that the distance between the start of the two instances of ATKHP in the ciphertext is 198, and between the start of the two instances of NHTUT is 282. (Counting the distance is not hard keeping in mind that there are 5 letters per block and 11 blocks per line, and so 55 letters per line.) Kasiski's test tells us that we expect both of these distances to be a multiple of the keyword length, since we expect that both of these repetitions happened because some repetitions in the plaintext lined up in the same way with the keystream. The only common divisors of 198 and 282 are 1, 2, 3 , and 6, and it makes sense to choose the largest of these as the guess for the keyword length (which is right in this case!)[2] Let's see how to put everything together to break an intercepted message which was encrypted by the Vigenére cipher.

[2] It turns out that even if the keyword length was actually not the greatest common divisor, but one of the other common divisors, everything would still work out even if we tried to do frequency analysis on letter groups with the greatest common divisor as our guessed keyword length. If, for example, the message had been encrypted with the 3-letter keyword KEY, this is the same anyways as having been encrypted with the 6-letter keyword KEYKEY, so our analysis would still work out choosing 6 as the length instead of 3.

Ex. 1.7.4. For such a short passage, the following ciphertext contains many long repeated strings. Use the Kasiski test to determine the length of the Vigenére keyword used to encrypt the following message. (You should find enough repetitions that you can get a distance GCD of 10 or less.)

```
KBPYU BACDM LRQNM GOMLG VETQV PXUQZ LRQNM GOMLG VETQV PXYIM HDYQL
BQUBR YILRJ MTEGW YDQWE GUPGC UABRY ILRJM XNQKA MHJXJ KMYGV ETQVP
XCRWV FQNBL EZXBW TBRAQ MUCAM FGAXY UWGMH TBEJB BRYIL RJMLC CAHLQ
NWYTS GCUAB RYILR JMLNT QGEQN AMRMB RYILR JMPGP BXPQN WCUXT GT
```

1.7.3 Breaking the Vigenére cipher

We want to decode the following intercepted message, which was encrypted with the Vigenére cipher:

```
KTPCZ NOOGH VFBTZ VSBIO VTAGM KRLVA KMXAV USTTP CNLCD VHXEO CPECP
PHXHL NLFCK NYBPS QVXYP VHAKT AOLUH TITPD CSBPA JEAQZ RIMCS YIMJH
RABPP PHBUS KVXTA JAMHL NLCWZ VSAQY VOYDL KNZLH WNWKJ GTAGK QCMQY
UWXTL RUSBS GDUAA JEYCJ VTACA KTPCZ PTJWP VECCB PDBEL KFBVI GCTOL
LANPK KCXVO GYVQB NDMTL CTBVP HIMFP FNMDL EOFGQ CUGFP EETPK YEGVH
YARVO GYVQB NDWKZ EHTTN GHBOI WTMJP UJNUA DEZKU UHHTA QFCCB PDBEL
CLEVO GTBOL EOGHB UEWVO GM
```

The first step is to identify some long repeated strings in the ciphertext so that we can apply the Kasiski test. The strings KTPCZ, HLNL, CCBPDBEL, VOGYVQBND are all repeated, at the positions underlined above.

VOGYVQBND (×2)	50
CCBPDBEL (×2)	120
KTPCZ (×2)	190
HLNL (×2)	70
OGH (×2)	334
ZVS (×2)	120
VTA (×2)	165
TAG (×2)	135
NLC (×2)	89
PPH (×2)	60
AKT (×2)	112
AJE (×2)	85
HBU (×2)	227
VOG (×4)	50, 55, 15
LEO (×2)	85
GHB (×2)	47

There are also many repeated strings of length three. All repetitions of length at least three are shown below on the left, along with their separating distances. Note that *it is not necessary to find all repetitions to apply the Kasiski test*, in fact, typically 3 or 4 repetitions will be plenty, and in some cases 2 repeated pairs may even be enough to give a reasonably small gcd.

In spite of the fact that the Kasiski test tells us that the keyword length should be a divisor of the distances between repeated strings, the distances in the table to the left actually don't have any common divisors bigger than 1! Notice, for example, that 47 is prime and has no divisors other than 1 and itself. This can happen because it is possible for some strings to be repeated just by chance, and not because they correspond to a repeated plaintext word for which the keyword has lined up in a consistent way.

This is particularly true for short repeated strings which are only repeated once. On the other hand, we should be confident that the repetitions of the strings CCBPDBEL, VOGYVQBND, KTPCZ, HLNL, and VOG are not just coincidences, since all of these examples are either longer than three characters, or are repeated several times (in the case of VOG). The greatest common divisor of the distances separating the instances of these sets of repeated strings is 5, so we will guess that 5 is the length of the keyword used for encryption.

Since the Kasiski test gives a keyword length of 5, the next step is to consider the ciphertext as 5 groups of letters, according to how each one lines up with the (as of yet unknown) keyword, and do frequency analysis on each group separately. For example, the first group consists of the letters K,N,V,V,. . . (the first letter from each group of 5), the second consists of the letters T,O,F,. . . (the second from each group of 5), and so on. We just need to count the frequency of letters from each of these groups. This has been done in the table below:

group	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
1	1	0	6	1	4	1	8	1	0	3	8	1	0	6	0	5	3	3	0	1	5	9	2	0	3	0
2	4	0	3	5	7	3	0	7	4	1	0	3	2	4	5	1	0	1	4	9	2	2	1	0	3	0
3	6	12	3	0	2	2	3	1	0	1	0	3	8	2	1	2	0	1	1	5	1	2	3	7	2	2
4	2	1	10	2	3	2	4	3	1	2	4	1	0	0	3	6	5	0	0	7	3	8	2	0	1	0
5	7	5	0	2	0	0	0	5	2	2	4	10	1	1	6	9	1	0	4	1	1	1	0	0	2	6

So, this table means that the ciphertext letter A lines up with the first letter of the keyword 1 time, the second letter of the keyword 4 times, and so on.

We begin with frequency analysis on the first group. The most common letters in this group are V, G, and K, occurring 9, 8, and 8 times, respectively. A first guess would be that V corresponds to the plaintext letter E. If this is the case, than this group was encrypted by a shift of 17. To check if this is reasonable, we can examine what this would mean for the frequency of other plaintext letters. For example, we expect T would be common in the plaintext, and under a shift of 17 this corresponds to the letter K, which occurs 8 times as a ciphertext letter in this group, which seems reasonable. We can also check an uncommon letter: Z would be encrypted to the letter Q under encryption by 17, which occurs 3 times in the first ciphertext group, which is quite often for this letter. While not impossible, this is perhaps enough to suggest that 17 is not the correct shift, prompting us try try some other possibilities. If we instead assume that the plaintext E corresponds to the ciphertext G, this would mean that this group was encrypted by a shift of 2. This seems to check out okay: T would be encrypted to V, which is common (occurring 9 times), Z would be encrypted to B, which doesn't occur at all, A would be encrypted to C, which is relatively common, and so on. Thus it seems we have successfully determined the first letter letter of the keyword: C (the keyword letter which would give a shift of 2).

