1 Cryptography: An Overview

1.1 Elementary Ciphers

Cryptography is the design and use of communication schemes aimed at hiding the meaning of the message from everyone except the intended receiver. *Cryptanalysis* is the effort to foil an encryption system, to crack the code. The study of cryptography and cryptanalysis is called *cryptology* and is the focus of this chapter.[1] Later we will study some fairly sophisticated cryptographic systems, but we begin with a few elementary examples.

1.1.1 Substitution ciphers

Substitution ciphers are the familiar sort of encryption that one finds in Sunday newspaper puzzles, in which each letter of the alphabet stands for another letter. A special case is the *Caesar cipher*, in which the alphabet is simply shifted by some number of places. In the version used by Julius Caesar, the alphabet is shifted forward by three places. For example, if a letter of the original message, or *plaintext*, is A, the corresponding letter of the encrypted message, or *cyphertext*, is D, and so on as indicated here:

$$\text{plaintext: } A\ B\ C \ldots X\ Y\ Z$$
$$\text{ciphertext: } D\ E\ F \ldots A\ B\ C$$

[1] For a compact overview of cryptography and cryptanalysis, including many practical issues, see Piper and Murphy (2002). A more mathematical information-theoretic approach is given in Welsh (1988). A popular historical account can be found in Singh (1999). For a more thorough treatment of the history, see Kahn (1967).

We can express this cipher mathematically by assigning a number to each letter: $A \to 0$, $B \to 1, \ldots, Z \to 25$. Then if x represents a letter of the plaintext and y the corresponding letter of the ciphertext, Julius Caesar's cipher can be expressed as

$$y = x + 3 \pmod{26},$$

where "(mod 26)" means that one takes the remainder upon dividing by 26. (Much more on modular arithmetic in later sections of this chapter.) If you are adept at cracking the substitution ciphers of the Sunday paper, you may find it surprising that Caesar was able to keep any messages secret with this simple strategy, but evidently it worked well enough.

A simple generalization of the Caesar cipher is expressed by the equation $y = ax + b \pmod{26}$, where a and b are integers.[2] It is interesting to ask whether some values of a and b are better than others, and indeed this question is the subject of one of the exercises below. A further generalization is to use an arbitrary permutation of the alphabet.

How does one go about cracking a substitution cipher? The standard technique, which is well known today but was not known in Roman times, is *frequency analysis*. Let us assume that the cryptanalyst knows what language the plaintext is expressed in; suppose it is English. In typical English text, each letter occurs with a certain frequency. The most common letter in English is E: if you blindly point to a letter on a page in a novel, the probability that the letter will be E is around 12.7%. The following table gives the frequencies of all the letters, as computed from a sample of over 300,000 characters taken from newspapers and novels.[3]

E	12.7%	D	4.2%	P	1.9%
T	9.0%	L	4.0%	B	1.5%
A	8.2%	U	2.8%	V	1.0%
O	7.5%	C	2.8%	K	0.8%
I	7.0%	M	2.4%	Q	0.1%
N	6.7%	W	2.4%	X	0.1%
S	6.3%	F	2.2%	J	0.1%
H	6.1%	G	2.0%	Z	0.1%
R	6.0%	Y	2.0%		

[2] The special case with $a = 1$ and $b = 13$, called "ROT13," is used nowadays in online settings to hide such things as joke punchlines and puzzle solutions.

[3] Piper and Murphy (2002). The authors write that the table is based on one originally compiled by H. J. Beker and F. C. Piper.

We can use this table to crack a substitution cipher as follows. Given the ciphertext, we count how many times each letter appears. If the message is long enough, the frequencies of occurrence will help us guess how each letter should be decrypted. For example, if v occurs around 13% of the time, we guess that v represents the letter *e*. Once we have correctly guessed a few of the letters, we look for familiar words and so on. A related technique is to look for *pairs* of letters that occur frequently together. Some of the exercises at the end of this section will give you practice with frequency analysis.