For the second group, the most common letters are T, E, and H. Actually, these are relatively common letters in English text overall, and a quick glance at group 2's row in the plaintext shows that common English letters are common in this group, while uncommon English letters are uncommon. Thus it seems that this group was not shifted at all, meaning that the second letter of the keyword should be A.

Ex. 1.7.5. Determine the remaining 3 keyword letters, and decrypt the beginning of the ciphertext (at least 20 characters).

Ex. 1.7.6. Use frequency analysis to recover the message from Exercise 1.7.4, whose keyword length you determined in that problem. The ciphertext from that problem was:

```
KBPYU BACDM LRQNM GOMLG VETQV PXUQZ LRQNM GOMLG VETQV PXYIM HDYQL

BQUBR YILRJ MTEGW YDQWE GUPGC UABRY ILRJM XNQKA MHJXJ KMYGV ETQVP

XCRWV FQNBL EZXBW TBRAQ MUCAM FGAXY UWGMH TBEJB BRYIL RJMLC CAHLQ

NWYTS GCUAB RYILR JMLNT QGEQN AMRMB RYILR JMPGP BXPQN WCUXT GT
```

Note that for a passage as short (and unusual) as this one, the most common English letters may not be all that common in the some of the plaintext positions, depending on how our luck goes. In cases like this, it is good to pay close attention to how the *uncommon* English letters line up. The plaintext in this case contains no j's, q's, x's, or z's at all.

1.8 The Hill Cipher

The Hill cipher was invented by Letser Hill in 1929. What distinguishes it from other ciphers we have covered so far is that it encrypts messages in blocks of letters, in such a way that how a particular letter in the message gets encrypted will depend on all the letters in the letter's block, rather than on just the letter itself[3]. To do this, Hill's cipher makes use of matricies and matrix operations. Before describing the cipher, let's quickly review basic matrix arithmetic.

1.8.1 Matrix Review

A *matrix* is just an array of numbers. For example,

$$\begin{pmatrix} 7 & 2.3 & \sqrt{2} \\ -50 & \pi & 0 \end{pmatrix}$$

is a 2×3 matrix. Any matrices of the same dimensions can be added in just the way you would expect. For example:

$$\begin{pmatrix} 7 & 2.3 & \sqrt{2} \\ -50 & \pi & 0 \end{pmatrix} + \begin{pmatrix} 3.7 & -1 & 5 \\ \frac{1}{\sqrt{2}} & 3\pi & 2 \end{pmatrix} = \begin{pmatrix} 10.7 & 1.3 & (5+\sqrt{2}) \\ (-50+\frac{1}{\sqrt{2}}) & 4\pi & 2 \end{pmatrix}$$

[3] This is kind of true for the Vigenére cipher, if we think of the message as broken into blocks of the same length as the keyword. But in that case, the way a letter gets encrypted only depends on its *position* in the block. For the Hill cipher, what the other letters are in blocks wil matter as well.

Matrix multiplication, however, is a bit different, and is what makes matrices different from just listings of numbers. We can multiply one matrix by another one whenever the number of *columns* in the first matrix matches the number of *rows* in the second matrix. The product matrix then has as many *rows* as the first matrix, and as many *columns* as the second matrix. The element of the product matrix in the ith row and jth column is computed as the product of the ith row of the first matrix times the jth column of the second matrix. To see what this means, consider the following example:

$$\begin{pmatrix} 3 & -2 & 11 \\ -5 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 4 \\ 0 & 3 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 3\cdot 1 - 2\cdot 0 + 11\cdot 2 & 3\cdot 4 - 2\cdot 3 + 11\cdot 1 \\ -5\cdot 1 + 1\cdot 0 + 1\cdot 2 & -5\cdot 4 + 1\cdot 3 + 1\cdot 1 \end{pmatrix}$$

$$= \begin{pmatrix} 25 & 17 \\ -3 & -16 \end{pmatrix} \qquad (1.15)$$

Thus, for example, element in the first row and first column of the product matrix is 25, since this is $3 \cdot 1 - 2 \cdot 0 + 11 \cdot 2$, which is the first row of the first matrix $(3, -2, 11)$ times the first column of the second matrix $(1, 0, 2)$. The product of two matrices will have as many rows as the first matrix, and as many columns as the second.

Notice that there is something which might seem rather unusual about matrix multiplication: order matters. For example, we have

$$\begin{pmatrix} 1 & 2 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} 3 & 2 \\ -1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 6 \\ 1 & -2 \end{pmatrix},$$

but if we multiply them the other way, we get

$$\begin{pmatrix} 3 & 2 \\ -1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ -1 & -4 \end{pmatrix}.$$

In fact, in many cases (like line (1.15)) reversing the order gives a problem which is not possible because the dimensions do not line up correctly.

The *identity matrix* is the matrix with 1s on the main diagonal and 0s everywhere else. For example,

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

is the 2×2 identity matrix. Such a matrix is called the identity matrix because multiplying it always gives back the same matrix. For example:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 13 & 5 & 22 \\ -20 & 3 & -30 \\ 15 & 13 & 15 \end{pmatrix} = \begin{pmatrix} 13 & 5 & 22 \\ -20 & 3 & -30 \\ 15 & 13 & 15 \end{pmatrix}.$$

(Notice that this works even if the order of the multiplied matrices was reversed.)

The *inverse* of a matrix is the matrix which gives the identity matrix when multiplied by the original matrix. For example, $\begin{pmatrix} -1 & -3 \\ 1 & 2 \end{pmatrix}$ is the inverse of $\begin{pmatrix} 2 & 3 \\ -1 & -1 \end{pmatrix}$ since

$$\begin{pmatrix} -1 & -3 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 2 & 3 \\ -1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Not all matrices have inverses, however. For example, it's not too hard to check that the matrix

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

can not be multiplied by any matrix to give the identity.

The *determinant* of a 2×2 matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is defined as $ad - bc$. Thus, the determinant of the 2×2 matrix in line (1.15) is $25 \cdot (-16) - 17 \cdot (-3) = -349$. Whenever the determinant of a 2×2 matrix is nonzero, the inverse is given by the formula

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \qquad (1.16)$$

Note that if the determinant is 0, this definition involves division by 0; in this case the matrix has no inverse.

Just like arithmetic with numbers, it is possible to carry out matrix arithmetic modulo 26. For example,

$$\begin{pmatrix} 3 & 2 \\ 10 & 2 \end{pmatrix} \cdot \begin{pmatrix} 13 & 20 \\ 0 & 12 \end{pmatrix} = \begin{pmatrix} 39 & 84 \\ 130 & 224 \end{pmatrix} \equiv \begin{pmatrix} 13 & 6 \\ 0 & 16 \end{pmatrix} \pmod{26}$$

Modulo 26 there is no division, so to adapt formula (1.16) for the inverse to matrices modulo 26, we replace division by $(ad - bc)$ with multiplication by the modulo 26 inverse of $(ad - bc)$:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} \equiv (ad - bc)^{-1} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \pmod{26} \qquad (1.17)$$

For the inverse of a matrix to exist modulo 26, it is not enough for the determinant to be nonzero. It must be relatively prime to 26, otherwise the inverse of the determinant does not exist modulo 26 and the formula in (1.17) cannot be applied.

Ex. 1.8.1. Carry out the following matrix multiplications modulo 26

(a) $\begin{pmatrix} 3 & 2 \\ 0 & 15 \end{pmatrix} \cdot \begin{pmatrix} 13 & 2 \\ 8 & 22 \end{pmatrix} \equiv$

(b) $\begin{pmatrix} 5 & 11 \\ 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} 22 & 8 \\ 4 & 19 \end{pmatrix} \equiv$

Ex. 1.8.2. Carry out the following matrix multiplications modulo 26

(a) $\begin{pmatrix} 11 & 3 \\ 2 & 5 \end{pmatrix} \cdot \begin{pmatrix} 19 & 2 \\ 3 & 2 \end{pmatrix} \equiv$

(b) $\begin{pmatrix} 0 & 5 \\ 25 & 15 \end{pmatrix} \cdot \begin{pmatrix} 13 & 3 \\ 4 & 4 \end{pmatrix} \equiv$

Ex. 1.8.3. Find the inverses of the following matrices (or indicate 'no inverse' when there is none'). When you find an inverse, check it by multiplying by the original matrix to get the identity.

$\begin{pmatrix} 11 & 3 \\ 2 & 5 \end{pmatrix}^{-1} \equiv$

$\begin{pmatrix} 19 & 2 \\ 3 & 2 \end{pmatrix}^{-1} \equiv$

$\begin{pmatrix} 0 & 5 \\ 25 & 15 \end{pmatrix}^{-1} \equiv$

$\begin{pmatrix} 13 & 3 \\ 4 & 4 \end{pmatrix}^{-1} \equiv$

Ex. 1.8.4. Check that the formula given for the inverse of a 2 × 2 is correct by carrying out following matrix multiplication (you should get the identity matrix).

$$\frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

1.8.2 Hill cipher encryption and decryption

When encrypting a message with the Hill cipher, the key is a square matrix modulo 26 (a square matrix is one where the number of rows and columns is the same). We will only deal with 2×2 matrices as keys, since we have not covered determinants and inverses of larger matrices.

Let's see how to encrypt the message

```
ETPHO NEHOM E
```

using the matrix

$$\begin{pmatrix} 3 & 6 \\ 1 & 3 \end{pmatrix} \tag{1.18}$$

as a key. The message is first split into blocks of 2, since this is the number of rows (and columns) in the matrix:

```
ET PH ON EH OM ET
```

1.8.1. $\begin{pmatrix} 3 & 24 \\ 16 & 18 \end{pmatrix}$, $\begin{pmatrix} 24 & 15 \\ 4 & 21 \end{pmatrix}$

We had to pad the last block since the message had an odd number of letters. Next, each block of 2 letters is treated as a 2×1 matrix of numbers modulo 26; for example, since `E` corresponds to the number 4 and `T` corresponds to the number 19, the first block corresponds to the matrix $\begin{pmatrix} 4 \\ 19 \end{pmatrix}$. After converting the rest of the blocks in this way, we see that the message corresponds to the following list of matrices:

$$\begin{pmatrix} 4 \\ 19 \end{pmatrix}, \begin{pmatrix} 15 \\ 7 \end{pmatrix}, \begin{pmatrix} 14 \\ 13 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix}, \begin{pmatrix} 14 \\ 12 \end{pmatrix}, \begin{pmatrix} 4 \\ 19 \end{pmatrix} \tag{1.19}$$

So far, we have just rewritten the message as numbers arranged in matrices. The encryption step is to multiply each of these matrices by the key matrix. For example, the first matrix will become:

$$\begin{pmatrix} 3 & 6 \\ 1 & 3 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 19 \end{pmatrix} \equiv \begin{pmatrix} 3 \cdot 4 + 6 \cdot 19 \\ 1 \cdot 4 + 3 \cdot 19 \end{pmatrix} \equiv \begin{pmatrix} 12 + 10 \\ 4 + 5 \end{pmatrix} \equiv \begin{pmatrix} 22 \\ 9 \end{pmatrix} \pmod{26} \tag{1.20}$$

After carrying out the same multiplication for the rest of the matrices in line (1.19), we get the matrices

$$\begin{pmatrix} 22 \\ 9 \end{pmatrix}, \begin{pmatrix} 9 \\ 10 \end{pmatrix}, \begin{pmatrix} 16 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 25 \end{pmatrix}, \begin{pmatrix} 10 \\ 24 \end{pmatrix}, \begin{pmatrix} 22 \\ 9 \end{pmatrix} \tag{1.21}$$

To get the ciphertext we simply convert back to letters:

```
WJ JK QB CZ KY WJ
```

and regroup:

```
WJJKQ BCZKYQ G
```

Decryption works by multiplying by the inverse of the encryption matrix. The inverse of the encryption matrix from line (1.18) is

$$\begin{pmatrix} 3 & 6 \\ 1 & 3 \end{pmatrix}^{-1} \equiv (3)^{-1}\begin{pmatrix} 3 & -6 \\ -1 & 3 \end{pmatrix} \equiv 9\begin{pmatrix} 3 & 20 \\ 25 & 3 \end{pmatrix} \equiv \begin{pmatrix} 1 & 24 \\ 17 & 1 \end{pmatrix} \pmod{26}$$

Thus to decrypt the first block `WJ`, we multiply the decryption matrix by the corresponding vector:

$$\begin{pmatrix} 1 & 24 \\ 17 & 1 \end{pmatrix}\begin{pmatrix} 22 \\ 9 \end{pmatrix} \equiv \begin{pmatrix} 4 \\ 19 \end{pmatrix} \pmod{26}$$

and we've recovered the first plaintext vector, corresponding to the first two plaintext letters `ET`.

Note that decryption works because multiplying by the inverse matrix reverses the original multiplication. For example, in the case just covered, we have

$$\begin{aligned} \begin{pmatrix} 1 & 24 \\ 17 & 1 \end{pmatrix}\begin{pmatrix} 22 \\ 9 \end{pmatrix} &\equiv \begin{pmatrix} 1 & 24 \\ 17 & 1 \end{pmatrix}\begin{pmatrix} 3 & 6 \\ 1 & 3 \end{pmatrix}\begin{pmatrix} 4 \\ 19 \end{pmatrix} \\ &\equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 4 \\ 19 \end{pmatrix} \equiv \begin{pmatrix} 4 \\ 19 \end{pmatrix} \pmod{26}. \end{aligned}$$

Ex. 1.8.5. Encrypt the message MATRI CES with the key $\begin{pmatrix} 7 & 3 \\ 5 & 2 \end{pmatrix}$.

Ex. 1.8.6. The message ZSUEI MCDDL TW was encrypted with the Hill cipher using the matrix $\begin{pmatrix} 3 & 2 \\ 11 & 3 \end{pmatrix}$ as the key. Find the corresponding decryption matrix and decrypt the message.