1.1.2 Vigenère ciphers

We now consider a cipher that is more sophisticated than simple substitution. It was invented by Giovan Batista Belaso in the sixteenth century but later incorrectly attributed to Blaise de Vigenère and given his name. (Vigenère devised a more powerful variation on this cipher, in which the message itself was used to generate the key.)[4] The secret key in this case is a word or phrase. It is easiest to explain the cipher by giving an example; in the following example the message is "Meet me at midnight," and the key is "quantum." (Not a key that Belaso or Vigenère is likely to have used.)

```
PLAINTEXT:   M E E T M E A T M I D N I G H T
KEY:         Q U A N T U M Q U A N T U M Q U
CIPHERTEXT:  C Y E G F Y M J G I Q G C S X N
```

To generate the ciphertext, we have associated an integer with each letter as before: A$\rightarrow$ 0, B$\rightarrow$ 1, etc.; in each column above we have added, mod 26, the numbers corresponding to the given letters of the plaintext and the key. For example, the first letter of the ciphertext is obtained as follows:

$$\mathrm{M} + \mathrm{Q} \;\rightarrow 12 + 16 \pmod{26} = 2 \rightarrow \mathrm{C}$$

In other words, each letter is encrypted with a Caesar cipher – the encryption is a cyclic shifting of the alphabet – but different letters can be shifted by different amounts. In the above example, six distinct Caesar ciphers

[4] Belaso's cipher is closely related to ciphers devised by others in the preceding century. A full account can be found in Kahn (1967).

are used in a pattern that repeats after seven letters. The intended recipient should know the key and can recover the plaintext by subtracting from each letter of the ciphertext the corresponding letter of the key.

Notice how this cipher improves on the simple substitution scheme. The letter M appears three times in our plaintext, and each time it is encrypted differently. Conversely, the letter G appears three times in the ciphertext, and each time it stands for a different letter of the plaintext. Thus a straightforward frequency analysis will not be nearly as effective as it is against a substitution cipher.

However, one can still use frequency analysis to crack the cipher if the message is long enough. Suppose that the cryptanalyst can somehow figure out the length of the repeated key. Let us say that the length is 7 as in the above example. Then every seventh letter is encrypted with the same Caesar cipher, which can be cracked by doing a frequency analysis on just those entries of the ciphertext. So the problem is not hard once we know the length of the key. But how might the cryptanalyst guess the length of the key? One method is to look for repeated strings of letters. For example, in a long message it is quite likely that the word "the" will be encrypted in the same way several times and will thus produce the same three-letter sequence several times. So if the cryptanalyst sees, for example, three instances of "rqv," the second instance displaced from the first by 21 steps and the third displaced from the second by 56 steps, he or she could reasonably guess that the repeated key is seven letters long, since 7 is the only positive integer (other than 1) that divides both 21 and 56. Of course such a guess becomes more trustworthy if more repetitions are discovered, since it is always possible for a string of letters of the ciphertext to be repeated by chance. This method of cracking the Vigenère cipher was discovered in the nineteenth century by Friedrich Kasiski.

An alternative version of the Vigenère cipher replaces the repeated key with a "running key," usually an easily accessible text that is at least as long as the message. For example, we might use as the key the Constitution of the United States, beginning with the preamble. Then our encryption of "Meet me at midnight" would look like this:

```
PLAINTEXT:  M E E T M E A T M I D N I G H T
KEY:        W E T H E P E O P L E O F T H E
CIPHERTEXT: I I X A Q T E H B T H B N Z O X
```

The recipient, knowing the key, again simply subtracts it, letter by letter, from the ciphertext to recover the original message.

Clearly the cryptanalytic method we just described will not work against this encryption scheme, because the key is no longer periodic. But the key does have some structure, and a cryptanalyst can use this structure to get a foothold on the plaintext. For example, if the cryptanalyst suspects that the key is a piece of English text, she can guess that the word "the" appears in it frequently. She can then try "the" as part of the key in various positions along the ciphertext and see if the resulting plaintext is plausible as part of the message. Let us try this in the above example, applying THE at each position of the ciphertext.

Trigram in ciphertext	Trigram minus THE
IIX	PBT
IXA	PQW
XAQ	ETM
AQT	HJP
⋮	⋮
ZOX	GHT

Most of the trigrams on the right-hand side of the table could not possibly be part of a message written in English. In fact the only plausible candidates are ETM and GHT. The latter is particularly helpful, because there are only a few combinations of letters that are likely to precede GHT in English. The cryptanalyst might try a few of these, to see what they would imply about the key. Here is a table showing what he or she would find:

Guess at plaintext	Ciphertext minus plaintext = key
OUGHT	NTTHE
NIGHT	OFTHE
FIGHT	WFTHE
RIGHT	KFTHE
LIGHT	QFTHE
EIGHT	XFTHE

Of these, only the first two make any sense as part of a passage in English, and of these the second is more likely. So the cryptanalyst might

tentatively guess that NIGHT is part of the plaintext and OFTHE part of the key. Continuing in this way, working back and forth between the unknown plaintext and the unknown key, he or she has a reasonable chance of cracking the cipher.