1.9 Breaking the Hill Cipher

Since the Hill cipher encrypts letters in blocks, frequency analysis on individual letters is not really useful for breaking the cipher. Recall that, for the example in the previous section, the message

ET PH ON EH OM ET

was encrypted to

WJ JK QB CZ KY WJ

Like the Vigenére cipher, individual letters are not always encrypted the same way: the first E became a W under encryption, while the second became a C. Unlike the with Vigenére cipher, with which letters *were* encrypted the same way if when they lined up the same way with the keyword, letters can be encrypted differently with the Hill cipher even if they occur at the same place in a block, as is the case with the E's just mentioned. For the Hill cipher, encryption of each letter in each block depends on every other letter in the block, and so there is no direct correspondence between individual letters.

To break the cipher, then, we have to do frequency analysis on blocks of letters rather than individual letters. Consider the following message, which was encrypted with the Hill cipher using a 2 × 2 matrix as the key:

```
FRAQR TFRLZ KEFPZ KXYWS XYGZX YZZZF WHPHM SMUMT GZFPK ECUWY BNVZG ZQOVA
TATDJ ROCUQ MTCCI KGZAG ONVZT MDIDJ BNVZF PZKVB BNCXZ KRBMY GZXLV ACLGZ
QOVLL ZNTXL UUVVR XYUCX ZKVMX TGZLZ UUEGN NVMKI QMFPJ CLIGZ EISWV AUGRY
HULMB TGZEI VZCLG ZBNRB XTJIC XTADY VNCSO RPSBV USCCU SVATE UTKSV MKIQM
AGBQU KGZKY UKEOX TGZBT PXRXD DZKCC OKMTJ ICXTA DYVNC SORPS BVUSC CUSVA
TEXOY CXTPS WMUKF JZFWH LFZFX TLSLM XLGZN NUKUT KSNEQ MFPZK GZRBU VZFWH
PHMSH STPGZ IQNEQ MYSHB ZBIZW KSYFR HDMSU BGZLF MSFEZ EXNJM NEFUD VVOHU
ROALI MLZZU WHZFW HPHMS KOOAK IYEFJ FREYM TMURJ ZKFKB AWHZF WHPHM SBTPS
BVUSC CZBXT HQVLO AGQUV FPYEB TTANW AGSJZ FWHPH MSVZT PSTDZ ZUMTT PYJDZ
ZUMTT PUHHU YGUSW RUSCC BTRCI NBRZK JCRBZ BASZJ BTFWZ JBTZJ BTMUN ZGZZZ
ZFWHP HMSHS VAHYU SBAFP ZKKMK PVCGB HURJZ KNEQM AGNAM TLZFP ZKFKB AWHZF
WHPHM SGZEI XNKTZ KVEWY FPDDZ KNEQM YSTEX OYCRD FPYEB TTANW AGSJZ FWHPH
MSVZT PSTDZ ZUMTT PYJDZ ZUMTT PUHHU YGUSW RUSCC BTRCI NBRZK JCRBZ BASZJ
BTFWZ JBTZJ BTMUN ZGZZZ ZFWHP HMSHS CXZKK MKPKN EIGZL ZNEQM AGYZW GNNPM
HGGZO ACUFR UVZFW HPHMS GZEIX NKTZK VEWYF PDDZK NEQMY SQJNG YCRDF PYEBT
TANWA GVENG TJGZE IXNDW NGIBL QGRNN VCIBZ WGRXO IBFVH UYGUS WRUSC CBTRC
INBRZ KJCRB ZBASZ JBTFW ZJBTZ JBTMU NZGZP HMSWM ZZORZ KKMKP LQGRT JNNKD
```

```
DZPCG BHBZK OKBLC ZDDKN CWTOT XGZLZ IZWKN PUSCC HSCXZ KKMKP GKMUR DUUCL
KMKPN NENCU JIZLT JRTFR GZXLW GKTXN WGFLV ZKSSW BTVAG ZLFNG YCNEQ MFPZK
JICXT ADYYI KSGZE IXNHY JIZLR TFRGZ XLWGZ UMTJI RYLSD VBTLV RBAGY ZZKFK
XYCRI EUQMT CRIET JFRZF FJDVU GOWCX TAZBG KHGRI ZKYZZ KKMKP FKNAI BXPUS
MTGZJ RXNAG YZZKZ JKEDI FTCUH QLTTJ TOTXN NDGXL HLDSB VNGTZ XOYZD DJCFS
FPWHY SJMLZ XLNNA LJCBQ UKGZK YMTZB PSXTL SNRTA WMZFU BZKLU VNZTK MKPGZ
HDVXQ RSWGZ ZZZFW HPHMS BTHWH GENIQ MYGZX LLHXL BVOAG QHIUS CCAGX OJZQM
RYSKV ACLUU HBZKG KBRQQ FPFJF RKOYZ ZKKMM ZVCXT RXDDP SLMMU WMZZD SEQUS
KIUXV ZTMNA HGVOD WSPJC WHZFW HPHMS ZKTAD VDIAL JCWHO AQWNA MTZKV YXTJI
ROENV RDYVO NTXLH XHGIX
```

Since the message was encrypted in blocks of size 2, we are interested in finding *bigrams* (pairs of letters) which would be common if this ciphertext were split up into blocks of two (as was done for encryption). The most common blocks of length 2 in this message are GZ and ZK, each occuring 35 times in the ciphertext.[4] (The next most common digram, ZZ, is significantly more rare, occuring only 22 times.) To make use of this information, we can make use of the list of common bigrams from Table 1.3. We might guess, for example, that the block GZ corresponds to the plaintext block TH. We can express this as an equation by writing

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 6 \\ 25 \end{pmatrix} \equiv \begin{pmatrix} 19 \\ 7 \end{pmatrix} \pmod{26}, \tag{1.22}$$

where here the matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is the (as-of-yet unknown) decryption matrix (and the inverse to the matrix used for encryption). If we find this matrix, we can use it to decrypt the rest of the message. By carrying out the multiplication in line (1.22) symbolically, we get the following matrix congruence:

$$\begin{pmatrix} 6a + 25b \\ 6c + 25d \end{pmatrix} \equiv \begin{pmatrix} 19 \\ 7 \end{pmatrix} \pmod{26}, \tag{1.23}$$

which gives the system of congruences

$$\begin{cases} 6a + 25b \equiv 19 \pmod{26} \\ 6c + 25d \equiv 7 \pmod{26} \end{cases} \tag{1.24}$$

This system cannot yet be solved, however, since we have four unknowns and only two congruences. We can get another pair of congruences, however, by making another guess about a bigram encryption: if we guess that ZK (the other most common bigram in the ciphertext) corresponds to HE in the plaintext (the second most common bigram in English according to Table 1.3), this implies that

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 25 \\ 10 \end{pmatrix} \equiv \begin{pmatrix} 7 \\ 4 \end{pmatrix} \pmod{26}, \tag{1.25}$$

[4] Note that not all occurrences of these strings in the ciphertext count towards this total; for example, the instance of ZK starting at the 9th character of the ciphertext would span two different blocks if the message were split into blocks of length 2, and so must just be a coincidence.

common bigrams			common trigrams		
1	th	3.88%	1	the	3.50%
2	he	3.68%	2	and	1.59%
3	in	2.28%	3	ing	1.14%
4	er	2.17%	4	her	0.82%
5	an	2.14%	5	hat	0.65%
6	re	1.74%	6	his	0.59%
7	nd	1.57%	7	tha	0.59%
8	on	1.41%	8	ere	0.56%
9	en	1.38%	9	for	0.55%
10	at	1.33%	10	ent	0.53%
11	ou	1.28%	11	ion	0.50%
12	ed	1.27%	12	ter	0.46%
13	ha	1.27%	13	was	0.46%
14	to	1.16%	14	you	0.43%
15	or	1.15%	15	ith	0.43%
16	it	1.13%	16	ver	0.43%
17	is	1.10%	17	all	0.42%
18	hi	1.09%	18	wit	0.39%
19	es	1.09%	19	thi	0.39%
20	ng	1.05%	20	tio	0.37%

Table 1.3: Common English bigrams and trigrams, useful for breaking the 2- and 3-dimensional Hill ciphers, respectively.

which gives that

$$\begin{pmatrix} 25a+10b \\ 25c+10d \end{pmatrix} \equiv \begin{pmatrix} 7 \\ 4 \end{pmatrix} \pmod{26}. \tag{1.26}$$

Combining the two resulting congruences from those in line (1.24) gives us the system

$$\begin{cases} 6a+25b\equiv 19 & \pmod{26} \\ 25a+10b\equiv 7 & \pmod{26} \\ 6c+25d\equiv 7 & \pmod{26} \\ 25c+10d\equiv 4 & \pmod{26} \end{cases} \tag{1.27}$$

Notice that the system can be grouped into two pairs of congruences on two variables each. We'll begin by solving the pair

$$\begin{cases} 6a+25b\equiv 19 & \pmod{26} \\ 25a+10b\equiv 7 & \pmod{26} \end{cases} \tag{1.28}$$

by subtracting congruences. To eliminate the b term in each congruence, we multiply each by the coefficient of b in the other and subtract:

$$\begin{array}{rrl} & 10(6a+25b)\equiv 10(19) & \pmod{26} \\ - & 25(25a+10b)\equiv 25(7) & \pmod{26} \\ \hline \end{array} \tag{1.29}$$

which, after simplification and subtraction, gives:

$$\begin{array}{rrl} & 8a+16b\equiv 7 & \pmod{26} \\ - & a+16b\equiv 18 & \pmod{26} \\ \hline & 7a+0b\equiv 15 & \end{array} \tag{1.30}$$

Since 7 has an inverse modulo 26 (namely, 15), we can solve for a by multiplying both sides by the inverse of 19:

$$a\equiv 15\cdot 15\equiv 17 \pmod{26}. \tag{1.31}$$

And now we can find b by plugging this back into one of the congruences from line (1.28). For example, plugging into the second one gives

$$25\cdot 17+10b\equiv 7 \pmod{26}. \tag{1.32}$$

Simplifying gives

$$10b\equiv 24 \pmod{26}. \tag{1.33}$$

Unfortunately, since 10 is not relatively prime to 26, it has no multiplicative inverse, and this congruence does not have a unique solution. By looking Table 1.2 (page 13) we can see that $10{\cdot}5\equiv 24 \pmod{26}$ and $10{\cdot}18\equiv 24 \pmod{26}$ both are true congruences. Thus we have only determined that $b\equiv 5$ or $18 \pmod{26}$. In some cases this might be the best we could do without further guesswork, in which case we might have to try both possibilities in the final decryption matrix to see which works out (by decrypting the ciphertext to something recognizable). In our case, however, plugging $a\equiv 17 \pmod{26}$ into the first congruence from line (1.28) instead of the second gives

$$6\cdot 17+25b\equiv 19 \pmod{26}, \tag{1.34}$$

which simplifies to

$$25b\equiv 21 \pmod{26} \tag{1.35}$$

and can be solved to give $b\equiv 5$ since 25 has an inverse modulo 26.

We'll solve the remaining two congruences for c and d by substitution, so that both methods of solving congruences have been demonstrated. Beginning with the system

$$\begin{cases} 6c+25d\equiv 7 & \pmod{26} \\ 25c+10d\equiv 4 & \pmod{26}, \end{cases} \tag{1.36}$$

solving the second congruence[5] for c gives

$$c\equiv 10d+22 \pmod{26}.$$

[5] note that, since 6 has no inverse modulo 26, the first congruence cannot be uniquely solved for c. Sometimes both congruences have this problem, in which case subtraction of congruences is easier to apply than substitution, although the final result will still involve multiple solutions which will have to be tried independently.

Plugging into the second congruence gives

$$6(10d + 22) + 25d \equiv 7 \pmod{26},$$

which simplifies to

$$7d \equiv 5 \pmod{26}.$$

Multiplying by 15 (the inverse of 7) gives

$$d \equiv 23 \pmod{26}.$$

Plugging this back into the second of the congruences from line (1.36) gives that

$$25c + 10 \cdot 23 \equiv 4 \pmod{26},$$

which gives that

$$c \equiv 18 \pmod{26}.$$

Thus we have found that the decryption matrix is

$$\begin{pmatrix} 17 & 5 \\ 18 & 23 \end{pmatrix} \tag{1.37}$$

(It would not be a bad idea at this point to check our congruence-solving work by checking that the blocks `ZK` and `GZ` decrypt to `TH` and `HE`, respectively.)

We can now try to use the decryption matrix to decode the ciphertext. The beginning of the ciphertext `FR AQ RT FR LZ KE FP ZK`... corresponds to the matrices

$$\begin{pmatrix} 5 \\ 17 \end{pmatrix}, \begin{pmatrix} 0 \\ 16 \end{pmatrix}, \begin{pmatrix} 17 \\ 19 \end{pmatrix}, \begin{pmatrix} 5 \\ 17 \end{pmatrix}, \begin{pmatrix} 11 \\ 25 \end{pmatrix}, \begin{pmatrix} 10 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ 15 \end{pmatrix}, \begin{pmatrix} 25 \\ 10 \end{pmatrix}, \ldots$$

which, upon multiplication by the decryption matrix in line (1.37), give

$$\begin{pmatrix} 14 \\ 13 \end{pmatrix}, \begin{pmatrix} 2 \\ 4 \end{pmatrix}, \begin{pmatrix} 20 \\ 15 \end{pmatrix}, \begin{pmatrix} 14 \\ 13 \end{pmatrix}, \begin{pmatrix} 0 \\ 19 \end{pmatrix}, \begin{pmatrix} 8 \\ 12 \end{pmatrix}, \begin{pmatrix} 4 \\ 19 \end{pmatrix}, \begin{pmatrix} 7 \\ 4 \end{pmatrix}, \ldots$$

corresponding to the plaintext

```
ON CE UP ON AT IM ET HE...
```

And it appears the encrypted message is a fairy tale.

Ex. 1.9.1. Decrypt some more of the message; enough to identify which fairy tale it is.