1.1.3 One-time pad

What makes the Vigenère cipher insecure, even with the running key of the last example, is that the key has some structure that can be exploited by the cryptanalyst: the key is a piece of English text, and English definitely has some structure. The natural way to avoid this problem is to use a running key consisting of purely random letters. The key used in the following example was generated, literally, by tossing coins.

```
PLAINTEXT:  M E E T M E A T M I D N I G H T
KEY:        P O V N H U J B K R C J D C O F
CIPHERTEXT: B S Z G T Y J U W Z F W L I V Y
```

Of course the intended recipient must also have a copy of the random key.

In this example, even though there is plenty of structure in the plaintext, the randomness of the key – if it is truly random – guarantees that there will be no structure whatsoever in the ciphertext. This cryptographic scheme can therefore not be broken by cryptanalysis. [5] (An eavesdropper could try other attacks such as intercepting the secret key on its way to the intended recipient.) We are assuming here that the random key is at least as long as the message, so that it will not have to be repeated. Also, for complete security it is important that the key be used only once. If it is used twice, an eavesdropper could compare the two ciphertexts and look for patterns. This method of encryption – a random key used only once – is known as a *one-time pad*, suggesting that the key might be copied on a pad of paper, delivered to the intended recipient, used once, and then destroyed.

Nowadays much of the information that is conveyed from place to place is in digital form and can be expressed as a sequence of zeros and

[5] A precise statement of this claim was proved by Shannon (1949).

ones. A one-time pad works fine for such an application, the key in this case being a random binary string. For example, one might see the following encryption of a rather uninteresting message. (Here again the key was generated by tossing a fair coin, despite what you may think.)

```
PLAINTEXT:  0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
KEY:        1 0 0 1 0 0 1 0 0 1 0 0 0 0 1 1
CIPHERTEXT: 1 1 0 0 0 1 1 1 0 0 0 1 0 1 1 0
```

In each column the two entries, one from the plaintext and one from the key, have been added mod 2: that is, the ordinary sum is replaced by its remainder upon division by 2, so that $1 + 1 \pmod 2 = 0$.

Though the one-time pad is perfectly secure against cryptanalysis, it is by no means the most widely used form of cryptography. The problem with the scheme is that the random key has to be generated and delivered securely to the intended recipient at a rate equal to or exceeding the rate at which messages are to be conveyed. If this rate is large, one might have to employ an army of trusted couriers to transport the key.

Later in this book we consider in some detail a recently invented scheme, quantum key distribution, that could potentially solve this problem by relying not on mathematics but on the laws of nature. But our study of quantum key distribution will have to wait until we have introduced the relevant laws in Chapter 2.

EXERCISES

Problem 1. We mentioned a substitution cipher in which each plaintext letter, represented by an integer x, is replaced by the letter corresponding to the integer $y = ax + b \pmod{26}$, where a and b are integers. If the alphabet we are using has n letters, where n is not necessarily 26, we can generalize this rule to $y = ax + b \pmod n$, where "mod n" means that we take the remainder upon division by n. In answering the following questions, assume that the integers a and b are restricted to the values $0, \ldots, n-1$.

(a) Suppose that n has the value 26, as it does if the plaintext is in English and we do not encrypt spaces or punctuation marks. Is there a

reason not to use certain values of the constant a or of the constant b? If so, which values are the bad ones and what makes them bad?

(b) If we also count "space" as a character to be encrypted, we have $n = 27$. Now what, if any, are the bad values of a? Of b?

(c) For a general n, make a conjecture as to what will be the bad values of a and b, if there are any.

Problem 2. The following ciphertext was encrypted by a simple shift of the alphabet. All spaces and punctuation marks were first deleted from the plaintext, which was then arbitrarily broken into five-letter blocks. Find the original plaintext.

```
VQFGE KRJGT VJKUU GPVGP EGUJK HVGCE JNGVV GTDCE
MYCTF DAVYQ UVGRU
```

Problem 3. The following ciphertext was generated by a Vigenère cipher with a repeating key. All spaces and punctuation marks were removed from the plaintext, and the resulting ciphertext was broken into six-letter blocks.

```
NRUATW YAHJSE DIODII TLWCIJ DOIPRA DPANTO EOOPEG
TNCWAS DOBYAP FRALLW HSQNHW DTDPIJ GENDEO BUWCEH
LWKQGN LVEEYZ ZEOYOP XAGPIP DEHQOX GIKFSE YTDPOX
DENGEZ AHAYOI PNWZNA SAOEOH ZOGQON AAPEEN YSWYDB
TNZEHA SIZOEJ ZRZPRX FTPSEN PIOLNE XPKCTW YTZTFB
PRAYCA MEPHEA YTDPSA EWKAUN DUEESE YCNJPP LNWWYO
TSKYEG YOSDTD LTPSED TDZPNK CDACWW DCKYSP CUYEEZ
MYDFMW YIJEEH WICPNY PWDPRA LSPSEK CDACOB YAPFRA
LPLLRA YTHJCK XEOQRK XAOZUN NEKFTO TDAZFK FROPLR
PSWYDE DMKCEI JSPPRE ZUO
```

(a) Look for strings of three or more letters that are repeated in the ciphertext. From the separations of different instances of the same string, try to infer the length of the key.