Ex. 1.9.2. The following ciphertext was encrypted with a 2×2 Hill cipher:

```
HOOYH BTXGW KMVDH KKBMC HOHOM TMIJN FBCPY PTRAW SCTCK XEZHO APUCT
MGYKS MTMCT CETQF ZTDJC YNVFW PPBMM GUURG PPGSX OEZHO MTIWY IQDPC
CMUBH OGEXN TCQID QPPPP QBIGK AZEYP HOAPE BEKWX HOCUW XKBAP MIMRI
JMTTL UEYPH OIJIL UDNZE IHNZR HOWXN
```

A quick count shows that the most common bigrams occurring in this ciphertext are `HO` (occurring 10 times), `PP` (occurring 4 times), and `MT` (occurring 4 times). Break the cipher.

1.10 Running Key ciphers, One-time pads, and perfect secrecy

Suppose I encrypt the message

```
WEWIL LINFI LTRAT ETHEI RTREE HOUSE ATDAW N
```

with the Vigenére cipher using the key:

```
THISISTHESUPERSECRETPASSWORDTHATHEYDONTKNOW
```

The encryption is shown below:

```
WEWIL LINFI LTRAT ETHEI RTREE HOUSE ATDAW N
THISI STHES UPERS ECRET PASSW ORDTH ATHEY D
PLEAT DBUJA FIVRL IVYIB GTJWA VFXLL AMWHA L
```

If an eavesdropper intercepted the encrypted message `PLEAT DBUJA FIVRL IVYIB GTJWA VFXLL AMKEU Q`, they would have a very difficult time breaking it. Even if they were told the length of the keyword, breaking the message into groups on which to do frequency analysis would result in groups of 1 letter each, which would be quite useless! In fact, it might seem that whenever the password used for the Vigenére cipher is at least as long as the message, the cipher is unbreakable.

The running key cipher

When the Vigenére cipher is used as above with a keyword as long as the message to be encrypted, it is called the *Running Key* cipher. Although at first glance it seems such a system would be unbreakable, the cipher can be broken by hand. Though the cipher is indeed substantially more secure than the Vigenére cipher, its weakness is the fact that the keyword is not typically random letters, but meaningful text. This bias can be exploited to find the original message. For example, suppose the attacker simply subtracts `E` from every letter in the ciphertext:

```
PLEAT DBUJA FIVRL IVYIB GTJWA VFXLL AMWHA L
EEEEE EEEEE EEEEE EEEEE EEEEE EEEEE EEEEE E
LHAWP ZXQFW BERNH ERUEX CPFSW RBTHH WISDW H
```

Since the keyword was English text and e a common letter in English, this should give a high fraction of correct letters from the plaintext, and in our example it gets nearly 14% of the letters correct. (Correct letters are underlined above.) Needless to say, while 14% isn't bad, it's not clear how we could possibly figure out the original message using just this technique, since we don't have a reliable way of distinguishing correct letters from incorrect ones.

The same idea is much more powerful, however, when applied to blocks of letters. For example, *that* is a very common English word, thus the attacker could try guessing that it appears at some point in the keystream. By subtracting THAT from each possible position in the ciphertext, the attacker can decide which partial decryptions make the most sense. For example, the first few subtractions would be:

```
PLEA  LEAT  EATD  ATDB  TDBU  DBUJ  BUJA  UJAF
THAT  THAT  THAT  THAT  THAT  THAT  THAT  THAT
WEEH  SXAA  LTTK  HMDI  AWBB  KUUQ  INJH  BCAM
```

and the complete list of four letter blocks which result from subtracting THAT from different positions of the ciphertext is

```
WEEH, SXAA, LTTK, HMDI, AWBB, KUUQ, INJH, BCAM, QTFP, HYIC, MBVY,
PORS, CKLP, YEIC, SBVF, POYP, CRII, FBBN, PUGA, IZTQ, NMJD, ACWH,
QPAC, DTVM, HOFE, CVXS, MQLS, EELH, SEAT, STMD, HFWO, TPHH, DAAS.
```

Of these, most seem like they would not be likely to arise in any English message. Some exceptions in this list are BCAM (*e.g., Bob. Camera please*), PORS (*e.g., soup or salad*), and of course, SEAT (not only *seat*, but also, as in the case of our plaintext, *house at*).

The diligent attacker would then have to build on these discoveries with further guesswork. For example, if she decides that SEAT is likely to actually occur in the plaintext, she has decided on the partial decryption

```
PLEAT DBUJA FIVRL IVYIB GTJWA VFXLL AMWHA L
                                 TH AT
                                 SE AT
```

At this point, she could try subtracting some other common words from other parts of the message. If she tried subtracting THE from different parts of the message, for example, she might find that

```
PLEAT DBUJA FIVRL IVYIB GTJWA VFXLL AMWHA L
                                 TH ATTHE
                                 SE ATDAW
```

was a likely decryption, (especially since the word *that* is often followed by *the*). At this point, lucky guessing might lead her to

```
PLEAT DBUJA FIVRL IVYIB GTJWA VFXLL AMWHA L
                                 TH ATTHE Y
                                 SE ATDAW N
```

and then to

```
PLEAT DBUJA FIVRL IVYIB GTJWA VFXLL AMWHA L
                            WO RDTH ATTHE Y
                            EH OUSE ATDAW N
```

and she is well on her way. It should be emphasized, of course, that this kind of attack requires a *lot* of trial and error, and cracking running key ciphers by hand is very labor intensive and dependent on luck. Computers can do quite well when programmed to take advantage of more sophisticated information about the language of the message and keystream (which words are likely to come after which other words, *etc*), and there is sophisticated software which can be used to break running key ciphers.

One-time pads

The weakness of running-key ciphers is that information about the likely properties of keystreams (for example, that they are likely to contain common words like *that*) can be used to deduce likely decryptions of the ciphertext. The one-time pad is a slight modification of the running-key cipher, which simply requires that the keystream be a *random* stream of letters. For example, assume we again want to encrypt the message

```
WEWIL LINFI LTRAT ETHEI RTREE HOUSE ATDAW N
```

The one-time pad demands that we generate a random keystream—for example, by drawing letters out of a hat (replacing them each time for the next draw). The letter sequence YYIVFQPUBVKCPKDGYJDSWFRTSGOMDXWXXVHR was generated 'randomly' with the help of a computer. We can use it as the key for a one-time pad to encrypt our message:

```
 WEWIL LINFI LTRAT ETHEI RTREE HOUSE ATDAW N
+YYIVF QPUBV KCPKD GYJDS WFRTS GOMDX WXXVH R
 UCEDQ BXHGD VVGKW KRQHA NYIXW NCGVB WQAVD E
```

Note that none of the techniques discussed for the running-key cipher would help an eavesdropper break the one-time pad if they intercepted our message, since there are no letters or blocks of letters which are more likely than others to appear in the keystream—since it was generated randomly, THAT is exactly as likely to appear in the keystream as are ZZZZ and MQPX.

Indeed, the one-time pad cannot be broken, because the randomness of the key used for one-time pad encryption means that *any plaintext can give rise to any ciphertext with equal probability.* For example, even though our message WEWIL LINFI LTRAT ETHEI RTREE HOUSE ATDAW N was encrypted to the ciphertext UCEDQ BXHGD VVGKW KRQHA NYIXW NCGVB WQAVD E, the plaintext THEPE OPLEI NTHET REEHO USEAR EOURF RIEND S could be just as easily be encrypted to the same ciphertext with a random key:

```
 THEPE OPLEI NTHET REEHO USEAR EOURF RIEND S
+BVAOM NIWCV ICZGD TNMAM TGEXF JOMEW FIWIA M
 UCEDQ BXHGD VVGKW KRQHA NYIXW NCGVB WQAVD E
```

In fact, if a message was intercepted with a one-time pad and someone claimed to know the message's contents, we could not even verify their claim! By subtracting their claimed message from the ciphertext we could get the keystream that would have been used for encryption. If the running-key cipher had been used, than we could verify their claim by verifying that the keystream used was English text. But for the one time pad, all keystreams are equally likely to occur, and no inference about the plaintext can be made from ciphertext, even with very lucky guessing. Because of this, the one-time pad is said to have *perfect security*, which means that the ciphertext gives the cryptanalyst no information at all about the plaintext, since any plaintext gives rise to any ciphertext with equal probability.

For the one-time pad to secure, it is fundamentally important that *a keystream can never be reused.* Suppose the keystream YYIVFQPUBV... used earlier to encrypt the message WEWIL LINFI LTRAT ETHEI RTREE HOUSE ATDAW N was also used to encrypt the message BEGIN PREPA RATIO NSIMM EDIAT ELY:

```
 BEGIN PREPA RATIO NSIMM EDIAT ELY
+YYIVF QPUBV KCPKD GYJDS WFRTS GOM
 ZCODS FGYQV BCISR TQRPE AIZTL KZK
```

If an eavesdropper has intercepted the ciphertexts from both of these encryptions, they can subtract them from each other:

```
 ZCODS FGYQV BCISR TQRPE AIZTL KZK
-UCEDQ BXHGD VVGKW KRQHA NYIXW NCG
 FAKAC EJRKS GHCIV JZBIE NKRWP XXE
```

Think now about what the resulting text represents. It is

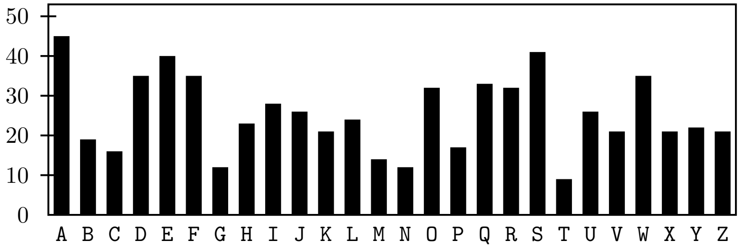
$$\begin{aligned}(\texttt{BEGIN PREPA}\ldots+\texttt{YYIVF QPUBV}\ldots)-(\texttt{WEWIL LINFI}\ldots+\texttt{YYIVF QPUBV}\ldots)\\ =(\texttt{BEGIN PREPA}\ldots-\texttt{WEWIL LINFI}\ldots),\end{aligned}$$

thus it is the difference (subtraction) of the two plaintexts. This means that the result is essentially a running-key cipher, using one of the plaintexts as a keystream to encrypt the other, and using subtraction instead of addition! Breaking this running-key cipher (which is quite possible) will reveal both original messages.

One final issue of fundamental importance with respect to the one-time pad is the issue of randomness. How can we generate a random keystream? Consider the following 'random' sequence of letters, which was generated by the author by banging on a keyboard haphazardly, and then removing nonletters and grouping into blocks of 5:

```
LKAJS DFPOI UZXBP UIOYQ WERMN YWERU YZXVT YIWER BNZVX MNZCX LKAHS DIOPY
QWERU YTWEQ IOPAJ KHASD NVZXC MNASJ KLHAS DFHJQ WEPIO UYQTR JKLSF DGZNX
ZXVCB NSDLA FFHGH ASDFY QOWER IOPER JKHAS DKLVB ZXCLH JASFD UYQWE RQTWR
EPUIO HBVAS DFGJL KQETU IOQWR EYPAJ KLSDF BNZGL ASDOF IUYQW ERKLJ HASDF
MBCNZ JOASK LJASG ASOIU YQWEL RKJHA MBCLV KJASH DFOIU YQWER HFVJD KASOI
UYQWE RHAJM ZXMZX CBVKL JHASO UIYQW ERLHA SMNZX BVALK JASDI OPQWE OIUYQ
WEROY LASDF PIOQW ERUIO XTRBV BVRME VBNRE WQGHJ HLQWR EOQPW REUIO PEUUI
OPASD FZVCX MBASD FUYOQ WREHL AGUYA SFDTQ WREYI OAFAS FDBMZ XVCLA SFDOQ
ETQWR EIASD BMZVK GJASF DQWRE OYIAS FGASF DZXVC ASFDF GFFGW REQWE UIOQW
EUIPO ASFDJ KLASF DHASF DZXCX VCHJB FDBMS RFDWR ETUWR EEQWU IOPQW EJKLA
SDHJA SFDGH ZXCZX CVXBN VCZXV CMASD FJKLQ WEUIO PQWRE ASFDY QWREP UIOJK
```

In spite of my best efforts, this letter stream is very far from random. For starters, here is the letter distribution:



The distribution is quite nonuniform. For example, letters accessible from "home position" on the keyboard (*a*, *s*, *d*, *f*, *j*, *k*, *l*) appear to be among the more common letters. While not quite as nonuniform as English text (see the graph on page 30), there is still substantial bias which might be exploited by an attacker. For example, if an attacker aware of the bias in my method of choosing random numbers is simply trying to decide whether an intercepted message YWW corresponds to the message YES or NO_ (where _ represents any letter, being used here as padding), they might observe that a plaintext of YES implies a keystream of ASE was used, which consists of letters which are commonly produced by my 'random' letter generation, and suggesting that this plaintext is likely, while the plaintext NO would imply a keystream beginning with LI, consisting of relatively less common letters.

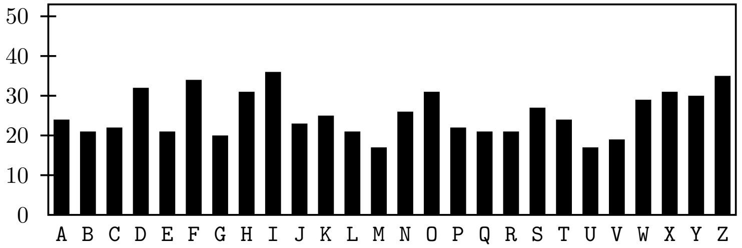
There is even greater bias in my 'random' stream in terms of long blocks. The stream has several long repeated blocks: for example, the blocks JKHASD, HJASFD, UYQWER, WREPUIO, IUYQWE, KLJHAS, ASOIUYQ, OIUYQW, YQWERH, IOPQWE, ASFDZX, QWEUIO, UIOPQW are all repeated in the 'random' stream, and some several times.[6] An attacker who knows these biases in the generation of our key material could use it very effectively to break a 'one-time pad' which used this keystream; the same techniques discussed in the section on running-key ciphers (now using common strings like UYQWER in place of common words like THAT) would be quite effective.