(b) Using frequency analysis or any other means, try to find the key and the plaintext. (You might find Section A.3 of the Appendix helpful.)

1.2 Enigma

Though our review of cryptography is by no means exhaustive, there is one historical example that we cannot pass by, namely, the Enigma cipher used by the German military before and during World War II.[6]

The Enigma cipher is more complex than the ciphers we have considered so far. Though it can be described in purely mathematical terms and could in principle be implemented by hand, the cipher is intimately tied to a mechanical device, the Enigma machine. In this section we describe a slightly simplified version of the Enigma machine and the cipher it generates.[7]

1.2.1 The Enigma cipher

The main cryptographic components of the machine are (i) the plugboard, (ii) the rotors, and (iii) the reflector. Each of these parts has the effect of permuting the alphabet, and in each case the permutation is achieved by electrical wires that we can imagine connecting the input letter to the output letter. The net effect of all the parts is obtained by following the wires through the machine, from the original input letter, typed on a keyboard, to the output letter, indicated by the lighting of a lightbulb labeled with that letter. We now describe briefly each of the components.

The *plugboard* includes an array of 26 jacks, one for each letter, and six electrical cables, each of which can be plugged into two of the jacks so as to interchange those two letters.[8] All the letters that are not part of such interchanges are left unchanged. Let us call the plugboard's permutation A; it is a function that maps the alphabet to itself. If x is an input letter, we will write Ax (without parentheses) to indicate the plugboard's output. Notice that the inverse function A^{-1}, which takes a given output of the plugboard to the corresponding input, is the same as A itself. This fact will be important in what follows.

[6] For more on the Enigma cipher, see for example Sebag-Montefiore (2000).
[7] Our main simplification is to avoid discussing the "rings," a feature of the Enigma machine that added some security but did not constitute one of the main cryptanalytic challenges.
[8] Each jack actually consists of a *pair* of holes – an input and an output – and each electrical cable consists of a pair of wires: if one wire sends the letter B to the letter J, for example, its companion wire sends J to B.

Each *rotor* is a disk, with 26 input locations arranged in a circle on one side, and 26 output locations arranged in an identical circle on the other side. Inside the rotor, a wire runs from each of the input locations to one of the output locations, and together, the 26 wires implement a complicated permutation with no special symmetries. The output of the plugboard becomes the input to the first rotor, the output of the first rotor becomes the input to the second rotor, and so on. In the original Enigma machine used by the German army, there were three standard rotors, each embodying a different permutation.

The *reflector* acts on the output of the last rotor and effects a permutation that, like that of the plugboard, simply interchanges letters in pairs. Unlike the permutation of the plugboard, the reflector's permutation is fixed and cannot be changed by the operator of the machine, at least not in the simple version of Enigma that we are considering here. (There were other versions allowing some freedom to adjust the reflector.) Also the permutation is not limited to six pairs of letters: *every* letter is sent to a different letter. We will call the reflector's permutation B, and we note that $B^{-1} = B$.

Let us now follow the path by which the input letter leads to a particular output letter. As we have implied above, the input letter first encounters the plugboard permutation, then each of the rotor permutations in turn, and then the reflector permutation. After that, the path goes *backwards* through the rotors (in reverse order) and finally through the plugboard again before the output is indicated by a labeled lightbulb. The whole path is diagrammed for a simplified alphabet in Fig. 1.1.[9] Notice that because the reflector leaves no letter unchanged, neither does the Enigma machine as a whole: it never encodes a letter as itself.

The most characteristic and subtle feature of the Enigma machine is this: though each rotor has a fixed permutation wired into it, its *orientation* with respect to the other rotors and with respect to the other components can change from one keystroke to the next. There is one special orientation of each rotor which we call the "standard" orientation. Let R_i be the permutation executed by the ith rotor when it is in its standard orientation. Then, if the rotor's orientation is rotated from its standard

[9] This figure and Fig. 1.2 were inspired by similar figures in Singh (1999).