To emphasize the bias of our key generation, we compare it to a more truly 'random' stream of letters, generated with the help of a computer:

```
BRAGZ QGMDB TNZKH LVFWQ OTHFX SYKNL GYYOY HOIAQ ODJOC PEKFW RDDOH QIGXP
SAFFX FXNOR GOWTO KTODH FACOQ DPJBB CTZFC XAHAO SUEXS BEUWR JINLJ IINTI
FCKFN BFTYZ YVHLN FOQFQ KSISB CATRJ ZAILH GGFVW YTCCD OZTNJ HYLNA MLUWF
CDLYS IOONH ZDZOP FISPM LEZSM EJXWR OMJLH EZXPW AOPFE ZECDK AZZIV MJDXV
ACZKX OGBYN WAIWW PKVLZ XYIEG MIRYX WNNBK WPJNI XKOAD UVBLA NANFT SJSHN
FSIML MJSSH SLRDK REYHB VPNZY EDDQY JGWMS ZBUFJ DVBAO XDDFS TVOZW HBXWW
TOVEX INFYB QGUAF QXMRV JGWHR QHGSX DSCIT NIXHF GCPZH ISALB BWMKC NCWOP
EIEOU IHZQE VTIXM DDSHD PRDGC IMKYS ZGOAF FJQDD UOPRL GLXPZ YXYJY TGZSH
WMRHU FAOHQ LKIDR FTVZQ KJQXI KBSZE OIUFO HNUID FBYWW EKYWF ZSKKZ LTUJJ
```

[6] It is instructive to sit at a keyboard and type some of these strings out—they feel very 'natural' and easy to type, and never require a finger to change position in the middle of a block. It is perhaps not surprising that blocks like this appear frequently when one tries to generate random streams using a keyboard.

```
RPKRU LEUSY EPYCC AMXXP EKQTK ZIVXA XHWHC DFIBZ YRPTD TVWZQ IGRNZ YNQZX
SZCYI IDBIY CYMTJ UWKHF PVVAD TNQRJ XZCHH VHQXK TUPRF DJNWW UGZXE PILYI
```



Note that the distribution of letters, shown above, is noticeably more uniform (as we produced more random key material, it would flatten out even more). Even more striking is the data on repeated strings. The above block of random text contains *no* repeated strings of length 7, 6, 5 or even 4. And it contains just 10 repeated strings of length 3, all of which occur just twice in the text. Compare this with the stream generated by banging on the keyboard, which contains 79 repeated blocks of length 3, many of which occur 10 or more times.

The issue of random letter (and number) generation is actually a very deep one. How can one program a computer to produce 'random' numbers? Computers and some calculators have functions to generate a random number, for example. This function does cannot actually generate a truly 'random' number, but carries out deterministic operations which should nevertheless produce sequences which have the properties of a random sequence. Designing good functions for this purpose can actually be quite difficult, and many software random number generators have weaknesses which would make them unsuitable to use for the generation of one-time pad key material.

The one-time pad has some significant drawbacks for practical use. Apart from the problem of generating truly random keys, the key used has to be at least as long as the message to be sent, and can never be reused. This means that parties wishing to exchange messages need to arrange for the secure exchange of key material, and need to exchange key material which is at least as long as the total length of messages to be sent. Of course, if it is possible to securely exchange so much key material, it seems like it might make more sense to securely exchange unencrypted messages, and avoid encryption altogether.

The one-time pad found real practical use by spies, however. In the past, spies obtained key material (books of random letters to be used for encryption) from their intelligence agencies before departing on their assignments. The pages from the books would be used to encrypt messages, and then destroyed so that they would not be discovered, and never be used again (key material was sometimes printed on special highly flammable paper to make it easy to burn them without a trace). When the key material was sufficiently random, this provided complete cryptographic security through a method which was easy to carry out by hand with nothing more than pencil and paper. For most modern applications, however, (for example, encrypting an email), there is no practical way for parties to securely exchange enough key material in advance for the one-time pad to be practical. In the next chapter, we will discuss modern encryption schemes, which attempt to provide excellent security in a practice, without requiring huge amounts of key material.

1.11 Known-Plaintext attacks

For every encryption scheme we covered in this chapter (other than the one-time pad), we learned how to break the cipher 'from scratch'; *i.e.,* without anything other than the ciphertext we are trying to read. This kind of attack on a cipher is called a **ciphertext-only** attack, because the cryptanalyst only has access to the ciphertext. These kinds of attacks typically rely on statistical information about likely messages (letter or bigram frequencies in the underlying language, for example). Obviously, the fact that the classical ciphers we have covered are vulnerable to ciphertext-only attacks is a serious flaw, and one which is to be addressed by the modern ciphers covered in the next chapter.

There is also another kind of attack on a cipher, called a **known-plaintext attack**. In this situation, the attacker has a ciphertext she wants to decrypt, but also has some ciphertext (which was encrypted with the same key) for which she knows the correct decryption. It is not surprising that in this case, the cryptanalyst can break the ciphers we have discussed!

For example, suppose we intercepted a message `NOKBT YRXCO XNSDC YYX`, encrypted with the Caesar cipher. If we know that the first word of the plaintext is `DEAR`, we can mount a known-plaintext attack: this tells us that `D` was encrypted to `N`, and so the shift is 10. This allows us to decrypt the message to recover the plaintext, which reads `DEARJ OHNSE NDITS OON`.

Other than the one-time pad (when used correctly, never reusing key material), every cipher we have covered in this chapter is extremely vulnerable to this kind of attack: even a small amount of known plaintext can allow one to completely break the classical ciphers.

Ex. 1.11.1. The ciphertext `HQGDJ GTQEL HGVQL BQGMQ` was encrypted with the affine cipher. Find the original message, which begins with the word `DEAR`

Ex. 1.11.2. The ciphertext `UENZH ZIMPW EPEVZ PETJR NI` was encrypted with the Vigenére cipher. Find the original message, which begins with `JANEDOE`.

Ex. 1.11.3. The ciphertext `GENMA NCMNJ WQHF` was encrypted with the 2×2 Hill cipher. Find the original message, which begins with the name `KARLA`

In fact, it is perhaps difficult to imagine how an encryption scheme could possibly be resistant to a known-plaintext attack. Intuition might suggest that if one knows a plaintext and the corresponding ciphertext and the encryption method being used, that the key which would transform the given plaintext into the given ciphertext can be 'figured out' in some way.

It turns out, however, that it is possible to develop encryption schemes that are resistant to attack even in this situation. And it's a good thing too, because, actually, opportunities for plaintext attacks arise in all sorts of situations in modern cryptography. If someone encrypts a computer file, for example, then the type of file they are encrypting may have standard identifying information that occurs regardless of the actual file content. For example, if someone wants to encrypt a webpage for transmission, the file source will start with tags like `<DOCTYPE...   >` and `<HTML>` that come at the beginning of html format pages, regardless of the webpage. (If you use the 'view source' option in your web browser, you can see that even completely different web pages share lots of structure in common). For this reason, it is of paramount importance for modern cryptographic needs to have encryption systems which are secure even against known-plaintext attacks. Such systems are the subject of the next chapter.